PLATEFORME INFORMATIQUE POUR L'ASSISTANCE À L'AUTONOMIE À DOMICILE DE PERSONNES ÂGÉES

Mémoire présenté

dans le cadre du programme de maîtrise en informatique

en vue de l'obtention du grade de maître ès sciences (M.Sc.)

PAR

©GUILLAUME GINGRAS

Décembre 2021

Composition of the jury :

Djamal Rebaine, jury president, Université du Québec à Chicoutimi

Mehdi Adda, research director, Université du Québec à Rimouski

Abdenour Bouzouane, co-director of research, Université du Québec à Chicoutimi

Marwen Bdiri, external examiner, Alithya

Initial submission : June 30th 2021 Final submission : December 8th 2021

I dedicate this to my mother and father. Thanks for always being there for me.

ACKNOWLEDGMENTS

I have received a tremendous amount of support and aid from the beginning of to the end of writing and preparing this work.

To begin, I am thankful for the support given from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds de recherche du Québec en Nature et technologies (FRQNT). These organisations are of tremendous help to stay motivated and aid the advancements of my academic progress.

I am extremely grateful and would like to show appreciation for my research director and supervisor Dr Mehdi Adda. You have created an immensely enjoyable and productive research environment through your discipline, positive vision and enthusiasm for the possible future as well as your clear and precise explanations. The insights, guidance and feedbacks keep pushing the envelope of my research methodology towards a professional and elite level.

I would like to thank my research co-director Dr Abdenour Bouzouane for your implication in supporting my advancements and helping me with your numerous quality feedbacks. In the same fashion, I would like to thank Dr Hussein Ibrahim and Dr Clémence Dallaire for their support and collaboration throughout my academic progress.

Finally, it is imperative to thank my parents who have always been there in times of need and to help me make wise decisions throughout my life.

RÉSUMÉ

Ambient Assisted Living (AAL) en général et Activity Recognition (AR) en particulier sont des domaines de recherche actifs qui visent à aider les personnes dans leurs activités de la vie quotidienne (AVQ). Au cours des dernières années, nous avons constaté un intérêt accru pour leur applicabilité aux personnes âgées vivant en milieu rural qui perdent lentement leur autonomie en raison du vieillissement et aux maladies chroniques.

Une avenue de recherche importante consiste à agréger et à rechercher des corrélations entre les données physiologiques qui servent à surveiller la santé des personnes âgées, leurs AVQ, leurs mouvements et toute autre donnée pouvant être recueillis sur leur environnement immédiat.

Dans ce travail, nous abordons la possibilité de développer une plateforme non intrusive et abordable en raison de l'absence d'une telle plateforme. Elle est basée sur des capteurs de santé, de mouvement, d'activité et de localisation.

En outre, nous discutons des principaux concepts derrière la création d'une architecture en couches, flexible et hautement modulaire qui se concentre sur la façon dont l'intégration de données de capteurs combinés peut être réalisée. À l'aide d'un prototype d'application de téléphonie mobile, nos travaux ont montré que nous pouvons intégrer de nombreuses technologies non invasives qui ne sont pas nécessairement les plus récentes, mais les plus abordables, évolutives et prêtes à être déployées dans des environnements réels.

Un autre domaine de recherche découlant de ces avancées est de savoir comment la technologie et l'analyse pourraient bénéficier à la prévention et au traitement des maladies chroniques chez le nombre croissant de personnes âgées ayant des problèmes de santé. De nombreuses architectures sont proposées dans la littérature, mais elles manquent de modularité et de flexibilité pour différents types de capteurs. À cette fin, nous proposons une architecture à quatre couches et hautement modulaire pour l'analyse de la santé des personnes âgées.

Finalement, nous évaluons l'approche en implémentant une partie de l'architecture sur des nœuds de brouillard et le cloud. De plus, nous déployons ces capteurs abordables, de qualité, et accessibles au grand public dans un appartement afin d'avancer vers l'utilisation du système proposé. Des données recueillies sont utilisées comme un test préliminaire pour évaluer les capacités de la plate-forme. En utilisant les données collectées lors de l'étape de validation, nous effectuons des prévisions d'une semaine dans le futur pour des séries univariées en utilisant des méthodes classiques populaires et les méthodes d'apprentissage en profondeur les plus récentes. Une comparaison de précision est présentée.

ABSTRACT

Ambient Assisted Living (AAL) in general and Activity Recognition (AR) in particular are active fields of research that aim at assisting people in their Activities of Daily Living (ADL). In recent years, we have seen an increased interest in their applicability to the rural seniors who are slowly losing their autonomy due to aging and chronic diseases.

One research venue is to aggregate and seek for correlations between the physiological data that serves to monitor the health of the elderly, their ADLs, their movements and any other data that may be collected about their immediate environment.

In this work, we are tackling the possibility of developing a non-intrusive and affordable platform due to the lack of such a platform. It is based on embedded health, movement, activity and location sensors.

Furthermore, we discuss the main concepts behind the creation of a layered, flexible and highly modular architecture that focuses on how the integration of newly combined sensor data can be achieved. Using a mobile phone application prototype, our work has shown that we can integrate many non-invasive technologies that are not necessarily the newest, but the most affordable, scalable and ready to be deployed in real life settings.

Another researched venue deriving from these advances is how the technology and analytics could benefit the prevention and treatment of chronic diseases in the escalating number of elderly people experiencing health issues. Many architectures are proposed in the literature, but they lack modularity and flexibility for different types of sensors. To that end, we propose a four layered and highly modular architecture for health analytics of elderly people.

In the final analysis, we evaluate the approach by implementing part of the architecture on fog nodes and the cloud. Moreover, we deploy these affordable consumer grade sensors in an apartment in order to move toward the use of the system proposed. The data collected from this experiment is used as a preliminary test of the capabilities of the platform. We perform univariate series forecasting using a popular classical methods and the more recent deep learning methods by using the data collected in the validation stage. An accuracy comparison is presented.

IoT, Remote Elderly Monitoring, Smart & Connected Health, Analytics, Ambient Assisted Living, Sensors, Artificial Intelligence

TABLE OF CONTENTS

ACKNOWLEDGMENTS vi

RÉSUMÉ vii

ABSTRACT viii

TABLE OF CONTENTS ix

LIST OF TABLES xiii

LIST OF FIGURES xiv

ABREVIATIONS LIST xvi

CHAPTER 1
INTRODUCTION 1

1.1 Research context 1

1.1.1 Ageing of the population 1

1.1.2 Field of research 3

1.1.3 History of smart health systems 4

1.1.4 Activity recognition 4

1.1.5 Biomarkers and the P4 Health Continuum 5

1.1.6 Quiet data gathering 5

1.1.7 Knowledge hierarchy 6

1.2 Research problem 7

1.2.1 The problem 7

1.2.2 Research hypothesis 8

1.2.3 Methodology 8

1.3 Contribution 9

1.4 Definitions and terminology 11

1.4.1 Activities of daily living (ADL) 11

1.4.2 Signal processing 11

1.4.3 Time series 11

1.4.4 Forecasting 13

1.5 Ethics 14

CHAPTER 2
STATE OF THE ART 15

2.1 Logical and physical architecture 15

2.2 Analytics architecture 17

2.3 Forecasting 19

CHAPTER 3
SYSTEM COMPONENTS AND DESCRIPTION 21

3.1 Physical architecture 22

3.1.1 Sensor layer 22

3.1.2 Edge layer 23

3.1.3 Cloud layer 23

3.2 Logical architecture 23

3.2.1 Sensing/data layer 23

3.2.2 Edge computing layer 24

3.2.3 Cloud computing layer 26

3.3 Analytics architecture 27

3.3.1 Layer 1: sensing 28

3.3.2 Layer 2: data preprocessing 30

3.3.3 Layer 3 : data processing pipelines 33

3.3.4 Layer 4: knowledge and insight 34

3.3.5 Automated algorithm selection 36

CHAPTER 4
IMPLEMENTATION AND DEPLOYMENT 38

4.1 Considered data 38

4.2 Tasks 40

4.3 Technologies 40

4.3.1 Protocols 40

4.3.2 Hardware 43

4.3.3 Operating systems . . . 50
4.3.4 Frameworks . . . 51
4.3.5 Delta Lake . . . 54
4.4 Pipeline . . . 54
4.5 Data . . . 56
4.5.1 Abstraction . . . 56
4.5.2 Payloads . . . 56
4.5.3 Sensor triggers . . . 59
4.6 Apartment . . . 59
CHAPTER 5
ANALYTICS EXPERIMENTS . . . 61
5.1 Forecasting model development process . . . 61
5.1.1 Problem definition . . . 61
5.2 Deep learning methods . . . 63
5.2.1 CNN . . . 64
5.2.2 LSTM . . . 65
5.3 An experimental study . . . 67
5.3.1 Dataset . . . 67
5.3.2 Data preparation . . . 69
5.3.3 Training and test data . . . 70
5.3.4 Assessment metrics . . . 70
5.3.5 Forecasting performance baseline . . . 71
5.3.6 ARIMA . . . 72
5.3.7 Univariate CNN model . . . 74
5.3.8 Univariate LSTM model . . . 75
5.4 Results . . . 76
5.5 Discussion . . . 77
5.6 Conclusion and future work . . . 77
GENERAL CONCLUSION . . . 79

REFERENCES 80

LIST OF TABLES

1 10 Common Chronic Conditions for Adults 65+. 2

2 P4H Continuum Stakeholders 6

3 Accelerometer data (sample data) 58

4 Gyroscope data (sample data) 58

5 Heart rate data (sample data) 58

6 Motion data (sample data) 59

7 Multipurpose data (sample data) 59

8 Description of the health and activity oriented dataset 68

9 Number of readings collected by the sensors 70

10 Separation of data for the training and test sets from the cleaned full dataset . 70

11 Comparison of RMSE scores between methods 76

LIST OF FIGURES

1 A high-level overview of the physical architecture. 22

2 A high-level overview of the logical smart health architecture. 24

3 Data analytics workflow and architecture proposed and separated into four layers. 28

4 Process of algorithm selection for a specific task. 37

5 BLE Architecture and Protocol Stack 41

6 ZigBee mesh network. 42

7 Raspberry Pi B+. 44

8 Motorola Moto G7 Play - 32GB. 45

9 Accelerometer and Gyroscope - MetaMotionC (MMC). 45

10 MetaMotionC (MMC) placed on the chest. 46

11 Mi Band 3. 47

12 Apple Watch (Series 4). 47

13 Multipurpose. 48

14 Motion. 49

15 Water Leak. 49

16 Aria 2. 50

17 Basic architecture of Kafka 53

18 Apache Spark components and API stack 54

19 Full data flow and pipelines. 55

20 The apartment where the prototype was deployed. 60

21 CNN architecture. 65

22 LSTM architecture. 66

23 Weekly RMSE scores for the weekly baseline forecast strategy. Overall RMSE: 686 readings. 71

24 Plot of the AutoCorrelation Function (ACF). 72

25 Plot of the Partial AutoCorrelation Function (PACF). 73

26 Weekly RMSE scores for the weekly ARIMA forecast. Overall RMSE: 526 readings. 73

27 Weekly RMSE scores for the weekly CNN forecast. Overall RMSE: 549 readings. 75

28 Weekly RMSE scores for the weekly LSTM forecast. Overall RMSE: 562 readings. 76

ABREVIATIONS LIST

IOT Internet Of Things

AI Artificial Intelligence

AAL Ambient Assisted Living

AR Activity Recognition

ADL Activities of Daily Living

CHSLD Centre d'Hébergement et de Soins de Longue Durée

UQAR Université du Québec À Rimouski

UQAC Université du Québec À Chicoutimi

ReLU Rectified Linear Unit

CNN Convolutional Neural Network

LSTM Long Short Term Memory

BLE Bluetooth Low Energy

GPS Global Positioning System

PPG Photoplethysmography

ECG Electrocardiogram

CHAPTER 1

INTRODUCTION

In this chapter, we introduce the ideas that have driven our research. First we present the context in which we eventually derive our research problem from. Then, we present our contribution to the problem. Next, we define some concepts that are used in our approaches. Since this study is closely related to humans and their health, we finish by presenting some of the ethics challenges in artificial intelligence in the healthcare sector.

1.1 Research context

This section presents our research context. We begin by presenting that the population is ageing and the projections say that there will be more elderly in the future. Considering these facts related to the ageing of the population, we present the field of research in general. We continue by going through an overview of the history of smart health systems, activity recognition, biomarkers, the P4 Health Continuum, quiet data gathering and the knowledge hierarchy.

1.1.1 Ageing of the population

In recent years and in most countries and regions, the number of people aged 65 years and over has been increasing and will keep increasing to an accelerated rate. The number of people 65 or over has reached a total of 703 million in 2019 United Nations (2019). We denote this age group as ”older persons”. By 2050, they are projected double and ultimately reach a pool of nearly 1.5 billion people. In addition, we find that the number of ”oldest-persons”, which is the group of people that are 80 years or over, is the sub-group of older persons

that grows at the fastest rate. Unsurprisingly, the local Québec population is following the same trend. For example, in 2016, the older person's pool represented 18 per cent of the Québec population and it is projected to reach 27.7 percent by 2066 according to Institut de la Statistique du Québec (2019).

In the light of these statistics, we can imply that the number of people that will be suffering from chronic diseases will be increasing substantially. According to the Centers for Medicare & Medicaid Services (2015), 80% of adults 65+ have at least 1 chronic condition and 68% have 2 or more chronic conditions. Chronic diseases do not follow a single uniform definition but they have a few common characteristics. Generally, they are persistent and recurring health problems that are not measured in days or weeks, but in months and years Goodman et al. (2013). In the table 1, we present the 10 most common chronic conditions for adults that are 65 and over. The two most common conditions are hypertension and high cholesterol with 58 per cent and 47 percent respectively.

Table 1: 10 Common Chronic Conditions for Adults 65+.

Chronic condition name	Percentage of adults 65+ (%)
Hypertension	58
High Cholesterol	47
Ischemic Heart Disease	29
Diabetes	27
Chronic Kidney Disease	18
Heart Failure	14
Depression	14
Alzheimer's Disease and Dementia	11
Chronic Obstructive Pulmonary Disease	11

Despite these health problems facing the elderly population, elderly people want to stay at home as long as possible. This may be more difficult to achieve in a rural setting where

distances require automobile transport, where nearby services are often limited and even if the municipality wants the elderly to continue to stay there. When a functional decline associated with aging, loss of autonomy and the presence of chronic diseases occurs, the elderly often have to move into a retirement home. In rural communities, this can lead to an exodus from the community, as residences for the elderly are often scarce.

1.1.2 Field of research

Ambient Assisted Living, which came into existence from a branch of Ambient Intelligence, tries to make optimal use of the technological advances in hardware, software infrastructure and artificial intelligence algorithms by assisting the human to accomplish tasks and support the population toward better aging Cook et al. (2009); Muñoz et al. (2011). This concept is closely related to the primary focus coming from the field of gerontechnology, although there is also no universal definition of Ambient Assisted Living as of now Blackman et al. (2016).

It is important to realize that this domain of research has been fuelled by many other research endeavours. For instance, in recent years, there has been a great deal of advancement in the Internet of Things (IoT) due to the increasing number of mobile devices used and the production of useful sensors. In 1999, Kevin Ashton coined the term during a presentation with the goal to collect data automatically and increase efficiency of supply chains Ashton (2009). A few years later, it was reintroduced with the idea that ”things” include a controller acting as the ”brain”, sensors, actuators and a network Gershenfeld et al. (2004). With the vast amount of data gathered by these ”things”, the concept of big data has to be introduced. Big data means that the data gathered is too big, too fast or too hard Madden (2012).

It is reasonable to imagine using these concepts to gather information in the home of a senior to try to help them live a better life on their own. Such a system would be considered smart, hence the name smart health system. If properly applied, smart health systems could

allow the elderly to stay at their homes longer, which would relieve pressure and stress on caregivers and their relatives. Such a system has the potential for practical health conditions monitoring, prevention of diseases, smart intervention and telemedicine.

1.1.3 History of smart health systems

We can establish that there are three generations of technologies for making health-oriented ambient intelligence systems Doughty et al. (1996). The first generation focused on alerts triggered by users to communicate an emergency situation. The person must take the action to be taken care of. The second generation tries to fix functional issues found in the first one by integrating sensors into the environment and allow automatic alerts to be sent if abnormal situations occur. At this stage, we begin to benefit from simple programmed rules to trigger these alerts. Next, the third generation is where the ambient system must make it possible to prevent, anticipate and predict important critical and emergency situations while remaining discreet toward users. This type of system relies heavily on artificial intelligence methods.

1.1.4 Activity recognition

A useful way of analyzing the health and activity of an elderly person is by knowing what they are doing, when and how well. Activity recognition is described by recognizing the actions or objectives of a person or a group of people in an environment D'Sa and Prasad (2019). The concept can be implemented by fetching raw data from the senior environment, process it and automatically associate an activity using different types of algorithms. The sensors fetching the data can vary in type Ramasamy Ramamurthy and Roy (2018).

1.1.5 Biomarkers and the P4 Health Continuum

In order to solve the issues surrounding chronic diseases in the elderly population with technologies deployed in their environment, it is imperative to assemble the system in accordance with current healthcare goals and objectives. The concept of using biomarkers to measure a biological state or condition could be useful for this use case. There are 4 biomarkers to think about in a delivery of systems: diagnostic, predictive, prognostic and staging.

The current health care system is focused on treating diseases mainly after it is diagnosed. The collection of data in the environment of elderly people has the capability to aim toward a more proactive way of assessing elders chronic conditions instead of a reactive one. The ”P4 Health Continuum” model was proposed to increase health span by applying the four P’s to healthcare systems: Predictive, Preventive, Personalized and Participatory Sagner et al. (2017).

The predictive component is used to gather enough information to predict events in order to intervene before it is too late. The preventive element tries to remove the risk factors instead of treating the individual. An example is to identify a risk at the time when it is most reversible. Precise or personalized means to remove the generality in healthcare and customize solutions for the individuals in order to keep them informed and prepared to make better health decisions. Finally, participatory is the idea to engage patients more in the healthcare process and more away front the ”top-down approach”.

1.1.6 Quiet data gathering

Given these past points, it is reasonable to believe that ubiquitous computing which encompasses mobile computing and pervasive computing is a relevant technology field that could be accepted by elderly people. One reason is because it focuses on bringing computers in our natural environment without being noticed. Elderly people would be able to see little

Table 2: P4H Continuum Stakeholders

Stakeholders Name
Educational systems
Employers
Food industry
Government - Policy makers
Health and fitness industry
Health care organizations and professionals
Individual and families - People
Insurance industry and payers
Medical outlets
Mobile health and technology companies
Nonprofit and community organizations
Professional organizations

change in their daily life because the pervasive concepts try to make the computer invisible in the environment while still making it intelligent.

1.1.7 Knowledge hierarchy

The embedded computers are installed in order to gather raw data. This raw data represents pure facts like numbers, text or symbols. From the raw data, it is now possible to retrieve basic information that represents what is actually happening in the environment. It is possible because we have the data in a better context and significance. At the level of information, if we add meaning to what we now know, the information becomes knowledge. By adding insights and strategies we can achieve our goals with a better understanding. This knowledge becomes wisdom when these decisions and judgments are based on a larger social

context and your values. This will lead to better decisions. Bernstein (2009) explores in detail the Knowledge Hierarchy that was just mentioned. In the context of elderly activity and biomarkers monitoring, there are many types of possible sensors with the potential to collect essential raw data.

1.2 Research problem

In this section, we will present different elements necessary to go through in the research process. First we are going to describe the problem clearly, precisely and completely. Next, we will deliver the research questions derived from the problem. Then, we put forward the methodology used for the research. Finally, we introduce the contributions of this work.

1.2.1 The problem

In recent years, there has been an increased interest in the applicability of ambient intelligence to seniors living in rural areas who are slowly losing their autonomy due to aging and chronic disease. By deploying wearable, non-wearable, environmental and context-aware sensors in the environment of an apartment and on the elderly, we will collect physiological data and their activities, which will make it possible to analyze the profiles of these and attempt to anticipate physical and psychological health problems.

To achieve this, we have to find low-cost, non-intrusive equipment to collect the data that we have targeted being key. They must produce data of sufficient quality and quantity. This data coming from different manufacturers has to be integrated into our system as seamlessly as possible. Additionally, the architecture of the system should be secure and efficient in its ingestion and processing of data.

With the lack of software architecture solutions that fit the description above, our main problem tackled in this work is to find out if we can design and develop such a software

system and deploy it in a real environment to validate its effectiveness.

In the end, this system will gather an enormous amounts of data effectively. It is built with components ready to be developed further in order to perform analytics on this data.

1.2.2 Research hypothesis

The idea behind this research is to build a software architecture that accepts many types of sensors to collect physiological, movement and other data that can eventually be used to monitor the health of the elderly during their daily activities in from their immediate environment. The architecture will be able to digest and process data rapidly by using the most recent software optimized for this kind of big data streaming use case. Also, we believe that the deployed sensor technology will be accepted by the elderly due to the fact that it is non-intrusive. Finally, we aim to add an analytics pipeline and architecture on top of the logical software architecture. We believe we will be able to test our architectures by deploying it in a real world setting such a as a small apartment to prove its feasibility and suitability. With this collected data we expect to be able to make preliminary experiments to validate the ability to do analytics with this software.

1.2.3 Methodology

As mentioned in the previous sections, we take into consideration the requirements of the health care system to build our smart health platform. Hence, we take into consideration that there are 4 biomarkers to consider in our delivery of platform: diagnostic, predictive, prognostic and staging. Moreover, we take into consideration the ”P4 Health Continuum” model while developing the platform. The aim of applying the 4 P’s (predictive, preventive, personalized and participatory) is to increase the length of good health. We use ubiquitous computing, which encompasses mobile computing. Ubiquitous computing is a relevant area of technology that could be accepted by the elderly. We intend to develop an architecture that

take into account the following constraints: The sensors were chosen based on price, battery life and ease of deployment to retrieve useful data and openness to have direct access to data. We develop a local architecture to process mobile data and to integrate sensor technologies normally used with their proprietary tools. The remote architecture has to be able to receive the data through a unified, high-speed, low-latency and close to real-time streaming service. We listen to these streams using a scalable and fault-tolerant stream processing engine. Finally, to permanently keep the data using an open source storage layer that provides ACID transactions, scalable metadata management, and unifies streaming and batch data processing.

1.3 Contribution

Several smart health approaches and systems are proposed in the literature, but none of them use non-intrusive and affordable devices, such as a consumer grade and long-lasting battery smart health band in order to retrieve the raw heart rate data directly without having to use the maker's servers. Additionally, we stay focused on affordability, comfort and accessibility by combining a smart band, accelerometer, gyroscope, multipurpose and motion sensors.

To that end, we introduce a new generic architecture to move toward a more accessible senior smart health monitoring system. This architecture takes into account different useful elements such as notifications to monitor rural seniors and ease the pressure on caregivers and relatives. It also combines these elements in a modular environment with the intention to accelerate the integration of ambient intelligence sensors. Subsequently, we implemented this architecture as a Flutter based mobile application.

The platform is meant to be *algorithm-agnostic*, as it will serve as our preliminary architectural work for the next steps of moving toward a fully functional smart health platform.

We follow with the analytics architecture. There exists interesting architectures for both

big data analytics and smart home analytics such as those described in the literature evaluated above but, they only seem to use a few techniques related to very specific use cases with few options and flexibility. To that end, we propose a different four layer modular and flexible analytics architecture that can accept different types of sensors and analytic approaches. The system can be deployed very rapidly in a home of elderly people to monitor their health and ADL's. This architecture is integrated with the logical architecture propositions mentioned earlier. The proposed architecture aims to provide a knowledge extraction environment for health monitoring. We describe each module with useful implementation details.

In addition to the four-layer analytics architecture, we introduce a novel automated algorithm selection process for different computing tasks necessary for a SmartHealth analytics.

To show the effectiveness of proposed architecture and system, we evaluated it by deploying non-intrusive, consumer grade and long-lasting battery sensors in an apartment. The sensors deployed include: SmartHealth bands, IMU's, multipurpose and motion sensors.

Our proposed architectures have resulted with three published articles. These are entitled:

The first, "Towards a non-intrusive and affordable platform for elderly assistance and health monitoring", was presented at the IEEE Computer, Software / Applications Conference (COMPSAC 2020), which was published by the Program Committee as a short paper (up to six pages). The second, "IoT Ambiant Assisted Living: Scalable Analysis Architecture and Flexible Process," was presented at the 10th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2020) Conference, which was published by the Program Committee as a full paper (nine pages). Finally, the most recent paper, "Forecasting Trends in an Ambient Assisted Living Environment Using Deep Learning" was presented to the IEEE Symposium on Computers and Communications (ISCC 2021): Workshop on ICT Solutions for eHealth (ICTS4eHealth).

1.4 Definitions and terminology

In this section, we define some important concepts that will help to understand the rest of this work. We targeted the topics of activities of daily living, signal processing, time series and forecasting.

1.4.1 Activities of daily living (ADL)

Long-term care for elderly people with chronic diseases is definitely taking a toll on the health care system in general, for the families and economically Katz (1983). These people are often sent home but need to be monitored to prevent further cumulative degree of disability. In the mid 1950's, standard useful functions to monitor were identified as being: bathing, dressing, toileting, transfer, continence, and feeding.

1.4.2 Signal processing

In this project, we are often gathering data in the form of a series. When these series are made up of numbers, it can be characterized as a signal. Furthermore if these numbers change in time, it is a time series signal. Often the time series signal represents a voltage that changes in time to represent a certain physical phenomenon. Some common operations to analyze time series signals are: resampling, interpolation, Fourier transform (Nyquist frequency, frequency-domain, bandpass filter) and harmonics.

1.4.3 Time series

Time series analysis and forecasting techniques are relevant in economics, business, and finance Siami-Namini et al. (2018). Often, it is difficult to perform because of the unpredictability in trends and incomplete data knowledge. For example, volatility in the studied

domain is a serious problem in time series forecasting.

As stated in Brockwell and Davis (2016), a time series is a set of observations x_t, each one being recorded at a specific time t. We can differentiate between two types of time series, either a discrete-time or continuous-time time series. A discrete-time time series is a time series where the set T_0 of times at which observations are made is a discrete set. An example of discrete-time time series is one when observations are made at fixed time intervals. On the other hand, and as the name suggests, to have a continuous-time time series, the observations are recorded continuously over some time interval.

1.4.3.1 Components of a Time Series

When looking at a time series, we can find a few noticeably important components that compose it. These include, trend, seasonality, cycle and irregular components Adhikari and Agrawal (2013). Trend is about what is happening to the time series over the long-term. Some example questions that concern trend could include: Is the time series going upwards, downwards or stagnates? For the most part, if there is no external events that trigger a break, the trend can be seen as a monotone function. Another important component of time series analysis is to see if there is a presence of recurring patterns within a regular period (e.g. climate customs or night and day phases). This is termed seasonality. Next, if there is no fixed frequency but nevertheless the time series rises and falls, we label this as cycles. These cycles have variable amplitude and duration. Finally, if we find an unpredictable part in the time series, we label it as an irregular component. Once the other components are withdrawn, the irregular component is equally known as the residual time series.

1.4.3.2 Stationarity

Stationarity is a characteristic crucial to the presence of a time series. That is why statistical forecasting methods usually assume the studied time series is stationary or can

be transformed to a stationary version. A stationary time series is easier to model as well as to forecast because its statistical properties (e.g. mean, variance, auto-correlation) are not changing over time. Although this is true in a perfect world, the real world often presents time series with trends and seasonal patterns. As a consequence, we have to transform, seasonally adjust, make them trend-stationary or difference-stationary.

1.4.4 Forecasting

In order to forecast a window of time in the future based on past observations, a few well-known traditional techniques can be utilized such as univariate Autoregressive (AR), univariate Moving Average (MA), Simple Exponential Smoothing (SES), and the noteworthy Autoregressive Integrated Moving Average (ARIMA) that has numerous variations. Within this group of methods, ARIMA has outperformed its peers in prediction accuracy and precision. Univariate ARIMA is a variation of the popular ARMA model which is a combination of an Auto-Regressive (AR) model and Moving Average (MA) model but where differencing is taken into account. One downfall of these techniques is that they work well on the short-term predictions but lack the ability to perform as well over the long term forecasts Siami-Namini et al. (2019).

New machine learning algorithms and approaches seek to make use of the advancements in computational power by building data-driven models instead of model-driven. Neural networks have made significant progress in recent years. For instance, Convolutional Neural Networks or CNNs have performed extremely well in the image recognition and classification for computer vision field Brownlee (2018). These models extract relevant features automatically instead of having them be handcrafted to solve the problem at hand. ”Convolutional networks combine three architectural ideas to ensure some degree of shift and distortion invariance: local receptive fields, shared weights (or weight replication), and, sometimes, spatial or temporal subsampling.” LeCun et al. (1995). With this ability, the CNN is apt to solve time series forecasting problems by treating the data as a one-dimensional image that

a CNN model. Another successful use case for this model type has been used in classifying human activities based on raw accelerator sensor data Yang et al. (2015).

A more common deep learning approach to time series forecasting is the use of recurrent neural networks such as the Long Short-Term Memory (LSTM). Contrary to the CNN, these take into account the order between observations. This means that the network will add the temporal dimension when mapping the input to the output. In other words, it will automatically learn the temporal dependence appearing in the data. Regularly, LSTMs that use fixed size time windows will be capable of solving tasks that would be otherwise unsolvable by regular feed forward networks Gers et al. (2002). Also, LSTMs are often applied to complex natural language processing problems (e.g. neural machine translation) or audio signals. Ultimately, the CNN and LSTM can be combined to build hybrid models that attempt to benefit from both of its capabilities. Such models include CNN-LSTMs or ConvLSTMs.

1.5 Ethics

A common and recent issue surrounding A.I. in general and more specifically in smart health is to figure out how much machines can assist to make decisions since they have been made almost exclusively by humans in the past Davenport and Kalakota (2019). Some issues often raised are of accountability, transparency, permission and privacy. In recent years, the issue of transparency has been one of the biggest challenges due to the advancements and usage of deep learning algorithms. The reason is that it is almost impossible to explain and interpret exactly how the algorithm reached a certain conclusion. Moreover, it is difficult for these algorithms to be accountable when they have made a mistake. A.I. algorithms can even be subject to algorithmic bias like basing their conclusion on gender or race without being the actual factors for cause of disease.

CHAPTER 2

STATE OF THE ART

In this chapter, we discuss the state of the art in three different sections. First we start off by analyzing the logical and physical software and hardware architectures described in the literature concerning smart health platforms for gathering health-related data. Once we have identified the current state of these types of platforms, we present the related works in the field of analytics architectures. These architectures intend to make use of the flow of collected data from our previous architectures and insert pipelines to perform analytics at the same time. Finally, as a preliminary step toward testing these pipelines, we demonstrate the current state of forecasting time series data. This is a building block to validate of analytics pipelines and move toward actual practical smart health analytics.

2.1 Logical and physical architecture

In order to move toward a smart health sensor-based platform, Cicirelli et al. proposed in to deal with the abstraction and visualization necessary to connect IoT devices. It also proposed an analytics layer in their architecture.

Similarly, a smart healthcare monitoring system (SW-SHMS) was introduced using a three-layer approach using sensors and a gateway such as a smartphone in the user layer Alkhafajiy et al. (2019). The cloud layer represents the database, user information and analytics based on the collected data. Lastly, they have added a monitoring platform that is the access point for health services. The experimental system used an Arduino Uno, a pulse sensor and a Bluetooth dongle.

Alexandru et al. (2019) introduced a new fog computation layer between the data gen-

eration layer composed of wearables, mobile devices and the cloud layer. As mentioned, the fog computation layer is used to help manage the complexity of resource management notably to access, aggregate, pre-process, analyze and filter data before being sent to the cloud.

These architectures also require the recognition of the activities of the seniors. With this in mind, Shoaib et al. (2016) have shown that activity recognition employing accelerometers, gyroscope and linear acceleration sensors from wrist wearable and pocket smartphone sensors can be reinforced in accuracy by joining the two positions readings together during the process of recognition.

In the light of these studies, Filippoupolitis et al. (2017) have proposed an indoor activity recognition framework by placing off-the-shelf wearables that embody tri-axis accelerometer readings in appropriate context using BLE Bluetooth beacons built with a Raspberry Pi. This is the closest study to our proposed solution, but their framework architecture is not described in detail using modularity. It is also not production ready as it is based on a setting with Raspberry Pi as location beacons. Lastly, they do not make use of the pulse of the user in any form.

Likewise, Fiorini et al. (2018) explored if a similar smart health system could be reinforced using vital signs such as ECG signals and 9 axis IMUs. As stated by the author, the devices used in the study were cumbersome and impractical.

Next, Pham et al. (2018) proposed Cloud-Based Smart Home Environment (CoSHE) aimed at putting in context physiological, motion and acoustic data to determine the user activity. The sensors used were Passive Infrared (PIR) sensors, Grid-EYE thermopile array sensors and an OptiTrack Camera system. On the user wearable side, they used an ECG, SpO2 pulse oximeter (blood oxygen concentration), respiration belt, throat microphone, SmartWatch 3D accelerometer and a thigh-worn IMU sensor. The accuracy of the system on the recognition of the task (drinking or not) arrived at a 91.5%.

When creating a smart health system it is important to note that out of 844 articles

analyzed in Granja et al. (2018), costs contributed most to the failure of the projects.

2.2 Analytics architecture

In order to move toward a SmartHealth sensor-based framework that tries to leverage the advantages of pervasive computing and big data, we first have to look to what has been proposed in the literature for generic big data analytics.

Iqbal et al. (2020) proposed a big data modelling methodology that has 6 layers. They present them in order: data input layer, data transformation layer, modelling layer, prediction layer, optimization and application layer.

Although the architecture proposed enumerates many interesting steps of their big data modelling methodology, they do not go into the details of each layer since they focus more on their universal generative modelling approach called Hierarchical Spatial-Temporal State Machine (HSTSM) which is applied to the layered architecture.

Siow et al. (2018) did a whole survey of the IoT and big data analytics that takes into account their utility in creating efficient, effective, and innovative applications and services. They examine the analytics components for health, transport, living, environment and industry domains. The health domain is the most interesting for our tasks. The authors found five main categories of analytics in the literature: descriptive, diagnostic, discovery, predictive, and prescriptive analytics.

Moreover, the paper describes the most current data mining techniques such as multi-dimensional data summary, association & correlation, classification, clustering and pattern discovery. Pattern discovery includes anomaly detection. Most of these techniques are undoubtedly part of the proposed architecture in this paper.

One of the applications of data analytics in healthcare information systems is Ambient Assisted Living (AAL). AAL tries to assist the human to accomplish tasks and move to-

ward better aging through the use of hardware, software and artificial intelligence algorithms Mukherjee et al. (2012).

Hossain and Muhammad (2016) tried to watermark electrocardiogram (ECG) sensed data to ensure integrity. The data is then sent to the cloud for further analysis using machine learning methods in real time.

Hassan et al. (2019) proposes an architecture that is meant for ambient and biomedical sensors to collect the data of an elderly patient. This architecture is the closest work in comparison to our work. Once collected they fuse it into context states with the goal to predict the health status of a patient in real time using context-awareness techniques.

Although the architecture is separated in 4 layers and it aims to put the data in context similar to our architecture, many components are differing in essence or they are not present in one or the other.

Syed et al. (2019) were able to recognize 12 physical activities of elderly people with the accuracy of 97.1%. The elder was wearing wearable sensors placed on the subject's left ankle, right arm, and chest. Their novel framework is made of three layers (the perception layer, the integrated cloud layer and the data analytics layer).

Yassine et al. (2019) presented another platform for IoT analytics from smart home captured data. They place fog nodes between the smart homes and the cloud in order to push the processing resources required from the cloud toward the edge of the network. They present their three-tier layers as the smart home, the fog node and the cloud. The activity recognition, event detection, behavioural and predictive analytics are performed by the fog and cloud computing systems. The results are then reported to the smart home.

Finally, Raghupathi and Raghupathi (2014) had explored the major challenges for the three most promising subjects: image, signals, and genomics based analytics. The signal processing workflow proposed in the corresponding section of the paper is especially inter-

esting. At a high level, a large number of wave form data is ingested at high speed. The data is enriched by adding situational and contextual awareness using the history of the patient or other relevant data. After, they perform nonlinear, linear and multi-domain analysis. This layer acts as a feature extraction and signal processing engine to produce insights. Finally, the last layer takes actions based on these insights, such as best action triggers, alarms, clinical decision support and more. The last layer is designed using the diagnostic, predictive and prescriptive analytics concepts.

2.3 Forecasting

To begin, a lot of recent research has been focused on comparing classical methods of time forecasting to recurrent neural networks such as the LSTM to see which performs best. For instance, Siami-Namini et al. (2018) and Siami-Namini and Namin (2018) try to figure out if and how the newly developed deep learning- based algorithms for forecasting time series data, such as the LSTMs, are superior to the traditional algorithm ARIMA. Furthermore, in a subsequent paper, they add the comparison of the two previous methods to the Bidirectional LSTMs (BiLSTMs) that adds training capabilities by traversing the input data twice instead of once Siami-Namini et al. (2019). In their final analysis, the LSTM model performed better than the ARIMA model and the BiLSTM added layer improved the accuracy of forecasts by 37.78% percent on average.

Given the recent COVID-19 pandemic, many have tried to forecast the number of active cases for a particular country.Papastefanopoulos et al. (2020) have developed 6 different time series methods to forecast the number of active cases by countries using two publicly available datasets. They compared the results and concluded that they could in fact be accurate. They looked seven days ahead and used the performance metric root mean squared error (RMSE) to make their comparison with each method. According to their results, the traditional statistical methods like ARIMA, TBAT, HWAAS performed better than the deep

learning methods such as DeepAR and N-BEATS. They note that this result did not come as a surprise because of the lack of data, which is necessary for deep learning training. Moreover, the Prophet Facebook library did not out perform the classical methods.

Cai et al. (2019) have compared the recurrent neural network (RNN) and convolutional neural networks (CNN) when applied to day-ahead multi-step load forecasting in commercial buildings. The two methods were compared with the Seasonal ARIMAX model. They found that the CNN model performed the best among the proposed approaches by improving the forecasting accuracy by 22.6% to that of the seasonal ARIMAX model.

Fawaz et al. (2019) has explored the state-of-the-art of deep learning algorithms for Time Series Classification (TSC). As a result, they give an overview of which domain has had these approaches be the most successful by performing TSC on the Univariate TSC benchmark (the UCR/UEA archive) and other multivariate time series datasets. They have compared nine architectures such as MLP, Fully Convolutional Neural Networks (FCNs), Residual Network (ResNet), Encoder, Multi-scale Convolutional Neural Network (MCNN), Time Le-Net (t-LeNet), Multi Channel Deep Convolutional Neural Network (MCDCNN), Time Convolutional Neural Network (Time-CNN) and Time Warping Invariant Echo State Network (TWIESN). For univariate datasets, ResNet has outperformed significantly the other approaches. On the other hand, for multivariate time series, the deep CNNs (ResNet, FCN and Encoder) outperformed the other approaches.

Finally, Brownlee (2018) has gone through the process of predicting the future with different types of MLPs, CNNs and LSTMs. He overviews and describes how to develop these architectures for univariate and multivariate forecasting as well as for single-step and multi-step forecasting. The two main domains explored were energy usage for household power consumption and human activity recognition (classification).

CHAPTER 3

SYSTEM COMPONENTS AND DESCRIPTION

In order to retrieve sensory data from a multitude of IoT devices communicating using different protocols, we need to provide the elderly with an easy and non-intrusive system that is flexible enough to gather environmental and physiological data. The data received from these devices is analyzed on the users' smartphone and then sends it to a cloud database. The system described below has the ability to accommodate BLE Bluetooth and ZigBee devices by abstracting the communication protocols. During the development of the architectures included in the system, we take into account scaleability, speed, performance, security, mobility.

The system is built on top of three main pillars. First we describe the physical architecture, then the logical architecture and finally the analytics architecture.

The proposed physical architecture is composed of the following layers: sensor layer, edge layer and cloud layer. To complement this physical architecture, we added the following logical layers: cloud and edge computing layers, cloud and edge analytic layers, cloud and edge alert/notification layers, and finally a sensing/data layer.

An Android mobile application named Gadgetbridge [1], inspired the approach as it proved the possibility to retrieve raw data from many affordable smart bands without the need to use the vendor's servers and applications.

The analytics architecture is introduced in its own section because it is more distinct than the physical and logical architectures.

---

1. Gadgetbridge: https://gadgetbridge.org/

3.1 Physical architecture

The following physical architecture presents three layers, one for each crucial physical system element used to run the system functionalities. Each of the layers' performance will rely on the device capabilities and specifications.

3.1.1 Sensor layer

It is the first layer of our physical system architecture diagram, starting from the bottom in Figure 1. This layer represents the physical and logical/virtual devices that are connected to the system. These devices can either be the data producers or consumers. Producers are the devices that can generate data while the consumers are the devices that can read from the cloud to take actions (actuators). For example, a producer could be a smart watch reading the pulse of a user. On the other hand, a consumer could be an actuator receiving a signal to perform an action, such as putting power ON/OFF or simply triggering an alarm.

In our setting, the sensor layer is composed of environmental and physiological sensors. While the former are fixed sensors placed in the user indoor space, the latter are mobile wearable health and movement sensors.

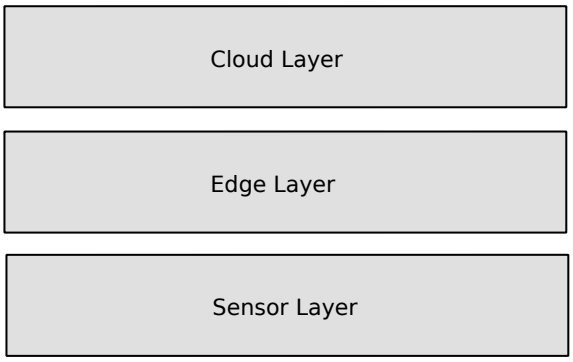

Figure 1: A high-level overview of the physical architecture.

3.1.2 Edge layer

This layer represents the computation device that is the closest to the user. In most cases, the device will be a smartphone. If the user possesses any recent smartphone, with Android or iOS operating system, the edge layer will benefit from a reasonable amount of computational power.

3.1.3 Cloud layer

The cloud layer represents the remote computation servers. On those servers, we find the main database for data persistence and computing power that is required for data analysis. These servers are connected and exchange data with the edge layer through the internet.

3.2 Logical architecture

On the physical architecture, we deploy the logical architecture of the system. It is where we place the layers necessary to implement the systems' high level functionalities and requirements to perform user smart health monitoring.

3.2.1 Sensing/data layer

The first layer of the logical architecture shown in Figure 2, is the sensing/data layer. This layer is placed inside the physical edge layer in order to deal with the reusability and ability to integrate new devices from the physical sensor layer. By adding this layer, we are able to add the necessary modularity that will eventually help us analyze a person's condition. The sensing/data layer is where the different communication protocols are handled. In general, we find that the BLE Bluetooth communication protocol is well suited for mobile applications since smartphones are able to communicate easily with it. Also, many off the

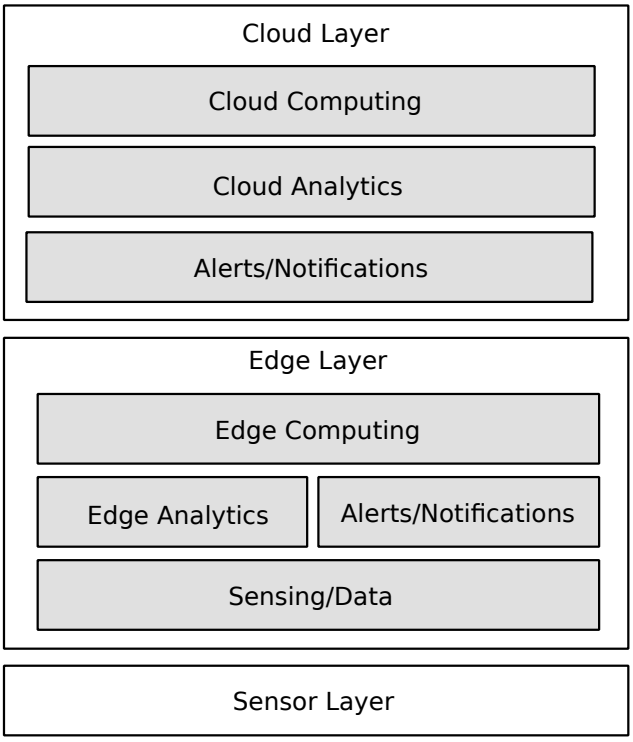

Figure 2: A high-level overview of the logical smart health architecture.

shelf wearables support this protocol. To summarize, the sensing/data layer is the one that is responsible of handling the raw data of the sensors and delivering it to the upper layer.

3.2.2 Edge computing layer

It is essential to our platform to perform as much as possible computation closer to the sensors. With this in mind, we have placed the edge computation layer right above the physical sensor layer and inside the physical edge layer to decrease latency, optimize local data usage as well as cleaning, filtering and collecting the data. The edge computation layer receives the raw data from the sensor layer. At this moment, all the data processes and rules we just described can be applied to it. It would be ideal to store raw data locally and attenuate the data privacy concerns, nevertheless we have to send absolutely essential data that

have meaning on the user state to the cloud. When we speak of a scalable application, the idea of bringing the computation closer to the users releases the computational stress on the cloud computational layer. As described in Shi et al. (2016), ”edge can perform computing offloading, data storage, caching and processing, as well as distribute request and delivery service from cloud to users”.

3.2.2.1 Edge analytics layer

The edge analytics layer is another subcomponent of the edge layer and located and on the same level as the edge computation layer. This component inspects the collected data with the intention to extract knowledge from it. The edge analytics layer accommodates the handling of multiple artificial intelligence algorithms. For example, when newly acquired data is stored, the analytics module can be notified automatically and start to execute simple rules or complex algorithms to predict, classify or analyze that data. This layer is deployed to mobile devices in order to facilitate the analysis of data in real time. Having analytics performed locally instead of the cloud benefits from individual node specific knowledge. However, when the analysis is done, the results can be stored on the cloud computation layer. Some analytics outputs may even be sent to the cloud while keeping a bit more of their privacy since they do not constitute the actual raw readings themselves.

3.2.2.2 Edge alert and notification layer

Additionally, an alert and notification layer is added in the smartphone application architecture. It is on the same level as the edge computation layer and the edge analytics layer. This layer comprises the components that will be used to create, modify and manage alerts. It is what takes care of the local alerts and notifications on the smartphone itself, be it push notifications, sounds and buzzing alarms as well as important messages. Furthermore, having a notification service permits us to handle the reception and publishing of remote alerts. The

external actors such as doctors and loved ones could be using the notification layer by sending useful information or directives to the seniors based on their processed data. Likewise, alerts are also sent to the caregivers to let them know of the senior's state or when the elder needs urgent attention.

3.2.3 Cloud computing layer

This layer is located inside the cloud layer. Our cloud computing layer is focused on pre-processing, storing the data and making it available to the cloud analytics layer where we will make a deeper analysis of the stored data. Again, the stored data received is the health, movement and location data that needs to be persisted. Furthermore, this layer also acts as a data service for the edge computation layer. It allows the edge computation layer to retrieve data and make its own computations when necessary. It is noteworthy that the user personal information such as authentication handling is performed at this layer.

3.2.3.1 Cloud analytics layer

Finally, the cloud analytics layer is a subcomponent of the cloud layer. It is where the cumulation of the data from all the users in the cloud computation layer can be put to use. By having all the data in one place, we can apply the same kinds of artificial intelligence algorithms that we used in the edge analytics layer. With that said, by being on the cloud layer, we can run the computation heavy and complex computations that are not possible on edge devices. Additionally, at this level, analytics models benefit from the aggregation of data from all the users. We may be able to compare output results from edge and cloud analytics as well. At the same time, the cloud analytics models can benefit from locally trained models by using the federation of local models sent over to the cloud layer.

3.2.3.2 Cloud alert and notification layer

Similar to the edge alert and notification layer located on the smartphone, this cloud alert and notification layer is used as a service to convey important information to either caregivers, loved ones or the user themselves. The reason we need a second alert and notification layer is because the cloud computation and analytics layers must also be able to communicate with the exterior based on their own analytics results. For instance, we can imagine having an analytics process that finds anomalies in a user activity or physiological information. This anomaly may fall in with a bunch of other anomalies forming a ”dangerous” pattern. This pattern will be communicated to the cloud layer. The cloud alert and notification layer will handle it by issuing alerts and or notifications.

3.3 Analytics architecture

In this section, we propose a novel and easy to deploy, affordable sensor and analytics system to extract knowledge from an elders' environment by applying the ”P4 Health Continuum” and move toward a more proactive healthcare of individuals. We detail each pillar necessary with the relevant modules to preform the task at hand. Additionally, we introduce a novel automated algorithm selection process for different computing tasks necessary for a SmartHealth analytics.

In order to perform health analytics for the elderly population, we need to design an architecture that is modular enough to be able to process many types of data such as signals, numerical, environmental, and contextual data. Each of them demands different techniques that are often described in big data analytics surveys Siow et al. (2018). Our architecture shown in figure 3 accepts these as inputs in the first layer. This layer is also known as the sensor layer. Next, this plain data received from each sensor is coming through at a fast rate and in enormous amounts. It is way too expensive and costly to take the sensor data and pass it directly to our knowledge extraction algorithms. Thus, we go through a data preprocessing

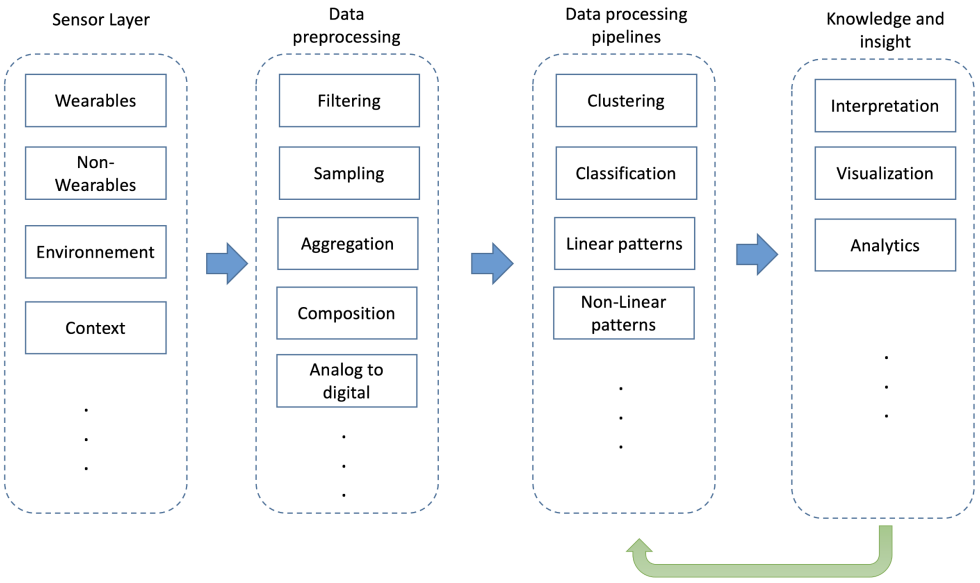

Figure 3: Data analytics workflow and architecture proposed and separated into four layers.

layer, which uses many techniques to reduce the amount of unnecessary data while keeping the most significant one. The layer is also parametrized to help get the type of data that you need and at which rate. The third layer accommodates the data processing pipelines available to extract information from the preprocessed data. Our last layer is the knowledge and insight layer. We use different modules to give the user the ability to make smarter decisions. We introduce modules for interpretation, prediction, visualization and analytics. It is important to note that the fourth and last layer is closely connected with the third layer because the knowledge and insights collected can be returned back through the data processing pipelines to retrieve even more insights or create a continuous learning cycle.

3.3.1 Layer 1: sensing

The first layer to implement in our architecture is the sensor layer which is used to capture the raw data from the sensors deployed in the elders' environment. These sensors can be of many types. We identify at least four types: wearables, non-wearable, environmental,

and contextual. These sensor types are not exclusive to only one category. They can belong into more than one type at a time.

3.3.1.1 Wearable

The wearable component is encapsulating the sensors that may be placed on the body of the monitored individual. Some examples include PPG and ECG smart bands that retrieve physiological information.

Recently, such smart bands have been successful in many studies. One very famous example is the Apple Heart Study (AHS) Turakhia et al. (2019). Another available information gathering sensor through wearables is the IMU that retrieves motion data. This inertial measurement unit is an umbrella sensor that often gathers data from accelerometer, gyroscope and magnetometer sensors. Finally, we can find the blood oxygen concentration (SpO2). These sensors are introduced in more and more innovative and affordable wearables like smart clothing, belts, rings and even shoes.

3.3.1.2 Non-wearable

Non-wearable sensors are the sensors that can retrieve information on the individuals with or without physical contact. Such sensor category includes 2D cameras, lidar cameras, smart scales, mattress sleep assessment sensors, localization beacons, etc. They are interesting sensors to include because of the added value brought without having to be worn at all times. These sensors were not included in our experimentations because they are considered intrusive to the privacy of the individual.

3.3.1.3 Environmental

It is beneficial for a health system to make use of environmental sensors because they can be placed and used without being intrusive. Such sensors usually need a way to communicate with a node in order to make use of what is sensed. We find that we can place ambient temperature, contact, tilt, vibration and audio sensors in the environment to gather its information and state.

3.3.1.4 Context-aware

The idea of context-aware sensors is that we are able to retrieve spatio-temporal information from them. We can find these when there are spatial and temporal dependencies between readings. For instance, a wearable accelerometer sensor allows us to get the spatial movement of the device often associated with a timestamp that determines when the reading was sampled. As we can see here, an accelerometer falls into both the wearable and context-aware categories.

3.3.2 Layer 2: data preprocessing

The second layer of the architecture is the data preprocessing layer. With a vast amount of incoming data, it is essential to build an efficient first data collection funnel. One reason is because the data collected comes in at a fast rate and with redundancies that can be eliminated if appropriate filtering techniques are used. This funnel helps to reduce the amount of processing necessary to extract knowledge in the upper layers. Finally, each modular component can get its own parameterization in order to produce a set of meaningful and versatile metrics to experiment with.

3.3.2.1 Filtering

The filtering component is used to maximize the amount of useful data and minimize the redundancies in incoming data. It is also used to remove noise in the data Donaghy (2017). Filtering is a form of lossy compression, which means that it reduces the size of the data, while erasing some of it at the same time. For instance, the Fourier transform is a popular mathematical transformation to filter out different frequencies that compose a signal. Following the Fourier transform rules permits us to recreate any signal by the sum of its sinusoids. This frequency information is crucial to extract the most important information in the signals. Another useful filter for the accelerometer is the bandpass filter.

3.3.2.2 Sampling

Digital sampling aims at listening to an analog continuous-time signal, quantize and discretize it in time, then store it in memory. An important concept to take into account when sampling is known as the Nyquist frequency. The Nyquist frequency is half of the given sampling rate.

3.3.2.3 Aggregation

In order to minimize the size of data storage necessary, aggregation can be used to take a table of data and highlight useful statistics from it. These statistics, such as mean, standard deviation and others, can then be used to perform analytics in the upper layers of our architecture Donaghy (2017).

3.3.2.4 Composition

In our architecture, the data is coming from numerous sensors. Often, it also has different data types coming from the same sensor. As a result, we can combine these using composition to produce new augmented data. A simple example of composition is seen when we combine temperature and humidity to obtain the humidex index using a chart.

3.3.2.5 Interpolation

Interpolation is a strategy that allows us to estimate what should be between data points when the signal is already sampled. Normalizing signals that are sampled unevenly is a useful way to utilize interpolation. Additionally, it can be used to compare two signals sampled at different rates. You can interpolate a signal to make use of another sampling rate.

3.3.2.6 Resampling

If we have a certain discrete signal, resampling means that we need to change the sampling rate for that signal. In other words we are changing the frequency of the observations. As we are generally gathering real world data, it may come in at different time intervals and will need this resampling to be mapped to uniform time intervals. In digital signal processing, we are either upsampling or downsampling our set of observations. We often use interpolation to perform upsampling. To downsample, summary statistics can be used to aggregate the data.

3.3.2.7 Analog-to-digital

An analog-to-digital converter (ADC) is used when an analog continuous-time signal needs to be encoded to a discrete digital numbers format. The analog signal is often a voltage.

This voltage at a specific time can be transformed to a digital output code. Some common analog signals include: temperature, pressure, liquid levels, forces and light intensity.

3.3.3 Layer 3 : data processing pipelines

This layer is focused on taking the preprocessed data from the previous layer and use it to recognize important patterns or classifying the data such as clustering, classification, linear and non-linear patterns, etc.

3.3.3.1 Clustering

The goal of clustering is to group the data collected into classes without any labels attached to these groups. Clustering techniques are unsupervised, which means that there is no known true labels for the model training. Moreover, we do not know how many clusters will be formed from the input data. Wong (2015) defines it formally as ”given a set of data instances, a data clustering method is expected to divide the set of data instances into the subsets which maximize the intra-subset similarity and inter-subset dissimilarity, where a similarity measure is defined beforehand”. They also present us with a survey of the different paradigms used in the field. Among them are partitional, hierarchical, density-based, grid-based, correlation, spectral, gravitational, herd, and other clustering paradigms.

3.3.3.2 Classification

When instances of data have a corresponding true class label with them, we are now into the field of supervised learning Kotsiantis et al. (2006). Classification algorithms constitute one category of supervised learning. The other one being regression algorithms. For classification problems, we can only classify outputs as unordered discrete values. We can split the

classification algorithms into a few subcategories: logic based algorithms, perceptron-based techniques, statistical learning algorithms and support vector machines.

3.3.3.3 Linear-patterns

The linear patterns are used to discover patterns that evolve linearly. For instance, a continuous linear trend in increasing or decreasing values will appear as a straight line across the data. Under this type of mining, the most common and simple method is linear regression Maulud and Abdulazeez (2020). Regression can either be used for forecasting prediction problems, causal relations between dependent and independent variables or between a fixed dataset collection of different variables and a dependent variable.

3.3.3.4 Non-linear patterns

When the independent variables are not showing a linear pattern, we can try to fit the data to a complex nonlinear function. This means that the best-fit line is not a straight line. Often, this technique is referred as curve fitting.

3.3.4 Layer 4: knowledge and insight

For our fourth layer, we find a layer that is aimed toward making use of all the information we have accumulated so far. At this level, we can present the data to a user, interpret it, predict future outcomes and perform analytics.

3.3.4.1 Interpretation

In the interpretation component of the architecture, we define rules or patterns that are meant to characterize the data and give it a signification that is in relation to the domain of

the data. The question that we are asking is: What does the information retrieved actually mean in a specific domain?

3.3.4.2 Visualization

The visualization component is aiming to provide a graphical view and presentation of the data. It is helpful to create these to let decision makers understand complex ideas and observe new structures or patterns. Letting the user interact with the visualizations helps to grasp more detail out of the underlying data Ajibade and Adediran (2016). The ultimate goal here is to show information and let the viewer try to extract knowledge out of it. To name a few of these methods as examples, we can use: histograms, line charts, tables, pie charts, bar charts, scatter plots, bubble plots, area charts, flow charts, Venn diagrams, data flow diagrams, time lines, multiple data series, entity relationship diagrams, cone trees, semantic networks, tree maps and parallel coordinates, etc.

3.3.4.3 Analytics

The analytics in this layer play a major role in attempting to gain knowledge from the information. We mentioned and described the five types of analytics in the state of the art chapter. These were: descriptive, diagnostic, discovery, predictive, and prescriptive analytics Siow et al. (2018). As an example, predictive analytics is about making use of current and historical data, apply statistics and modelling methods to it, and anticipate what the future may become. Predictive analytics is closely related but distinct from many quantitative approaches: statistics, forecasting, optimization, discrete event simulation, applied probability, data mining and analytical mathematical modelling Waller and Fawcett (2013).

3.3.5 Automated algorithm selection

In the previous sections, we have described a four layered analytics architecture with the goal to achieve SmartHealth monitoring of elderly people. With the large amount of data coming in at a fast rate, the complexity of knowing which components to use and when becomes increasingly difficult to figure out. For this reason, we are proposing a flow process to help with the selection of appropriate algorithms and data combination in order to get optimal results from our architecture to achieve a given task.

The flow proposed and displayed in figure 4 begins on the left side with four inputs sent to the gear symbol, which represents the Matching Engine component of the process. Starting from the top of the three inputs represented in blue, the profiles of algorithms component is acting as a repository of metadata, information, algorithms and tools available and pre-assembled. Secondly, the specification of tasks component is a standardized way of defining tasks that are accessible to achieve. Third, the data profiles component represents a standardized description of the data. Ultimately, in orange, we find the task at hand that we would like to perform. It is marked in orange because it is a dynamic input to our matching engine. With the three repositories accessible and the current task at hand, the matching engine has the purpose of automatically selecting a list of candidate algorithms. Once the list of candidate algorithms is produced, another flow component handles the selection and validation of the final algorithm. At this level, an expert (human, program or hybrid) could make the final choice of the algorithm. The selected algorithm is going to be treating the input data from our previous architecture, to produce our results.

In this chapter, we presented a physical, logical and analytics architecture that is able to easily and rapidly integrate different ambient intelligence sensors. These architectures are made up of many layers, each of them with their specific purpose to ultimately make up a system which is going to be used for deploying the sensors in a real environment, gather environmental and physiological data of individuals living in their homes and perform analytics

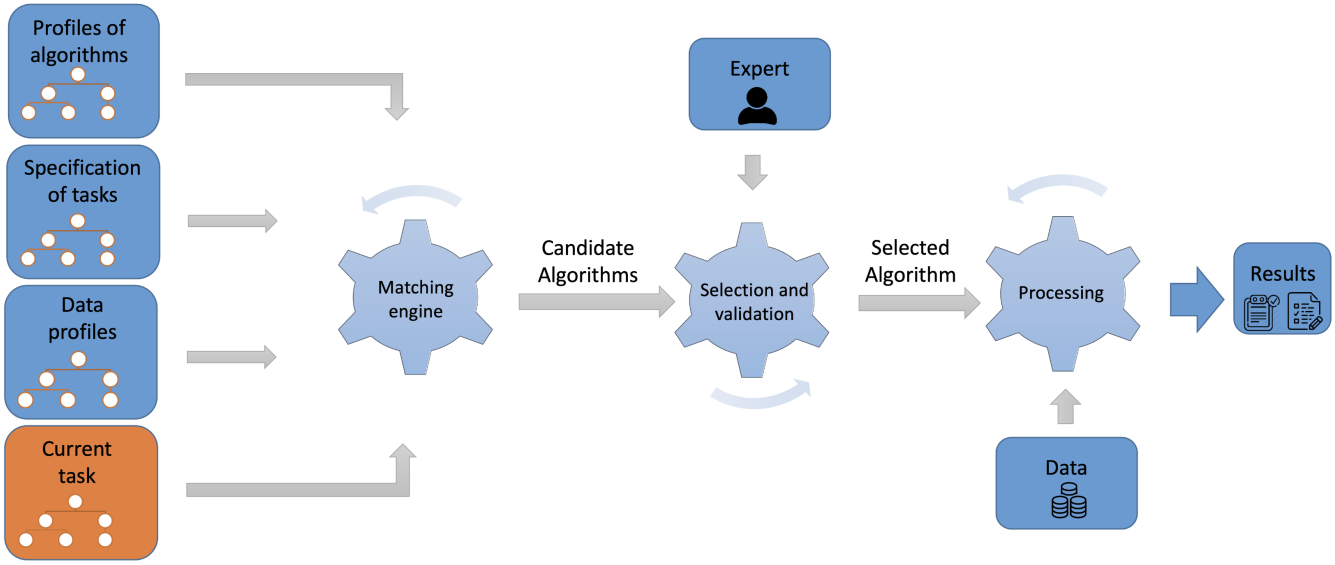

Figure 4: Process of algorithm selection for a specific task.

on the data. In order to use the system, the specific modules that compose each architecture still need to be implemented in a concrete manner.

CHAPTER 4

IMPLEMENTATION AND DEPLOYMENT

Up to this point, we have described which important modules compose the physical, logical and analytics architectures of a larger health oriented data collection system. These were presented theoretically in order to move toward implementing it concretely. In other words, it is crucial to deploy this architecture in the real world in order to prove its effectiveness. To that end, we implemented the main modules of the proposed architectures. We started by finding the sensors and figure how they communicate the data. This data is subsequently passed through our architectures before it is stored permanently.

When looking for sensors to recognize the ADL's of elderly people, we look for sensors that belong and conform to the sensing layer described earlier. Additionally, we made the choice to integrate sensors that are not intrusive, affordable and accessible. The reason is that, for the platform to be easily and quickly deployed in their environment, off-the-shelf products are a perfect fit by being ready to use and optimized for commercial use Gingras et al. (2020a).

4.1 Considered data

We have identified interesting data that could give us important information on the daily activities of the individual being studied and analyzed.

Motion detection: We would like to know when the individual is within which zone or region of the smart home. This would help us identify and reduce the range of possible activities that are being performed at that specific moment and also to deduce how active the individual is.

Object interaction detection: We want to know when the individual interacts with the objects in his surroundings, by having some sort of sensor that is triggered when the interaction with that important object we will have a better understanding of what is actually happening in a zone in the apartment. Putting these sensors on important appliances is crucial here.

Body movement reading: Additionally, the user body movement can indicate what the user is actually doing. To get this information, we are going to go with accelerometer and gyroscope sensors attached to the body of the individual.

Heart rate reading: With the latest advancements in smart wearable devices, we can measure heart rate using the integrated PPG or ECG sensors. The heart rate sensor can help in the recognition because a resting cardiac rhythm is simply different than one while participating in an activity.

Lighting state change: By having smart lights deployed in the apartment, we can get the daily, weekly, and monthly usage of the lights to detect trends and anomalies. On another note, their states at any specific moment can help to identify if a user is arriving or leaving a zone.

Ambient temperature reading: When placing ambient temperature sensors in the environment of the individual, we will find that if a user is in a room with appliances running, we will detect these changes and understand better what the activity actually is being performed. For example, the temperature may change when watching the tv, starting the dishwasher, cooking with the oven turned on, doing the laundry, taking a shower, etc.

Weather report querying: By getting the daily weather values and summary, we may want to understand how it affects the mood or the activities of the individual and seek for correlation between weather conditions and indoor activities.

Outdoor localization: Furthermore, in order to put the raw data from wearable sensors

in its context, it is interesting to see how we can locate the position of the individual in their environment and recognize the activity they are performing. Outdoor human localization is easily achievable using the global positioning system (GPS).

4.2 Tasks

Following the gathering of all this data, we may want to eventually use it in a supervised learning manner by doing activity detection and classification. To ensure we can perform supervised learning, we want to allow the user to label what they are doing at any specific time.

4.3 Technologies

In this section, we introduce several of the different protocols used by the devices deployed in our system. Each device is presented separately. If applicable, we give their associated operating systems. Next, we present the frameworks and our data persistence choice.

4.3.1 Protocols

In our system, the many devices deployed have to communicate with each other. In order to communicate, we have utilized Ethernet, Bluetooth Low Energy (BLE), Zigbee and Wi-FI. The following sections give an overview for each of them.

4.3.1.1 Ethernet

Partially described in the IEEE 802.3 specifications, ethernet has a data transmission of around 100 Mbps Poellabauer (2020). It is a popular, inexpensive and an easy to install LAN

architecture solution.

4.3.1.2 Bluetooth Low Energy (BLE)

The Bluetooth Low Energy (BLE) has lower power consumption and lower cost than Bluetooth classic Poellabauer (2020). It is an open short-range radio technology optimized for ultra-low power consumption. It is used in many domains and use cases, including health-care. It uses 40 channels to communicate. It is capable of reducing the power consumption by transferring short packets, has fewer RF channels to have a better discovery and connection time and finally it is a simple protocol. A Health Device Profile was approved in 2008 that aimed at developing a standard for existing and emerging medical devices.

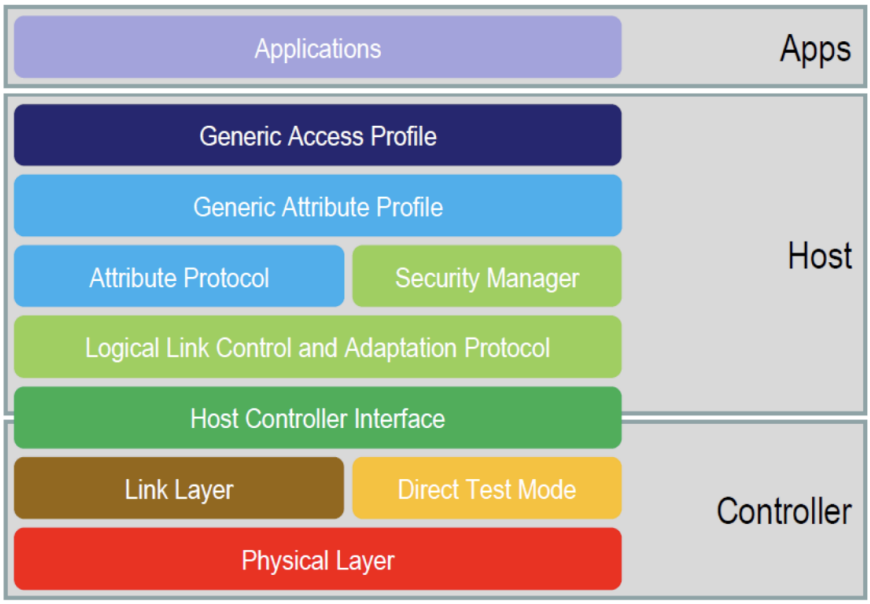

Figure 5: BLE Architecture and Protocol Stack

4.3.1.3 Zigbee

ZigBee-style networks began around 1998. IEEE 802.15.4 was first completed in 2003 Poellabauer (2020). The ZigBee Alliance was established in 2002. It was pushed and backed by many well-renowned companies such as Honeywell, Mitsubishi, Motorola, Philips, Samsung, etc. It is now supported by more than 260 members. It aims for low power consumption, simple design and low cost. The ZigBee protocol is a simpler but similar protocol to Bluetooth Ergen (2004). It is different in that it has a lower data rate and is snoozing most of the time. As a result, it can often be deployed for months to years without having to change the battery. Moreover, it benefits from having a longer range over Bluetooth by operating at approximately 10 to 75 meters. Another advantage is to allow the use of up to 254 nodes. Finally, the network allows flexibility by supporting many types of topologies: star topology, peer-to-peer topology and cluster tree. An example of this kind of mesh network is shown in Figure 6.

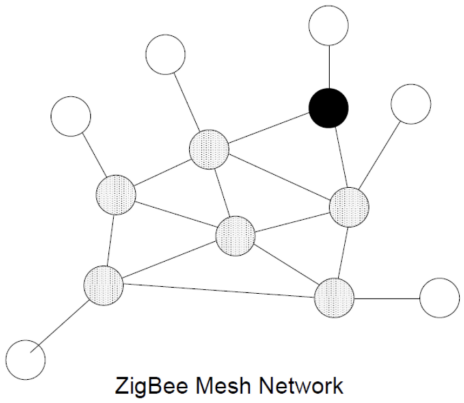

Figure 6: ZigBee mesh network.

4.3.1.4 Wi-FI

Based on IEEE 802.11 family of standards, it uses Wireless Local Area Network (WLAN) technology (often used synonymously with Wi-Fi) Poellabauer (2020). It usually uses either the 2.4 and 5 GHz bands. The 802.11 is primarily concerned with the lower layers of the OSI model. Data Link Layer: Logical Link Control (LLC), Medium Access Control (MAC). Physical Layer: Physical Layer Convergence Procedure, Physical Medium Dependent (PMD).

4.3.2 Hardware

When looking for sensors to recognize the ADL's of elderly people, we look for sensors that belong and conform to the sensing layer described earlier. Additionally, it is beneficial to look for sensors that are not intrusive, affordable and accessible. The reason is that, for the platform to be easily and quickly deployed in their environment, off-the-shelf products are a perfect fit by being ready to use and optimized for commercial use Gingras et al. (2020a).

4.3.2.1 Raspberry Pi

Many of our environmental sensors communicate via Zigbee. For that reason, we connected them to a hub such as a Raspberry Pi with a ZigBee USB Gateway running the Home Assistant software. A Raspberry Pi B+ is shown in Figure 7.

4.3.2.2 Cellphone

It is used to keep a connection with the Bluetooth devices. It is also used to retrieve GPS coordinates when outdoor.

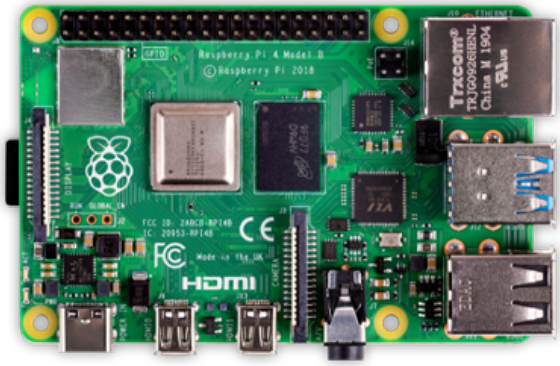

Figure 7: Raspberry Pi B+.

Additionally, in the category of contextual sensing, we can use the mobile application of the user to know if the user is home or not. We can even use the mobile application GPS module to get data about where they are exactly and at which speed they are travelling. This spatiotemporal information has the potential to put other data sources in a better context. The Motorola Moto G7 Play mobile phone that was used in our experiments is shown in Figure 8.

4.3.2.3 MetaMotionC (MMC)

In the wearables category, we added a MetaMotionC (MMC) from the MbientLab company. The MMC has many useful sensors included in the device such as: 6-axis accelerometer and gyroscope, 3-axis magnetometer, ambient temperature, barometer, pressure, altimeter and ambient light. It is attached to the chest of the senior using a small clip. Using Bluetooth, we can stream raw sensor data at up to 100 Hz. The MMC used in our experiment is shown in Figure 9 and it is attached to the chest of the person as shown in Figure 10.

Figure 8: Motorola Moto G7 Play - 32GB.

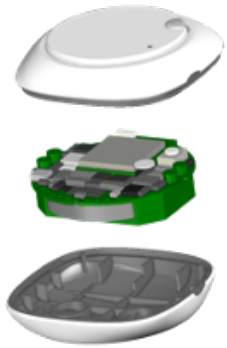

Figure 9: Accelerometer and Gyroscope - MetaMotionC (MMC).

4.3.2.4 Mi Band

For the wearables component of the sensing layer, the first wearable item we picked is the Mi Band shown is Figure 11. The Mi Band was a good fit for our application at a weight

Figure 10: MetaMotionC (MMC) placed on the chest.

of 22.1 g and width of 18mm [1]. It stands at a modest approximated $50 USD. Moreover, the band is advantageous because it benefits from a battery life of up to 20 days. A long battery life is necessary in an application for elderly people because they could forget easily to charge their devices overnight. It has a 3-axis accelerometer, 3-axis gyroscope, PPG heart rate sensor and capacitive proximity sensor. The smart bracelet needed for our application needed to be small in size and weight with BLE Bluetooth communication capabilities. It also needed to be reasonably cheap to increase the chance of large-scale adoption. Ultimately, the main piece of the smart band that collects data can be removed from the band itself so it may be placed at other positions on the body of a person for potentially more accurate activity recognition.

1. Mi Band 4 specs: https://www.mi.com/global/mi-smart-band-4/specs

Figure 11: Mi Band 3.

4.3.2.5 Apple Watch

Similarly, another wearable technology was used on the wrist of the user as an alternative during the data collection period. As shown is Figure 12, an Apple watch series 4 was connected through Bluetooth to an iPhone XS Max. This watch is much more expensive at around 300$. Essentially, it is a 44mm square attached with a velcro strip. It has a built-in rechargeable lithium-ion battery that lasts up to 18 hours. Just like the Mi Band, the watch retrieves heart rate data.

Figure 12: Apple Watch (Series 4).

4.3.2.6 Samsung SmartThings

We have found that the ZigBee sensors are very interesting by being easily accessible on the market and relatively cheap as well. We can deploy them in the vicinity of important areas in the apartment of the user and close to the appliances we want to would like to track. We decided to connect the popular Samsung SmartThings environmental sensors using Zigbee and a hub such as a Raspberry Pi with a ZigBee USB Gateway running the Home Assistant software. Each of them emits changes in ambient temperature which can be useful when placed near appliances that generate heat. We deployed three types of Samsung SmartThings environmental sensors: multipurpose, motion and water leak.

4.3.2.7 Multipurpose

The multipurpose sensor shown in Figure 13, emits changes in vibration, tilt and contact. The contact sensor uses a magnet to detect if a separate piece is close by. For example, we can place the main piece of the sensor on the door, close to one of its edges, then the smaller part of the magnet right next to it, but on the wall where it won't be able to move.

Figure 13: Multipurpose.

4.3.2.8 Motion

We also deployed motion sensors like the one shown in Figure 14. Most commonly known as a passive infrared (PIR) motion sensor, this sensor uses body heat (infrared energy)

to know if there is someone in the vicinity. The sensor is looking for changes in temperatures. With adhesive strips and the magnet ball, we can mount the sensor on a wall and angle it toward the region we need to monitor.

Figure 14: Motion.

4.3.2.9 Water leak

The water leak sensor shown in Figure 15, detects water that touches its moisture sensor. It has 2 moisture sensors on the bottom and on the top of the device.

Figure 15: Water Leak.

4.3.2.10 Body weight scale

For non-wearable sensors, we have added a Fitbit Aria 2 smart scale like the one shown in Figure 16. It connects to the Wi-Fi of the apartment. We can easily retrieve data from the Fitbit database using their api. This scale is able to measure our weight and digitally sends the data to the cloud associated with the linked account of the scale.

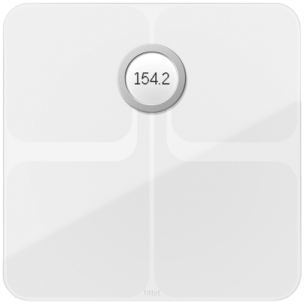

Figure 16: Aria 2.

4.3.3 Operating systems

Our system uses many operating systems because it needs to be flexible to the user needs. Moreover, we picked some of them due to the benefits and advantages of using them.

The Android and iOS operating systems allow us to install mobile applications on smartphones to enhance user experience, add the ability to retrieve valuable information from its built-in sensors and communicate with other sensors via the Bluetooth Low Energy (BLE) module. Moreover, the watchOS was used on the Apple Watch, but no apple watch application was developed for this project. We ran the iOS 14, watchOS 7 and Android 10.

A laptop running Ubuntu 20.04 was installed in the middle of the apartment to connect the MetaMotionC (MMC) using Bluetooth. This could eventually be removed by having a different configuration on the Raspberry Pi.

Indeed, a Raspberry Pi was placed in the apartment of the user with the Home Assistant Operating System installed on it. The Home Assistant OS is an open source home automation that puts local control and privacy first[2]. This operating system helps us track all the states

2. Home Assistant: https://www.home-assistant.io/

of devices connected to it via many protocols and plugins. Another key point is that it allows us to create rules and automations to either control our smart home or share the events by sending the state changes to a cloud server.

4.3.4 Frameworks

In the following few sections, we present each framework that we have used during our implementation. We begin by presenting Flutter, which was used for the development of a mobile application. Next a REST API was developed using the Springboot framework. Then we present briefly Apache Kafka, Apache Spark, Spark Structured Streaming and Delta Lake.

4.3.4.1 Flutter

To develop on iOS and Android, we used Google's Flutter which is an open-source framework "for crafting beautiful, natively compiled applications for mobile, web, and desktop from a single code base." [3]. The UI toolkit is optimized for 2D mobile applications and uses the object-oriented language Dart to program its logic. Flutter is definitely a distinguishable development platform to fast programming for versatile mobile applications. A few crucial details were necessary for us to go in its direction. First we need a framework that runs tasks in the background. Dart code can be run in both iOS and Android background processes using what is named as an *isolate.* [4]. Secondly, it is interesting to us that we can

3. Flutter: https://flutter.dev/docs/resources/faq

4. Background processes: https://flutter.dev/docs/development/packages-and-plugins/background-processes

run platform specific code when necessary using a ”flexible message passing style”[5].

4.3.4.2 Springboot

For over 10 years, the Spring framework has been the go to for developing Java applications Walls (2016). Using this framework adds a bit of magic by giving us an automatic configuration for many types of functionalities we actually seek for this a project like ours. Spring gives us starter dependencies for these functionalities. Finally, the actuator helps us understand the internal workings of running our application.

For this project, we used Spring to develop a web application with a controller class that will receive and respond to HTTP requests. Another reason for picking this configuration is that by adding the Kafka Spring dependency, we can interact with the Kafka core APIs. We developed many common types of classes to accept the raw data coming in from the HTTP requests. We obviously have the main RestController, then we receive the data using data transfer object classes, we transfer the object to a domain object using a mapper. Once the data is transferred to the domain model objects, we can produce an event message to the appropriate Kafka topic. This event message is formatted as a string with its values separated by commas.

4.3.4.3 Apache Kafka

Apache Kafka was created to have a mechanism that gives access to information coming from multiple sources and would be needed in other applications that act as receivers Garg (2013). Essentially, it gives a way to give a real-time connection of messages between applications. The data creators are named producers and those who listen to this data are named

5. Writing platform-specific code: https://flutter.dev/docs/development/platform-integration/platform-channels

consumers. In other words, Apache Kafka is an open source, distributed publish-subscribe messaging system. Some of its characteristics include: persistent messaging, high throughput, distributed, multiple client support and being real time. Likewise, we can describe it in short by being a distributed, partitioned and replicated commit log service. Figure 17 presents the basic architecture of Kafka.

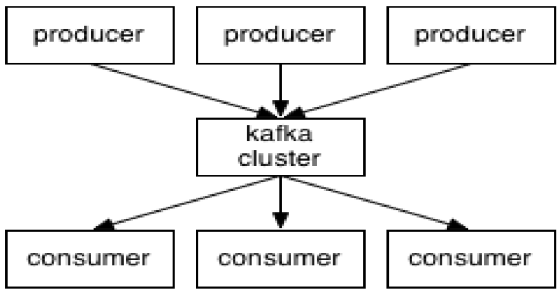

Figure 17: Basic architecture of Kafka

In our case, our topics created include: accelerometer, activity, geolocation, gyroscope, heartrate, isHome, light, motion, multipurpose, placemark, waterleak, weather, weight.

4.3.4.4 Apache Spark and Spark Structured Streaming

”Apache Spark is a unified engine designed for large-scale distributed data processing, on premises in data centers or in the cloud. Spark provides in-memory storage for intermediate computations, making it much faster than Hadoop MapReduce” Karau et al. (2020). As shown in Figure 19, it incorporates libraries with composable APIs for: Machine learning (MLlib), SQL for interactive queries (Spark SQL), stream processing (Structured Streaming) and Graph processing (GraphX). For our implementation and since our use case is essentially a continuous stream of data, we used the Spark Structured Streaming model and APIs built on top of the regular Spark SQL engine and DataFrame-based APIs. Using this model, the stream is represented as a continually growing table, with new rows of data appended at the end.

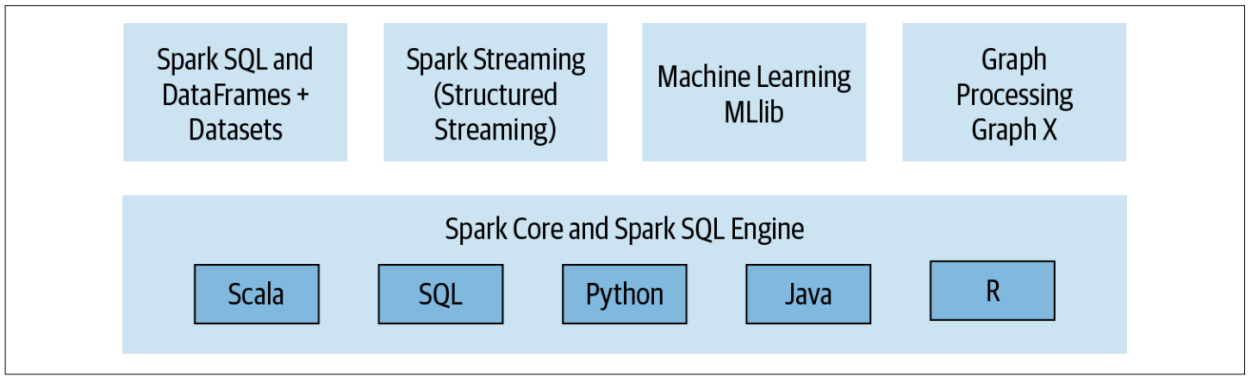

Figure 18: Apache Spark components and API stack

4.3.5 Delta Lake

Some of the most cost-effective cloud object stores include Amazon S3, but because of the way they use key-value stores it is difficult to perform high-performance ACID transactions Armbrust et al. (2020). Delta Lake uses the Apache Parquet format to compact a transaction log that provides those ACID properties and fast metadata operations for large tabular datasets. Using the previously mentioned Apache Spark, we can access these large tables. In our system, we have a single table for each of our Kafka topics to store them permanently.

4.4 Pipeline

The full data gathering and processing pipeline deployed in the apartment of the user is shown visually in Figure 19. First on the left-hand side of the diagram, we can see that the many different types of environmental sensors that use the ZigBee protocol communicate with the Raspberry Pi. The accelerometer and gyroscope sensors in the MMC device communicate the readings via Bluetooth to the laptop installed in apartment vicinity. The mobile application and the resulting data and metadata from it are stored locally on the SQFlite data

database. The firebase module of the mobile application is only used for metadata and login information for the moment. Eventually it could disappear and be treated within the application database and server itself. Therefore these edge devices have to then send their readings and payloads to the distant REST service running on our server. This service can then produce messages to our Apache Kafka instance running on the same server. The messages produced are assigned to a topic for each specific type of data coming in. These messages stay in the Apache Kafka instance for a total of 7 days. At the same time an Apache Spark Structured Streaming instance is listening (consuming) to these topics. At this step, we can take advantage of the various processing modules of Apache Spark Structured Streaming. Not to mention that it is where we insert the data in Delta Lake tables for long-term storage.

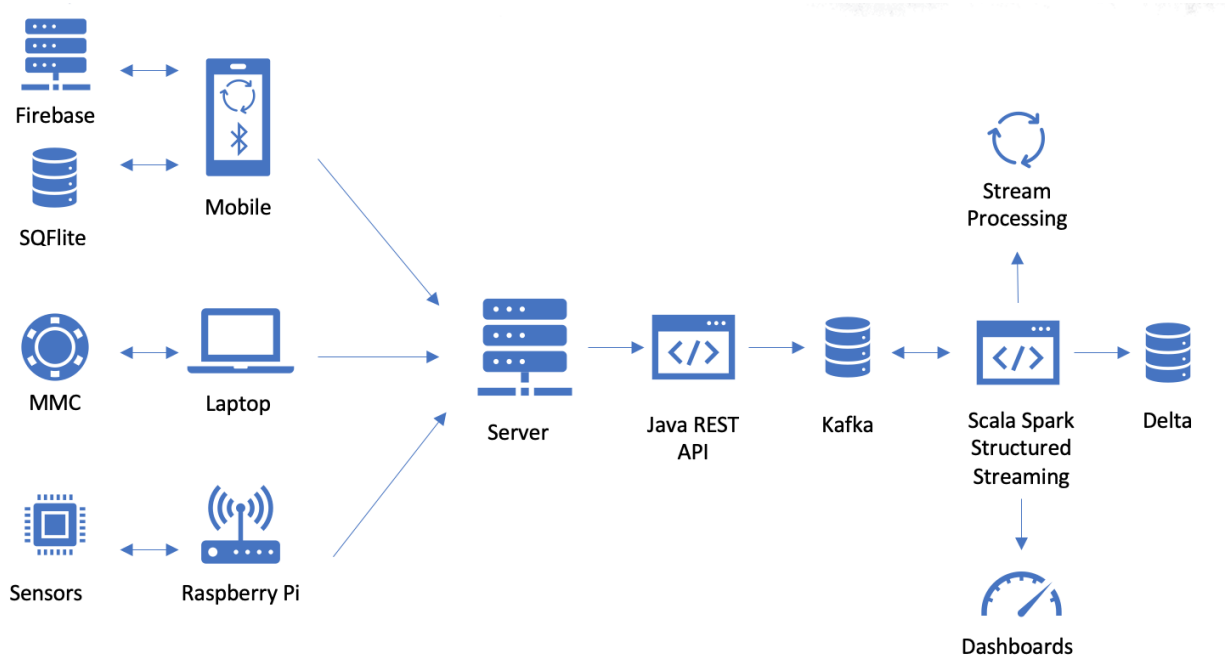

Figure 19: Full data flow and pipelines.

4.5 Data

In this section, we will explore the different data possible to capture with our sensors. The data comes from a 3-axis accelerometer, gyroscope, apple watch, motion and multipurpose sensors. We are going to describe the kind of data and its fields coming into our software architecture directly from the sensor layer.

4.5.1 Abstraction

Much of the incoming data has to have some type of similar data. Every payload coming from each sensor must be assigned to a specific user so it is not lost with the data from other users. For this reason, each payload comes with an ”ownerId” field that is a unique identifier for the user assigned when the user is created. Equally important, in our Apache Spark Structured Streaming Application, we assign a unique identifier for every row that is inserted in the Delta Lake tables. Next, for every payload, there is some sort of timestamp. These are not coming as a unified format because not all sensor manufacturers use the same format. We briefly mentioned above that we deployed the architecture in a real world setting inside a small apartment. When describing the apartment, we virtually separated the area to 8 regions. These regions include a kitchen, bathroom, bedroom, living room, dining area, office, entrance and balcony. In our code implementation, these regions are named locations.

4.5.2 Payloads

Activity: When the user begins an activity by pressing on the start activity button of the mobile application, the timestamp ”dateFrom” is initialized with the current time. Also, the field ”activity” is assigned the name of the activity that has been started. When the user ends its currently active task (activity), it initializes the ”dateTo” field. In summary, this means we have the time ”from” and ”to” that a user has performed a specific activity.

The activities that were taken into consideration were: Arriving, Leaving, Sleeping, Relaxing, Working, Stretching legs, Peeing, Pooping, Showering, Getting a drink, Eating, Cooking.

Accelerometer: The accelerometer data payloads coming in from our MetaMotionC (MMC) hold the X, Y and Z decimal gravity values.

Gyroscope: The Gyroscope data payloads coming in from our MetaMotionC (MMC) hold the X, Y and Z decimal angular speed values.

Geolocation: We can use the mobile application GPS module to get data about where they are exactly and at which speed they are travelling. The approximate accuracy of the measurements for the other fields are available in meters. Additionally, we find the latitude, longitude, speed, speed accuracy and heading.

IsHome: In the category of contextual sensing, we can use the mobile application of the user to know if the user is home or not. We are able to get this information based on the geolocation data.

Heart rate: The heart rate received from the watch or band is a simple integer value.

Light: The important value coming from the light payload is the state of the light. The state is simply a boolean value that gives us if the light has been turned on or off at a specific time.

Motion: The motion data payload includes the ambient temperature decimal value and a boolean value that indicates if the sensor has captured movement activity or not in the last 15 seconds.

Multipurpose: The multipurpose data payload includes the ambient temperature decimal value and a boolean value that indicates if the magnet sensor is open and if it is sensing vibration.

Table 3: Accelerometer data (sample data)

Epoch (ms)	X-Axis (g)	Y-Axis (g)	Z-Axis (g)
1533078186498	-0.311	-0.192	0.936

Table 4: Gyroscope data (sample data)

Epoch (ms)	X-Axis (°/s)	Y-Axis (°/s)	Z-Axis (°/s)
1533078186408	0.427	2.134	-0.366

Waterleak: The waterleak sensor payload includes the ambient temperature decimal value and a boolean value that indicates if the water sensor is sensing water at the moment.

Placemark: The placemark data payload is based on the geolocation data as well. The mobile application module gets the address from the latitude and longitude values.

Weather: The weather payload includes.

Weight: The weight received from the wight scale is a simple integer value.

Table 5: Heart rate data (sample data)

Timestamp (UTC)	Value
2020-11-09T17:55:55.956	75

Table 6: Motion data (sample data)

Timestamp (UTC)	Location	Temperature	Motion
2020-10-31 12:21:50.583435-04:00	office	22.31	true

Table 7: Multipurpose data (sample data)

Timestamp (UTC)	Location	Temperature	Vibration	Contact
2020-11-07 15:56:26.535037-05:00	office	23.6	true	true

4.5.3 Sensor triggers

To receive the changes in states from the Zigbee sensors, we can set up rules to send HTTP Post requests to our web server inside the automation and configuration YAML files in the Home Assistant settings.

4.6 Apartment

We briefly mentioned above that we deployed the architecture in a real world setting inside a small apartment. When describing the apartment, we virtually separated the area to 8 regions. These regions include a kitchen, bathroom, bedroom, living room, dining area, office, entrance and balcony.

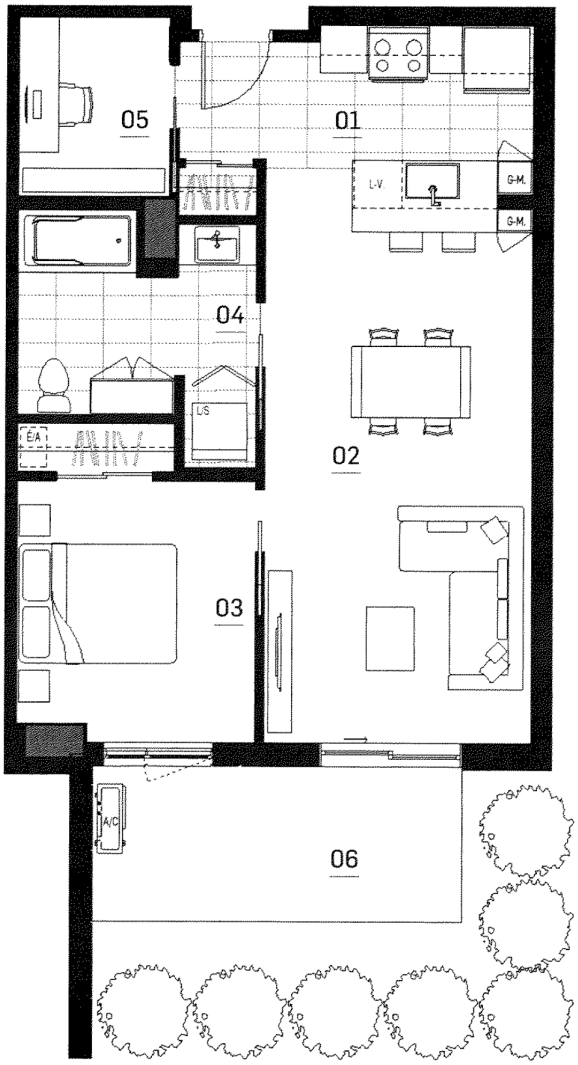

Figure 20: The apartment where the prototype was deployed.

CHAPTER 5

ANALYTICS EXPERIMENTS

In the previous chapters we have created a platform that gathers large amounts of data by deploying different types of sensors in the living environment of an individual as well as creating the pipeline to process that data. This pipeline is augmented by adding an analytics architecture and pipeline on top of it. We tested the data gathering capabilities of the platform by implementing and deploying it in a real environment setting. In this chapter, we explore and validate the analytics section of the platform by using this data for a few preliminary forecasting experiments. We identify the problem, methods, processing steps and compare the results with a forecasting baseline model.

5.1 Forecasting model development process

The following steps from Brownlee (2018) present a process to have a ”good enough” forecast model that may not be at the pinnacle of the possible accuracy, but still a good 80% to 90% of the optimal model that could be created for a specific problem.

5.1.1 Problem definition

The first phase of the forecasting model development process is the most challenging de la Combé (2019). In this phase, we need to define the problem at hand thoroughly. By accurately defining the purpose of the forecast, we clarify who it is for and how the forecasting process fit within the bigger picture, system or organization.

Brownlee (2018) presents many questions to answer during the problem definition phase to orient the next steps of the forecasting model development process in an informed

direction.

Inputs vs. outputs: When predicting the future, we most often use past observations as inputs to our model and try to forecast one or more probable future observations. The past observations are the historical data consumed to perform a forecast. Here, we are not describing the data for the purpose of training the model yet. Instead, we use this data to perform a single forecast (e.g. the last three days to predict the next day). This step is basically to get you to identify the variables you need to get together to produce your forecasts.

Endogenous vs. exogenous: We try to go deeper into the analysis of the input data by examining the relationship between the input variable and the output variable. An endogenous variable is one that is influenced by other variables in the system and the output variable is dependable on it as well. On the other hand, an exogenous variable is one where it is not influenced by other ones in the system. Most often we will use endogenous variables because the output is based on prior timesteps. Exogenous variables are often ignored because we focus a lot on the time series aspects of the forecast.

Regression vs. classification: A regression problem is one where the value we try to predict is a quantity. In other words, we try to forecast a numerical value. Some practical examples of regression problems can be to try to predict a price, count, volume, etc. On the opposite side, classification tries to predict a category. More accurately, a category is a label from a small group of explicit and precise labels. Although these two are often seen separately, we can reformulate a regression problem as a classification problem and vice-versa.

Unstructured vs. structured: It may be possible to find patterns in a time series by examining the plot it may generate visually. If there is no obvious systematic time-dependent pattern in a time series variable, we denote it as unstructured variable. Comparatively, if trends or seasonal cycles are obviously present in the time series we can define it as structured.

Univariate vs. multivariate: A univariate time series means that we are measuring a

single variable over time. On the opposite, a multivariate time series is simply that we are measuring multiple variables over time. These two characteristics can be assigned to either the inputs or the outputs separately. For example, you can have univariate or multivariate inputs and univariate or multivariate outputs.

Single-step vs. multi-step: Here we differentiate if we want to predict the next time step using a one-step forecast model or if we want to predict more than one time-step using a multi-step forecast model. One important challenge to note is that the more time steps in the future we would like to project, the uncertainty of each forecasted time-step increases because the compounding effect it has.

Static vs. dynamic: When a model is developed and reused to make predictions without being updated or changed between forecasts, this model is characterized as being static. Versus, by having new observations before making the next forecasts, we may want to create or update a new model. This is known as being a dynamic forecasting problem.

Contiguous vs. discontiguous: We define a time series where the observations are made uniformly as contiguous. For instance, they could be made at a specific time such as each hour, day month or year. If they are not uniformly taken, we describe it as discontiguous.

5.2 Deep learning methods

In this section, we present a small overview of the two main deep learning methods that were used in the experimental study. We begin with the Convolutional Neural Networks (CNN) followed by the Long Short-Term Memory (LSTM).

5.2.1 CNN

The Convolutional Neural Networks (CNN) is a network that can learn to extract features from the raw data instead of from handcrafted features Brownlee (2018). ”Convolutional networks combine three architectural ideas to ensure some degree of shift and distortion invariance: local receptive fields, shared weights (or weight replication), and, sometimes, spatial or temporal subsampling” LeCun et al. (1995). The CNN uses different processing units to give an effective representation of the most important local data, then the stacked deep architecture can characterize the most important representations at different scales.

The CNN is often made of two architectural parts Zhao et al. (2017). The first part of the architecture are often the convolution and pooling operations. As mentioned earlier, they characterize deep features of the data. These operations are used alternatively. The second part is taking these features as input to a multilayer perceptron (MLP) to ultimately make the predictions. A common CNN architecture often has five different layer types: input layer, convolutional layer, pooling layer, feature layer, output layer. Figure 26 shows the architecture of the CNN [1].

— Input layer: It has $N \times k$ neurons. The number of variate time series is defined by k and the light of each univariate time series by N.
— Convolutional layer: This layer is where the convolutional operations are performed. It does this by calculating the scalar product of the weights and the region connected to the input volume O’Shea and Nash (2015). This layer utilizes the kernel parameter which spreads along the depth of the input. As result of convolving each filter on the input, it ultimately outputs a 2D activation map.
— Pooling layer: The goal of this operation is to down sample the output of the convolutional operations.

1. Figure inspired from: Python For Finance Cookbook published: January 31st, 2020. By Eryk Lewinson.

— Feature layer: At this point, the original input is characterized as a series of feature maps. To recreate a new long-time series we can connect these features maps together in the feature layer.

— Output layer: In the context of predicting numeric values, the number of output neurons is the number of days we want to predict in the future. The feature layer and output layers are fully connected.

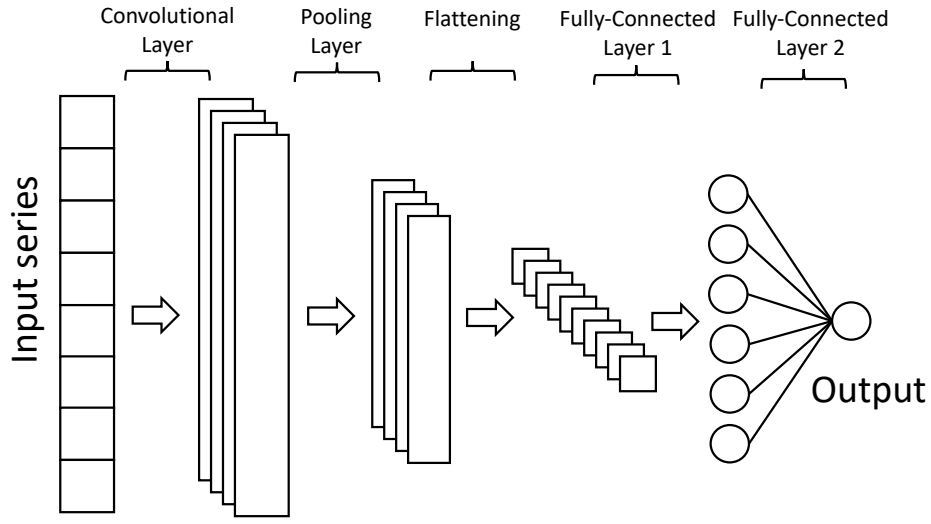

Figure 21: CNN architecture.

5.2.2 LSTM

The Long Short-Term Memory (LSTM) based models have been developed because the regular Recurrent Neural Networks (RNN) lacks in performance when it tries to learn long-term dependencies in the data. This problem is also known as the vanishing gradient problem. The LSTM is able to keep these long-term dependencies by allowing its memory to be modified over time during its training. The memory is a line of cells that transport the data. Inside these cells, many gates allow for certain operations such as reading, writing and deleting the memory of each cell. The final treated output is fed to the next cell of the network.

A LSTM usually has three types of gates to perform these operations of letting through or not data through the cell Siami-Namini et al. (2018). Figure 22 shows the architecture of the traditional LSTM cell [2].

— Forget gate: A gate based on a *sigmoid* function that outputs a number between 0 and 1. A value of 0 means to completely forget the learned value. On the other hand, a value of 1 means to completely retain the value.
— Input gate: This gate has two layers, the *sigmoid* layer and the *tanh* layer. The first is used to decide which value to change. The second creates a vector of possible values to be kept in memory.
— Output gate: As the name suggests, it is the gateway to what comes out of the cell based on the cell state and the other treated values of the state.

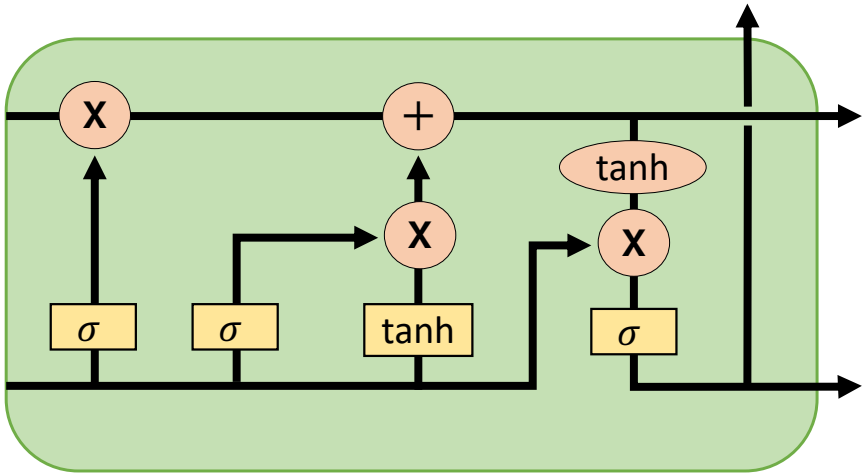

Figure 22: LSTM architecture.

2. Figure inspired from: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

5.3 An experimental study

In this section, we go through the machine learning process that was used to get our analytics experiment results. In our process, we first begin by giving an overview of the dataset, then we split this dataset into a train and test set. Next, we present our chosen assessment metric. Finally, we compare a forecasting performance baseline model with the ARIMA, CNN and LSTM models.

5.3.1 Dataset

The health-related data was collected by deploying non-intrusive sensors in a small apartment in Québec City, CA. When describing the apartment, we have separated it into 8 regions Gingras et al. (2020b). These regions include a kitchen, bathroom, bedroom, living room, dining area, office, entrance and balcony.

The dataset has been collected by deploying ZigBee sensors which are very interesting by being easily accessible on the market and relatively cheap as well. We deployed them in the vicinity of important areas in the apartment of the user and close to the appliances we want to would like to track. We connected the popular Samsung SmartThings environmental sensors using Zigbee and a hub such as a Raspberry Pi with a ZigBee USB Gateway running the Home Assistant software. We deployed two types of Samsung SmartThings environmental sensors: multipurpose and motion.

The multipurpose sensor emits changes in vibration, tilt and contact. The contact sensor uses a magnet to detect if a separate piece is close by. For example, we can place the main piece of the sensor on the door, close to one of its edges, then a smaller part with a magnet is placed on a fixed part of the appliance or door. The multipurpose data payload includes the ambient temperature decimal value and a boolean value that indicates if the magnet sensor is open and if it is sensing vibration or tilt.

Most commonly known as a passive infrared (PIR) motion sensor, the sensor uses body heat (infrared energy) to know if there is someone in the vicinity. The sensor is looking for changes in temperatures. With adhesive strips and the magnet ball, we can mount the sensor on a wall and angle it toward the region we need to monitor. The motion data payload includes the ambient temperature decimal value and a boolean value that indicates if the sensor has captured movement activity or not in the last 15 seconds.

A wearable technology was used on the wrist of the individual living in the apartment to collect the heart rate as a simple integer value. An Apple watch series 4 was connected through Bluetooth to an iPhone XS Max. We use its Photoplethysmography sensor to retrieve the heart rate.

The dataset contains three separate CSV files, one for each of the sensor types discussed previously. The heart rate file contains 114606 readings, the motion file 193712 readings and the multipurpose file 41702 readings. Combined the files amount to 104.73 MB of data. Table 8 describes the size of the dataset file and what the starting and end dates were for both the original dataset and the preprocessed one. Moreover, table 9 presents the number of readings collected for each type of sensor.

Table 8: Description of the health and activity oriented dataset

Dataset size	104.73 MB
Start date	November 4th 2020
End date	May 9th 2021
Start date (after preprocessing)	November 8th 2020
End date (after preprocessing	May 8th 2021

5.3.2 Data preparation

Since we receive the data from different sources we need to standardize the timestamps for each reading as well as removing the metadata that is not necessary for our use case (e.g. id, ownerId, uuid). Moreover, we have transformed the boolean columns as an integer representing 0 and 1. As a result of the question we are trying to answer, we will only keep the readings for motion and multipurpose where it was read to the boolean value true (1). Since we know the placement of the sensors, there are two exceptions to this processing. The multipurpose placed on the trash will be open very often because of the excess of trash coming out of it. As a result, we will only use the vibration readings for this sensor. Moreover, the multipurpose sensor placed on the toilet pipe is constantly closed because it is on top of the metal pipe vibrating when it has been flushed. We will only keep the vibration for this sensor as well. The next step is to resample our data to a daily sum for the motion and multipurpose data.

The Apple Watch on the wrist of the individual allows for workouts to be recorded. This results in the number of heart rate readings to be increased during the workouts. To mitigate this problem, we resampled the heart rate data as a mean per hour. Then, we can resample it as a daily mean.

As in many datasets, when batteries of the sensors had to be changed and the watch had to be recharged, data was not recorded. This means that there is missing data that can be filled using the forward and backward fill techniques. After all, we can join the different data reading types together to form a new cleaned dataset.

We calculated the sum of all observations for each day and created a new data set that represents the general daily activity for each of the variables. The last step is to calculate the sum of all the activity readings for each day and only keep that value for each day.

Table 9: Number of readings collected by the sensors

Motion data	193712 readings
Multipurpose data	41702 readings
Heart rate data	114606 readings

5.3.3 Training and test data

Since we try to forecast the week ahead, we will trim the data to begin a week on Sunday and end on Saturday. Therefore, the data has been trimmed slightly to begin on November 8, 2020 and end on May 8, 2021. This gives us a total of 26 full weeks which the first 17 weeks will be used for training (train set) and the last 9 for evaluation (test set). Table 10 shows the separation of data for the training and test sets from the cleaned full dataset.

Table 10: Separation of data for the training and test sets from the cleaned full dataset

Total weeks	26
Training set (weeks)	17
Test set (weeks)	9

5.3.4 Assessment metrics

The root mean squared error (RMSE) is going to be calculated to assess the difference between our model predictions and the true values. In other words, RMSE calculates the square root of the average of the squared differences between the forecasts and the actual values Brownlee (2018). In the scenario of forecasting, the RMSE metric is more punishing than the mean absolute error (MAE). The formula is described in the equation 5.1.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{d_i - f_i}{\sigma_i}\right)^2} \tag{5.1}$$

We will evaluate our models using walk-forward validation. This type of validation is where a model makes a prediction for one week, then the real data for that week is going to be available to the model to use for its next prediction. This gives the model the ability to use the best data available and to be trained the in a similar way to what it will predict in practice.

5.3.5 Forecasting performance baseline

To compare our deep learning models, we have built a forecast performance baseline. We can create a point of comparison using naive forecasting strategies. If a model has fewer error rates than the chosen simple forecasting strategy, we can say that this model is more skilful Brownlee (2018). We have chosen to use a weekly persistent forecast as the baseline for comparison. To do this, we are going to forecast the upcoming standard week based on the prior standard week by assuming that the next week is going to be similar. In the figure 25, we see that an overall RMSE of 686 readings. We can also see that Mondays, Tuesdays and Fridays seem to be the easiest to predict.

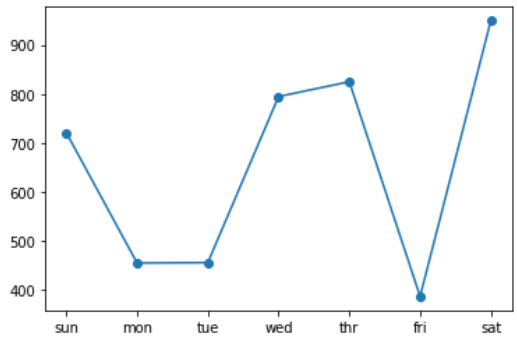

Figure 23: Weekly RMSE scores for the weekly baseline forecast strategy. Overall RMSE: 686 readings.

5.3.6 ARIMA

In this section, we present the development of our ARIMA model for multi-step forecasting. We begin by making a statistical autocorrelation analysis. We assume that the variable representing the daily activity is distributed following a Gaussian (bell curve) distribution. By assuming this, it allows us to use the Pearson's correlation coefficient. We plot the autocorrelation by using the AutoCorrelation Function (ACF) and a partial autocorrelation function (PACF).

5.3.6.1 Autocorrelation Analysis

In the ACF plot, we can see that there is a strong autocorrelation, meanwhile the PACF plot shows that this component is distinct for approximately the first five to seven observations. We can argue that a value of seven lag observations as input would be a fair starting point to build our model.

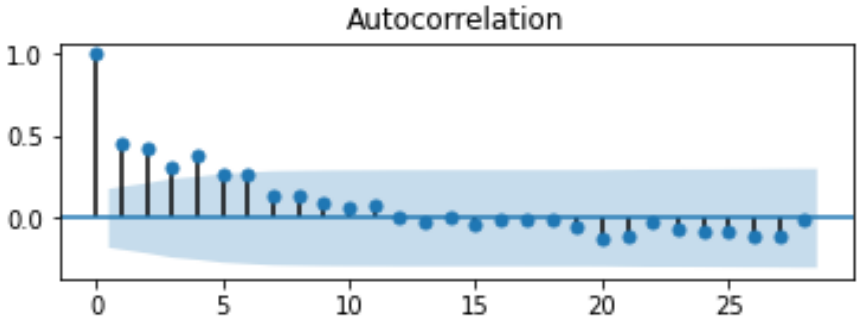

Figure 24: Plot of the AutoCorrelation Function (ACF).

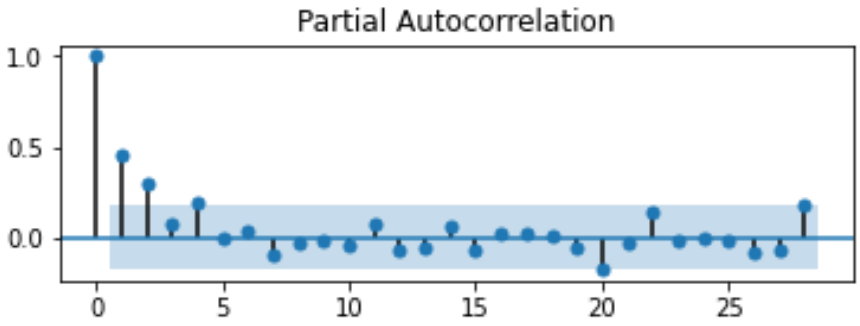

Figure 25: Plot of the Partial AutoCorrelation Function (PACF).

5.3.6.2 Univariate ARIMA Model

Using the Statsmodels library will allow us to develop an autoregression model for our daily activity data. This library provides multiple models (AR, ARMA, ARIMA and SARIMAX). We will select the ARIMA model with a constructor input of three parameters. The final model will be ARIMA(7,0,0).

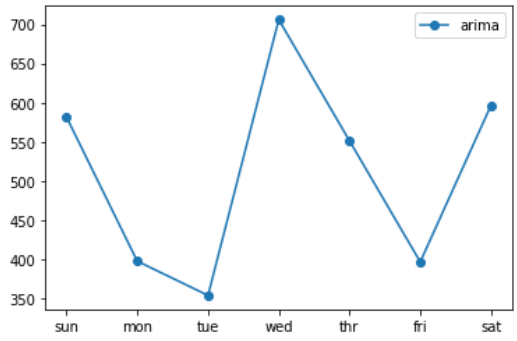

Figure 26: Weekly RMSE scores for the weekly ARIMA forecast. Overall RMSE: 526 readings.

5.3.7 Univariate CNN model

In this section we present the development of our univariate CNN model. We are going to use seven prior days as the one-dimensional (1D) subsequence as input to the model. Our 1D CNN has to receive the data with the shape of : [samples, timesteps, features]. With the training dataset, this means that we will have 17 samples made up of seven timesteps with one feature each. Under those circumstances, it is evident that 17 samples is not a huge amount of data for a neural network to train properly and use the benefits of deep learning. To mitigate the issue, we have modified the problem during the training phase to predict the next seven days regardless of the starting day being a Sunday. This means that the data for the training phase will be modified by having overlapping windows, moving along one timestep. On the other hand, the problem will stay as trying to forecast the next week based on the previous one.

The model we built has to be small because we do not have a large amount of data. As a foundation, the model was inspired by the work of Brownlee (2018). We use the Keras sequential API to create our model layer-by-layer. The first layer is a convolutional layer with 16 filters and a kernel size of 3. In other words, a window of three timesteps is created to read the seven days as input and performs the convolution using this window. The filter parameter defines how many times the convolution will be performed. Next, the objective of the next layer is to reduce the size of the feature maps by 14. Once that is done, we flatten the data as a single long vector. A dense layer of 10 neurons follows to interpret the vector. Finally, another dense layer is added with seven neurons to make the prediction for the next week. The model is fit for 20 epochs and a batch size of 4 using the Adam implementation of stochastic gradient descent. In the figure 27, we see that an overall RMSE of 549 readings.

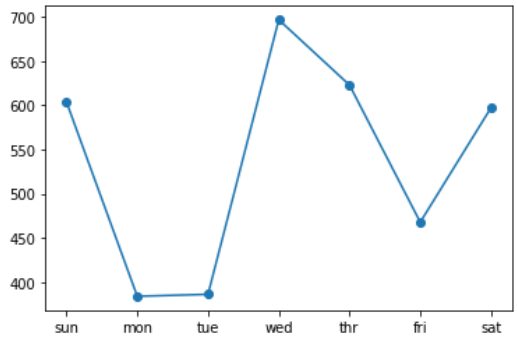

Figure 27: Weekly RMSE scores for the weekly CNN forecast. Overall RMSE: 549 readings.

5.3.8 Univariate LSTM model

In this section we are building a Vanilla LSTM model. This model has to take in the seven prior days and outputs a prediction for the next 7 days. The input one-dimensional subsequence is read through and tries to extract features. Just like the CNN, our LSTM has to receive the data with the shape of : [samples, timesteps, features]. The number of samples is still not a lot to train a neural network, so we will use the same overlapping window technique as with the CNN.

The model will be built layer-by-lay using the Keras sequential API. The first layer takes in the input vector with a single hidden LSTM layer with 200 units. Next, the features coming from the LSTM layer are interpreted by a dense layer with 200 nodes. Finally, we end the network by adding a dense layer with a unit for each day of the week we try to predict. The model is fit for 70 epochs and a batch size of 16 using the Adam implementation of stochastic gradient descent. In figure 27, we see that an overall RMSE of 562 readings.

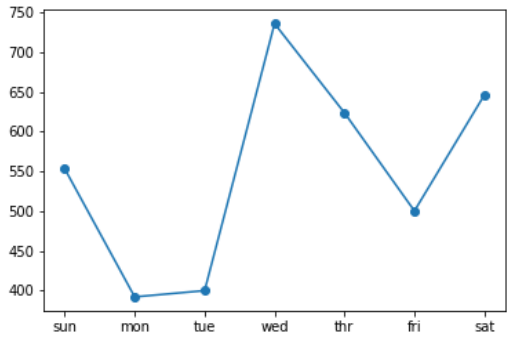

Figure 28: Weekly RMSE scores for the weekly LSTM forecast. Overall RMSE: 562 readings.

5.4 Results

The overall RMSE and each day of the week for each method of forecasting on the daily general activity dataset are reported in table 11. The Naïve forecasting baseline has a significantly higher RMSE score, which means that the two deep learning forecasting techniques have learned to extract important features from the input sequences. Although the CNN performs better than the LSTM, it is only a slight 2.6% difference. On the other hand, the CNN performed 20% better than the naïve forecasting baseline strategy. Also, the LSTM performed better than the Naïve model with an increase in accuracy of around 18%.

Table 11: Comparison of RMSE scores between methods

Models	**RMSE per day of the week**							**Overall**
	S	***M***	***T***	***W***	***T***	***F***	***S***	
Naïve	720	454	454	794	824	385	949	**686**
CNN	603	384	386	696	622	467	597	**549**
LSTM	553	391	399	736	623	500	646	**562**

5.5 Discussion

The results show that the deep learning methods such as our one-dimensional CNN and LSTM perform significantly better than the forecasting performance baseline. Although it has proven to learn and extract features by increasing the accuracy of the week-ahead predictions, we can expect that the two deep learning models would perform considerably better if we had given it more data to train with. The fact that only 26 weeks of data was given for the complete dataset is relatively small to train a deep neural network. In this case, the traditional forecasting techniques may perform better since they are usually better with smaller amounts of data. Moreover, we only trained the model once. The deep learning algorithms are stochastic in nature so it would be interesting to fit the model many times to see the differences in learning capabilities. As can be seen in figure 27 and 28 there are similar results for both the CNN and LSTM forecasts as Mondays and Tuesdays seem to be the easiest to forecast. On the other hand, the Wednesday is, generally speaking, the hardest day to predict accurately. The results for the baseline forecasting strategy vary in that the Fridays seem to be the easiest and the Thursdays and Fridays the hardest to forecast.

5.6 Conclusion and future work

In this paper, we present a comparison of two deep learning forecasting models for univariate inputs and multi-step outputs trained on a new health-related dataset collected by deploying non-intrusive sensors in a small apartment in Québec City, CA. The dataset is comprised of 26 weeks of data collected from popular ZigBee sensors which are very interesting by being easily accessible on the market and relatively cheap as well. We deployed them in the vicinity of important areas in the apartment of the user and close to the appliances we want to would like to the activity. Also, it includes the heart rate of the individual living in the apartment. In order to perform health analytics for the elderly population, we followed our past research by deploying and collecting our dataset through our designed analytics

architecture that is modular enough to be able to process many types of data such as signals, numerical, environmental, and contextual data.

The new dataset was used to train our developed one-dimensional CNN and LSTM models. They were evaluated using the metric root mean squared error, walk-forward validation and compared with a forecasting baseline strategy. The CNN and LSTM were approximately equal in performance at a minimal 2.6% difference in accuracy, but both performed well comparatively to the forecasting baseline strategy with an increase of 20% and 18% in accuracy respectively. The results show that there are some days that are easier than others to forecast. The models developed could potentially increase in accuracy due to the relatively small number of weeks of data collected.

In this work, we ingest properly the raw data coming from the sensors, but looking forward we want to find out what types of knowledge we are looking for in the detection of health problems. Also, we will have to find more artificial intelligence algorithms that have already proven themselves to help us analyze the data. If the algorithms do not exist already, we may need to create our own algorithms. These algorithms shall often take into account the temporal element because most of the data is gathered as a time series.

For each of these cases and for each data input to our system, we will have to find the feature extraction techniques that best fit this type of sensed data. Furthermore, we have to find the appropriate testing and validation techniques for these techniques. At the same time, the process must allow for the analysis of data locally for each individual as well as the aggregation of data for all other users because we would like to create individual and generic behaviour profiles. Moreover, the gathered data will serve analytics and processing modules so that they may help to prevent and detect physical or psychological health issues as well as to recognize activities, changes in activities and anomalies in those activities. These different types of experiments will be performed in a pilot project including eight to ten rural seniors living in their own homes.

GENERAL CONCLUSION

In this work, we first began by presenting a multi-layer architecture and platform that could incorporate health and location sensorial readings in order to monitor rural seniors' activities and health. The platform is geared to incorporate non-intrusive, low-cost, long-lasting, accessible and comfortable sensors to increase the probability of large-scale adoption by seniors for further research and smart health applicability.

Next, we proposed a modular four layer architecture to perform SmartHealth analytics using the different components needed when working with multiple types of data sources. Firstly, the input data is received through out sensing layer. The subsequent layers represent data preprocessing, data processing pipelines and knowledge and insight layers.

Furthermore, we presented an automated algorithm selection process to pick the best algorithm for a specific task at hand. To evaluate the effectiveness of our approach, we deployed a set of sensors as a preliminary version of an implementation of the architecture in a single apartment in the region of Québec, Canada. Our architecture was validated by gathering large quantities of data and was able to be used to train a classical ARIMA model, and two deep learning models such as a CNN and LSTM models. These models performed better than our baseline forecasting model fore one week ahead forecasting. The sensors deployed vary in type and the architecture is a base to build upon for future works. To further solidify and validate the methods proposed, we plan to continue implementing more of the architecture, such as adding more filtering methods, apply more machine learning and complete the full cycle of life of data.

Finally, we still need to apply and implement our automated selection process and deploy the solution in a pilot project in the region of Quebec, Canada. The objective is to monitor between eight and ten rural seniors for health issues and their ADL's. The ethics certificate has already been obtained by our collaborator Dr. Clémence Dallaire at the Université Laval.

REFERENCES

Adhikari, R., Agrawal, R.K., 2013. An introductory study on time series modeling and forecasting. arXiv preprint arXiv:1302.6613 .

Ajibade, S.S., Adediran, A., 2016. An overview of big data visualization techniques in data mining. International Journal of Computer Science and Information Technology Research 4, 105–113.

Al-khafajiy, M., Baker, T., Chalmers, C., Asim, M., Kolivand, H., Fahim, M., Waraich, A., 2019. Remote health monitoring of elderly through wearable sensors. Multimedia Tools and Applications 78, 24681–24706.

Alexandru, A., Coardos, D., Tudora, E., 2019. Iot-based healthcare remote monitoring platform for elderly with fog and cloud computing, in: 2019 22nd International Conference on Control Systems and Computer Science (CSCS), IEEE. pp. 154–161.

Armbrust, M., Das, T., Sun, L., Yavuz, B., Zhu, S., Murthy, M., Torres, J., van Hovell, H., Ionescu, A., Łuszczak, A., et al., 2020. Delta lake: high-performance acid table storage over cloud object stores. Proceedings of the VLDB Endowment 13, 3411–3424.

Ashton, K., 2009. That "internet of things" thing: In the real world things matter more than ideas. RFID journal 22, 97–114.

Bernstein, J.H., 2009. The data-information-knowledge-wisdom hierarchy and its antithesis .

Blackman, S., Matlo, C., Bobrovitskiy, C., Waldoch, A., Fang, M.L., Jackson, P., Mihailidis, A., Nygård, L., Astell, A., Sixsmith, A., 2016. Ambient assisted living technologies for aging well: a scoping review. Journal of Intelligent Systems 25, 55–69.

Brockwell, P.J., Davis, R.A., 2016. Introduction to Time Series and Forecasting. Springer International Publishing. URL: https://doi.org/10.1007%2F978-3-319-29854-2, doi:10.1007/978-3-319-29854-2.

Brownlee, J., 2018. Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery.

Cai, M., Pipattanasomporn, M., Rahman, S., 2019. Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. Applied Energy 236, 1078–1088.

de la Combé, B., 2019. From data to insights : An advice to improve the capacity planning of temporary employees at ceva logistics benelux. URL: http://essay.utwente.nl/79742/.

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

Davenport, T., Kalakota, R., 2019. The potential for artificial intelligence in healthcare. Future healthcare journal 6, 94.

Donaghy, M., 2017. Sensor data storage for industrial iot. https://kx.com/media/2017/06/Sensor-Data-Storage-for-Industrial-IoT.pdf.

Doughty, K., Cameron, K., Garner, P., 1996. Three generations of telecare of the elderly. Journal of Telemedicine and Telecare 2, 71–80.

D'Sa, A.G., Prasad, B., 2019. A survey on vision based activity recognition, its applications and challenges, in: 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), IEEE. pp. 1–8.

Ergen, S.C., 2004. Zigbee/ieee 802.15. 4 summary. UC Berkeley, September 10, 11.

Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A., 2019. Deep learning for time series classification: a review. Data Mining and Knowledge Discovery 33, 917–963.

Filippoupolitis, A., Oliff, W., Takand, B., Loukas, G., 2017. Location-enhanced activity recognition in indoor environments using off the shelf smart watch technology and ble beacons. Sensors 17, 1230.

Fiorini, L., Bonaccorsi, M., Betti, S., Esposito, D., Cavallo, F., 2018. Combining wearable physiological and inertial sensors with indoor user localization network to enhance activity recognition. Journal of Ambient Intelligence and Smart Environments 10, 345–357.

Garg, N., 2013. Apache kafka. Packt Publishing Ltd.

Gers, F.A., Eck, D., Schmidhuber, J., 2002. Applying lstm to time series predictable through time-window approaches, in: Neural Nets WIRN Vietri-01. Springer, pp. 193–200.

Gershenfeld, N., Krikorian, R., Cohen, D., 2004. The internet of things. Scientific American 291, 76–81.

Gingras, G., Adda, M., Bouzouane, A., 2020a. Toward a non-intrusive, affordable platform for elderly assistance and health monitoring, in: 2020 IEEE 44th Annual Computers, Software, and Applications Conference, IEEE. pp. 683–687.

Gingras, G., Adda, M., Bouzouane, A., Ibrahim, H., Dallaire, C., 2020b. Iot ambient assisted living: Scalable analytics architecture and flexible process. Procedia Computer Science 177, 396–404.

Goodman, R.A., Posner, S.F., Huang, E.S., Parekh, A.K., Koh, H.K., 2013. Peer reviewed: defining and measuring chronic conditions: imperatives for research, policy, program, and practice. Preventing chronic disease 10.

Granja, C., Janssen, W., Johansen, M.A., 2018. Factors determining the success and failure of ehealth interventions: systematic review of the literature. Journal of medical Internet research 20, e10235.

Hassan, M.K., El Desouky, A.I., Badawy, M.M., Sarhan, A.M., Elhoseny, M., Gunasekaran, M., 2019. Eot-driven hybrid ambient assisted living framework with naïve bayes–firefly algorithm. Neural Computing and Applications 31, 1275–1300.

Hossain, M.S., Muhammad, G., 2016. Cloud-assisted industrial internet of things (iiot)–enabled framework for health monitoring. Computer Networks 101, 192–202.

Iqbal, R., Doctor, F., More, B., Mahmud, S., Yousuf, U., 2020. Big data analytics: computational intelligence techniques and application areas. Technological Forecasting and Social Change 153, 119253.

Karau, H., Konwinski, A., Wendell, P., Zaharia, M., 2020. Learning spark: lightning-fast big data analysis, 2nd Edition. " O'Reilly Media, Inc.".

Katz, S., 1983. Assessing self-maintenance: activities of daily living, mobility, and instrumental activities of daily living. Journal of the American Geriatrics Society 31, 721–727.

Kotsiantis, S., Zaharakis, I., Pintelas, P., 2006. Machine learning: A review of classification and combining techniques. Artificial Intelligence Review 26, 159–190. doi:10.1007/s10462-007-9052-3.

LeCun, Y., Bengio, Y., et al., 1995. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks 3361, 1995.

Madden, S., 2012. From databases to big data. IEEE Internet Computing 16, 4–6.

Maulud, D., Abdulazeez, A.M., 2020. A review on linear regression comprehensive in machine learning. Journal of Applied Science and Technology Trends 1, 140–147.

Mukherjee, A., Pal, A., Misra, P., 2012. Data analytics in ubiquitous sensor-based health information systems, in: 2012 Sixth International Conference on Next Generation Mobile Applications, Services and Technologies, IEEE. pp. 193–198.

Muñoz, A., Augusto, J.C., Villa, A., Botía, J.A., 2011. Design and evaluation of an ambient assisted living system based on an argumentative multi-agent system. Personal and Ubiquitous Computing 15, 377–387.

O'Shea, K., Nash, R., 2015. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458 .

Papastefanopoulos, V., Linardatos, P., Kotsiantis, S., 2020. Covid-19: A comparison of time series methods to forecast percentage of active cases per population. Applied Sciences 10, 3880.

Pham, M., Mengistu, Y., Do, H., Sheng, W., 2018. Delivering home healthcare through a cloud-based smart home environment (coshe). Future Generation Computer Systems 81, 129–140.

Poellabauer, C., 2020. Lecture notes in wireless networks i and ii.

de la Statistique du Québec, I., 2019. Perspectives démographiques du québec et des régions .

Raghupathi, W., Raghupathi, V., 2014. Big data analytics in healthcare: promise and potential. Health information science and systems 2, 3.

Ramasamy Ramamurthy, S., Roy, N., 2018. Recent trends in machine learning for human activity recognition—a survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8, e1254.

Sagner, M., McNeil, A., Puska, P., Auffray, C., Price, N.D., Hood, L., Lavie, C.J., Han, Z.G., Chen, Z., Brahmachari, S.K., McEwen, B.S., Soares, M.B., Balling, R., Epel, E.S., Arena, R., 2017. The p4 health spectrum - a predictive, preventive, personalized and participatory continuum for promoting healthspan. Progress in cardiovascular diseases 59 5, 506–521.

Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L., 2016. Edge computing: Vision and challenges. IEEE internet of things journal 3, 637–646.

Shoaib, M., Bosch, S., Incel, O.D., Scholten, H., Havinga, P.J., 2016. Complex human activity recognition using smartphone and wrist-worn motion sensors. Sensors 16, 426.

Siami-Namini, S., Namin, A.S., 2018. Forecasting economics and financial time series: Arima vs. lstm. arXiv preprint arXiv:1803.06386 .

Siami-Namini, S., Tavakoli, N., Namin, A.S., 2018. A comparison of arima and lstm in forecasting time series, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE. pp. 1394–1401.

Siami-Namini, S., Tavakoli, N., Namin, A.S., 2019. A comparative analysis of forecasting financial time series using arima, lstm, and bilstm. arXiv preprint arXiv:1911.09512 .

Siow, E., Tiropanis, T., Hall, W., 2018. Analytics for the internet of things: A survey. ACM Computing Surveys (CSUR) 51, 1–36.

Syed, L., Jabeen, S., Manimala, S., Alsaeedi, A., 2019. Smart healthcare framework for ambient assisted living using iomt and big data analytics techniques. Future Generation Computer Systems 101, 136–151.

Turakhia, M.P., Desai, M., Hedlin, H., Rajmane, A., Talati, N., Ferris, T., Desai, S., Nag, D., Patel, M., Kowey, P., et al., 2019. Rationale and design of a large-scale, app-based study to identify cardiac arrhythmias using a smartwatch: The apple heart study. American heart journal 207, 66–75.

United Nations, D.o.E., 2019. World population ageing. Department of Economic and Social Affairs .

Waller, M.A., Fawcett, S.E., 2013. Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management. Journal of Business Logistics 34, 77–84.

Walls, C., 2016. Spring Boot in action. Manning Publications.

Wong, K.C., 2015. A short survey on data clustering algorithms, in: 2015 Second international conference on soft computing and machine intelligence (ISCMI), IEEE. pp. 64–68.

Yang, J., Nguyen, M.N., San, P.P., Li, X., Krishnaswamy, S., 2015. Deep convolutional neural networks on multichannel time series for human activity recognition., in: Ijcai, Buenos Aires, Argentina. pp. 3995–4001.

Yassine, A., Singh, S., Hossain, M.S., Muhammad, G., 2019. Iot big data analytics for smart homes with fog and cloud computing. Future Generation Computer Systems 91, 563–573.

Zhao, B., Lu, H., Chen, S., Liu, J., Wu, D., 2017. Convolutional neural networks for time series classification. Journal of Systems Engineering and Electronics 28, 162–169.