Thesis for the Degree of Doctor of Philosophy

SEMANTIC-AWARE DATA IMPUTATION FOR UNOBTRUSIVE COMPLEX HUMAN ACTIVITY RECOGNITION

Muhammad Asif Razzaq

Department of Computer Science and Engineering
Graduate School
Kyung Hee University
Republic of Korea

February 2021

SEMANTIC-AWARE DATA IMPUTATION FOR UNOBTRUSIVE COMPLEX HUMAN ACTIVITY RECOGNITION

Muhammad Asif Razzaq

Department of Computer Science and Engineering
Graduate School
Kyung Hee University
Republic of Korea

February 2021

SEMANTIC-AWARE DATA IMPUTATION FOR UNOBTRUSIVE COMPLEX HUMAN ACTIVITY RECOGNITION

by

Muhammad Asif Razzaq

Supervised by

Prof. Sungyoung Lee

Submitted to the Department of Computer Science and Engineering and the Faculty of Graduate School of Kyung Hee University in partial fulfillment of the requirements of the degree of Doctor of Philosophy

Dissertation Committee:

Prof. Eui-Nam Huh

Prof. Tae-Seong Kim

Prof. Yong-Ik Yoon

Prof. Seungkyu Lee

Prof. Sungyoung Lee

for my beloved

Father *(may **ALLAH** rest his soul in peace)*, *Mother,*

Wife, Kids, Brothers & Family, who played a pivotal role

by their continuous encouragement and support in

all hard times for keeping me in the position

to complete my Ph.D. degree

Abstract

The recognition of activities of daily living (ADL) in smart environments is well-known and is an important branch of human-centred research, which presents the real-time state of humans in pervasive computing. Advances in wearable and unobtrusive technologies offer many opportunities for Human activity recognition (HAR). While much progress has been made in HAR using wearable technology, it still, however, remains a challenging task using unobtrusive (non-wearable) sensors. The process of recognizing human activities generally involves deploying a set of obtrusive and unobtrusive sensors, pre-processing the raw data, and building classification models using machine learning (ML) algorithms. Integrating data from multiple sensors is a challenging task due to the dynamic nature of data sources. This is further complicated due to semantic and syntactic differences in these data sources. These differences become even more complex if the data generated is imperfect, which ultimately has a direct impact on its usefulness in yielding an accurate classifier. In this thesis, we propose a semantic imputation method to improve the quality of obtrusive and unobtrusive sensor data. We propose two different approaches to deal with obtrusive and unobtrusive data sources. As a first approach we propose Multi-strategy Imputation method, which uses a set of proposed ontologies "SemImputOnt" for data modelling and further utilizes proposed semantic similarity learning. This is achieved by identifying semantic correlations among sensor events through SPARQL queries, and by performing a time-series longitudinal imputation. Furthermore, we applied deep learning (DL) based artificial neural network (ANN) on public datasets to demonstrate the applicability and validity of the proposed approach. Secondly, to deal with unobtrusive data sources, we also propose a vision-based Multioccupant State Imputation method for accurate detection and tracking of multi-occupant HAR. This method uses a novel low-resolution Thermal Vision Sensor (TVS) in a smart-home environment. Specifically, for this, we propose an unobtrusive Multioccupant Detection and Tracking (uMoDT) method. The uMoDT is a two-step framework, consisting of a Computer Vision (CV) based method to accurately detect and track multiple occupants combined with Convolutional Neural Network (CNN) based HAR. The proposed algorithm uses frame-difference over consecutive frames for occupant detection, a set of morphological operations to refine identified objects, and features are extracted before applying a Kalman filter for tracking of missing frames. These missed frames are imputed in order to draw a comparison with the ground-truth to prove the robustness of the method. Laterally, a 19-layer CNN architecture is used for HAR and afterwards the results from both methods are fused using time interval based sliding window.

Keeping in view the relevance of afore-mentioned facts and to improve complex HAR using proposed semantic imputation techniques, this research, provides insights with the contribution in the following areas:

(1) Multi-strategy Data Imputation; designed and developed a semantic data imputation method for understanding the semantics of the human activity monitoring sensor data, select different variables, model data using a domain-specific ontology, identify semantic relations among variables, validate the data against the ground-truth, fill in the missing variables and their values. This method has the ability to deal with the sensor-based data of structured form.

(2) Multioccupant State Imputation; a method to perform an unobtrusive detection and occupant state estimation in a vision-based approach. This method deals with an unobtrusive thermal vision sensor, which generates low-resolution grayscale frames. The proposed and developed method identifies human activities within the smart-home environment using low resolution thermal frames in order to identify, detect, track and classify single, as well as, multioccupants.

Throughout this thesis, we performed both qualitative and quantitative evaluation of our proposed methodology on publicly available benchmark datasets. We performed extensive experimentation to evaluate *(1) SemImput* and *(2) uMoDT* methods. The extensive experimental results show a higher accuracy with semantically imputed datasets using ANN for SemImput method. We also presented a detailed comparative analysis, comparing the results with the state-of-the-art from the literature. We found that our semantic imputed datasets improved the classification accuracy with 95.78% as a higher one thus proving the effectiveness and robustness of learned models.

In order to validate uMoDT method, it was also evaluated through a series of experiments based on benchmark Thermal Infrared datasets (VOT-TIR2016) and multi-occupant data collected from TVS. Results demonstrate that the proposed method is capable of detecting and tracking 88.46% of multi-occupants with a classification accuracy of 90.99% for HAR.

Acknowledgement

Alhamdulillah, By the Grace of Almighty ALLAH, who is the Most Beneficent and the Most Merciful. Who gave me the strength, patience, courage, opportunity, ability during my doctoral study and showering of HIS Blessings upon me and my family. Without HIS Support, such an achievement would not have been possible.

I am highly grateful to respected advisor Prof. Sungyoung Lee from Kyung Hee University (KHU), South Korea, for giving me a chance to pursue Ph.D. degree, his boundless moral and technical supervision, guidance, motivation, and courage in coping with the difficult challenges throughout the educational period of my doctoral studies. He trained me in multi-directions to face the challenges of practical life in a professional manner. His lively nature, clear assistance, and direction enabled me able to complete my thesis. He has refined the key ingredients for high-quality research, namely my skills of creativity, thinking, and technical understanding. Moreover, I would like to acknowledge his valuable guidance and support to refine the problem statement as well as that to streamline my research direction.

I appreciate co-advisor Prof. Seungkyu Lee. I also acknowledge my dissertation evaluation committee for their valuable observations and insight recommendations during the dissertation defense. These comments enhanced the presentation and contents of the dissertation.

My sincere thanks also go to Prof. Christopher Nugent and Dr. Ian Cleland from the Ulster University, Northern Ireland, United Kingdom for their continuous support, contribution of valuable time, manifold research conversations, and provision of experimental environment. Without their precious encouragement and continuous feedback, it was not possible to complete research satisfactorily. I will keep these in remembrance.

I am also grateful to my KHU fellows who have supported me with their precious time, tech-

nical expertise, and brotherly encouragement during my study period. They were always ready to direct me in tough situations during my studies. I would like to thank Dr. Wajahat Ali Khan, Dr. Thien Huynh-The, Dr. Bilal Amin, Dr. Jamil Hussain, Dr. Taqdir Ali, Dr. Shujaat Hussain, Hafiz Syed Muhammad Bilal, Syed Imran Ali, Usman Akhtar, Musarrat Hussain, Ubaid Ur Rehman, Fahad Ahmed Satti, Muhammad Sadiq, and Asim Abbas. They supported me in successfully performing various personal and academic tasks that presented hurdles to me during my stay at South Korea. I would like to thank Mrs. Kim for being available and helping me in many regards.

I also appreciate all of my current and former UCLab fellows for their kind support to my personal and academic life at Korea. I am highly gratified to the following excellent researchers - Dr. Rehman Ali, Dr. Maqbool Hussain, Dr. Muhammad Afzal, Dr. Tae Ho Hur, Dr. Dohyeong Kim, Dr. Jaehun Bang, Hua-Cam Hao, Muhammad Zaki Ansaar, Dr. Kifayat Ullah, Dr. Waqas Nawaz, Dr. Saeed Ullah, and Aftab Alam. This journey would have been quite difficult without their support. They helped my personal and academic life to boost myself. I also appreciate all my Korean and international friends who worked collaboratively with me and inculcated teamwork skills in me. I would also thank Dr. Claudia, Dr. Javier, Dr. Aftab, Dr. Naveed Khan, Hanifullah, and Zeeshan for their research collaboration, unconditional support in research and during my stay abroad.

Muhammad Asif Razzaq
February 2021

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This dissertation mainly focuses on investigating the problem of missing data in the multimodal environment for human activity recognition (HAR). HAR has been studied for several decades using real-time ambient environments or by utilizing publicly available datasets. In practice, the collected dataset contains gaps in the shape of missing attribute values. This incompleteness may arise due to a variety of sources, improper configuration, power failure or network error. Removal of such data has a significant effect on analysis results for drawing conclusions. Therefore, a strategy has to be derived to identify missingness and provide a suitable solution based on the nature of sensors and their data. In this way, the quality of data is enhanced, which ultimately will improve the classification accuracy for activities. The main motivations are discussed in the opening chapter, the process of recovering the data missingness in Section 1.1, the problem statement along with posed research questions are described in Section 1.2, key contributions to address challenges of this research in Section 1.3, and lastly, the dissertation summary is outlined in Section 1.4.

1.1 Motivation

Over the past few decades, a rapid advancement has been observed in pervasive computing for the assessment of cognitive and physical well-being of older adults. For this purpose, monitoring of Activities of Daily Living (ADLs) is often performed over extended periods of time [4]. This is generally carried out in intelligent environments containing various pervasive computing and sensing solutions. Sensors data from these intelligent environments has been playing a vital role in machine learning tasks such as model learning which is complementary to the human-annotated training data. However, due to misoperations, sensor data may have quality issues with a variety of missing values, which could result in the performance degradation of the learned model.

It is, therefore, an important step to understand the needs as to why data are missing and what strategy would be suitable to treat the missingness using the state-of-the-art statistical analysis approaches. As in literature there exist numerous approaches which includes: list-wise deletion based on complete case analysis, single imputation methods (mean replacement / substitution or single regression replacement), and multiple imputations. All aforementioned approaches make mathematical-statistical assumptions which may hinder invalid empirical results. However, any incorrect categorization or any invalid assumption may have substantial effects in terms of (1) decreasing the prediction power to estimate models usually resulting from a decreased sample size (2) leading to biased results with potentially incorrect inferencing, (3) and overestimating errors mostly occurring through relative bias. So it evident that studying these substantial effects is more important as if a large number of observations are not included in the statistical analysis, the model may loose the statistical power and the resulting predictions may of lesser significance. To keep the variance intact importance to the dependent variables and their relationships has to be ascertained. As if any subset or a whole block of observations is ignored or dropped due to missingness, would create systematically unacceptable results from the statistical analysis. As this missingness will induce bias in the samples as well as in the subsequent estimates generated from them [5]. Moreover, the usability of semantic imputation and feature extraction using ontological methods in combination with deep neural networks for recognizing complex activities remains to be investigated. Previous studies have not provided a comprehensive analysis of the impact of imputation on the classification accuracy. To this end, we present research proving the applicability of semantic imputation for missing sensors and their states on activity classification in a controlled environment using deep-learning-based Artificial Neural Networks (ANNs). This combination of semantic imputation with neural networks in a supervised learning method using public datasets not only increases accuracy but also reduces the complexity of training data. The presented work is, to the best of our knowledge, the first to exploit ontologies, semantic imputation, and neural networks.

Over several decades, advances in pervasive computing have offered great promise towards the potential of indoor localization and Human Activity Recognition (HAR) [6]. Over this period, significant research effort has been targeted towards the creation of solutions that can reliably monitor individuals through the use of on-body wearable sensors, dense sensors, and vision sensors [7]. Whilst results utilizing on-body sensors has improved greatly, wearable solutions are often said to be impractical, as they can be difficult to carry or inconvenient to wear continuously [8]. Additionally, vision sensors capable of capturing RGB or grayscale images have been studied intensively within the Computer Vision (CV) domain. The use of cameras, however, raises serious privacy concerns [9].

Recently, researchers have been investigating the potential of deploying unobtrusive, inexpensive and low-resolution Thermal Vision Sensors (TVS) for occupant detection and pervasive sensing [10]. Similar to traditional vision-based approaches, TVS suffer from same limitations for handling complex object appearances due to shape deformation, low resolution, varying number of objects, pose variation, motion estimation, and object re-identification [11] due to missing information. TVS do, however, address some of the challenges as they tend to be more robust to illumination changes, can operate even in complete darkness and offer less intrusion on user's privacy [12].

The majority of research into HAR has focused on single-occupant environments. Nevertheless, living environments are usually inhabited by more than one person. Therefore, HAR in the context of multi-occupancy would provide a more practical solution, however, also more challenging. The difficulty with multi-occupant HAR stems from two related challenges in occupant identification, known as data association, and the diversity of human activities.

In CV, object tracking remains one of the most significant research challenges [13, 14]. This becomes even more complex when using TVS for monitoring multi-occupants, as data only corresponds to variation in temperature. Therefore a different strategy is required for identification and re-identification of the occupants [15] when the information is missing. The aforementioned challenges are addressed by proposing and implementing a robust CV-based integrated framework for multi-occupant detection, tracking and HAR based on TVS.

The key objectives being addressed in this study are to: (1) design and development of a practical scheme for modelling time-series data into an ontology, (2) perform semantic data expansion using the semantic properties, (3) identify suitable semantic data imputation measure, (4) design and train an effective deep learning model for Human Activity Recognition (HAR), and (5) undertake a comparative analysis using public datasets with each having different rates of missing data and imputation challenges.

1.2 Problem Statement

In the real world environment, building an accurate activity recognition model remains a problem due to missing data, which reduces an overall performance and robustness. There are three types of problems which are usually analogous with MVs in any multivariate dataset and their analysis tasks [16]: (1) prediction efficiency loss; (2) complexities while handling, manipulating and analyzing multivariate data; and finally (3) bias occurring due to the differences between missing and complete data. A simple solution where there is an abundance of data is to delete but this process becomes challenging when the data is of small size. So rejecting a variable may result in the loss of predictive power. It also effects the ability to detect statistical significance to both missing and complete data and may become a source of bias affecting the conclusions through conventional classifications or data mining tasks. For these reasons, data variable selection representing sensors need to be identified as a source to the missing data. Thus imputation can be performed after selection of missing data variables. Most of the statistical methods adapt general steps for handling missing data such as (1) Identify the patterns; (2) observe the portion of missing data; (3) choose the imputation method based on a similarity measure, maximum likelihood or any other metric. In this thesis, the goal is to improve HAR by recovering missing values by understanding inter-variable semantics in the observed data through an accurate data imputation methodology. To achieve the goal, the objectives are to design and develop a semantic data imputation methodology for understanding the semantics of the HAR sensor variables and fill in the missing values. It also requires a process to evaluate the proposed methods by deep learning HAR classification for its accuracy and robustness while avoiding the bias in the complete dataset.

Following are the prominent challenges to the proposed methods.

- How to minimize the missing values and maximize the quality of datasets by keeping semantics intact? [7,8]

- How to provide an empirical method to prove data consistency and improved accuracy for both structured and unstructured HAR data? [16]
- How to prove the robustness of proposed methods? [4]

In this section, we first introduce key definitions, which are carried throughout the paper. These definitions are necessary for understanding concepts referred to in this paper. Later, a robust illustrative example is presented to represent the research problem for HAR referred in this study.

1.2.1 Some Definitions

In this section, we first give preliminary definitions of problems that the methodology aims to address. Laterally, we introduce the notion of Semantic imputation.

Definition 1 (*Formal Notation*) Let $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n\}$ be the set of multimodal sensory data of the form $(p \times q)$ matrices modelled over the domain ontologies $\{\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n\}$ respectively, where *p* represents the number of observations for *q* concepts (variables).

Definition 2 (*Training Tuples*) Let T_d = $\{t_1$, ..., $t_p\}$ be the set of training tuples for dataset $\mathcal{D}_n$ containing missing attributes or their values. Let t_m is a tuple with q attributes $\{A_1, \ldots A_q\}$, which may have one or more missing attributes or its value where $t_m \in T_d$. Let t_{ma} be the missing attribute A and t_{mv} be the missing value on attribute A where A $\in A_q$. Given a candidate imputed set, $t_m = \bigcup_1^m (t_{ma} \cup t_{mv})$ for a possible missing attributes or its value for t_m.

Definition 3 (*Ontology*) A core ontology is a structure $\mathcal{O} := (C, \leq_c, R, \sigma, \leq_r)$ consisting of two disjoint sets concept identifiers 'C' and relation identifiers 'R', a partial order $\leq_c$ on C, called concept hierarchy or taxonomy, a function σ representing signature, and a partial order $\leq_r$ on R defining relation hierarchy.

Definition 4 (*Ontology-based Tuples*) Given o_k and o_l in $\mathcal{O}$, (o_k, o_l) is called an ontology-based tuple, if and only if: (1) $\exists$ A, B $\in$ C | o_k $\in$ A and o_l $\in$ B; (2) A $\mapsto$ B; and (3) $\lambda_{o_k}(o_l)$ $\leq \gamma$.

Definition 5 (*Knowledge-base*)A Knowledge Base $\mathcal{K}$ is conceptually referred to a combination of intentional terminologies TBox ($\mathcal{T}$) part and extensional assertion ABox ($\mathcal{A}$) part modeled over an ontology $\mathcal{O}$. $\mathcal{T}$ includes concept modeling and the relations in ontology $\mathcal{O}$ and $\mathcal{A}$ includes concept instances and roles.

Definition 6 (*Conjunctive Query*) Conjunctive queries $\mathcal{Q}$ enable answers by identifying attributes or their values, which are rewritten as

$$\forall \bar{A}\bar{R}(\bar{A},\bar{C}_k) \wedge not(\bar{N}(\bar{A},\bar{C}_k)) \tag{1.1}$$

where $\bar{A}$ represents vector of attributes $(A_1,\ldots,A_q)$, vectors of concept instances $\bar{C}_k$, conjoined predicates (relations) $\bar{R}$, and a vector of disjoined predicates (relations) $\bar{N}$.

1.2.2 Problem Formulation: Semantic Imputation

A Knowledge Base is a consistent structure $\mathcal{K} = (\mathcal{T},\mathcal{A})$, and we revise the Abox $\mathcal{A}$ to $\mathcal{A}^I$ such that $\mathcal{K} = \left(\mathcal{T},\mathcal{A}^I\right)$ should also be consistent:

$$\mathcal{A}^I = \mathcal{A} \cup \mathcal{I}(A_m) \qquad \text{since } (\mathcal{A}_m = D_n \setminus \mathcal{A}) \tag{1.2}$$

$$\mathcal{I}(A_m) = \mathcal{I}_{SS}(A_m) + \mathcal{I}_{SI}(A_m) + \mathcal{I}_L(A_m) \tag{1.3}$$

where A_m represents missing attributes or their values and $\mathcal{I}_{SS}(A_m)$, $\mathcal{I}_{SI}(A_m)$, $\mathcal{I}_L(A_m)$ measure structural-based, instance-based and longitudinal imputations for missing attributes and their values, respectively.

Hence, we define our problem in a 4-tuple ($\mathcal{D}$, $\mathcal{K}$, $\mathcal{Q}$, $\mathcal{I}$) such that $\mathcal{D}$ denotes the input data, modeled over the ontology $\mathcal{O}$ having assertion set $\mathcal{A}$ which are retrieved using conjunctive queries $\mathcal{Q}$ with the results used to perform semantic imputation $\mathcal{I}(A_m)$ introducing improved assertions $\mathcal{A}^I$. We ensure that, during the whole process, $\mathcal{K}$ remains consistent with the addition of imputed assertions $\mathcal{A}^I$.

1.3 Key Contributions

We summarize the main contributions of this thesis as below:

1.3.1 Time-Series SemImput Ontology Modeling

Design and development of a practical scheme for modelling time-series data into an ontology to facilitate the discovery of data, its sources and relationships amongst them. It becomes challenging due to the data complexity and its variety of data producing data sources. Moreover, the ontology is also used to provide additional support in the realization, intelligently retrieval management, and imputation methods.

1.3.2 Novel Semantic Data Expansion

For performing semantic data expansion, the semantic properties of the sequential, concurrent, and parallel activities of daily living are inspected and retrieved using ontology models. This was achieved by using specially designed semantic queries for finding, inspecting, retrieving the sensor events and underlying sensors.

1.3.3 Improved Semantic Data Imputation

Utilized ontology modelling supported with reasoning to perform high-quality imputation to fill the missing data with minimal bias and computational efforts. The proposed method successfully improves the accuracy of imputed public HAR datasets when compared to HAR without imputation. Semantic imputation is performed based on the nature of the specific sensor and understanding the semantics of categorical or continuous data. This is achieved by exploiting ontology-based complex activity structures, and conjunction separation. Our proposed method includes different imputation measures such as: (1) Structure-based, (2) Instance-based, and (3) longitudinal imputation. These measures deal with the imputation of a single value or multiple values by keeping the focus on the data repair, which is closed to the initial values based on the minimum change principle.

1.3.4 Refined Morphological Characteristics

We rely on refined morphological characteristics, which ensure efficient detection and tracking accuracy over the dynamic patterns for non-rigid moving targets per-frame. For this, we propose a technique, which obtains effective and optimal results in terms of its quality, efficiency, and time.

A binarizing operation is suggested on grayscale thermal frames with global thresholding. The morphological features of grayscale thermal frames are further enhanced using morphological operators erosion and dilation for selected features and suppressing noise. An appropriate threshold value plays a major role in separating background and foreground, an essential task required for detection and tracking of multioccupant in frames.

1.3.5 Robust CV-based State Imputation

To overcome the missing objects in the frames, We propose an online method, which uses a CV-based algorithm, with improved morphological features with a low computational load. It performs an automatic multi-target initialization using frame differencing with an optimum threshold. Kalman Filter is applied with modified features for occupant tracking, however, for any missed state, estimated occupant location is also obtained.

1.3.6 Accurate Track Assignment

We use the Hungarian method for track assignment problem with an approach for maintaining an association history of re-identified tracks of individual moving objects per-frame. Both of the Kalman filter and the Hungarian method are used collectively to predict the occupant's position in the frame especially for the cases where there is a missing observation due to a false negative or imperfect communication.

1.3.7 Comprehensive HAR Evaluations and Comparative Analysis

Design and train an effective deep learning model to study the impact of imputation on classification performance for Human Activity Recognition (HAR) using both sensor-based and vision-based datasets. The statistical significance in terms of accuracy and robustness performances were obtained for unaugmented models over non-imputed datasets and augmented DL models trained on complete datasets with imputation effect. The consistency in the accuracies for each of the activity in the HAR dataset provides enough evidence for the model learning robustness. Similarly, a comparative analysis was also undertaken by using public sensor-based and vision-based datasets with each one having a different set of activities, rate of missing data and imputation challenges.

The proposed methodology is validated using a thermal vision sensor dataset gathered at Smart Environments Research Group SERG) laboratory from the Ulster University, UK. It also proved to be computationally robust and achieves a promising tracking accuracy in comparison with other MOT methods. We also demonstrated quantitative evaluations on the publicly available vision-based dataset for the VOT-TIR 2016 challenge proving the practicality and efficacy of the proposed framework with the state-of-the-art. Additionally, we propose to apply a CNN architecture to extract and learn spatial features from multiple successive Thermal Vision Sensor Frame (TVS-F) for individual action recognition.

1.4 Thesis Organization

The dissertation aims at investigating an accurate and robust data imputation methodology to impute the missing values in sensor-based and vision-based thermal camera HAR datasets to recognize complex human activities in a smart-home environment. Figure 1.1 provides the dissertation overview, summarizes the dissertation structure and its flow. This thesis is further organized into the following chapters.

- **Chapter 1: Introduction**. Chapter 1 encompasses the introduction of the research work for robust unobtrusive human activity recognition methods by using suitable semantic imputation method. It focuses on the motivation for this research, preliminary definitions, problems in areas, the goals to achieve to solve these problems, key contributions, the objectives achieved through this research work, and finally the overview of dissertation and organization.

- **Chapter 2: Related work**. Chapter 2 provides a detailed review of previous research related to semantic data imputation for unobtrusive sensing streams in an organized manner. This research focuses on presenting a flexible and comprehensive semantic imputation methodology to deal with sensor missing data for recognizing accurate human activities. Therefore, we conclude with an overview of several methodological studies of semantic imputation sensor-based and computer vision-based approaches. Various research directions related to (1) sensing technologies involved in recognizing unobtrusive human activities, (2) proposed

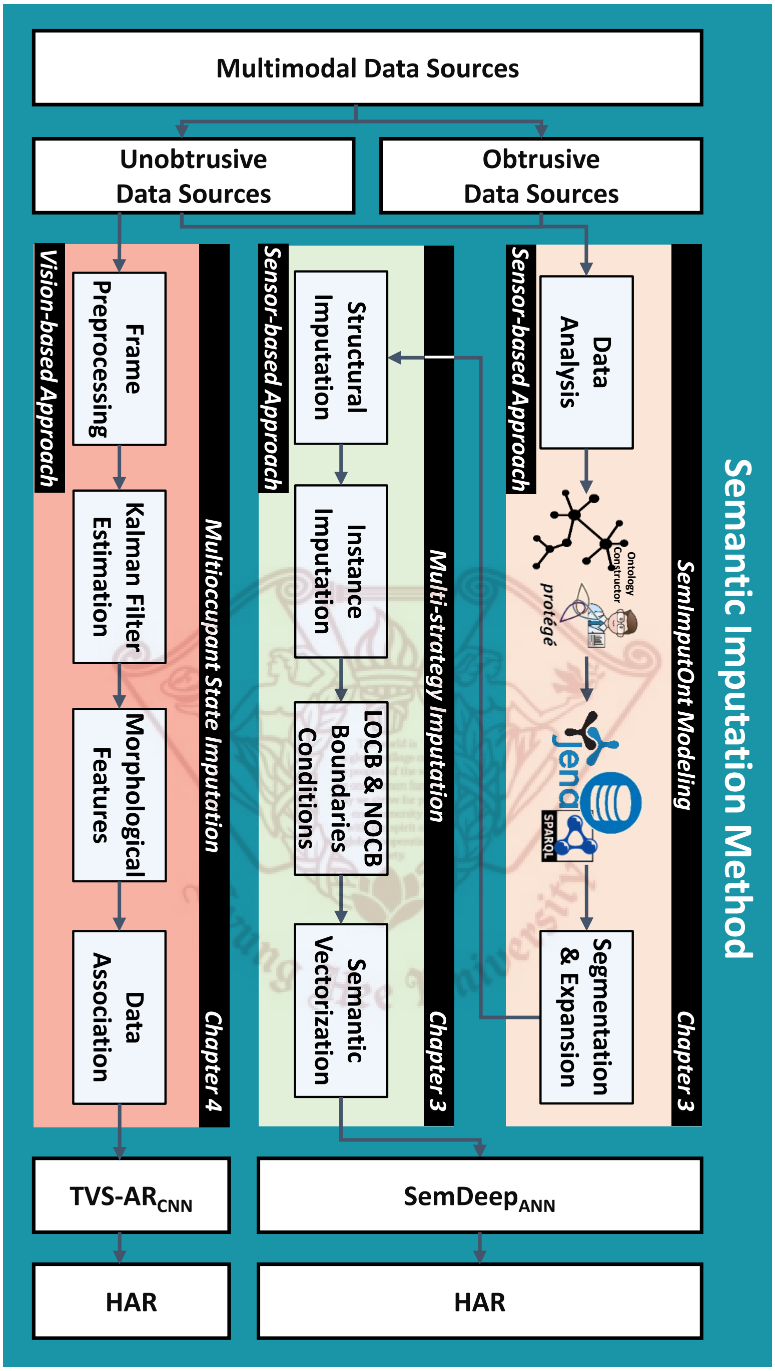


Figure 1.1: Idea diagram representation and mappings of chapters with proposed research studies.

HAR learning models in the literature, (3) a detailed review for existing methods for performing data imputation techniques, and (4) summary of related literature to the proposed methodology for semantic imputation for both HAR and Computer vision datasets are discussed in each subsection. Finally, we summarize the related works that identified missing value states and providing semantic imputation solution for the proposed methodology.

- **Chapter 3: Multi-strategy Data Imputation**. In this chapter, we present high-level overview of the *Semantic Imputation* (SemImput) functional framework to fully utilize a methodology for identifying missing sensors and their states from given HAR datasets. For the SemImput framework, we first propose a *SemImput* ontology to model time-series HAR sensors and their datasets. For this, expert knowledge has to be established regarding the nature of sensors and their states for recognizing activities. The SemImput framework involves a sequence of proposed ontology-based data alignment using *Data alignment and re-sampling algorithm*, preprocessing using proposed *Semantic Data Expansion* methods *conjunction separation* technique, identification of missing sensor states, a methodology to fill in those states using proposed *Semantic Data Imputation* method through SPARQL queries, and finally completion of HAR datasets for missing sensors and their states using *Instance-based* and *Longitudinal imputation* measure. We also present the preparation of Semantically Imputed datasets for classification using *One-Hot Code Vectorization*, which is an experimental requirement for applying proposed *Semantic Deep Learning-based Artificial Neural Network* (SemDeep-ANN).

- **Chapter 4: Vision-based Multioccupant State Imputation**. This chapter describes a methodology to construct the *Computer Vision* (CV)-based occupant detection, tracking, predicting for missing objects using proposed *unobtrusive Multi-occupant Detection and Tracking* (uMoDT) framework. A detailed description is provided for the proposed *Thermal Vision Sensor multi-occupant frame vector* (TVS-MoFV) algorithm, which improves the morphological features in grayscale thermal frames. This chapter also describes the experimental setup and implementation details for *uMoDT* framework to evaluate collected TVS and benchmark datasets.

- **Chapter 5: Evaluations and Results**. This chapter presents the application of deep learning algorithm for training of model for both incomplete and semantically complete datasets and finally, perform HAR test data classification. In this chapter we introduce (1) Predicted data validation against ground-truth using suitable metrics (2) the *SemDeep-ANN*, which measures the impact of both imputation against non-imputed datasets using evaluation metrics such as accuracy, precision and f-measure, (3) classification performance of SemImput framework for public datasets, (4) confusion matrix per-class HAR for non-imputed and imputed datasets, (5) evaluation of *uMoDT* framework with Thermal Sensor Frames and benchmark sequences, and (6) a comparison between uMoDT robustness and accuracy.

- **Chapter 6: Conclusion and future directions**. This chapter draws the concluding remarks for the dissertation and also provides future directions in this research area. Furthermore, it also suggests the potential applications areas of the proposed methodology.

Chapter 2

Related Work

This chapter provides detailed insights for various existing studies related to data missingness in structured and unstructured forms, mechanisms for addressing them through missing data imputation methods and lastly its impact on HAR. This research targets various areas related to (1) complex human activity recognition through structured and unstructured data, (2) unobtrusive sensing technologies, (3) deal with data missingness for obtrusive and unobtrusive data sources and (4) provides a summarized picture for all modalities in the last section.

2.1 Overview of Complex Human Activity Recognition

An automated recognition of a set of human activities requires data acquisition from suitable sensors, to find patterns through optimal methods for describing them in terms of the data gathered by available sensors, and to bifurcate them from each other. In turn, this requires suitable sensors, which could identify these set of activities, which are of particular interest in a way that they can be distinguishable and recognizable [17]. The collected sensors data can provide a good approximation to the user interaction with the surroundings and these recognized activities are an essential ingredient of ubiquitous and pervasive computing systems [18] acting as a key technology, which results for applications to be aware of the situation of their users and interaction with the environment. Nowadays, the most dominant user interaction information consumers are mobile applications, which engage different capabilities of sensors in the shape of HAR [19].

2.2 Overview of Unobtrusive Sensing

Activity recognition and monitoring processes can be executed using wearable sensors and non-wearable video cameras at smart-homes, rehabilitation nursing homes or even at medical care units. These sensors may pose several problems ranging from battery life and wearability to a feeling of discomfort and privacy concerns [20]. The subsequent subsections, therefore, discusses the use of a non-charged, non-wearable, unobtrusive and a privacy-enhanced sensing solution in more detail. In addition to cameras, there exist potential unobtrusive sensing solutions using thermal, radar, optical, ultrasound sensing, and various object-based sensing technologies. These technologies provide solutions aimed at monitoring and recognizing human activities using a single or a combination of unobtrusive sensing devices. The identification of these sensor types, which better suites for HAR over given set of human activities is a complex and still unsolved issue. There exist several commonly adopted sensing strategies for HAR, which can be grouped mainly in three categories [17]:

• Body-worn inertial sensors, embedded in smartwatches and smartphones [21] and possibly complemented by other wearable sensors, have recently gained increasing popularity for HAR.

The need for considering this dataset is to tailor a framework for ADLs recognition and perform the research using sensors, as most of the technologies nowadays are underscored by the elderly people, their health and importance of their occupancy state. So for them, associated activities might affect their functionality of daily life. So ADLs recognition from multimodal sensors for each segment of a daily routine i.e. morning, afternoon and evening in a controlled environment is the primary motivation behind this study.

• HAR systems mainly based on distributed environmental sensors usually aim at design and development of cost-effective [22] unobtrusive solutions for behavioural anomaly detection or activities requiring the interaction with home appliances.

The selected aggregation strategy for data-level fusion determines the way in which multimodal data reach the fusion node [23]. At the fusion node, different ADLs can be best be recognized by selecting the appropriate fusion strategy with variable window lengths [24].

• Similarly, Vision-based HAR applications describe activities and their information captured by a single or more than one camera device, statically placed in the environment [25].

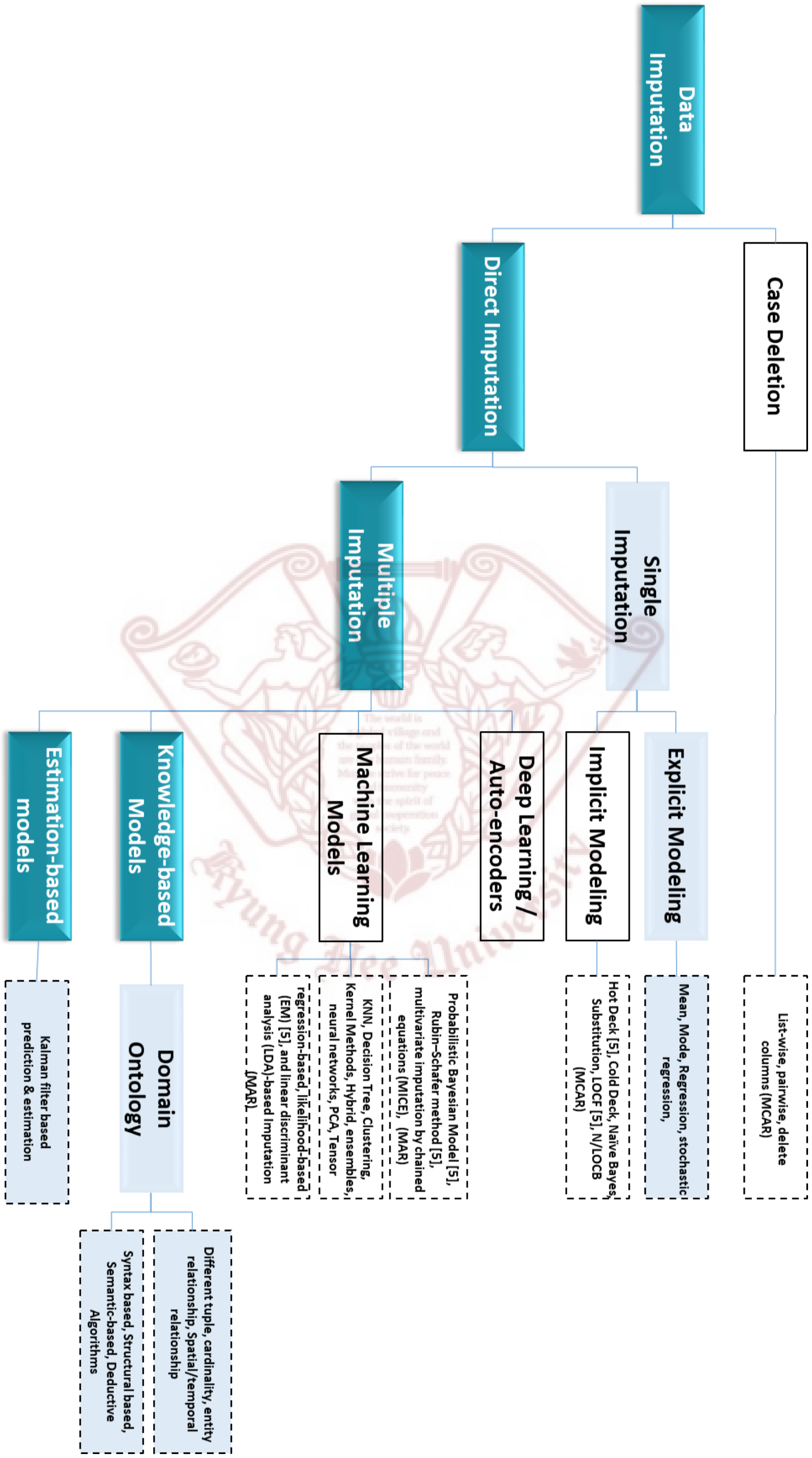

Data Imputation
Direct Imputation
Case Deletion
Multiple Imputation
Single Imputation
Estimation-based models
Knowledge-based Models
Machine Learning Models
Deep Learning / Auto-encoders
Implicit Modeling
Explicit Modeling
List-wise, pairwise, delete columns (MCAR)
Mean, Mode, Regression, stochastic regression,
Hot Deck [5], Cold Deck, Naïve Bayes, Substitution, LOCF [5], N/LOCB (MCAR)
Probabilistic Bayesian Model [5], Rubin–Schafer method [5], multivariate imputation by chained equations (MICE), (MAR)
KNN, Decision Tree, Clustering, Kernel Methods, Hybrid, ensembles, neural networks, PCA, Tensor
regression-based, likelihood-based (EM) [5], and linear discriminant analysis (LDA)-based Imputation (MAR)
Domain Ontology
Kalman filter based prediction & estimation
Different tuple, cardinality, entity relationship, Spatial/temporal relationship
Syntax based, Structural based, Semantic-based, Deductive Algorithms

Figure 2.1: Research Taxonomy - Semantic Imputation along with other Data Imputation Methods and approaches.

We aim to perform semantic imputations to increase the accuracy of HAR using unobtrusive data obtained from public repositories. However, for vision-based HAR, we use thermal sensor to detect and track multioccupant whereas their missing locations are predicted using Kalman Filter.

2.3 Overview of Semantic Imputation

In many real-world HAR domains, the collected datasets are usually incomplete, containing missing attributes and their values such as *UCamI* [1], *Opportunity* [26], and *UCI-ADL* [27] public datasets. Many techniques have proposed to process them and recognize activities. By directly using them with missing states for recognizing activities can have a significant impact on final conclusions drawn from them [28]. Consequently, data quality has become an important feature before applying any technique in obtaining decisions. Therefore missing value recovery for the datasets with different distributions becomes an important preprocessing step [29, 30].

However, to properly identify the categories of the data to be classified, it is essentially a basic requirement to understand the nature of missing data and their treatment mechanism [31]. Researchers have studied such data loss due to missingness and proposed numerous techniques to deal with missing values based on the losses. These can be categorized as: (1) missing completely at random (MCAR); (2) missing at random (MAR); and (3) missing not at random (MNAR). In MCAR, the missing data occurs completely at random with having no relationship amongst other data point missing or observed. In MAR, the missing values have no relationship with other missing data points, however, does relate to observed data points. Lastly, the MNAR category does refer to such missing data points, which are not random and can be dealt with minimum bias using MAR mechanism [32]. Generally, most widely used term for missing value replacement is "data imputation", which refers to the process of replacing the missing values of attributes within a given incomplete dataset with their potential or actual values according to a specific strategy [33]. A comprehensive survey provides an overview of the challenges, opportunities and approaches for using machine learning and Semantic Web to cover aspects related to scalability, contradicting and missing data [34].

Missing values are treated with unconditional mean imputation or median imputation where data complexities are not challenging. However, for time-series characterized data univariate time-series imputation bridges data together by filling missing gaps using last observation carried forward (LOCF). The challenges where predictor variables relies on observable variables for missing value identification, multivariate time-series imputation is performed [35]. As this thesis deals with HAR public datasets, which is time-series so a different strategy needs to be explored to impute missing real-time monitoring data. Any of the missing sensor states may affect any HAR statistics resulting into some serious anomaly for human behaviour. Curley et al. [5] demonstrated with different techniques of data imputation while offering several rationale in order to prove the impact of different remedies. They suggested list-wise deletion, single imputation and multiple imputation methods to deal with MCAR, MAR, MNAR classification. Most recently Liu et al. [28] proved that combining suitable data imputation with feature selection method is a better choice to deal with any incomplete and high-dimensional data resulting in useful results.

In an IoT environment, however, Kim et al. [36] proposed seamless and effective integration of machine learning and semantic technology to compensate each other for data imputation and its analysis. Nishihara et al. [37] highlighted latency and high throughput as the most important requirements for emerging machine learning applications, which keep on evolving using dynamic data-flow parallelism. Moran et al. [38] also illustrated comprehensive, highly inter-operable, reproducible and exchangeable classification methodology for ontology-based knowledge management and machine learning approaches using spatial data sources. Danylenko et al. also provided sufficient evidence towards use of different learning strategies for improving the decision accuracy [39]. The data from a multi-user environment, while dealing with missingness in real-time for the recognition of HAR is extremely challenging. Such data may also suffers delay and data loss without no mechanism of retransmission for sensors. Therefore, to meet HAR QoS, there is dire need to strengthen real-time applications by addressing the missing values and overcome unnecessary accuracy issues. We discuss various data imputation techniques in detail as presented in the Figure 2.1.

2.3.1 Case Deletion

A most common approach to deal with data missingness is "do not impute" (DNI) [16]. DNI includes list-wise data deletion, pairwise deletion, delete columns for MCAR classified data. The

List-wise Data Deletion method also known as complete-case analysis deletes records that are MCAR incomplete remaining data remains at large after removal of individual observations. This suggests, such a method does not work with small scale dataset classified as MAR and MNAR. It may lead to serious bias and inconsistency by increasing the probability for any of the remaining observations [32, 40]. On the other hand, *Pairwise Data Deletion* is a more selective method which will not delete those records with missing values but it determines the missing information based on the existing information on a case by case basis. Such method tries to minimize the data loss even if the data size is small with large samples of missing data [31]. Aforementioned approaches are naive and easy-to-use. They omit the missing data from the observations but still not an attractive choice amongst researchers as they may loose the most important information and leave the remaining data unbalanced [29].

2.3.2 Direct Imputation

The data imputation methods are broadly classified into single imputation and multiple imputation methods. The former method deals with statistical methods such as mean and mode techniques whereas the later on deals with the limitation so single imputation methods with the distribution of possible values. Following sections discuss some of the semantic imputation methods for pre-processing the data along with their semantics.

Single Imputation

Univariate or single imputation methods deal with replacing a single missing value. They are further categorized into explicit modelling and implicit modeling-based methods.

- Explicit Modeling: A widely used statistical approach which is considered to be the efficient and simplest one is mean imputation. In which missing values are replaced with the mean or mode of observed values for any particular variable. This method, however, is not recommendable though it is easy to use but produces biased results, especially when the data are not MCAR. A slightly better method named as conditional mean imputation was proposed to overcome the biassed results, it replaces missing values through the predicted value using regression model from the observed data. This model depends on the relationships between observed attributes. This method however, also underestimates variability thus not recommendable [41].

- Implicit Modeling: Hot deck imputation method is based on similar instances which are replaced by missing values. It identifies similar complete cases from the pool of similar cases and replaces missing values as a complete instance. The similarity is measured between similar cases and case with missed values in terms of the distance between its covariates. This approach is better than mean or mode imputation, however, it still introduces some bias as well as variability underestimation [42]. Cold deck imputation is similar to the hot deck, but the instances or cases used to fill the missing values are chosen from external sources, not from the current dataset [43].

  Naïve Bayes(NB) algorithm based imputation makes rational decisions using probability theory under uncertainty. It works with categorical attributes as probability computation since it can only work within the discrete domains. Such methods assume that the effect of values for given attributes of a certain class is independent of the values for any other attribute. NB imputation methods are very sensitive to usefulness and redundancy of some of the attributes. At the same time, they are more sensitive to outliers and noise, for this reason, find it challenging to deal with missing values.

  Last observation carried forward (LOCF) method is for longitudinal studies in which missing value is imputed at one-time point with the last observation in the dataset under that particular variable [31]. There is another approach similar to this is Last observation carried backward (LOCB) or Next Observation Carried Backward (NOCB). These methods, however, seem unrealistic for filling in gaps of missing data as they can potentially introduce bias as well as underestimate variability.

Little and Rubin [44] conclude that the aforementioned missing values substitution and casewise methods are proven less effective and inferior when compared with other methods a part of multiple imputations.

Multiple Imputation

Multiple imputation methods are considered to be modern statistical type methods for dealing with incomplete datasets having missing values. These are further subdivided into machine-based learning techniques, Deep Learning methods or auto-encoders, knowledge-based models and model-based estimation. All of these multiple imputation methods involve four important phases: (1) data imputation; (2) analysis; (3) pooling (4) aggregation. Each of these methods is itemized below with detailed insights:

- Machine Learning Models (ML): ML methods are modern algorithm underpinned with computational learning theory and pattern recognition. Such algorithms derive models, which are built on data for the prediction of missing values. Some of the ML methods include: Probabilistic Bayesian Model, which learns the probabilities using graphical models for categorical data. The missing values are identified by employing a recursive method and a partial augmentation of posterior means [45]; Multiple imputations was also performed through the standard regression methods, logistic regress and regression-based nearest neighbor hot decking methods [46]; For MAR multiple imputation linear discriminant analysis (LDA)-based method [47] seems to be a useful technique, however it resulted better for the large nature of datasets; Expectation–Maximization (EM) algorithm using the maximum likelihood (ML) method, an iterative technique, which is widely used to estimate model parameters to approximate the expected probability distribution of numerical datasets. Random values are selected from the dataset for missing observations to estimate the second dataset. The iteration process continues until estimates get converged to certain fixed values [48]; In order to treat MAR, a likelihood-based (EM) method was devised for the missing response problem. The values were imputed by the kernel regression imputation at the first place, which was used to construct a complete data using empirical likelihood, both the datasets and processes are handled independently. It was, however, noted that empirical log-likelihood ratio was distributed as a scaled chi-square variable asymptotically [49], which cannot be utilized for statistical inference; Some of the researchers also utilized Fuzzy C-Means clustering hybrid approach in combination with a genetic algorithm and support vector regression [50]. Their proposed method optimized the weighting factor, cluster size and fuzzy clustering parameters for estimating the missing values with high-performance results for imputation; Krishna et al. [51] minimized the error function by using mean squared error between the covariance matrix for a complete dataset versus a covariance matrix over the dataset include the imputed values. They used Particle Swarm Optimization (PSO) and calculated the absolute difference between for the determinants of both covariance matrices; Self-organizing map (SOM) uses the concept of distance object per one weight along with several other parameters from the incomplete datasets to give estimates and predict the results. Such estimation methods proved to be reasonable better and time-saving while performing imputation [52]; Buuren et al. [53] proposed and developed a software-based method using multivariate imputation by chained equations (MICE) in order to deal MAR for imputing incomplete multivariate data. In their technique, they used Fully Conditional Specification (FCS) method to separate each of conditional model for each of the observed variable; Few of the researchers involved decision tree-based classifiers to model missing variables through supervised induction [54]. Later on, similarity such as Euclidean distance, Minkowski distance or their variants were used to deal with MAR. Such distance measures are used for imputation using K-Nearest Neighbour (KNN) or Random nearest-neighbour (RKNN) imputation [55]; Similarly a kernel-based method to find the similar instances with no missing values from the complete data was proposed to deal with missing values in the target attributes using clustering [56]; Kernel methods were also proposed to deal with missing values for binary variables even without the data preprocessing methods by involving sophisticated multiple imputation techniques. These proposed techniques were supported with logistic regression for model learning [57]; An ensemble method for bagging and stacking was also proposed to deal with multiple imputations through an integrated approach [58]; Josse et al. [59] proposed a method using Principal component analysis (PCA), which provided point estimation of the variables. In their study, they also assessed the variability caused by missing values for continuous variables. However, their PCA based framework may be extended to mixed or categorical variables; Tan et al. [60] also performed PCA to capture the correlation amongst several variables, which also suggested the tensor patterns for a high volume of data. They proved their methodology by using traffic data imputation for PCA and multi-correlations; A Gaussian Process Regression [61] was also proposed to addressing the missing data imputation issues. In which a nested Gaussian imputation was suggested in a high-dimensional tensor. These regression models also handled time-based correlations among several variables to fill in the missing values by proposing a hierarchical structure.

- Deep Learning Models: Deep Learning Models/Auto-encoders neural network is also proposed as the imputation method to train the auto-encoders for better prediction of missing values [62]. They used latent space with higher dimension instead of a bottleneck layer for handling missing data. They trained auto-encoders based on the hypothesis that a complete dataset and missing values can be used to reconstruct missing data using these auto-encoders. For the reconstruction of missing data, the nearest neighbour rule was applied with precision quality enhanced through minimizing the reconstruction error. Turabieh et al. [63] applied Dynamic Layered Recurrent Neural Network (Dynamic L-RNN) to impute missing data for medical devices involved in the Internet of Medical Things (IoMT). The authors proved enhanced improvement through missing data recovery for medical cases in IoT based real application.

- Knowledge-based Models:

  Data imputation has been addressed using symbolic methods such as: (1) rule learning and (2) decision trees (3) Ontology models, which are briefly discussed as under:-

  Garcia et al. [16] deployed rule learning-based method also known as "separate-and-conquer" using covering rule algorithms for missing data imputation. This algorithm searches the data based on the query supported with the rule, which retrieves some part of the data or whole instances. Such process separates rule-based similar examples from the data through an iterative method. It can be a simpler data retrieval or may use the inference mechanism. In general missing data or incomplete retrieval can be of the nominal or discretized form. Again the performance of rule learning method is prejudiced by the outliers and noisy examples. There exists several algorithms using rule learning methods such as RIPPER, AQ, PART, FURIA, and CN2.

In addition to the above literature also exists for another similar method "Decision Trees" (DT) for imputing missing values. DTs comprise of predictive models arranged in hierarchical shape containing decision constructed through the iterative divide-and-conquer scheme. This method analyzes and splits the data into homogeneous subgroups using any of the selected independent variables. These homogeneous subgroups are translated into if-then-else rules with decisions at the leaf nodes. However, DT based missing values identification runs into the same problems and disadvantages as of rule learning. In literature, some of the well known DTs are recommended such as C4.5, PUBLIC, CART [16].

Besides symbolic models, other important Knowledge-based models are also used to find similar instances or tuples using ontologies. To better represent the data, this is modelled using ontologies, later on, different tuples are retrieved with the variables having a temporal or spatial relationship. These hierarchical models provide a better understanding to the semantics of the data, which are exploited using structural as well as syntax properties through deductive algorithms [24]. A comprehensive ontology covering a specific domain can effectively handle missing or unknown data. In literature ontology-based methods have been built for biological knowledge, which by using gene ontology can assist the system in estimating the unknown or missing values in micro-array data [32].

- Kalman filters, an advanced model-based framework, were also involved to estimate the missing values using an autoregressive integrated moving average (ARIMA) model [64]. The model predicts the missing values or data states based on the previously observed values or states. These Kalman filter-based techniques are suitable for univariate missing values handling methods, especially for univariate time-series methods. The performance Kalman filters were also studied by Hadeed et al. [35] who proved it to be best where this missingness of the data is low to moderate. It was also suggested that Kalman filters based imputation yielded exponentially best performance with strong trends and to be considered most viable for univariate time-series data of missing nature.

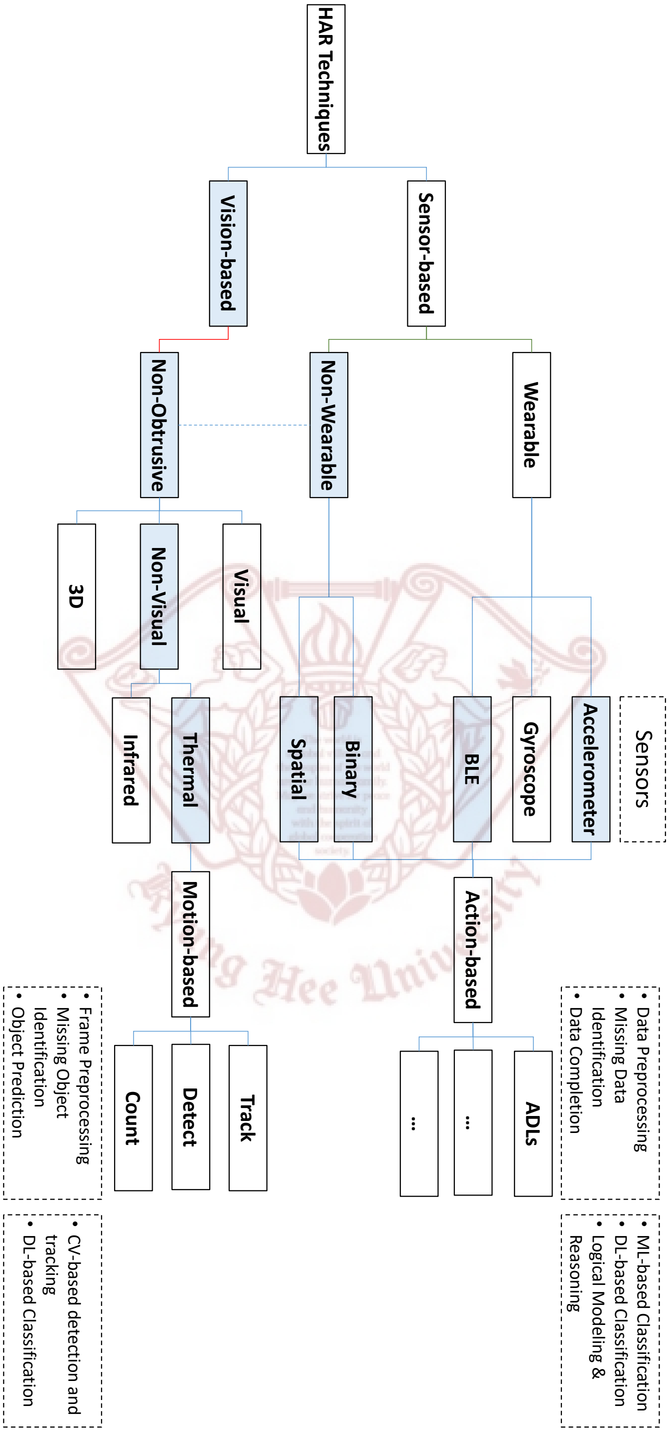


Figure 2.2: Human Activities Recognition Techniques using Obtrusive and Unobtrusive Sensors.

2.4 Sensing Technologies for Unobtrusive HAR

In this section, we present an up-to-date discussion of the state-of-the-art in activity recognition. We describe various existing approaches and their underlying sensing technologies. In particular, we discuss and analyse semantically the impact of missing states for unobtrusive sensors for recognizing human activities with their collected public datasets. A special emphasis is also placed on the unobtrusive vision-based thermal sensor for capturing and predicting missing multioccupant states for recognizing their activities. For this, we consider public dataset, as well as, we also collected HAR dataset using thermal vision sensor under smart-home controlled environment. We discuss each one of them one-by-one in the following subsections.

2.4.1 Background for Unobtrusive Sensing

A wide variety of applications are underpinned with state of the art Machine Learning (ML) algorithms for recognizing ADLs in smart environments using obtrusive and unobtrusive sensors. The wearable devices, also called as obtrusive devices, are most commonly engaged by the users for Activity Recognition (AR), however, such devices may not be practically applicable for long-term use because of their maintenance cost, battery life, and discomfort caused by continuously wearing them. This may also lead to the noisy and imprecise state, causing an erroneous classification and recognition. This study explores how to recognize activities based on the available sheer amount of discrete and continuous multimodal data produced by obtrusive, in particular with unobtrusive sensing devices. Above-mentioned factors affect the performance of ML-based AR from multimodal sensory data sources thus the appropriate solution is required, which can lift the performance of the ADLs in monitoring applications [65].

Because of the promising features of unobtrusive non-wearable sensing devices to recognize human pervasive activities using smart-home applications [66], this study gives a brief overview and usage of human identification technologies categorized namely as, object-based, footstep-based, body shape-based and gait-based identification technologies. Among all, the first type of unobtrusive human identification uses a signal pattern of interaction with an object. The second type of identification strategy uses footstep's pressure, their patterns, sounds, and vibration to identify the ADLs around the home. The third category, body shape-based human identification

Table 2.1: Unobtrusive sensing technologies applied as non-wearable sensor [2, 3]

	Object-based technologies	**Footstep-based identification**	**Body shape-based identification**	**Gait-based Identification**
Sensing technologies	• Pressure Sensor • RFID • Accelerometer	• Sensor switching • Microphone • Pressure sensor • Electromechanical film • Accelerometer • Piezoelectric • Transducers • Photo-interrupter	• Ultrasonic	• Passive infrared • RF Transceiver • Electric Potential Sensor • Wi-Fi Transceiver
Features	• Object use pattern • Object use acceleration	• Walking Pattern • Footstep Sound • Footstep induced vibration • Centre of pressure trajectory • Geometric and holistic information	• Height • Width • Area • Perimeter • Radius	• Body Heat Emission • Disruption of RF & Wi-Fi Signals • Body Electric Charge Changes

captures an individual's information based on their body shape, height, and width using an ultrasound technology. Lastly, gait-based technologies use passive infrared (PIR) detector and Wi-Fi PIR to observe the human body heat emission to recognize the individual and ADLs. Some of the further details and features for the aforementioned technologies are mentioned in Table 2.1.

It appeared very challenging to identify public datasets which cover the unobtrusive HAR in particular. For this in this section, we describe the nature of available HAR public datasets $\mathcal{D}_n$ with underlying sensing technologies. Throughout the study, these sensing technologies are differentiated into two broad categories of *unobtrusive* and *obtrusive* activity sensing based on the wearables and data sources. We, therefore, provide a brief description of both categories using *UCamI* [1], *Opportunity* [26], and *UCI-ADL* [27] public datasets for their distinct sensing functionalities, signal type, sampling frequencies, and protocols. The *UCamI* dataset is shared by the University of Jaén's Ambient Intelligence (UJAmI) Smart Lab [67]. The UJAmI Smart Lab measures approximately 25 square meters divided into five regions: entrance, kitchen, workplace, living room and a bedroom with an integrated bathroom. We also illustrated sensing unobtrusiveness by using highly imbalanced Opportunity dataset covering a wide range of realistic daily life activities. The Opportunity dataset is very complex with around 75% missing values, which has been widely explored by the research community for HAR [26]. The dataset uses common unobtrusive sensing modalities such as a magnetometer, reed switches, proximity infrared, and RFID. Additionally, it also uses on-body accelerometers or gyroscopes. In addition to the above mentioned public datasets, we also tested our proposed methodology over *UCI-ADL* [27]. It provides recognition of 22 activities covering functional mobility through obtrusive as well as unobtrusive monitoring technologies.

2.4.2 Learning Models for Unobtrusive Sensor-based HAR

The use of conventional Machine Learning (ML) for recognizing human movements is well-documented research in the literature. Most of these ML methods rely on pattern recognition approaches and use heuristics or hand-crafted features for training HAR models [68, 69]. Researchers have used various approaches for HAR. These presented approaches utilized wearable, non-wearable, device-free or hybrid sensing devices. The usage of numerous conventional ML algorithms have become an essential part of HAR since long. Some of them include Support Vector Machine (SVM), Random Forest (RF), k-Nearest Neighbor (KNN), Naive Bayes (NB), Hidden Markov Model (HMM), and Decision Tree (DT). Most of the HAR methods use supervised learning, which requires training data for model generation, to be required for labelling new incoming data. In this study, we consider only those set of HAR, which are mentioned in public datasets gathered through sensors falling under the category of unobtrusiveness or in the device free environment. It includes benchmarks *Opportunity* [26, 70], *UCI-ADL* [27], *UCamI* [67], and *Thermal sensor frames* [71] datasets.

The HAR datasets mostly suffer from missing data or most often there exists imbalanced amongst classes [72]. As some of the classes may include a large number of tuples whereas others may have fewer ones. The *Opportunity* dataset too is extremely unbalanced with around 70% of data containing *NULL* class [73]. They filled missing values of sensors for 18 classes by using linear fitting however, it has proved to be less optimality properties than maximum likelihood. Yang et al. [74] proposed DNNs, built on network architectures in combination with convolutional and non-recurrent layers to the HAR domain. They used raw signal data for non-hand-crafted feature extraction and considered *NULL* class due to its dominance in the *Opportunity* dataset. Ordóñez et al. [75] preprocessed sensor data using linear interpolation to fill in the missing values and performed interval-based per channel normalization to [0,1]. They evaluated *Opportunity* dataset using *DeepConvLSTM* for gesture recognition and proved performance improvement without the *NULL* class as compared to results reported in the study [74].

Iwana et al. [76] demonstrated the effectiveness of their *Dynamic Time Warping* algorithm supported by dynamic weight alignment. They evaluated their strategy quantitatively and compared it with baseline methods using the UCI-ADL recognition dataset for 7 classes. In the work, Salguero et al. [77] proposed the ontology-based data-mining technique by using the class expression learning (CEL) for ADL recognition. For this, they used *UCI-ADL* datasets. They converted the ontological dataset into ontology models before recognition of several activities from different datasets. The results obtained through the proposed ontology only describe ADLs as sequences of events, they do not, however, considered any mechanism to deal with the missing sensor states or data imputation methods. In the methodology proposed by [78], they used temporal *UCI-ADL* data mining technique to identify associations and extracted potential relations among different entities, which were utilized to produce a network of hypothesised causal relations. The class imbalance and missing sensor states were also highlighted by [79], which lead to inappropriate results without obtaining optimal results. Razzaq et al. [21] dealt class imbalance and missing sensor states using the semantic methodology. In their study, they proposed a hybrid approach using ontology modelling and reasoning to increase ADL accuracy by using dataset [67]. They also provided a detailed view of the importance of unobtrusive sensing technologies with innumerable interdisciplinary ADL applications.

Despite conventional ML/DL methodologies, there exit plenty of approaches, which utilize semantic web technologies underpinned with ontologies to enable and facilitate activity recognition. In the field of pervasive computing, researchers have also used rule-based approaches in which domain experts make a decision on manually created rules. In such approaches, researchers utilize the concept of a domain knowledge-driven approaches to activity recognition [80]. For these, they adopt ontological models as the conceptual backbone covering the life-cycle of activity recognition in a sensorised smart-home environment. The compelling feature of the proposed approach is that activity recognition is performed through direct semantic reasoning making extensive use of semantic descriptions and domain knowledge. In this, thesis we cover various unobtrusive sensors along with their characteristics for progressive activity recognition at both fine-grained and coarse-grained levels.

In particular, HAR accuracy has always been a challenge as there always exits missing data while dealing with real-time or even offline classification. The drawback of both the pattern recognition approaches and the rule-based approaches are overcome by using Deep Learning (DL) methods. This study also suggests semantic imputation methods using ontology modelling [81,82] based on HAR public datasets before applying DL methods for recognizing activities. The robustness and accuracy is proved in such cases which are discussed in later sections.

2.4.3 Review of existing Methods for Unobtrusive Vision-based HAR

Vision-based HAR most often involves camera-based sensing facilities for monitoring occupant's behavior. Such facilities are usually equipped with camera-based visual modalities to capture images, camera-based non-visual modalities such as thermal or infrared devices and 3D depth, multiview or skeletal capturing devices [83]. All of these modalities generate visual data of the form such as digitized video frames. HAR using these frames, however, is performed by employing various computer vision approaches and techniques involving several steps such as segmentation, feature extraction, and occupant detection with their movement tracking by observing their patterns [84]. The importance of such approaches cannot be denied even for the non-visual sensors such as thermal sensors or infrared equipped sensors. As we are dealing with thermal vision sensors which can be used as a single unit or in multiple combinations for investigating the diversity of ADLs. These approaches have been studied by several researchers who have published their outcomes over the years to detect and track multiple people in a smart environment.

Nigam et al. [85] discuss several reasons as to how the detection of human activities is a complicated task using computer vision. They identified various reasons such as variable object shapes, occlusion, abrupt motion, luminance issues, occlusion, real-time analysis and action recognition. These issues have been investigated by several researchers and have proved to be an important area of research. Several indoor personal localization and their recognition techniques using neural networks have been discussed in a comprehensive survey [86]. In such a work Kawashima [87] proposed a Deep Learning-based approach using a grid of 16×16 infrared thermal sensor array sequence for daily life action recognition. The authors combined CNN structures for feature extraction with long short term memory (LSTM) for obtaining spatio-temporal representation. It is worth pointing out that significant progress has been made in recognizing ADLs with visual monitoring. The information collected is rich and intuitive. Activity recognition, however, suffers from issues relating to ethics and privacy [88] with these camera gadgets, which are being perceived as event recording devices. Murakami et al. [89] used 8×8 RGB converted frames of infrared array sensors for posture classification and residents monitoring while keeping protecting their privacy. They promised a higher accuracy for posture classification by using Deep Convolution Neural Network (DCNN). By using low-resolution 4×16 thermal image frames, dynamic changes due to shape changing effect from the sequence of frames were also studied by Shelke et al. [90] for classifying indoor activities. They used conventional machine learning classifier and performed a frame-wise classification. An automatic occupant detection and tracking is challenging and are the most important elements in most of the computer vision applications. The tasks related to action recognition for these detected occupants are also widely studied in these applications, for these numerous tracking algorithms are proposed [91]. These algorithms are broadly categorized into two main groups: (1) Learning-based; (2) Model-based. Learning-based or feature-based methods mostly use training frames to extract discriminative features for recognizing gestures or actions. Model-based methods, however, uses point tracking for deterministic or probabilistic based methods, kernel tracking-based methods for multiview or template-based methods and silhouette tracking using contour matching or shape matching [92]. In point tracking, most of the researchers used Kalman Filtering algorithm to track point having low-to-moderate computational time, recursive Bayes filtering for handling occlusion and complex background but higher computational time. Lastly, multiple hypothesis based point tracking algorithms have also been introduced for tracking objects. In this thesis, we focused on point point-based tracking methods using Kalman Filtering algorithm as other methods have high computational cost as well as compromised accuracies.

2.5 Robust Methods for Semantic Imputation

Over the past, several machine learning-based statistical methods have been introduced and employed for addressing the incompleteness of datasets by approximating the missing attributes and their values [93]. An extensive research is available to address missing data issues, the solutions for imputation are, however, generally categorized as (1) single imputation; (2) Machine-learning based imputation and (3) multiple imputation methods [41, 93]. In a single or univariate imputation method, a single missing value for each attribute is replaced with imputed one. For this, various statistical single imputation methods are proposed in the research, which includes mean and mode method, where missing values are replaced with mean or mode of observable values for the same variable [94]. For multivariate attributes, the relationship among various attributes is estimated supported by the regression coefficients by using regression-based imputation methods [95]. Another method referred to as hot-deck uses probabilistic information from a similar set of observations to replace missing data [45]. Finally, an iterative method expectation-maximization with each step containing expectation for estimating missing values and by likelihood maximization was also introduced for imputing missing values in the incomplete numerical datasets [96].

Machine learning (ML) based imputation methods are dependent on predictive learning models, which are built on observed data for estimating the missing values for target attributes. There exist top six most used ML-based imputation methods such as: (1) clustering; (2) decision trees; (3) k-Nearest Neighbors (k-NN); (4) Random Forests; (5) Support Vector Machines (SVM); and (6) Artificial Neural Networks. The cluster analysis technique uses unsupervised learning for categorizing similar objects in the same clusters. To impute the missing values a distance-based measure is applied over the missing data and cluster centroids [97]. Decision Tree models the data into a tree-like structure represented by internal nodes for test attributes and leaf nodes representing classes whereas each branch represents the outcome of target missing values [98]. About k-NN, which is a form of supervised learning in which a distance is calculated between observed values and target missing values [99]. Various distance metrics are used in the literature under k-NN based imputation, which includes Hamming distance, Euclidean distance, Minkowski distance or Manhattan distance. On the other, hand Random Forests based imputation identifies important variables with their masking scores, ensembles the variables and adjust the scores for target missing values. Support Vector Machines based imputation using learning theory concepts for the estimation of missing condition attribute values where the data is linearly separable [100, 101]. Furthermore, Artificial Neural Networks provide powerful mathematical models for estimating missing values using probabilistic approximation using neuron, layers, activation functions and weights [102].

In addition to above ML-based methods these, a plethora of extended ML-based imputations algorithms are also presented and discussed in [16], which include: (1) Imputation with K-Nearest Neighbor (KNNI): KNNI computes a missing value by taking the average of corresponding numeric entries in the example expression vectors whereas for the nominal values most common value is taken for imputation [103]. (2) Weighted Imputation with K-Nearest Neighbour (WKNNI): The WKNNI method selects different distances strategies as in the case of KNN. It uses weighted mean or frequently repeated value as a similarity measure for missing value imputation measure [104]. (3) K-means Clustering Imputation (KMI): In KMI, the intra-cluster dissimilarity is measured by the summation of distances between the objects and the centroid of the cluster they are assigned to. A cluster centroid represents the mean value of the objects in the cluster [100]. (4) Imputation with Fuzzy K-means Clustering (FKMI): In order to extend the original K-means clustering method to a fuzzy version FKMI, a membership function is added to describe the degree of missing data to which it this belongs to [105]. Such a FKMI approach provides a better description of clusters where they are overlapped. (5) Event Covering (EC): EC a three-decade old algorithm, works based on a mixed-mode probability model, which is approximated by a discrete one. Such a method does not require scale normalization or the order of discrete values [106]. (6) Singular Value Decomposition Imputation (SVDI): SVDI based imputation utilizes a set of mutually orthogonal expression patterns that are identical to principal components of data values named as eigenvalues and these are linearly combined approximate values for all the attributes in a dataset. The most significant eigenvalue is identified through sorting them and finding the corresponding ones [101]. (7) Local Least Squares Imputation (LLSI): LLSI represents missing values with target gene which is a linear combination of similar genes. This method uses only similar genes by computing local least square similarity measure in which L_2-norm is used lead by regression and estimation [107].

Lastly, the third category to impute the missing data involves multiple imputation method is again a statistical-based technique originally proposed by Rubin [108]. It is mainly aimed at solving the limitations of the previous two categories i.e. single imputation method and machine learning-based method especially the former one. In multiple imputation methods, each missing value is treated with multiple imputations, which can be two or more acceptable values representing the distribution of possibilities [109]. Thus, multiple imputation method promises better results in terms of modelling uncertainty in three steps: (1) imputation of missing values, (2) statistical analysis on imputed datasets, and (3) pooling of results across multiple imputed datasets. Imputation of missing values may involve specific statistical machine learning technique such as single or multiple regression models like linear, poison or logistic as described in [110]. These underlying techniques are effective, however, are complex, higher computational cost and large data storages for analysis and pooling for multiple imputations as it involves the construction of *N* number of imputed datasets and performs multiple substitutions to obtain a single final dataset [111]. Another important aspect has to be considered while performing multiple imputations is selection of *N* as how many imputations have to be performed, which is usually take as 3 or 5 [112]. It has been proven by Liu et al. [113] that imputation provides robustness to the model learning, which leads to the increased performance for simple and complex scenarios.

All of the above-mentioned techniques have their own merits and demerits. Semantic imputation based techniques, however, utilize the prior knowledge of sensors and sensory data before applying them for some applications. Building semantic knowledge from the human activity data requires domain knowledge, and applying it further for identifying missing values for semantic imputation is still a challenging problem and requires further studies [114].

Regardless of the method used, imputation is considered both an essential and sensitive step of data preprocessing [16], especially for HAR and ADLs. This clearly affects the performance of the HAR and ADLs in a smart-home environment further explored by data mining task [44].

2.5.1 Review of existing Sensor-based Semantic Imputation Methods

Unobtrusive sensing enables continuous monitoring of activities and physiological patterns during the daily life of the subject. These wearables most often involve binary sensors (BinSens), PIR sensors, and pressure sensors embedded within smart objects or the ambient environment. *BinSens* generate an event stream comprising of binary values, working on the principles of the Z-Wave protocol. Such protocols are implemented through some unobtrusive wireless magnetic

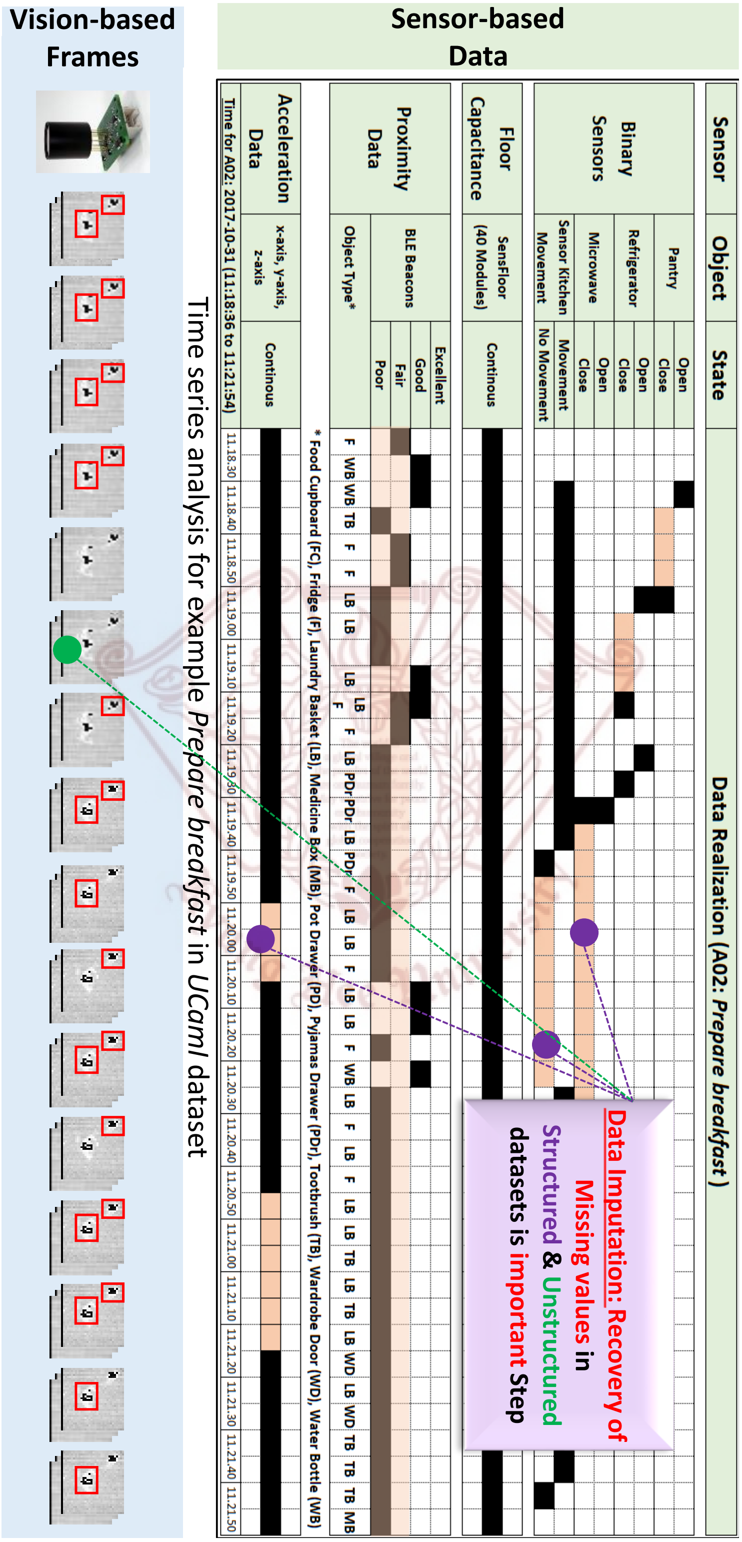

Vision-based Frames
Sensor-based Data
Sensor
Object
State
Data Realization (A02: Prepare breakfast)
Binary Sensors
Pantry
Open
Close
Refrigerator
Open
Close
Microwave
Open
Close
Sensor Kitchen Movement
Movement
No Movement
Floor Capacitance
SensFloor (40 Modules)
Continous
Proximity Data
BLE Beacons
Excellent
Good
Fair
Poor
Object Type*
F WB WB TB F F LB LB LB LB F F LB PDr PDr LB PDr F LB LB F LB LB F WB LB F LB F LB LB TB LB TB LB WD LB WD TB TB TB MB
* Food Cupboard (FC), Fridge (F), Laundry Basket (LB), Medicine Box (MB), Pot Drawer (PD), Pyjamas Drawer (PDr), Toothbrush (TB), Wardrobe Door (WD), Water Bottle (WB)
Acceleration Data
x-axis, y-axis, z-axis
Continous
Time for A02: 2017-10-31 (11:18:36 to 11:21:54)
11.18.30
11.18.40
11.18.50
11.19.00
11.19.10
11.19.20
11.19.30
11.19.40
11.19.50
11.20.00
11.20.10
11.20.20
11.20.30
11.20.40
11.20.50
11.21.00
11.21.10
11.21.20
11.21.30
11.21.40
11.21.50
Data Imputation: Recovery of Missing values in Structured & Unstructured datasets is important Step
Time series analysis for example Prepare breakfast in UCamI dataset

Figure 2.3: Time series analysis for example *Prepare breakfast* in *UCamI* dataset [1].

sensors. This can be explained through the *Prepare breakfast* example in Figure 2.3. For 'Pantry', 'Refrigerator', and 'Microwave' objects, *Open* state means magnets are detached and they are in use, whereas *Close* state shows they are not in use. The inhabitant's movements are recorded at a sample rate of 5 Hz, using the ZigBee protocol implemented in 'PIR sensors' such as the 'Sensor Kitchen Movement' [115]. It also produces binary values with *Movement* or *No Movement*. The presence of an inhabitant on the 'Sofa', 'Chair', and 'Bed' objects are collected via the Z-Wave sensing protocol, implemented through the 'Textile Layer Sensors', which produce binary values *Present* or *Not present*. Similarly, a continuous stream of data are also observed for unobtrusive spatial data gathered through the suite of capacitive sensors installed underneath the floor.

The dataset generated through the *BinSens* is of a challenging nature as the duration of the generated stream may be instantaneous, lasting for a few seconds or may continue for hours. As shown in Figure 2.3, filling the gaps between two states for *BinSens* is of a challenging nature since every *BinSens* has a different operation nature and state transition time depending on the activities performed. The proximity data from the Bluetooth Low Energy (BLE) beacons is collected through an android application installed on the smart-watch at a sample rate of 0.25Hz [115]. BLE beacons are measured through RSSI. The value of the RSSI is higher if there is the smaller distance between an object and the smart-watch and vice versa. BLE beacons are used for 'Food Cupboard', 'Fridge', 'Pot Drawer', etc., for the *Prepare breakfast* activity example in Figure 2.3. Ambulatory motion is represented by *Acceleration* data, which is again gathered through the android application installed on the smart-watch. The 3D acceleration data are collected in a continuous nature using a sampling frequency of 50Hz. Such acceleration data [26] is also measured through body-worn sensors, object sensors and ambient sensors, which measure 3D acceleration using inertial measurement units, 3D acceleration with 2D rate of turn and 3D acceleration with multiple switches, respectively.

2.5.2 Review of existing Vision-based Imputation Methods

The foundation of most of the visual tracking systems is built on generally three functional models [116]: (1) Appearance model, it characterizes and distinguishes between objects and non-objects; (2) motion model, which detects object's motion trail and predicts future location; (3) and searching strategy to find the object in the video streams. Many methods have been proposed in the literature to deal with appearance and motion models. An online model learning algorithm, however, is required to deal with dynamic appearances and trajectories. To cope with the aforementioned challenges several strategies have been proposed depending on features required to build appearance models. These features are also necessary to address the searching strategies as well which also play an important role in searching for missing objects. Kalman Filter [117] has proved to be an optimal tracker for practical applications by ensuring a good compromise between computational complexity and tracking performance. It uses a series of inaccurate and noisy data observed over time to estimate unknown variables by ensuring increased accuracy [118]. For each input tracklet, such noise is alleviated from inaccurate detections by refining the positions and by estimating the sizes and velocities of its detection responses of the shape bounding boxes [119]. For a higher accuracy and minimal computation cost, the bounding boxes are considered as small as possible by the Kalman filter [120]. In literature, there are numerous real-world applications, which show that the Kalman filter allows a suitable treatment of incomplete and noisy data [121]. Especially in frame-based detection and prediction Kalman filter is used to predict the location of an object in the current frame from the previous one. The system, however, also compares the previously detected position with the newly predicted one [122]. Many researchers also focused on the spatial features of multiple objects required for their tracking [123]. However, prediction in such cases become more challenging. Some of the traditional methods developed earlier use frame-by-frame for making the prediction of objects by involving multiple hypothesis tracking (MHT). Some of the researchers also employed joint probability data association filter (JPDAF) in which a single state is generated for prediction purpose using relationship among individual measurement and likelihood associations. However, literature shows both MHT and JPDAF are computationally expensive for identifying missing objects during detection and tracking.

Despite all the efforts, however, such methods still consider two problems such as target initialization and how to deal with missed targets while tracking. Most recently Dimitrievski et al. [124] proposed vision-based imputation methods to deal with the missing data on the detector as well as tracking side. They adapted imputation theory to recover from lost information through lower-level likelihood information in cases of missing detections. They proposed multiple particle filters for handing missing detections and estimated the missing detection information by using imputation proposal function. In this thesis, we are intended to deal with the missed targets and predict the new location as to perform vision-based imputation.

2.6 Summary of literature

2.6.1 Complex HAR literature

In the past few decades, a rapid rise in the advancement of pervasive computing has been observed through gathering Activities of Daily Livings (ADLs). This is generally carried out in a controlled environment equipped with the inexpensive wireless sensors. A wide variety of applications are underpinned with state of the art machine learning algorithms for recognizing ADLs in smart environments. In such environments, the real-world streaming dataset is almost of the same content as near-duplicates [125]. This leads to the noisy and imprecise state, causing an erroneous classification and recognition. On the other hand, it also becomes infeasible to perform comprehensive data cleaning step before the actual classification process. This study explores how to recognize activities based on the available sheer amount of discrete and continuous multimodal data. In order to recognize the activities, time-based sub-window re-sampling techniques were adapted, as they keep a partial order in the multi-modal sensor data stream for recognizing each individual activity [126, 127]. The sampling techniques [128] have widely been used to handle approximate results by accommodating the growing data size. These samples form a basis for statistical inference about the contents of the multimodal data streams [129]. As relatively little is currently known about sampling from the time-based window [130] and is still a nontrivial problem [131]. It is pertinent to mention that sampling becomes more challenging if the sensory data is of a highly dynamic nature [132] for the activity recognition models. Moreover, high data arrival rate can also suppress the robustness of such models as near-duplicate speeding data may get multiple re-asserts [133]. However, some research studies also suggest a compromise be made between efficient sampling rates for such dynamic nature [130]. As above mentioned factors affect the performance of Machine-Learned (ML) based activity recognition from multimodal sensory data sources thus appropriate aggregation strategy can lift the performance of the

ADLs monitoring applications [65]. The selected aggregation strategy for data-level fusion determines the way in which multimodal data reach the fusion node [23]. At the fusion node, different ADLs can be best recognized by selecting the appropriate fusion strategy with variable window lengths [24, 134, 135]. Current tendencies prove the acceptance of high attention by sensing modalities for improving the recognition of ADLs performance and overall application robustness [136, 137].

2.6.2 Unobtrusive Sensing literature

Multi-object Tracking (MOT) in CV domain has been studied for decades and has attracted a lot of research attention. It is, however, still far from solved regarding HAR [138]. Many solutions exist for HAR in a controlled environment. These solutions mostly involve the deployment of numerous wearable and pervasive sensors [139], which can lead to increased cost, privacy concerns and more often an inconvenience. To alleviate these challenges, attention of the research community has directed to low-cost unobtrusive sensors [140].

TVS are an excellent candidate for pervasive sensing due to their inexpensive nature, portability, limited maintenance requirement and lower privacy issues compared to traditional cameras. Hevesi et al. [141] have illustrated that such a sensing modality can be deployed for indoor HAR and monitoring of sedentary behavior of a single occupant in an office environment. Solutions based on TVS mostly require CV based approaches for locating moving objects by identifying them as a region of interest (ROI) in a frame sequence. Detection of an ROI is deemed as the first step in most CV-based applications [142]. It may involve various techniques such as: (1) thresholding, which yields low accuracy and is of lesser use in current applications [143]; (2) multi-resolution processing which faces challenges for detecting objects during congestion [144]; (3) edge detection which has challenges in deriving an ROI where the shape of the object is highly dynamic; (4) inter-frame differencing which uses consecutive frames for detecting an ROI but can only be considered for a sequence of shorter duration [145]; (5) an optical flow-based detection which requires a large number of frames resulting in poor performance; (6) background subtraction which extracts objects not belonging to the background, however, this technique requires a static background as an initialization.

2.6.3 Semantic Imputation literature

While most of the attention and research focus has been given to the development of machine learning (ML) imputation methods, they do not handle the semantics of data well at all. Besides ML algorithms statistical methods have also been proposed to deal with the missing data. In contrast to statistical methods, the ML algorithms perform classification for missing values using the trained model, which is built on the complete data. Several algorithms were built such as rule-based methods, probabilistic models, or decision trees, however, their underlying methodology remained the same [146]. Data imputation for HAR is also not widely studied research topic, which we aim in this study by keeping the semantics of data after imputation.

Recognition of ADLs has been undertaken across a wide variety of applications including cooking, physical activity, personal hygiene, and social contexts. Generally, solutions for recognizing ADLs are underpinned with rule-based or knowledge-driven supported by conventional Machine Learning (ML) algorithms [74, 147]. In such environments, the embedded or wireless sensors generate high volumes of streaming data [134], which in a real-world setting can contain huge amounts of missing values or duplicate values [125]. Such noisy and imprecise data may lead to one of the major causes of an erroneous classification or imprecise recognition. Conversely, several challenges also exist while coping with missing values hence an efficient mechanism for imputation of the sensory data are thus required. Issues in missing data become even more difficult when considering multimodal sensor data to recognize real-time complex ADLs. In this case, some of the sensors may generate continuous streams of data whilst others generate discrete streams [7].

Several statistical-based approaches are reported in the literature to deal with missing values. The majority of these propose data imputation solutions, the nature of which can vary depending on the size of the actual data and the number of missing values [148]. Most of them, however, use model-based imputation algorithms i.e., likelihood-based or logistic regression to encounter the missing values. The impact of imputation is determined by the classification performance, which may lead to biased parameter estimates, as most of the ML classifiers deal with the missing information implicitly. For this reason, complications whilst handling missing sensor states is still considered to be a non-trivial problem [146]. An appropriate strategy is therefore needed to improve the quality of data imputation with minimal computational efforts. Current approaches must also address data imputation in multimodal sensor streams, which not only improves the recognition performance but also increases overall robustness of the applications [149, 150].

Despite the gain in statistical power, more recently, ontology-based modeling and representation techniques have been introduced [151]. These ontological models can discover, capture, encode rich domain knowledge, monitor patterns of ADLs, and provide heuristics in a machine-processable way [152, 153]. Ontologies represent rich structured hierarchical vocabularies and can be used to explain the relations amongst concepts or classes. The coded knowledge is made accessible and reusable by separating sub-structural axioms, rules and conjunctions among the concepts [154]. In addition to separation logic, use of a query language, SPARQL also provides support for disengaging these semantics and assertions for interpreting any rule-based complex activities [155]. In work by Amador et al. [156], the authors used SPARQL for retrieving class entities and their types, which were later transformed into vector form before using deep learning approaches. Similarly, Socher et al. [157] have bridged neural networks with an ontological knowledge-base for the identification of additional facts. Only a limited amount of work, however, has been undertaken to account for semantic imputation using ontological models and SPARQL [158].

2.6.4 Review of existing methods for Vision-based HAR

Regarding MOT various techniques [159] have also been proposed by the research community. These techniques focus on addressing common challenges such as frequent occlusions, identical appearances, track management and interaction among objects. No single approach currently exists which can address all of these challenges. MOT in any visual tracking system usually involves three functional models [116]: (1) appearance model, which describes the object and distinguishes it from the non-objects; (2) motion model which characterizes the current and predicts the future states of an object by tracking their trajectories; (3) searching strategy which helps to identify and match an object based on the appearance model in a frame sequence.

Motion models have gained significant attention for object state estimation. They operate by producing accurate motion affinity models in a linear motion space, which can be used to predict object position [160]. Thus, it reduces the search space by capturing the dynamic behavior of the object. To solve the linear tracking problem, where continuity of moving objects is not abrupt, Kalman filtering (KF) is often used [161]. This approach can track moving objects using their center of gravity [162]. KF is a linear state-space motion model proved to be an optimal tracker suitable for practical applications. It promises a good compromise between computational complexity and performance for object tracking by utilizing a point-based approach in learning statistical features [163]. It uses identified features and uncertainty information to estimate different states of an object through the successive frames. KF may, however, experience object drifting due to the loss of an object's appearance information in a frame sequence. The object drifting complexities require efficient object refinement schemes to analyze object motion properties leading to proper data association [164]. Yilmaz et al. [160] addressed some of the issues and complexities related to data associations through a joint solution for state estimation. Choi et al. [165] formulated the problem of multi-occupancy and resolved it through multiple target tracking. They merged the problems of HAR and tracking into a single probabilistic graphical model for tracking individual actions. Similarly, an adaptive framework was also proposed by Shen et al. [166] to identify the correct state of the targets. They suggested the use of an adaptive detection algorithm for MOT task to refine the detection targets and minimize the detection errors. In order to classify Activities of Daily Living (ADLs), it has been observed that CNN have shown superior performance over the traditional Machine Learning (ML) approaches such as Support Vector Machines [167] and feed-forward neural networks [168]. The visual object recognition tasks [169] can be performed over the raw low-resolution TVS frames using CNN, which is easier to train by adjusting a few parameters and inter-layer connections. It extracts meaningful features without requiring domain knowledge and with minimum preprocessing over a stacked sequence of frames [170]. The CNN model has the capability to extract multiple motion features encoded in the adjacent TVS frames for automatic classification of ADLs [171].

The current work is closely related to [9] in which the authors proposed a system for indoor player tracking captured through the thermal camera at a sports arena and pedestrian tracking in a courtyard. Ray et al. [172] proposed a detection algorithm, which does not depend on any prior background knowledge for object detection and also does not require initialization. Similarly, Leira et al. [173] considered the problem of small unmanned aerial vehicles equipped with thermal cameras for real-time target detection and tracking at sea using the KF based technique. Tiwari et al. [92] highlighted the research gaps for video-based HAR. They suggested designing an approach to improve the robustness of the detection and tracking algorithms by increasing the number of occupants, which can be tracked over a sequence.

The purpose of this study is to propose a framework for moving object detection, tracking and classification of ADLs with increased performance using low-resolution thermal video frames. To achieve this goal, an implementation using a KF was devised by building a robust object appearance model with morphological feature refinements [174]. It also involves the Hungarian algorithm for data association per frame [166]. Additionally, this study evaluates the robustness of the integrated framework to detect and track ADLs of the users using low-resolution TVS. For this, the solution was tested using a comprehensive experimental analysis drawing quantitative and qualitative comparisons. Robust tracking systems, such as [175], mostly involve an appearance and motion model to track the candidate states of the target. Computational complexity, however, increases proportionally with the increase in the number of targets to be tracked [176]. Therefore, joint optimization is essential for MOT. Most MOT research focuses on tracking-by-detection methods, however, an extension to it, by classifying the activities may result in boosting the overall effectiveness of these methods.

Chapter 3

Multi-strategy Data Imputation

3.1 SemImput Methodology

In this Chapter, we demonstrate the proposed methodology for Sensor-based semantic imputation, an overall functional architecture and a workflow in Section 3.1.1. An ontology model to represent the activities is presented in Section 3.1.2 and a detail of specially designed SPARQL queries for semantic segmentation in Section 3.1.6. Ontology-based complex activities identification and conjunction separation for semantic data expansion is explained in Section 3.1.7. An algorithm to perform semantic imputation is then described in Section 3.1.8. Lastly, the classification method describing HAR using DL based ANNs is presented.

3.1.1 High-Level Overview of the Multi-strategy Data Imputation Method

The presented work describes a layered Semantic-based Imputation (*SemImput*) method, which supports an innovative means to synchronize, segment, and complete the missing sensor data. This is achieved by automatically recognizing the indoor activities within the smart environment. The architecture depicted in Figure 3.1 comprises of (a) *Data Sensing and Representation Layer* designed to capture data; (b) the *Semantic Segmentation Layer* segments the data based on the timestamps for over 1-second; (c) the *Semantic Expansion Layer* segregates the concurrent activities represented by separate features into a sensor event matrix; (d) the *Semantic Imputation Layer*, responsible to fill the missing data, sensor states, which are of periodic nature and provides continuity to the data by using the proposed strategies; (e) the *Semantic Vectorization* receives the filled sensor event matrix and generates vector sets; (f) and finally the *Classification Layer*, which uses a neural network to classify the augmented Semantic Vectors for evaluation purposes.

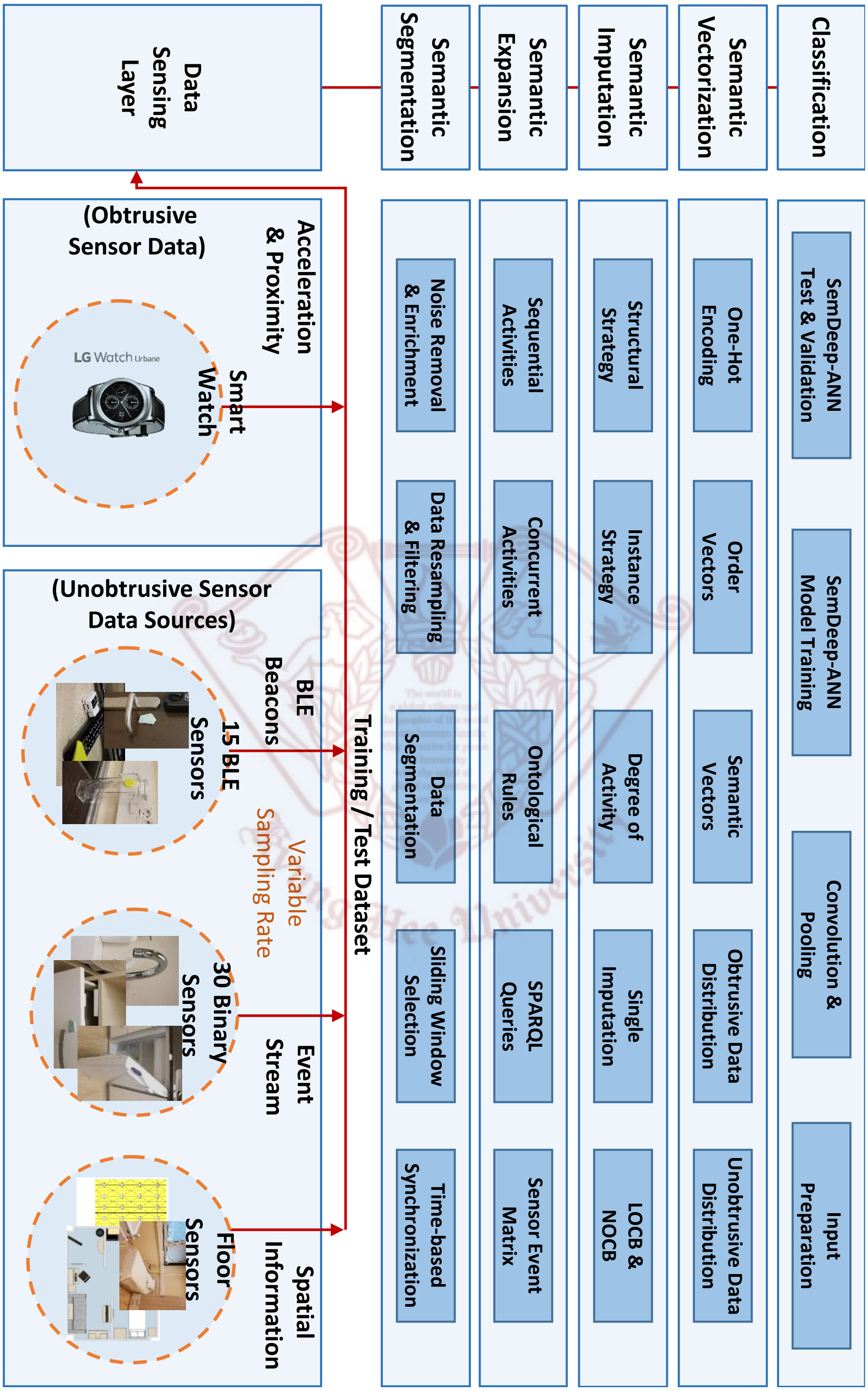


Figure 3.1: A detailed view of *SemImput* framework.

3.1.2 Data Sensing and Representation

The *Data Sensing and Representation* layer utilizes the sensor streams which are simulated over a dynamic sliding window. We used ontological constructs, which are derived through the data-driven techniques for representing sequential and parallel activities. This layer is encapsulated by the newly modeled set of OWL2 Semantic Imputation Ontologies (*SemImputOnt*) to map sensory data. It models sensor streams, identifies patterns, and discovers the overlapping temporal relations in them. It supports generality in terms of data semantization [177], offers more expressiveness, and helps in decoupling the concurrent fragments of sensor data rather than using non-semantic models. It not only provides a basic model for representing the atomic and complex ADLs but also supports the expansion of dataset instances through the SPARQL queries.

3.1.3 Dataset Description

The multimodal dataset comprises of data collected over the period of 10 days. This dataset preserved different sensor functionalities: signal type varying from continuous to discrete; sampling frequency high, low and very low; sensor placement of the type wearable and non-wearable categories. However, the information related to the underlying sensor technologies and dataset collected is briefly discussed in the following subsections:-

Binary sensor event streams

In this challenge, an event stream of 30 binary sensors (BinSens) is shared by UJAmI Smart Lab comprising of binary values along with the timestamps. These BinSens are the wireless magnetic sensor that works on the principles of Z-Wave protocol. For example 'Medication Box' in use is considered 'Open' and not in use will be 'Close'. The inhabitant movement is monitored using the 'PIR sensors' working with underlying ZigBee protocol with the sample rate of 5Hz. The binary values, in this case, are represented by 'Movement' or 'No movement'. The sofas, chairs, and beds are equipped with the textile layer to observe the inhabitant pressure and transmits 'Present' or 'No present' binary values using the Z-Wave protocol. These BinSens gather information about the interaction with objects or sensors with a set of the associated activities. This interaction is relatively associated with the sparse BinSens streams, as it is defined by rapid firing for an event.

While on the other hand, a pressure sensor stream can even last for minutes to hours, so this makes it challenging for handling the diversified streams.

Spatial data

UJAmI Smart Lab has shared the SensFloor dataset generated by the suite of capacitance sensors beneath the floor. The SensFloor dataset has the description of eight sensor fields identified with an individual ID. The challenging part is that spatial data information is generated with different sample rate.

Proximity data and Bluetooth beacons

The Proximity data is provided for a set of 15 BLE beacons at 0.25Hz sample rate by an android application installed on smart-watch. The BLE beacons are generated for the objects like TV controller, fridge, Laundry basket etc.

Acceleration data

The physical activity frequency, ambulatory movements, and motion intensity are captured using an acceleration data stream which is gathered by the android application installed on smart-watch worn by the inhabitant. The accelerometer dataset is continuously generated at the 50Hz sampling rate in the tri-orthogonal directions such as x, y, and z-axis.

3.1.4 UCAmI Cup Challenge Design

The 1^{st} UCAmI Cup studies the problem of recognizing activities based on the shared UJAmI Smart Lab dataset for 24 activities in the smart-home environment as mentioned in Table 3.1. The total samples for each individual activity which was performed following several routines over the period of 7 days is plotted in Figure 3.2. Reflective of the real-world nature of the training dataset, the imbalance in the number of data samples for the different activities can be observed by observing the total duration for each individual performed activities. However, the challenging task is to map the pre-segmented training and test dataset from different sensor modalities having different sample rate. The challenge involves handling of missing values, identification mechanism to fill in the gaps since some sensors generate continuous streams while others discrete. So there is a need to devise an algorithm to handle more realistic and complex activity classification tasks.

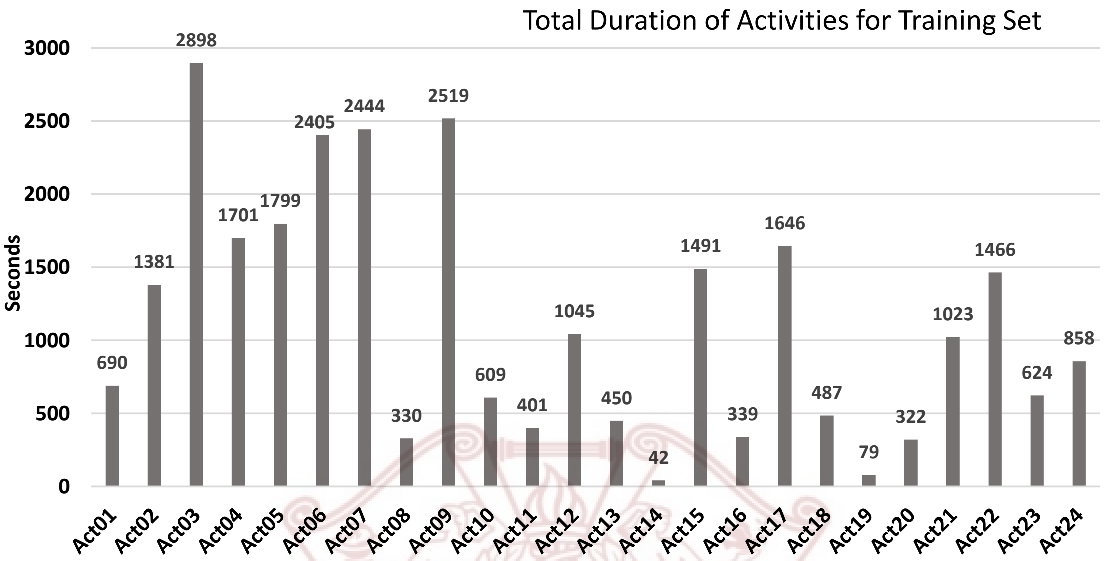


Figure 3.2: Life-time for the individual activities for UJAmI Smart Lab Training dataset

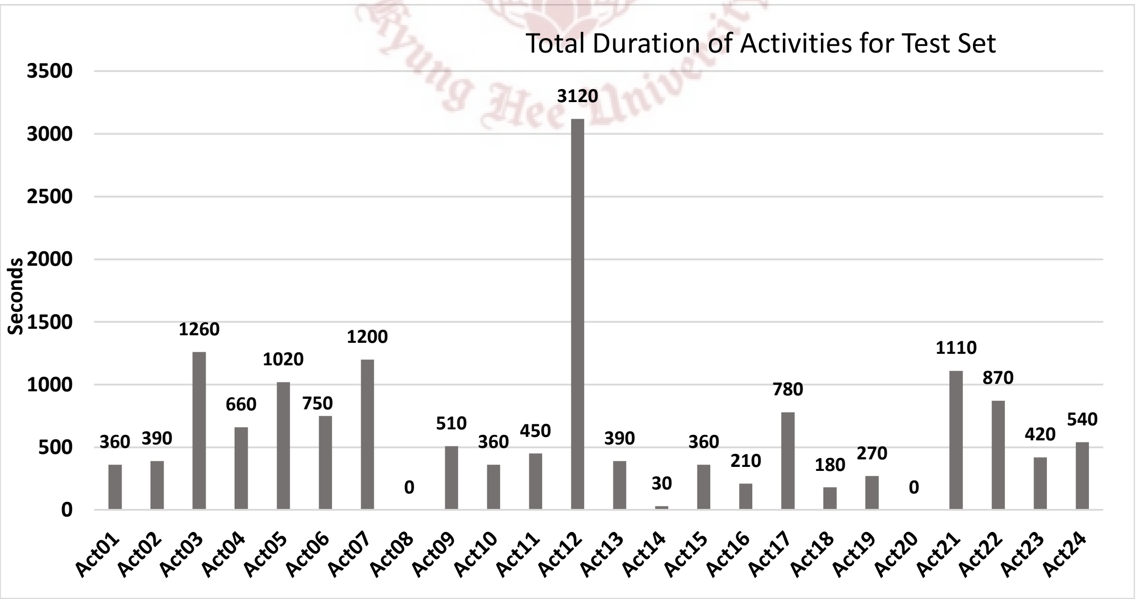


Figure 3.3: Life-time for the individual activities for UJAmI Smart Lab Test dataset

Main Tasks

The main aim of this work was recognizing activities based on the test dataset, evaluating the results and identifying the challenges associated with the UJAmI Smart Lab dataset. The training dataset has been automatically cleaned and preprocessed for activity classification, which is further discussed in subsequent sections. The overall UJAmI Smart Lab dataset is categorised as follows:

- Labelled training dataset with seven days of recordings that contained 169 instances followed by different daily life routines.
- Unlabelled test dataset with three days of recordings that contained 77 instances obtained by following a set of daily life routines.

3.1.5 SemImputOnt Modeling

As shown in Figure 3.1, the activity recognition framework is a sequence of data alignment, preprocessing, application of machine learning techniques, training of model based on the training dataset and classifying the test data.

Data alignment & mapping

UJAmI Smart Lab dataset corpus contains independently gathered data from multimodal sensors. For the main task, each of the sensor data was reordered into a set of 1-second window slot for the *time-stamp coherence*. In order to generate time-stamp, the Activity[n] (where n represents the number of activities performed for each time slot) was segmented based on *DateBegin* and *DateEnd* for over 1-second time window so that the rest of Sensors, Floor, Proximity and Acceleration dataset can be mapped based on timestamps. A sliding window segmentation technique [178] with the step size of 1-second was chosen in order to keep the maximum number of instances [74] and better performance [179]. The complete work-flow with details of the whole process of data alignment and mapping is presented in the proposed Algorithm 1.

Data preprocessing

The accelerometer data consisted of instances at the rate of 50Hz, which was preprocessed to filter out the unwanted noise and avoid the computational complexities. These were re-sampled by

Algorithm 1 Data alignment and re-sampling algorithm for UJAmI Smart Lab dataset (D)

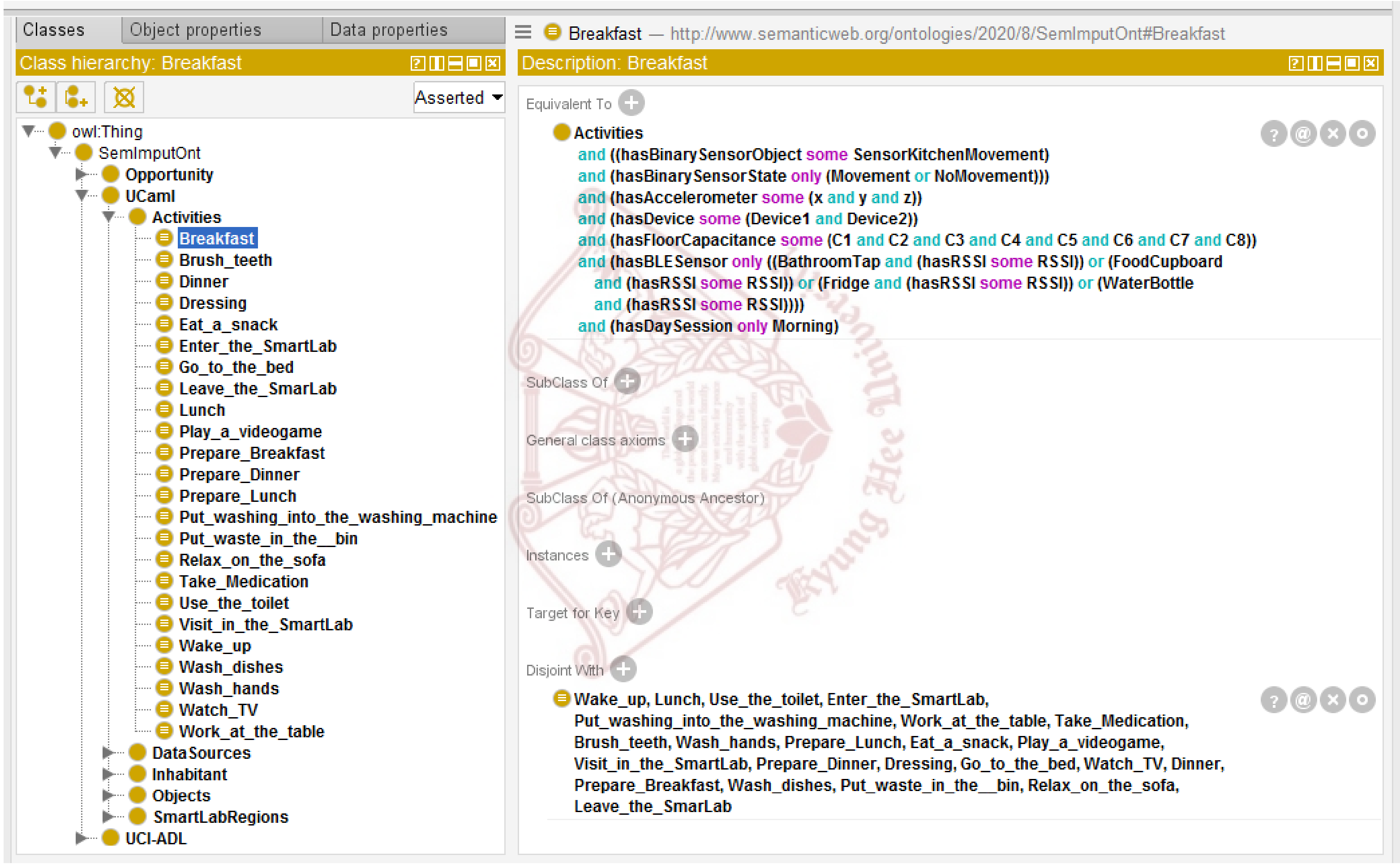
```
Input: $D_{seq}$ ▷ Sequence (Activity Act, Sensor Sen, Floor Flo, Proximity Prox, Acceleration Acc)
Output: $D_{seg}$ ▷ Segmented 1-sec re-sampled data tuples.
procedure DATAPROCESSING
    function $UpdateTimestamps(Activity\_FilesAct)$
        for all $Act_i$ to $Act_n$ do
            Load $Activity\_FileAct_i$
            Read $Activity\_FileAct_i$
            while $DateBegin(Act_i) < DateEnd(Act_i), Step$ do
                Generate $timestamp\ tuple$
                Add $Columntimestamp_1 sec$
                Set $Step = 1sec$
            end while
            $ActivityFileA1second \leftarrow ActivityFileA_i$
            Append $Activity\_FileA\_1\_second$
        end for
        Return $Activity\_FileA\_1\_second$
    end function
    function $ReSample(Acceleration\_FilesAcc, Step)$
        for all $Acc_i$ to $Acc_n$ do
            Load $Acceleration\_File\_Acc_i$
            Read $Acceleration\_File\_Acc_i$
            while $NotEOF$ do
                $Tuple(TS_j, X_j, Y_j, Z_j)_{1sec} \leftarrow (TS_i, \frac{1}{n}\sum_{i=1}^{n} x_i, \frac{1}{n}\sum_{i=1}^{n} y_i, \frac{1}{n}\sum_{i=1}^{n} z_i)$
                Set $Step = 1sec$
            end while
            $AccelerationFileAcc\_1\_second \leftarrow AccelerationFileAcc_i$
        end for
        Append $Acceleration\_File\_Acc\_1\_second$
    end function
    function $ReSample(Floor\_FilesFlo, DeviceID, mean, Step)$
        for all $Flo_i$ to $Flo_n$ do
            Load $Floor\_File\_Flo_i$
            Read $Floor\_File\_Flo_i$
            while $NotEOF$ do
                $Tuple(TS_j, DeviceID, C_1, , , C_8)_{1sec}$
                $\leftarrow (TS_i, DeviceID_i, \frac{1}{n}\sum_{i=1}^{n} c_1, , , \frac{1}{n}\sum_{i=1}^{n} c_8)$
                Set $Step = 1sec$
            end while
            $FloorFileFlo\_1\_second \leftarrow Floor\_File\_Flo_i$
        end for
        Append $FloorFileFlo\_1\_second$
    end function
    Load Files $ActivityAct; SensorSen; FloorFlo; ProximityProx; AccelerationAcc$
    Read Files $ActivityAct; SensorSen; FloorFlo; ProximityProx; AccelerationAcc$
    for all $TS_{Ai}$ to $TS_{An}$ do
        if $TS_{Act} = TS_{Sen} = TS_{Flo} = TS_{Prox} = TS_{Acc}$ then
            for each tuple in Act: $Tuple\_Vector \leftarrow Act \bowtie Sen \bowtie Flo \bowtie Prox \bowtie Acc$
        end if
        Append $MainTable \leftarrow Tuple\_Vector$
    end for
    $D_{seg} \leftarrow MYSQL(MainTable)$
end procedure
```

applying the commonly used time-domain statistical features, such as a mean filter for each of x, y, and z tri-orthogonal values over a duration of 1-second. After observing the Floor data, which was also generated at a variable rate, was re-sampled within the duration of a 1-second window. In this case, the method considered was to take the mean for floor capacitances and also at the same time, to keep the characteristics intact for individual data generating sensing device, identified by the device ID.

Taxonomy Construction

We followed and utilized the data-driven techniques to model sensor streams for identifying complex concurrent sensor temporal state patterns. These state patterns become the basis for the parallel and interleaved ADLs, which are of static and dynamic nature as mentioned in Table 3.1. An ontology engineer utilizes the complete knowledge of involved sensors and the nature of the data produced by them. In addition, the core vocabulary required to model and design the *SemImputOnt* is obtained through the temporal patterns of sensor stream data, describing the complex ADL's main class definitions. The descendants of these main classes, however, have been described to model each sensor object, which generates discrete or continuous sensory data. These primitive classes are related to ADLs using "SensorStateObject" properties. These object properties such as *hasBinarySensorObject* shows the relationship between the ADL and the core sensor object defining its state. Again, the state is linked by a property *hasBinarySensorState* with *SensorStateObjects*. Similarly, the other obtrusive sensor objects have the properties *hasAccelerometer*, *hasBLESensor* with the *hasRSSI* data property. All these sensor objects define the ADL with open intervals without any prior knowledge of *Start-time* or *End-time* [4]. The temporal relations for each sensor object are obtained using object properties *hasStartTime* and *hasEndTime*.

How comprehensive *SemImputOnt* is at representing disjoint ADLs can be visualized and explained through an example of the activity *Breakfast* modeled in Figure 3.4. In this example, an ADL *Breakfast* is represented as a class. The ADL *Breakfast* is a descendant of the *Activities* class, defined as being an equivalent class relating to the instances of BinarySensorObject, BinarySensorState, Accelerometer, Devices, FloorCapacitance, BLESensors, and DaySession. This means that, to be a member of the defined class *Breakfast*, an instance of the *Activities* class must have a property of type *hasBinarySensorObject*, which relates to an instance of the *SensorKitchenMovement* class, and this property can only take as value an instance of the *SensorKitchenMovement* class. The instance of the *Activities* class must also have a property of type *hasBinarySensorState*, which relates to an instance of the *Movement* class, or the *NoMovement* class, and this property can only take as value an instance of one of them. The instance of the *Activities* class must also have a property of type *hasAccelerometer*, which relates to an instance of the *x* class, *y* class, and *z* class. This property must only relate to the instances of these three classes. The instance of the *Activities* class must also have a property of type *hasDevice*, which relates to an instance of the *Device1* class, and *Device2* class. This property must only relate to the instances of these two classes. The instance of the *Activities* class must also have a property of type *hasFloorCapacitance*, which relates to an instance of the *C1* class, *C2* class, *C3* class, *C4* class, *C5* class, *C6* class, *C7* class, and *C8* class. This property must only relate to the instances of these seven classes. The instance of the *Activities* class must also have a property of type *hasBLESensor*, which relate to an instance of the *Tap* class, *FoodCupboard* class, *Fridge* class, and *WaterBottle* class for this example. This property must only relate to the instances of these four classes and every class must also have a property *hasRSSI*, which relates to the instance of *RSSI* class. Moreover, the instance of the *Activities* class must also have a property of type *hasDaySession*, which relates to an instance of the *Morning* class and only to an instance of the *Morning* class. Thus, if an instance of the *Activities* class fulfills the seven existential restrictions on the properties *hasBinarySensorObject*, *hasBinarySensorState*, *hasAccelerometer*, *hasDevice*, *hasFloorCapacitance*, *hasBLESensor*, and *hasDaySession*, the instance will be inferred as being a member of the *Breakfast* class.



```
owl:Thing
  SemImputOnt
    Opportunity
    UCamI
      Activities
        Breakfast
        Brush_teeth
        Dinner
        Dressing
        Eat_a_snack
        Enter_the_SmartLab
        Go_to_the_bed
        Leave_the_SmarLab
        Lunch
        Play_a_videogame
        Prepare_Breakfast
        Prepare_Dinner
        Prepare_Lunch
        Put_washing_into_the_washing_machine
        Put_waste_in_the__bin
        Relax_on_the_sofa
        Take_Medication
        Use_the_toilet
        Visit_in_the_SmartLab
        Wake_up
        Wash_dishes
        Wash_hands
        Watch_TV
        Work_at_the_table
      DataSources
      Inhabitant
      Objects
      SmartLabRegions
    UCI-ADL
```

Equivalent To

```
Activities
 and ((hasBinarySensorObject some SensorKitchenMovement)
 and (hasBinarySensorState only (Movement or NoMovement))
 and (hasAccelerometer some (x and y and z))
 and (hasDevice some (Device1 and Device2))
 and (hasFloorCapacitance some (C1 and C2 and C3 and C4 and C5 and C6 and C7 and C8))
 and (hasBLESensor only ((BathroomTap and (hasRSSI some RSSI)) or (FoodCupboard
   and (hasRSSI some RSSI)) or (Fridge and (hasRSSI some RSSI)) or (WaterBottle
   and (hasRSSI some RSSI))))
 and (hasDaySession only Morning)
```

Disjoint With

Wake_up, Lunch, Use_the_toilet, Enter_the_SmartLab, Put_washing_into_the_washing_machine, Work_at_the_table, Take_Medication, Brush_teeth, Wash_hands, Prepare_Lunch, Eat_a_snack, Play_a_videogame, Visit_in_the_SmartLab, Prepare_Dinner, Dressing, Go_to_the_bed, Watch_TV, Dinner, Prepare_Breakfast, Wash_dishes, Put_waste_in_the__bin, Relax_on_the_sofa, Leave_the_SmarLab

Figure 3.4: *SemImputOnt*: Class hierarchy with a definition axiom for the activity *Breakfast*.

Table 3.1: A list of activities, locations, and dependent sensor objects identified from UCamI dataset utilized for *SemImputOnt* constructs.

Type	ID	Activity Name	Location	Activity Dependencies Sensors' Objects
Static	Act01	Take medication	Kitchen	Water bottle, MedicationBox
Dynamic	Act02	Prepare breakfast	Kitchen, Dining room	Motion Sensor Bedroom, Sensor Kitchen Movement, Refrigerator, Kettle, Microwave, Tap, Kitchen Faucet
Dynamic	Act03	Prepare lunch	Kitchen, Dining room	Motion Sensor Bedroom, Sensor Kitchen Movement, Refrigerator, Pantry, Cupboard Cups, Cutlery, Pots, Microwave
Dynamic	Act04	Prepare dinner	Kitchen, Dining room	Motion Sensor Bedroom, Sensor Kitchen Movement, Refrigerator, Pantry, Dish, microwave
Dynamic	Act05	Breakfast	Kitchen, Dining room	Motion Sensor Bedroom, Sensor Kitchen Movement, Pots, Dishwasher,Tap, Kitchen Faucet
Dynamic	Act06	Lunch	Kitchen, Dining room	Motion Sensor Bedroom, Sensor Kitchen Movement, Pots, Dishwasher,Tap, Kitchen Faucet
Dynamic	Act07	Dinner	Kitchen, Dining room	Motion Sensor Bedroom, Sensor Kitchen Movement, Pots, Dishwasher,Tap, Kitchen Faucet
Dynamic	Act08	Eat a snack	Kitchen, Living room	Motion Sensor Bedroom, Sensor Kitchen Movement, Fruit Platter, Pots, Dishwasher,Tap, Kitchen Faucet
Static	Act09	Watch TV	Living room	RemoteControl, Motion Sensor Sofa, Pressure Sofa, TV
Dynamic	Act10	Enter the SmartLab	Entrance	Door
Static	Act11	Play a video game	Living room	Motion Sensor Sofa, Motion Sensor Bedroom, Pressure Sofa, Remote XBOX
Static	Act12	Relax on the sofa	Living room	Motion Sensor Sofa, Motion Sensor Bedroom, Pressure Sofa
Dynamic	Act13	Leave the SmartLab	Entrance	Door
Dynamic	Act14	Visit in the SmartLab	Entrance	Door
Dynamic	Act15	Put waste in the bin	Kitchen, Entrance	Trash
Dynamic	Act16	Wash hands	bathroom	Motion Sensor Bathroom, Tap, Tank
Dynamic	Act17	Brush teeth	bathroom	Motion Sensor Bathroom, Tap, Tank
Static	Act18	Use the toilet	bathroom	Motion Sensor Bathroom, Top WC
Static	Act19	Wash dishes	Kitchen	dish, dishwasher
Dynamic	Act20	Put washing into the washing machine	Bedroom, Kitchen	Laundry Basket, Washing machine, Closet
Static	Act21	Work at the table	Workplace	
Dynamic	Act22	Dressing	Bedroom	Wardrobe Clothes, Pyjama drawer, Laundry Basket, Closet
Static	Act23	Go to the bed	Bedroom	Motion Sensor bedroom, Bed
Static	Act24	Wake up	Bedroom	Motion Sensor bedroom, Bed

Concurrent Sensor State Modeling

The object properties introduced in *SemImputOnt* as an existential restriction support management of concurrent and sequential sensor states as explained in the *Breakfast* activity model example. These properties not only describe the hierarchy of sensor object states, and their actions by establishing object–data relationships but also support in augmenting the incomplete sensor sequences using SPARQL queries. Moreover, the relationship also supports, while generalizing data-driven rules as shown in the anonymous equivalent class for the activity *Breakfast*. These rules map sensor states in *SemImputOnt* to model an activity rather than tracking rigid sensor state patterns. These sensor state patterns are identified and linked to their respective timestamps using temporal datatype properties such as *hasStartTime* and *hasEndTime*. *SemImputOnt* comprehensively models sensor situations using sensor state concepts independently and concurrently by exploiting their relationships using *Allen's temporal operators* [155].

3.1.6 Semantic Segmentation

The Semantic Segmentation Layer in the *SemImput* framework describes the ontological operations to illustrate the modeling patterns of ADLs, by observing them in a sliding window. The first step is to retrieve and synchronize the non-segmented sensor state instances obtained from obtrusive and unobtrusive data sources along with their temporal information. We used a non-overlapping and static sliding time windows [180] approach, in which each sensor state is identified by a timestamp. For this, we used a set of 9 SPARQL-based query templates for retrieving and interpreting rules to deal with underlying temporal sensor state relations, as well as their structural properties. Moreover, the SPARQL queries require additional parameters in order to correlate, interpret, and aggregate sensor states within the endpoints of the sliding window [24]. Some of the initializing parameters include *start-time*, *end-time*, and a list of sensors within the sliding window identified based on the *start-time* and datatype properties. These parameters provide support for manipulating concurrent sensors states, which are expanded and imputed as illustrated in further sections. *SemImputOnt* is also used for validating temporal constraints and for the verification of property values within a sliding window [181]. The sensor state endpoints are retrieved through the following custom set of conjunctive ABox SPARQL queries $\mathcal{CQ}$ where ($cq_i \epsilon \mathcal{CQ}$) over the sliding time window:

- cq_1: Valid *Open* sensor state
- cq_2: Valid *Closed* sensor state
- cq_3: *Start-time* of *Next,* sensor state
- cq_4: Sensor having *Open* state within the sliding window

- cq_5: Sensor having *Closed* state within the sliding window

whereas the concurrent sensor states are retrieved through following SPARQL-based query templates, which are also coincidental at their:

- cq_6: *start-time* and still *Open* sensor states
- cq_7: *start-time* but *Closed* sensor states
- cq_8: *end-time* but still *Open* sensor states
- cq_9: *end-time* but *Closed* sensor states

The SPARQL query, cq_1, refers to the identifiers from the *SemImputOnt* retrieved instances, which are still active but are yet to be finished. These states are identified based on their initialization timestamps represented by the *start-time*. The query cq_2 retrieves *SemImputOnt* instances having both endpoints identified by *start-time* and *end-time*. The query cq_3 retrieves the *start-time* of the sensor initialization, which may deactivate and at the same time becomes active in a current sliding time window. The query cq_4 retrieves sensor state, which has just started in the sliding window; this query provides the *start-time*. The query cq_5, a specially designed query to monitor the sensor state, which is currently active in the sliding window and changes its states to *deactivation* or *off state*. This query retrieves the *end-time* for such state transition. The query cq_6 retrieves active concurrent sensor states for more than one sensor, based on the *start-time* within the current sliding time window which is yet to finish. The query cq_7 on the other hand fetches the *start-time* for such concurrent sensors, which have *closed* states with valid *end-times*. Similarly, the queries cq_8 and cq_9 retrieve the active and inactive concurrent sensor states based on some *end-time* data value, respectively. The above-mentioned queries cq_3, cq_4, and cq_6 are responsible for initializing a separate thread to monitor and keep the track for sensor states which are to become inactive by identifying the *end-time*.

The segments returned through the SPARQL queries may be considered complete if they contain both the endpoints represented by dissimilar sensor states. If one of the end points goes missing, however, the segment becomes anomalous or erroneous in the sensor stream data. Such erroneous behavior is identified by using semantic data expansion and resolved through the semantic imputation.

3.1.7 Semantic Data Expansion

The proposed set of *SemImputOnt* models sensor objects (concepts and properties) and their states (instances) from the segmented D_n datasets. It not only maps sensor streams but also captures structure, preserving the associations within the sensor state instances using a data-driven approach. A structure-preserving transformation encompasses each sensor object, their associations, and subsumptions relating to different concurrent activities [182]. These preserved semantics and associations are separated by understanding the complex activity structures. The separation process includes conversions of these semantics into distinct columns while conjunctions in between them provide essential existential conditions for representing activities in a matrix.

Ontology-Based Complex Activity Structures

To encode more detailed structure, the *SemImputOnt* uses primitive and defined concepts with value-restriction and conjunctions as concept-forming operators. These value restrictions are enforced through classifiable attributes (roles) and non-classifiable attributes (non-definitional roles) to model HAR datasets. In *SemImputOnt*, primitive-concepts (Activities) provide necessary conditions for membership, whereas defined concepts (Sensors, Objects, Data sources) provide both necessary and sufficient conditions for membership as mentioned below:

$$\mathcal{A} \sqsubseteq C; \tag{3.1}$$

$$\mathcal{A} \equiv C; \tag{3.2}$$

where $\mathcal{A}$ is any *Activity* name, and *C* defines a primitive concept or a defined concept as mentioned in Equations (3.1 and 3.2), respectively. These concepts are used to form an expression, which can be either a sensor state, or conjunction of sensor states with or without a value-restriction as described below:

$$C \rightarrow A_1; C \rightarrow (\forall R.A_2 \sqcap \exists R); C \rightarrow C_1 \sqcap C_2 \tag{3.3}$$

Here, A_1, A_2 are attribute, R is a conjoined predicate, and C_1, C_2 are concept instances forming expressions.

Utilizing the Description Logic (DL) notations, an example of *Breakfast Activity* from *UCamI* dataset can be described in DL expression as:

Breakfast $\equiv$ *Activities* $\sqcap$ $\exists$ *hasBinarySensorObject.SensorKitchenMovement* $\sqcap$ $\forall$ *hasBinarySensorState.(Movement* $\sqcup$ *NoMovement)* $\sqcap$ $\exists$ *hasAccelerometer.(x* $\sqcap$ *y* $\sqcap$ *z)* $\sqcap$ $\exists$ *hasDevice.(Device1* $\sqcap$ *Device2)* $\sqcap$ $\exists$ *hasFloorCapacitance.(C1* $\sqcap$ *C2* $\sqcap$ *C3* $\sqcap$ *C4* $\sqcap$ *C5* $\sqcap$ *C6* $\sqcap$ *C7* $\sqcap$ *C8)* $\sqcap$ $\forall$ *hasBLESensor.(Tap* $\sqcap$ $\exists$ *hasRSSI.RSSI* $\sqcup$ *FoodCupboard* $\sqcap$ $\exists$ *hasRSSI.RSSI* $\sqcup$ *Fridge* $\sqcap$ $\exists$ *hasRSSI.RSSI* $\sqcup$ *WaterBottle* $\sqcap$ $\exists$ *hasRSSI.RSSI)* $\sqcap$ $\forall$ *hasDaySession.Morning*

whereas the same activity *Breakfast* using the DL attributes from *UCI-ADL* dataset is described as:

Breakfast $\equiv$ *UCI-ADL* $\sqcap$ $\exists$ *hasPlace Kitchen* $\sqcap$ $\forall$ *hasPlace Kitchen* $\sqcap$ $\exists$ *hasSensorLocation (Cooktop* $\sqcup$ *Cupboard* $\sqcup$ *Fridge* $\sqcup$ *Microwave* $\sqcup$ *Seat* $\sqcup$ *Toaster)* $\sqcap$ $\forall$ *hasSensorLocation (Cooktop* $\sqcup$ *Cupboard* $\sqcup$ *Fridge* $\sqcup$ *Microwave* $\sqcup$ *Seat* $\sqcup$ *Toaster)* $\sqcap$ $\forall$ *hasSensorType (Electric* $\sqcup$ *Magnetic* $\sqcup$ *PIR* $\sqcup$ *Pressure)*

In both the expressions, the activity *Breakfast* is represented by different concept attributes modeled into their corresponding ontologies in the *SemImputOnt*. It is evident that this activity is represented by different sets of underlying ontological concepts depending upon the nature of sensors deployed for acquiring the datasets for that activity. Keeping the same definition of each activity represented by different underlying constructs may result in recognition performance degradation. For this reason, they are defined separately, as the focus of the study is to fill in the gaps for missing sensor states.

The primitive concepts are mapped into partial concepts using Web Ontology Language (OWL), which are encoded with *rdfs:subClassOf* construct (Equation (3.1)). In addition, the defined concepts are mapped to complete concepts in OWL, which are encoded as class equivalence axioms represented as *owl:equivalentClass* (equation 3.2). The concept names and concept conjunctions are mapped to class names and class intersections in OWL, respectively, whereas roles are mapped with object properties. These primitive and defined concepts definitions map the data instances into *SemImputOnt* models for representing complex activities.

Conjunction Separation

The concepts expressed in the DL for *Breakfast* definition uses conjunctions for relating the sensor state events [183]. The *Breakfast* equivalent class forming a complex activity with the involvement of several *Class* concepts, relationships (object & data properties), and data instances. All the involved *Class* concepts coupled with conjunctions defining the *Activity* equivalent classes are transformed into independent entities by separating them based on involved conjunctions [154]. Conjunction separation emphasizes the idea of concept ($\varphi, \psi, \omega, \chi \ldots$) separation over the intention *I* such as:

$$\models I\left(\varphi \wedge \psi \wedge \omega \wedge \chi \ldots\right) \rightarrow I\left(\varphi\right) \wedge I\left(\psi\right) \wedge I\left(\omega\right) \wedge I\left(\chi\right) \ldots \tag{3.4}$$

These independent entities are transformed into multi-dimensional vectors representing the features from all sensor states for a particular activity w.r.t. associated timestamps. The size of the multi-dimensional vector may vary for each activity based on the conjunctive class concepts learned through the data modeled over *SemImputOnt*.

Feature Transformation

The predicates separated in the previous step produces a row vector identified by a single activity label, whereas column represents the class concepts with states as an instance. These predicates in the feature space represent activities along with the timeline. These features ensure the reliability of activities through mappings with the *SemImputOnt* [152, 183]. In our case, *SemImputOnt* supports essential properties while generating and validating the data into ABox $\mathcal{A}$ features as provided using an example from the *UCamI* dataset.

$$\mathcal{A}_n \leftarrow \{BinSens_1, BinSens_2, \ldots BinSens_{30}, BLE_1, \ldots BLE_{15}, C_1, C_2, \ldots C_8, x, y, z\}$$

(3.5)

where n = $\{1, 2 \ldots 24\}$, $BinSens$ can have one of the states at a unit time T_{1sec} from { *Open*, *Close*, *Present*, *No present*, *Pressure*, *No Pressure*, *Movement*, *No Movement*}. These state mappings result into a matrix representing each row with a single activity and every column with *Class* concepts. Each of the separated concept supports modification of one segment independent of the others column-wise.

Algorithm 2 Semantic Imputation Using $\mathcal{I}_{SS}(A_m)$, $\mathcal{I}_{SI}(A_m)$, and $\mathcal{I}_{L}(A_m)$ through SPARQL Queries

```
Input: Incomplete Segmented Data $A_m, \mathcal{A}, D_{seg}$
Output: Complete Data with Imputation $A_m^{Imp}$    ▷ Segmented Imputed Dataset.
procedure SEMANTICIMPUTATION
    for all timestamp t = 1 to T do
        function $ImputeBinSens(A_m, CQ, A, T)$    ▷ $BinSens_{attrib}$ with their state imputation
            for ($cq_i \epsilon \mathcal{CQ}$) do
                $BinSens_{Attrib} \leftarrow execute(cq_i).filter(BinSens, A_m)$    ▷ using SPARQL Queries
                $BinSens_{Target} \leftarrow execute(cq_i).filter(BinSens_{Attrib}, \mathcal{T})$
                $A_{BS_{att}} \leftarrow BinSens_{Attrib}$
                $A_{BS_{tar}} \leftarrow BinSens_{Target}$
                $max(\mathcal{I}_{SS}) \leftarrow Compute\ \mathcal{I}_{SS}(A_{BS_{tar}}, A_{BS_{att}})$    ▷ Equation (10)
                $A_{BS_{att}} \leftarrow A_{BS_{att}} \cup (A_{BS_{tar}} \setminus A_{BS_{att}})$    ▷ Update missing BinSens Attribute
                $BinSens_{mappings} \leftarrow retrieve.mappingsLists(BinSens_{LOCF}, BinSens_{NOCB})$
                while $A_{BS_{att}}(state) = \phi$ do    ▷ Load Updated BinSens attributes
                    if ($A_{BS_{att}}$ in $BinSensList_{LOCF}$) then    ▷ based on BinSens characteristics
                        $A_{BS_{state}} \leftarrow execute(cq_i).retrieveLastState.(A_{BS_{att}})$
                        $A_{BS} \leftarrow \mathcal{I}_L(A_{BS_{att}}, A_{BS_{state}})$
                    else if ($A_{BS_{att}}$ in $BinSensList_{NOCB}$) then
                        $A_{BS_{state}} \leftarrow execute(cq_i).retrieveNext, State.(A_{BS_{att}})$
                        $A_{BS} \leftarrow \mathcal{I}_L(A_{BS_{att}}, A_{BS_{state}})$
                    end if
                end while
            end for
            Return Imputed $A_{BS}$
        end function
        function $ImputeProximity(A_m, CQ)$ ▷ Imputation for Proximity Sensors and their values
            for ($cq_i \epsilon \mathcal{CQ}$) do
                $A_{Prox} \leftarrow execute(cq_i).filter(Proximity, A_m)$
                $Prox_{max} \leftarrow maxValue(A_{Prox})$
                $A_{Prox} \leftarrow Update\ A_{Prox}(Prox_{max})$
            end for
            Return Imputed $A_{Prox}$
        end function
        function $ImputeFloor(A_m, CQ, A)$ ▷ Imputation for Floor sensors and their values
            for ($cq_i \epsilon \mathcal{CQ}$) do
                $A_{mfloor} \leftarrow execute(cq_i).filter(Floor, A_m)$
                $A_{tfloor} \leftarrow execute(cq_i).filter(A_{mfloor}, A)$
                $mean(floortuples) \leftarrow$ Compute $\mathcal{I}_{SI}(A_{tfloor}, A_{mfloor})$    ▷ Equation (13)
                $A_{floor} \leftarrow$Update $A_{mfloor} \cup mean(floortuple)$    ▷ update using mean for tuples
            end for
            Return Imputed $A_{floor}$
        end function
        function $ImputeAccelerometer(A_m, CQ, A)$    ▷ Imputation for accelerometer values
            for $(cq_i \epsilon \mathcal{CQ})$ do
                $A_{mAcc} \leftarrow execute(cq_i).filter(Acc, A_m)$
                $A_{tAcc} \leftarrow execute(cq_i).filter(A_{mAcc}, A)$
                $mean(acctuples) \leftarrow$ Compute $\mathcal{I}_{SI}(A_{tAcc}, A_{mAcc})$
                $A_{Acc} \leftarrow$Update $A_{mAcc} \cup mean(acctuples)$    ▷ update using mean for last 10 tuples
            end for
            Return Imputed $A_{Acc}$
        end function
    end for
    $A_m^{Imp} \leftarrow A_{BS} \parallel A_{Prox} \parallel A_{floor} \parallel A_{Acc}$
    increment $t$ by 3 sec
end procedure
```

3.1.8 Semantic Data Imputation

The resulting n-dimension feature vector matrix has missing sensor states (*Null*), which lead to the loss in efficiency for the activity classification model. Such losses can be dealt with suitable imputation techniques, which enriches the expanded data semantically by filling in the missing sensor states. We propose a *Semantic Imputation* algorithm to capture the temporal missing sensor states semantically and perform an overall feature vector matrix enrichment [184]. We adapt two similarity-based methods and a time-series longitudinal imputation strategy to assess similarity of the concepts $\mathcal{T}$ and instances $\mathcal{A}$ for imputation $\mathcal{I}(A_m)$ as described in Algorithm 2.

Structure-Based Imputation Measure

The structural patterns in TBox ($\mathcal{T}$) are identified and exploited using SPARQL queries over the *SemImputOnt*. These queries could retrieve $\mathcal{T}$ assertions based on the query criteria to measure semantic similarity with target activity patterns. However, choosing a suitable pattern from target activities and selecting the appropriate sensor state to fill in the missing ones is addressed through structure-based similarity measure. We define structural similarity function for a target set of description A_n and activity A_m with missing attributes to identify maximum probability as:

$$Sim_{ss} : A_n \times A_m \longmapsto [0 \ldots 1] \tag{3.6}$$

It returns semantically equivalent sensor states where the child nodes for two concepts are similar [185]. We use the *Tanimoto* coefficient between A_n and A_m for measuring the structural similarity. A_n gives the binary description for the involved sensors and A_m are the available sensor predicates for the activity with missing predicates mentioned below:

$$\begin{aligned} \mathcal{I}_{SS}(A_m) &= Sim_{ss}(A_n, A_m) \\ &= \frac{\sum_{j=1}^{k} A_n \times A_m}{\left(\sum_{j=1}^{k} A_n^2 + \sum_{j=1}^{k} A_m^2 - \sum_{j=1}^{k} A_n \times A_m\right)} \end{aligned} \tag{3.7}$$

The $\mathcal{I}_{SS}(A_m)$ function determines the structural similarity among the target A_n and A_m, the higher the numerical value is, a more closer structural description of A_m instance is with A_n description [186, 187]. As a result, structural attributes are suggested for a tuple A_m with missing attributes.

Instance-Based Imputation Measure

The ABox $\mathcal{A}$ is comprised of a finite set of membership assertions $\mathcal{A}$ referring to the concepts and membership roles to their respective TBox $\mathcal{T}$. The set of assertions $\mathcal{A}$ for the *UCamI* dataset is represented as:

$$\mathcal{A} \leftarrow (ts, r_s, \mathcal{R}_i, \mathcal{V}_i) \tag{3.8}$$

Each of the assertion is a combination of sensors r_s with their certain states $\mathcal{V}_i$ at a timestamp *ts*.

$$(r_s, \mathcal{R}_i, \mathcal{V}_i) \leftarrow \langle binsens_{1\ldots30}, \mathcal{R}_\alpha, \mathcal{V}_\alpha\rangle \bigcup \langle ble_{1\ldots15}, \mathcal{R}_\beta, \mathcal{V}_\beta\rangle \bigcup \langle c_{1\ldots8}, \mathcal{R}_\epsilon, \mathcal{V}_\epsilon\rangle \bigcup \langle acc_{x,y,z}, \mathcal{R}_\varphi, \mathcal{V}_\varphi\rangle \tag{3.9}$$

where $binsens_{1...30}$ are the object names referring to the concept *BinarySensor* in the *SemImputOnt*, ranging from 1 and 30 with binary states $[0, 1]$ represented as $\mathcal{V}_\alpha$. $ble_{1...15}$ refers object names, which are members for *Proximity* concept having values $\mathcal{V}_\beta$, *Intelligent Floor* concept having assertions $c_{1...8}$ with values $\mathcal{V}_\epsilon$ and accelerometer *SmartWatch* concept having membership for with values as $\mathcal{V}_\varphi$. Instance-based similarity $\mathcal{I}_{SI}(A_m)$ is measured [188] between target activity instance $\mathcal{A}_n$ and instance with missing states $\mathcal{A}_m$ as:

$$\begin{aligned}\mathcal{I}_{SI}(A_m) &= Sim_I(\mathcal{A}_n, \mathcal{A}_m) \\ &= max_m \frac{overlap\,(\mathcal{A}_n, \mathcal{A}_m, m)}{\mathcal{A}_n \uplus \mathcal{A}_m}\end{aligned} \tag{3.10}$$

where *m* is the mapping between $\mathcal{A}_n$ and $\mathcal{A}_m$ in conjunction with concept-to-concept and roles-to-roles. In addition, $\mathcal{A}_n \uplus \mathcal{A}_m$ represents the disjoint union of memberships pertaining to concepts and their roles between them. Instance-based similarity exploits neighborhood similarity by measuring similarity through $Sim_I(\mathcal{A}_n, \mathcal{A}_m)$ function. Thus, an instance with high similarity value is chosen for attribute states to be imputed for a tuple $\mathcal{A}_m$ with missing states.

Longitudinal Imputation Measure

The quality of data, resulting from structure and instance-based imputation in a matrix form, is further improved by using classical techniques of Last Observation Carried Forward (LOCF) and Next Observation Carried Backward (NOCB). LOCF and NOCB are applied to the data in an observable manner by analyzing each longitudinal segment, as described in Equation (3.4), for activity states retrieved through SPARQL queries. While observing the binary sensors and their states in a time series longitudinal segments, it is observed that the sensor states are triggered once either for activation or deactivation. For example, an object *Washing Machine* in *UCamI* dataset has a *contact* type sensor with *Open* state at T_1 = 2017-11-10 13:37:56.0 and *Close* state at T_2 = 2017-11-10 13:38:39.0. In this case, while synchronizing this sensor data with other states per unit time, *Null* values appear after T_1 till T_2 as the states triggered for once. For this LOCF, a *sample-and-hold* method is activated, which carries forward the last state and imputes the *Null* values with this last available sensor state. Similarly, NOCB imputes the missing values from next available state, which is carried backwards. The missing states for Proximity sensors in the

Algorithm 3 Semantic Vectorization Using One-Hot Coding Technique

```
Input: $A_m^{Imp}$    ▷ Extract scalar sequence (BinSens, Proximity)
Output: $M$    ▷ Vectorized feature Matrix.
procedure SemanticVectorization
    for all timestamp t = 1 to T do
        function $BinSensVectorization(CQ, A_m^{Imp})$
            for ($cq_i \epsilon \mathcal{CQ}$) do
                $BinSens_{Attrib} \leftarrow execute(cq_i).filter(BinSens, A_m^{Imp}))$    ▷ using SPARQL Queries
                $BinSens_{states} \leftarrow execute(cq_i).filter(BinSens_{Attrib}))$
                while $BinSens_{states} \neq \phi$ do
                    $BinSens_{Vec} \leftarrow Map(BinSens, BinSens_{Attrib})$
                    $BinSens_{fCol} \leftarrow Transform(n \times p, BinSens_{Vec})$    ▷ transform rows into columns
                    $BinSens_{stride} \leftarrow StateReplace(BinSens_{Vec})$    ▷ 1 for Active BinSens or 0, otherwise
                end while
            end for
            Return $BinSens_{stride}$
        end function
        function $ProxVectorization(CQ, A_m^{Imp})$
            for ($cq_i \epsilon \mathcal{CQ}$) do
                $Prox_{Attrib} \leftarrow execute(cq_i).filter(Prox, A))$    ▷ using SPARQL Queries
                $Prox_{states} \leftarrow execute(cq_i).filter(Prox_{Attrib}))$
                while $Prox_{states}(state) \neq \phi$ do
                    $Prox_{Vec} \leftarrow Map(Prox, Prox_{Attrib})$
                    $Prox_{fCol} \leftarrow Transform(n \times p, Prox_{Vec})$    ▷ transform rows into columns
                    $Prox_{stride} \leftarrow StateReplace(Prox_{Vec})$    ▷ Set 1 for highest RSSI and 0 for rest
                end while
            end for
            Return $Prox_{stride}$
        end function
    end for
    $M \leftarrow BinSens_{stride} \| Prox_{stride} \| A_{floor} \| A_{Acc}$
    increment t by 3 sec
end procedure
```

case of the *UCamI* dataset are imputed in a slightly different way as elaborated in Algorithm 2. It identifies the proximity sensors and their respective RSSI values within the sliding window. The proximity sensor utilizes maximum value imputation in which the LOCF method is applied until some other proximity sensor with a value greater than the already known value is identified. For continuous data such as *Floor* and *Acceleration*, a statistical approach is adopted to replace the missing states with the mean of corresponding observed attributes. Mean imputation method tends to be robust and easy to substitute the missing values.

3.1.9 Classification

To cross examine the effectiveness for imputed datasets using proposed *SemImput* framework, we used a Deep Learning-based Artificial Neural Network (*ANN*) classifier [189]. The experimental results proved to be suitable for multimodal, multi-sensory, and multi-feature datasets for HAR. For this, an *ANN* model is trained with the labeled 2D training matrix instances for the *UCamI*, *Opportunity* and *UCI-ADL* datasets. The computational complexity and recognition accuracies are then assessed.

One-Hot Code Vectorization

It has been observed as advantageous to transform categorical variables using suitable feature engineering before applying neural network [190]. For this, we used *one-hot* encoding, a robust feature engineering scheme, for generating the suitable feature vector indices [156]. These categorical features are mapped into sensor state vector indices representing the concurrent sensor activation patterns for a particular activity. This scheme expands the dimension of the feature matrix for 2^n possible combinations based on the binary states for the "n" sensors involved in the feature vector. As described in Algorithm 3, n-dimensional sparse vector per unit time is obtained for populating feature matrix required for classification. The value 1 is encoded where the sensor has an active state and the value 0 is assigned for *missing* state in a row vector [190]. The missing value indicator *r* in the matrix is represented as $r_{n,p}$ with n_{th} row and p_{th} column:

$$r_{n,p} = \begin{cases} 1, & \text{value is observed} \\ 0, & \text{if value is missing} \end{cases} \tag{3.11}$$

Algorithm 4 Semantic Deep Learning-based Artificial Neural Network (SemDeep-ANN)

```
Input: Labeled Dataset $\mathcal{M}_{lab}$,Unlabeled Dataset $\mathcal{M}_{unlab}$, and labels    ▷ Scalar sequence equation 3.5
Output: Activity Labels $A_n$ for the $\mathcal{M}_{unlab}$    ▷ HAR.
procedure DEEP LEARNING HAR
    Forward Propagation
    for all timestamp t = 1 to T do    ▷ Sliding Widow Process
        $D_F \leftarrow \mathcal{M}_{lab}$    ▷ Retrieve Data (Feature Vectors Matrix)
        $x \leftarrow normalize(D_F)$    ▷ Preprocessing, reordering, filtering examples with no missing labels
        Sample, Split, FE, TV
        Initialize random weights $\{w_1, w_1, \dots w_n\}^T$ and biasness b
        $y = \sigma\left(\sum_{k=1}^{n} w_k x_k + b\right)$    ▷ applying nonlinear transformation $\sigma$ using $y = \sigma\left(w^T x + b\right)$
        $fc_y \leftarrow fully\_connected\_NN(y)$
        $A_n \leftarrow soft\_max(fc_y)$    ▷ Update weights in the network
        Backward Propagation
        Compute Cross entropy gradient    ▷ Use trained network to predict Activity labels
        Apply gradient descent    ▷ Update network parameters
    end for
    Activity Labels ← Use trained network model    ▷ Predict labels
end procedure
```

Artificial Neural Networks for HAR

We introduced a Semantic Deep Learning-based Artificial Neural Network (*SemDeep-ANN*) having the ability to extract hierarchy of abstract features [191, 192] using a stack of convolutional operators, which are supported by Convolutional Neural Networks (*CNN*). *SemDeep-ANN* consists of three layers namely *input layer*, *hidden layers*, and *output layer*, which use vectorized data to train model for probability estimation over the test data. The estimated probabilities are obtained from the output layer through the *soft_max* activation function in addition to gradient descent algorithm. Further details of the *SemDeep-ANN* are given in Algorithm 4.

Chapter 4

Vision-based Multioccupant State Imputation

This Chapter initially outlines the design challenges as how to impute missing objects in the thermal frames. The proposed Vision-based Multioccupant State Imputation methodology uses multioccupant previous states and predict the missing states, which results in increased HAR recognition accuracy. It further presents the algorithmic solutions and their inner details to deal with vision-based challenges.

4.1 Introduction

The main challenges in CV-based object detection and tracking applications are correct identification of ROI, reliable and efficient handling of moving objects along-with their inter-frame associations. These challenges, however, become even more complex for interacting multi-objects, which may have erratic movements represented by low-resolution appearances in a frame sequence. For this, an efficient method is required to predict their motion and manage data association [193]. Additionally, recognition of interaction amongst objects and classification of activities is also a computationally intensive task and requires a more robust process. This further requires a trade-off when implementing the above-mentioned methods in a more efficient manner for a complete, coherent and correct detection, tracking and classification of an occupant's activities. To address the aforementioned challenges, as presented in Fig. 4.1, we propose a unified scalable *unobtrusive Multi-occupant Detection and Tracking* (*uMoDT*) method, which detects, tracks and recognizes different indoor activities under multioccupancy using TVS.

The *uMoDT* method addresses six strategies as described below:

- We propose an online method, which uses a CV-based algorithm, with improved morphological features, for an automatic multi-target initialization using frame differencing with an

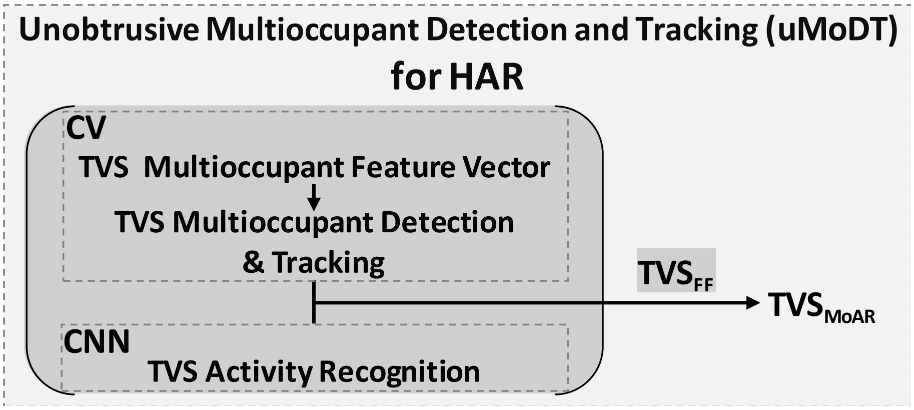

Unobtrusive Multioccupant Detection and Tracking (uMoDT) for HAR
CV
TVS Multioccupant Feature Vector
TVS Multioccupant Detection & Tracking
CNN
TVS Activity Recognition
TVS$_{FF}$
TVS$_{MoAR}$

Figure 4.1: Overview of proposed solution strategies as *uMoDT* framework

optimum threshold.

- We rely on refined morphological characteristics, which ensure efficient detection and tracking accuracy over the dynamic patterns for nonrigid moving targets per-frame.
- We use the Hungarian method for track assignment problem with an approach for maintaining an association history of re-identified tracks of individual moving objects per-frame.
- The proposed method is validated using a dataset gathered at Smart Environments Research Group (SERG) laboratory from the Ulster University, UK. It proved to be computationally robust and achieves a promising tracking accuracy in comparison with other MOT methods.
- We also demonstrated quantitative evaluations on the publicly available dataset for the VOT-TIR2016 challenge proving the practicality and efficacy of the proposed method with the state-of-the-art.
- Additionally, we propose to apply a CNN architecture to extract and learn spatial features from multiple successive Thermal Vision Sensor Frame (TVS-F) for individual action recognition.

The focus of the presented work is to simultaneously detect multi-occupants as well as recognize their activities frame-by-frame from TVS. It also requires a solution for resident data association in a smart-home environment, which is accomplished by unifying two different approaches.

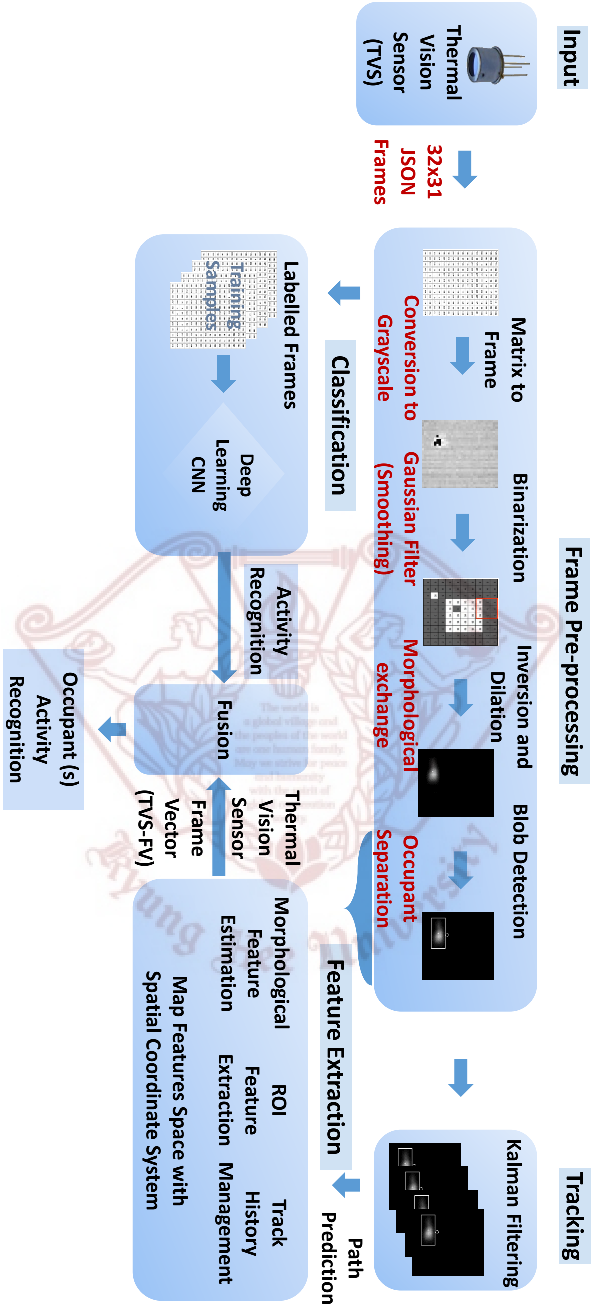


Figure 4.2: Proposed unobtrusive Multi-occupant Detection and Tracking (uMoDT) method for HAR

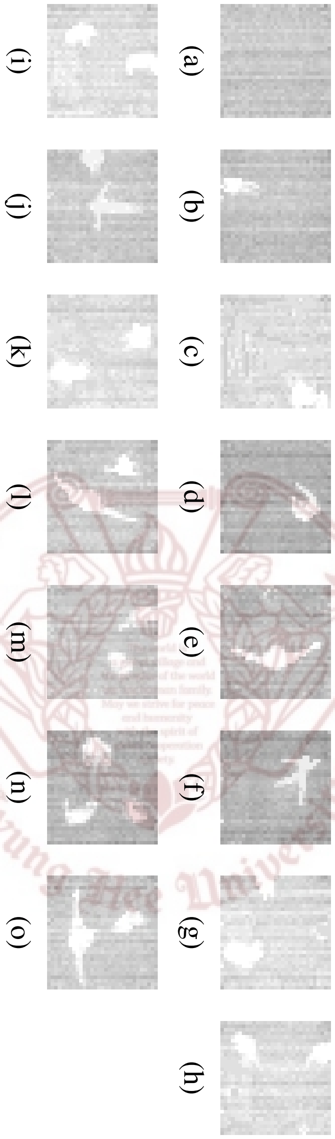


Figure 4.3: (a) Empty smart living room. Single occupant activities shown as (b) *Sitting* (c) *Standing* (d) *Walking* (e) *Stretching* (f) *Fall Down*. Multi-occupant activities shown as (g) Two persons *Sitting* (h) One person *Sitting* while other *Standing* (i) One person *Sitting* while other *Walking* (j) One person *Standing* while other *Fall Down* (k) Both persons *Standing* (l) One person *Standing* while other *Stretching* (m) Both persons *Walking* (n) All are *Walking* (o) one person *Walking* while other one *Stretching*

Firstly, using the CV-based technique, which detects, tracks, and monitors the occupant within the controlled area by observing a robust frame difference between the consecutive frames. Secondly, the CNN layers are invoked by the TVS frame sequence (TVS_F_{seq}), which recognizes the occupant's individual activities such as *Walking*, *Standing*, *Sitting*, *Fall down*. Finally, the recognized activities are associated with each occupant using the proposed Thermal Vision Sensor Feature Fusion (TVS_{FF}) method per frame.

Algorithm 5 TVS-MoFV: Thermal Vision Sensor multi-occupant frame vector algorithm

```
Input TVS_F: Thermal Vision Sensor grayscale sequence frames;
Output: Multi-occupant Frame Vector $TVS_{MoFV}$.
procedure TVS_MatPreProcessing
    Load $TVS\_F_{seq} \leftarrow \{TVS\_F_1, TVS\_F_2 \ldots TVS\_F_n\}$ where i = $\{1, 2, \ldots n\}$
    Read Matrix $TVS\_F_{seq}$                          ▷ Reads sequence of frames $TVS\_F_{seq}$
    for all $TVS\_F_i$ to $TVS\_F_n$ do
        function $Low\_thresholding(TVS\_F_i)$
            $TVS\_F_i - TVS\_F_{i-1} > TVS_{Th}$        ▷ Frame differencing sensitive to threshold
            $\mathbf{B}_n \leftarrow TVS\_F_i$              ▷ Identify 'n' Occupants as Blobs
            $TVS\_F_i \leftarrow \mathbf{Gauss}_{k,l}(TVS\_F_i)$   ▷ Smoothing by Gaussian blur k=l=3
        end function
        function $morphologicalTVSPreProcessing(TVS\_F_i)$   ▷ Morphological filtering
            $TVS\_F_i \leftarrow \mathbf{E}_{k_w,k_h}(TVS\_F_i)$   ▷ Erode: width 'w' & height 'h' =8
            $TVS\_F_i \leftarrow \mathbf{D}_{k_w,k_h}(TVS\_F_i)$   ▷ Dilate: width 'w' & height 'h' =8
        end function
        function $Detect\_Contour(TVS\_F_i)$
            $\mathcal{C}nt_n \leftarrow TVS\_F_i$
            Find $\mathcal{C}nt_n Contours$
            for all i = 1 to n do
                $min(\mathbf{B}) < \mathcal{C}nt_i < max(\mathbf{B})$   ▷ $min\_Blob\_Area < ContourArea < max\_Blob\_Area$
                $\mathbf{P}_{x_i,y_i} \leftarrow \mathcal{C}nt_i(p_{x_i}, p_{y_i})$
                $\mathbf{BR}_n \leftarrow boundingrectangle(\mathbf{P}_{x_i,y_i})$   ▷ Assign $Bounding\_Rectangle$
                array $[\mathbf{BR}] \leftarrow \mathbf{BR}_n$         ▷ Populate $Rectangle\_Array$
                $\mathcal{P}_{\square} \leftarrow$ array $[\mathbf{BR}]$   ▷ GetContourFeatures $Perimeter$
                $\mathcal{A}_n \leftarrow area(\mathbf{P}_{x_i,y_i})$   ▷ GetContourFeatures $Area$
                $\mathcal{A}_{\blacksquare} \leftarrow$ array $[\mathbf{BR}]$
                $\overline{\mathcal{P}}_{\mathbf{avg}} \leftarrow Average_{pixels}(\mathbf{BR}_n)$   ▷ Compute pixel p, average avg for $Bounding\_Rectangle$
            end for
        end function
    end for
    return $TVS_{MoFV} \leftarrow [\mathbf{P}_{x_i,y_i}, \mathcal{P}_{\square}, \mathcal{A}_n, \mathcal{A}_{\blacksquare}, \overline{\mathcal{C}\mathbf{nt_n}}]$
end procedure
```

4.2 Materials and methods

4.2.1 Computer vision-based occupant detection and tracking

This Chapter describes the inner details of the proposed method to detect the presence of multi-occupants in real-time, and track them throughout the duration of TVS_F_{seq} by following them from frame-by-frame. Fig. 4.2 illustrates the overall *uMoDT* method with underlying several components, namely *TVS sensor* as an *Input* device, *TVS-F Preprocessing*, *Occupant Tracking*, and *TVS-F Feature Extraction*. These components are connected in series whereas the information flow between subcomponents is discussed further in the following subsections.

Input frames

In this study, we propose to mount the Heimann HTPA TVS [194] in the ceiling of the smart-home's living room and kitchen at the height of 3m. The monitored space is a quadrilateral area with dimensions 4×3.5m. This setting provides a clear field of view and collects an aerial view of the multi-occupants as seen in Fig. 4.3. It also overcomes the challenges related to occupant-to-occupant and, occupant-to-scene occlusion, whilst avoiding camera motion and is operative even in complete darkness. The TVS ensures a high degree of user's privacy by capturing low-resolution grayscale TVS_F_{seq} with the dimensions of $32\text{h}\times31\text{v}\times1$. Each of the 992 pixels correspond to an area within the smart living room and kitchen represented by each pixel value ranging between 0 and 255. This range sets a correspondence of every pixel with an average temperature characteristic to that area. The TVS_F_{seq} is managed by using RESTful HTTP services, which are processed by the server.

Multi-occupant Feature Vector (TVS-MoFV)

The frames represent the presence of heat sources within the TVS_F_{seq}. The characteristics of identified heat sources are calculated by using the proposed Thermal Vision Sensor Multi-occupant Feature Vector (TVS-MoFV) algorithm. It gathers multi-occupant feature vectors in TVS_F_{seq} frame-by-frame. The series of tasks performed by TVS-MoFV are described in Algorithm 5, which are summarized as follows:-

- Converts the JSON 32×31 matrices into the sequence of frames TVS_F_{seq}.
- Segments the TVS_F_n frames in order to detect foreground (multi-occupant) and background (static smart living room or kitchen) per frame.
- Applies the *Low_thresholding* TVS_{Th} function with a background subtraction method sensitive to threshold [195].
- Convolves the TVS-F using Gaussian Kernel $Guass_{k,l}$ for smoothing and reducing noise with the kernel k=l=3.
- Performs morphological filtering and binarization on TVS_F_n to reduce the thermal noise using operations such as Erode $\mathbf{E}_{k_w,k_h}$ and Dilate $\mathbf{D}_{k_w,k_h}$.
- Determines the presence of multi-occupant using connected pixels termed as the contours $\mathcal{C}nt_n$ represented by blobs in the sequence of binary frames TVS_F_n.
- Assigns and encapsulates each identified $\mathcal{C}nt_i$, within the ROI, represented by *Bounding Rectangles* i.e. $\mathbf{BR}_n$.
- Estimates the centroid $\mathbf{P}_{x_i,y_i}$ for the identified $\mathcal{C}nt_i$ surrounded by $\mathbf{BR}_n$, which acts as a pivot for further tracking.
- Computes an array of the morphological feature vector for every TVS_F_i frame, which includes Perimeter $\mathcal{P}_{\square}$, Area $\mathcal{A}_{\blacksquare}$, and Contour Pixel Average $\overline{\mathcal{P}}_{\mathbf{avg}}$ for every $\mathbf{BR}_n$ in the TVS_F_i.

The learned frame vector TVS_{MoFV} from every TVS-F comprises of the morphological states of the detected occupant. These states represent the occupant's thermal area, a center of contour, a perimeter of the bounding box, and the area enclosed within the bounding box encapsulating the occupant. These multiple features become the basis for TVS_{MoAR} with the support of the proposed method TVS_{FF} required for the data association before recognizing and associating individual activities.

Algorithm 6 TVS-MoDT: Thermal Vision Sensor multi-occupant detection and tracking algorithm

```
Input TVS_F: Thermal Vision Sensor gray-scale frame sequence;
Output: Multi-occupant tracks $T_{MoDT}$.
procedure TVS_MATPREPROCESSING
    Load $TVS\_F_{seq} \leftarrow \{TVS\_F_1, TVS\_F_2 \dots TVS\_F_n\}$ where i = $\{1, 2, \dots n\}$
    Read Matrix $TVS\_F_{seq}$    ▷ Reads Sequence of Frames $TVS\_F_{seq}$
    for all $TVS\_F_i$ to $TVS\_F_n$ do
        function VECTORPOINT $\mathcal{V}_p(TVS\_F_i, \mathcal{C}nt_n)$    ▷ Detect_VectorPoint
            for all i = 1 to n do
                $\mathcal{P}_c^+ \leftarrow \mathbf{BR}_n \{\mathcal{C}nt_n\}$    ▷ Iterate $Contours$
                array $[\mathbf{D}] \leftarrow \mathcal{P}_c^+$    ▷ Array of detections
                $TVS\_F_i \leftarrow$ Draw $(\mathbf{BR}_n, TVS\_F_i)$    ▷ $drawRectangle \leftarrow Contours$
                $TVS\_F_i \leftarrow$ Draw $(\mathcal{P}_c^+, TVS\_F_i)$    ▷ $drawCenterPoint \leftarrow Contours$
            end for
        end function
        function TRACK $\mathcal{T}_i(\mathcal{C}nt_n, \mathbf{D}, TVS\_F_i)$    ▷ initialize $(NoOfTracks, TrackSize)$
            for all i = 1 to Size ([D]) do
                $\mathcal{T}_i \leftarrow$ new$(\mathcal{T}, \mathbf{D})$
                $Cost\,[i]\,[i] \leftarrow Euclid(\mathcal{T}_i^{pred}, \mathbf{D})$    ▷ Euclidean distance between prediction & detection
                $\mathcal{C} \leftarrow Cost\,[i]\,[j]$
                $\overrightarrow{\mathcal{A}} \leftarrow Vector(Assignment)$
                $\mathcal{T}_i^{assign} \leftarrow HungarianAssignment\ (\mathcal{C}, \overrightarrow{\mathcal{A}})$
                if $(\mathcal{C} > \mathcal{D}_{threshold})$ then    ▷ Identify $unAssigned\_tracks$
                    $[\mathcal{T}_i^{unassinged}] \leftarrow$ add$(\mathcal{T}_i^{unassinged})$    ▷ Search $Un\_Assigned\_Tracks$
                end if
                if $([\mathcal{TVS}\_\mathcal{F}_i^{skipped}] > max_f)$ then
                    $\mathcal{TVS}\_\mathcal{F}_i \leftarrow$ remove $(\mathcal{TVS}\_\mathcal{F}_i)$    ▷ Remove not detected tracks
                    $\overrightarrow{\mathcal{A}} \leftarrow$ remove $(\overrightarrow{\mathcal{A}_i})$    ▷ Remove assignments
                end if
                if $(size(\mathbf{D}_i^{unassigned}) > 0)$ then
                    $\mathcal{T}_i \leftarrow$ add$(\mathcal{T}_i, \mathbf{D}_i^{unassigned})$    ▷ Initialize New_Tracks for un_Assigned_Detects
                end if
                $\mathcal{T}_i \leftarrow \mathcal{T}_i^{skipped} > \mathcal{TVS}_{SkippedAllowed}$
                /* Update Kalman for All Detected Contours */
                $\mathcal{TVS} \leftarrow UpdateKalman(TVS\_F_i, \mathbf{D})$ ▷ Predict, Update Kalman Occupant State
                /* Iterate the No of contours, detections in the $\mathcal{TVS}\_\mathcal{F}_i$ */
                for all t = 1 to Size $(\overrightarrow{\mathcal{A}})$ do
                    $\mathcal{T}_{id} \leftarrow \mathcal{T}_i(t)$
                    $\mathcal{TVS}\_\mathcal{F}_i \leftarrow \mathcal{TVS}\_\mathcal{F}_{append}(\mathcal{TVS}\_\mathcal{F}_i, \mathcal{T}_{id}, \mathcal{P}_c^+)$    ▷ Draw tracks
                    $[\mathcal{TVS}\_\mathcal{F}_i]_{history} \leftarrow \mathcal{TVS}\_\mathcal{F}_{append}(\mathcal{TVS}\_\mathcal{F}_i, \mathcal{P}_c^+)$ ▷ Contours & Tracks History
                end for
                /* Update $\mathcal{TVS}\_\mathcal{F}_i$ with Kalman Prediction and Correction */
                $\mathcal{I}t \leftarrow n\,(\mathcal{C}nt_n)$    ▷ Number of Contours
                while $\mathcal{I}t.hasnext$ do
                    $TVS\_F_i \leftarrow$ update $(TVS\_F_i, \mathcal{P}_c^+, [\mathcal{TVS}\_\mathcal{F}_i]_{history})$    ▷ Kalman Effect
                    $TVS\_F_i \leftarrow$ draw $(\mathcal{P}_c^+, [\mathcal{TVS}\_\mathcal{F}_i]_{history})$ ▷ Kalman prediction updation
                    $TVS\_F_i \leftarrow$ draw_line $(\mathcal{P}_c^+, \mathcal{T}_{i-1}, \mathcal{T}_i, [\mathcal{TVS}\_\mathcal{F}_i]_{history})$
                end while
            end for
        end function
    end for return $T_{MoDT}$
end procedure
```

Multi-occupant Detection and Tracking (TVS-MoDT)

Algorithm 6 describes the TVS Multi-occupant Detection and Tracking (TVS-MoDT) method to identify, predict, plot, visualize, and maintain the occupant's tracks within TVS_F_{seq}. Some of the key features for this algorithm are summarized as below:-

- The TVS_F_{seq} is read as input simultaneously as in the case of Algorithm 5.
- The detected contours $\mathcal{C}nt_i$ through Algorithm 5 are iterated within TVS_F_{seq} for computing the vector point $\mathcal{V}_p$ responsible for tracking and maintaining the history of the tracks as shown in Line 4-11.
- For every detection $\mathbf{D}$ for $\mathcal{C}nt_i$ the tracks $\mathcal{T}_i$ are initialized as shown in Line 15.
- We used two classical efficient methods, Hungarian method, and KF to handle the occupant's data association and smoother motion refinement with position prediction of the multi-occupant respectively.
- The optimal assignment $\overrightarrow{\mathcal{A}}$ and cost $\mathcal{C}$ computation task for tracks $\mathcal{T}_i^{assign}$ is performed using the Hungarian method.
- We employed KF to generate multi-occupant motion trajectories i.e. estimation and position prediction for the blob representing each of the individual occupants as mentioned in the Line 32.
- The *UpdateKalman* prediction function predicts the position of the occupant based on the history from previous TVS-F whereas the update function rectifies the state of the multi-occupant from the current TVS-F (Lines 39-45).
- Every multi-occupant being tracked is assigned *Tracking ID* ($\mathcal{T}_{id}$) representing tracklets. The morphological features such as position, size and other statistical measurements are also calculated for blob.
- $\mathcal{T}_{id}$ is dynamically assigned (or reassigned) to blobs with rapidly varying sizes. The array with tracking identifiers represents each occupant's motion model and state history.

4.2.2 CNN-based activity classification

The CNN has been utilized for real-time multi-occupant AR from the TVS_F_{seq}. It is computationally built on five major mathematical functions such as *Convolution*, *Batch Normalization*, *Rectified Linear Unit (ReLU)*, *Pooling*, and *Soft-max*. These functions are applied in a hierarchical residual block within an architecture, which provides fully connected layers for processing TVS_F_{seq} to get multi-occupant activity classification output per frame. These are briefly discussed in the following subsections.

Input layer

An input layer for the CNN architecture reads the grayscale TVS_F_{seq} of the fixed dimensionality, requires TVS_F_{Train} to train the model while producing an output $TVS\_F_{labelled}$, representing "n" activities performed by the multi-occupants.

$$TVS\_F_{labelled} \leftarrow \{TVS\_F_{seq}, TVS\_F_{Train}, act_n\}_{CNN} \tag{4.1}$$

Convolutional layer

The Convolutional Layer is responsible for extracting the pixel-wise features from the input TVS-F. To learn the TVS-F features, the kernel weights are adjusted automatically through back-propagation training. The convolution is obtained by taking dot product (•) between sub-part of the TVS-F and the convolutional kernel K. In response, a feature map f_c is computed by sliding the convolutional kernel over the TVS-F spatially. The output $x_i^{l,j}$ for the l^{th} convolutional layer having the j^{th} feature map on the i^{th} unit can be presented mathematically as:

$$x_i^{l,j} = \sigma\left(b_j + \sum_{a=1}^{m} w_a^j x_{i+a-1}^{l-1,j}\right) \tag{4.2}$$

where σ is a non-linear mapping, it uses hyperbolic tangent function, $\tanh(\cdot)$ [196].

Batch normalization layer

The input channel x across the mini-batch is normalized $\hat{x_i}$ by the introduction of a batch normalization layer [197]. Normalized activation is computed using mini-batch mean μ_B, standard deviation σ_B^2 for input channel x, and ϵ to provide the numeric stability for mini-batch variances, described as:

$$\hat{x_i} = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{4.3}$$

It increases the performance of CNN training and reduces sensitivity of the neural nets.

ReLU layer

Rectified Linear Unit (ReLU), a nonlinear activation function responsible for introducing a point-wise non-linearity to the CNN by resolving the vanishing gradient problem [75]. ReLU layer processes an element-wise activation function over each individual input x, wherever the value is less than zero, is set to zero and it also linearly conveys the input for positive inputs described by Eq. 4.4:

$$f_\tau = ReLU(x_i) = \begin{cases} x_i, x_i \geq 0; \\ 0, x_i < 0; \end{cases} \tag{4.4}$$

A rectified feature map f_τ is obtained as an outcome.

Max-pooling layer

The max-pooling layer produces compact feature space by taking the sub-samples of f_τ thus reducing the spatial dimensionality and sensitivity of the output. The pooling operation derives maximum value from the set of nearby inputs as mentioned in equation 4.2, which can also be represented mathematically as [170]:

$$f_i^{l,j} = \max_{r \in R}(x_{i \times T + r}^{l,j}) \tag{4.5}$$

where R represents pooling size and T as a pooling stride. The soft-max classifier is placed at the final layer for HAR. The TVS-F features obtained from the stacked convolutional and pooling are represented as:

$$f^l = [f_1, f_2, f_3.....f_K] \tag{4.6}$$

where K represents the number of units learned from the last pooling layer, which acts as a feature map for the soft-max classifier.

Table 4.1: List of 16 activities recorded for data collection

Activity ID	Activity Type	Activity Name	No. of Occupants
Act_1	Single	FallDown	1
Act_2, Act_3	Single, Multi	Sitting	1, 2
Act_4	Multi	SittingStanding	2
Act_5	Multi	SittingWalking	2
Act_6, Act_8	Single, Multi	Standing	1, 2
Act_7	Multi	StandingFallDown	2
Act_9	Multi	StandingStretching	2
Act_{10}	Multi	StandingWalking	2
Act_{11}	Single	Stretching	1
$Act_{12}, Act_{15}, Act_{16}$	Single, Mutli	Walking	1, 2, 3
Act_{13}	Multi	WalkingFallDown	2
Act_{14}	Multi	WalkingStretching	2

Training process

The CNN is trained in a supervised learning fashion by selecting the parameters using *Gradient-based optimization* method. For faster convergence, the *stochastic gradient descent* method is applied [198]. The training process involves a series of steps such as *propagation* and *weight update*. The gradients are computed in the propagation step by using *standard forward* [196] and *back-propagation* algorithms [199], by minimizing the objective function, which is given mathematically as:

$$x_i^l = \sum_j w_{j,i}^{l-1} \sigma(x_i^{l-1}) + b_i^{l-1} \tag{4.7}$$

where x_i^l represents the output feature and *w* is the weight vector. The output feature map is passed to every subsequent layer till it reaches the output layer, which is formulated as:

$$\frac{\partial L}{\partial y_{i,j}^{l-1}} = \sum_{a=0}^{m-1} \frac{\partial L}{\partial x_{i-a}^{l}} \frac{\partial x_{i-a}^{l}}{\partial y_{i,j}^{l-1}} = \sum_{a=0}^{m-1} \frac{\partial L}{\partial x_{i-a}^{l}} w_{a,b} \tag{4.8}$$

It applies *chain-rule* for computing the propagation error and the whole process remains cyclic until the CNN reaches a satisfactory validation state or attains the stopping criterion.

Classification

The soft-max regression function in the final layer of the neural network leads to the multi-occupant HAR using *TVS-based Activity Recognition* (TVS-AR) method. It normalizes the output, which is computed by fully connected layers, and more often is a combination of a set of positive numbers with their sum equivalent to one, and value ranges between $[0 \ldots 1]$. These ranges are further transformed into classification probabilities through the *Classification* layer in the CNN residual block. The *i-th* probability value for soft-max function $p(y_i)$ [200] is computed as:

$$\hat{y}_i = p(y_i) = softmax(x_i) = \frac{exp(x_i)}{\sum_{k=1}^{n} exp(x_k)}, i = 1 \ldots N_c \tag{4.9}$$

The cross-entropy [199] is minimized between the output probability vector $\hat{y}$ and total number of class labels 'y' as follows:

$$E = -\sum_{i=1}^{N_c} (y_i log(\hat{y}_i) + (1 - y_i) log(\hat{y}_i)), i = 1 \ldots N_c \tag{4.10}$$

where y_i represents binary indicator if the class label 'c' is correctly classified from the i^{th} neuron and $\hat{y}$ is the predicted probability of the i^{th} class.

4.3 Experimental Results

The complete real-time prototype application for our proposed *uMoDT* method is built for multi-occupant detection, tracking and AR. To demonstrate the functionality of the *uMoDT* method, we first discuss the dataset and later the implementation insights.

Table 4.2: List of benchmark dataset sequences and their details

ID	Dataset	Sensor	Resolution	Frames	Object	Threshold
1	ETHZ-CLA [201]	FLIR TAU320	324×256	659	Human	115
2	Soccer [9, 202]	3×AXIS Q-1922	1920×480	3,000	Human	120
3	Crouching [202]	FLIR A655SC	640×480	625	Human	125
4	Depthwise Crossing [202]	FLIR A655SC	640×480	858	Human	135
5	Crowd [202]	FLIR Photon 320	640×512	78	Human	110
6	TVS_F_{seq}	Heimann	32×31	57,290	Human	155

Table 4.3: Processing time for benchmarks and TVS_F_{seq} with TVS-MoDT and TVS-AR algorithms

Algorithm	Dataset	Duration(s)
TVS-MoDT	ETHZ-CLA	3.91×10^{-6}
	Soccer	2.99×10^{-6}
	Crouching	6.35×10^{-6}
	Depthwise Crossing	2.93×10^{-6}
	Crowd	2.93×10^{-6}
	TVS_F_{seq}	4.88×10^{-6}
TVS-AR	$TVS\_F_{seq}(O=1)$	7.1×10^{-2}
	$TVS\_F_{seq}(O=2)$	8.3×10^{-2}
	$TVS\_F_{seq}(O=3)$	9.0×10^{-2}

4.3.1 Dataset

We collected 57,290 frames in a sequence from three healthy male volunteers aging 25±7 [yrs]; height 1.55±0.7 [m] and weight 68±8 [kg]. Each volunteer performed different ADLs individually and collectively in a smart living room over a duration of at least 3 minutes each, reported in Table 4.1. During the entire collection, the application was neither reparameterized nor recalibrated, which means this setting remained valid for all kind of ADLs performed during this study. Additionally, TVS_F_{seq} was annotated with *LabelImg*, an open source annotation tool [203]. During labeling, multi-occupants were approximated by using bounding rectangles over the subsequent frames by assigning them unique identifiers referred as ground-truth $\mathbf{G}_i$ in the TVS_F_{seq}. This process followed a strict annotation protocol by qualified researchers.

The goal is to quantitatively evaluate the proposed *uMoDT* method and prove its accuracy and robustness. For this, we tested and compared it, also on five challenging, publicly available annotated sequences from VOT-TIR2016 challenge [202, 204]. These sequences were mostly captured with the help of static FLIR and thermal cameras.

Table 4.4: TVS-AR: Activity recognition for multi-occupants using Convolution Neural Networks

Layer	Layer Type	Activation	Parameters (No. of units, Size, Stride)
1	TVS_F_{seq}	Image Input	32×32×1 images with zerocenter normalization
2	conv1	Convolution	16 3×3×1 convolutions with stride [1 1] and padding [1 1 1 1]
3	batchnorm1	Batch Normalization	Batch normalization with 16 channels
4	relu1	ReLU	ReLU
5	maxpool1	Max Pooling	2×2 max pooling with stride [2 2] and padding [0 0 0 0]
6	conv2	Convolution	32 3×3×16 convolutions with stride [1 1] and padding [1 1 1 1]
7	batchnorm2	Batch Normalization	Batch normalization with 32 channels
8	relu2	ReLU	ReLU
9	maxpool2	Max Pooling	2×2 max pooling with stride [2 2] and padding [0 0 0 0]
10	conv3	Convolution	32 3×3×32 convolutions with stride [1 1] and padding [1 1 1 1]
11	batchnorm3	Batch Normalization	Batch normalization with 32 channels
12	relu3	ReLU	ReLU
13	maxpool3	Max Pooling	2×2 max pooling with stride [2 2] and padding [0 0 0 0]
14	conv4	Convolution	64 3×3×32 convolutions with stride [1 1] and padding [1 1 1 1]
15	batchnorm4	Batch Normalization	Batch normalization with 64 channels
16	relu4	ReLU	ReLU
17	fc	Fully Connected	16 fully connected layers
18	soft-max	soft-max	Bayesian binary classifier
19	classoutput	Classification Output	crossentropyex with FallDown and 15 other classes

4.3.2 Implementation details

The proposed *uMoDT* method, comprising of *TVS-MoFV* (Algorithm 5), *TVS-MoDT* (Algorithm 6) and *TVS-AR* method, was implemented. The former algorithms utilize the Java-based standard libraries OpenCV (an open-source API) [205] while the latter method requires MATLAB interfaces (machine learning toolbox API). The *uMoDT* method was implemented and evaluated using the PC system equipped with AMD A10-5800K APU with Radeon(tm) HD Graphics (4 CPUs 3.8GHz), 16GB RAM, and NVIDIA GeForce GTX 750 GPU 4GB.

Proposed algorithms, *TVS-MoFV* for feature extraction and *TVS-MoDT* for multi-occupant detection and tracking were tested. Both of them used stored TVS_F_{seq}, which was retrieved from the intermediate repository as JSON object arrays, by a pull-based web-service. In TVS-MoFV, TVS-F vector was obtained by varying binary threshold values and finding the best value, suitable for each of VOT-TIR2016 benchmark datasets and the TVS_F_{seq} as mentioned in Table 4.2. The parametric settings also involved finding the optimal value for the contour area in order to predict the maximum number of occupants in the benchmarks and TVS_F_{seq} as shown in Fig. 4.4. These TVS-F feature vectors support while iterating the multi-occupant represented as Blobs predicted as bounding rectangles, implemented through the TVS-MoDT algorithm. The *Euclidean distance* was calculated between the detected and predicted bounding rectangles for multi-occupant tracking frame-by-frame. The processing time for each algorithm and method to process a single frame is referred to in Table 4.3. The source code for *uMoDT* method and TVS_F_{seq} is available on GitHub at [71].

To recognize multi-occupant's ADLs from TVS_F_{seq}, a supervised CNN model was trained. For this the entire collection of TVS_F_{seq} was sorted into two subset groups i.e. training and test categories, each having sixteen classes. The training set is further split with random TSV-F distribution into two halves i.e. 70% for training samples (TVS_F_{Train}) and remaining to validate each class. We used 28,485 TVS-F samples to train CNN model and 1,920 TVS-F test samples (120 TVS-F for each of 16 classes) to evaluate the prototype *uMoDT* method application.

The nineteen-layer, CNN architecture is designed based on the findings from the systematic comparison and benchmarking to achieve an affordable classification time and computation cost [206]. The implemented CNN architecture comprises of two units i.e. feature extractor and a non-linear classifier [168]. The former unit encapsulates fifteen layers (Layer2. . . Layer16) whereas the latter unit i.e. non-linear classifier is built on all fully connected layers along with the soft-max classifier. During the model training process, the CNN hyper-parameters were set with the help of input functions, by adjusting the learning rate effectively to 0.01, every 10 epochs using Stochastic Gradient Descent with Momentum (SGDM) algorithm with the maximum 20 number of epochs size [199]. For every iteration, a mini-batch of size 16 (64) was applied for which the details are mentioned in Table 4.4.. The output of the last ReLU (relu4) at *Layer 16*, is given to fully connected layer *Layer 17*, which uses the features and processes it for class prediction based on the TVS_F_{Train}. The classification layer i.e. *Layer 18* uses the soft-max activation function, which squashes the output probability vector between sixteen multi-occupant activities and returns the binary indicator to them.

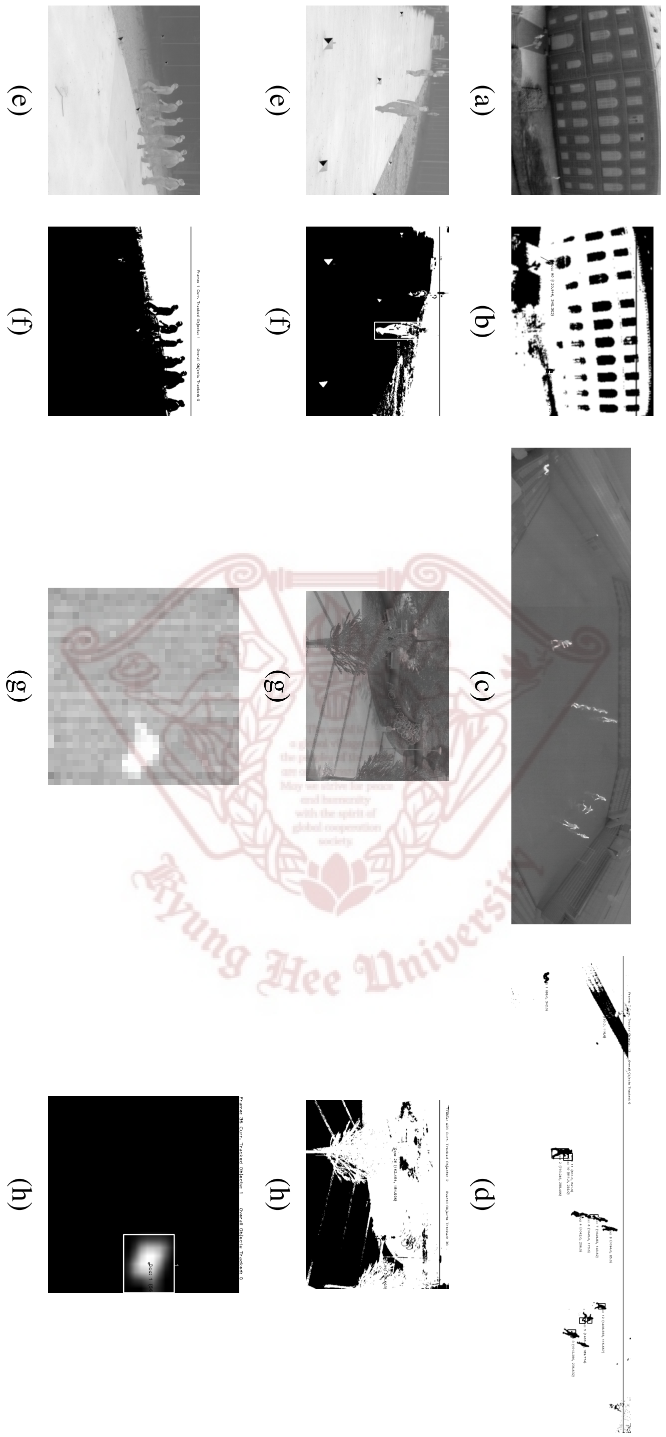


Figure 4.4: Examples of raw Input (I) frames and processed Output (O) frames using proposed method. (a) & (b) ETHZ-CLA (I&O) (c) & (d) Soccer (I&O) (e) & (f) Crouching (I&O) (g) & (h) Depthwise Crossing (I&O) (i) & (j) Crowd (I&O) (k) & (l) TVS-F (I&O)

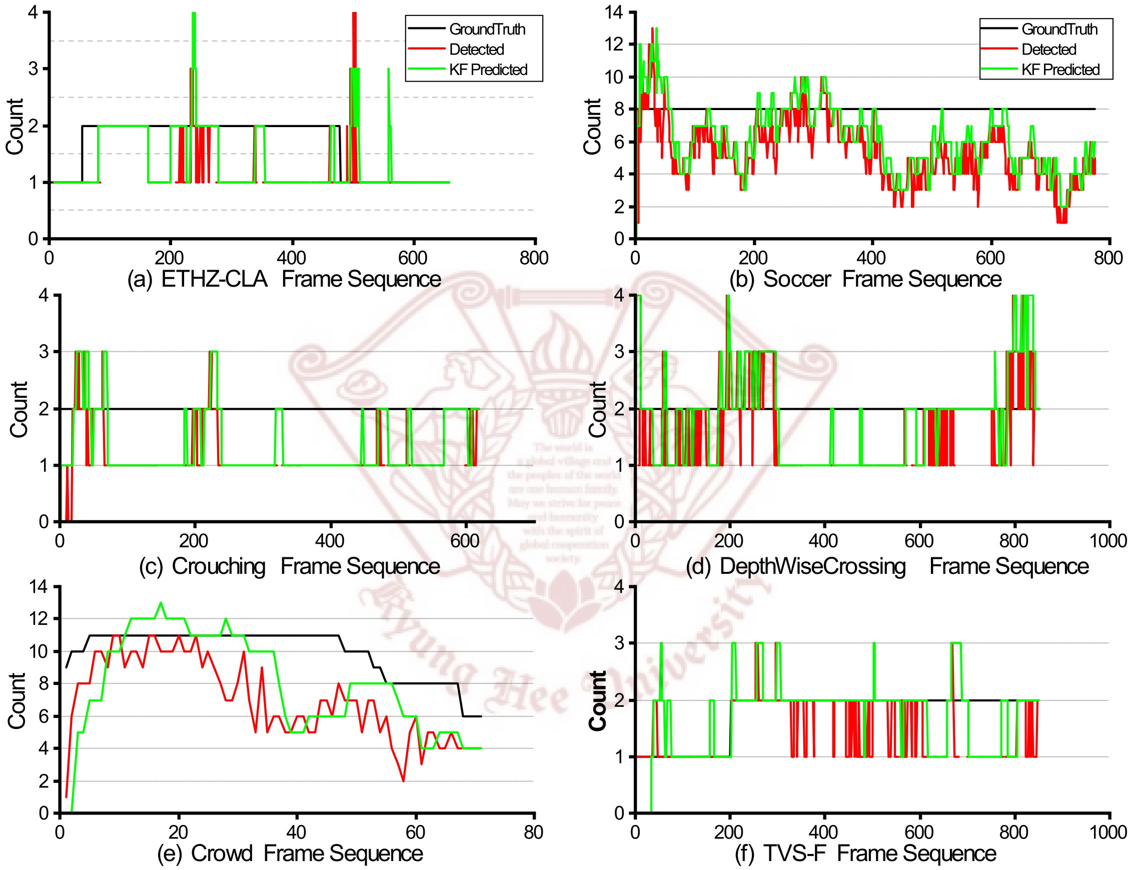


Figure 4.5: Quantitative evaluations shown in (a) ETHZ-CLA (b) Soccer (c) Crouching (d) Depth-wise Crossing (e) Crowd (f) TVS-F

Chapter 5

Evaluations and Results

This chapter presents data validation techniques, evaluation and results comparisons for the proposed Semantic Imputation framework for sensor-based and vision-based imputation for HAR. The results of the proposed framework were also compared with the state-of-the-art methods and performed experiments, show significant improvement in recognized activities. It is expected that semantic imputation methods would be a practical solution as compared to existing counterparts.

5.1 Data Validation

This thesis also outlines a strategy to evaluate the statistical properties of incomplete data, validation methods to evaluate modeling of missing data against the amount of missingness. The validation strategies also provide the basics as to how the observed data with strong statistical properties have more predictive power for classification models.

In this study, we validated the quality of SemImput solution for obtaining the valid inference from complete publicly available datasets. It is evident that the quality of multiple imputation solutions solely depends on statistical properties of complete data as well as for incomplete dataset. Another important aspect of how much missing data exists in the incomplete datasets. Generally, quality of missing data inference performs inversely with respect to the amount of missingness. However, the relationship among strong attributes in the missing data with respect to complete data or observed data can support the inference and predictive power of the classification models [207].

5.1.1 Performance Evaluations

Data imputation performance issues are mostly linked with the data generation, missingness generation, missing values identification and evaluation methods. For this study, however, publicly available HAR datasets are used which are already labelled, containing timestamps and missingness. We considered several evaluation criteria which are discussed in detail. At the same time a brief detail for the state of the art data validation techniques are also discussed as under.

Accuracy

As a performance evaluation measure, most of the researchers used "accuracy" to measure how well the imputation model, method or strategy reproduce the original data with a valid reference usually coming from complete data with all possible instances. The objective, in this case, is to compute as close as possible data from the reference data using the imputation method whereas the distance differences are measured between both data. This difference is analyzed through the accuracy measure between imputed data and observed data, which provides the percentage of correctly imputed missing values. We computed the accuracy metric for measuring the distance between the imputed data and reference data using SemImput model. The valid inferences using the designed ontologies not only deal with bias but also can deal with the artificially created missingness in the data. So the main goal is of the evaluation is to measure the effect of missingness on imputation in accordance to the percentage of missingness. Whether the proposed strategy successfully handles both in an efficient manner accurately and exhaustively.

Distributional Properties

The data distribution properties can also provide a very effective clue as to how the distribution of the ground truth values is compared with the distribution of the imputed values. Alternatively how the distribution of the observed values are compared with the distribution of all values including the imputed values.

In practice, it cannot be promised that the distribution of the incomplete data will be same as of the observed data. They can differ greatly from each other. It is important to observe and perform detailed analysis to keep a check on the distributional shale and properties of the data under consideration. Thus a diagnostic evaluation has to be performed for the datasets coming from the controlled environment so that proper data imputation strategy is devised to avoid any anomalies.

Sampling Variation

During the evaluation process sampling variation can be dealt with two possible approaches such as: (1). Model-based Simulation and (2). Design-based simulation approach [207]. The aforementioned Model-based simulation draws the samples from the multivariate normal distribution. During the simulations, the theoretical parameters, which are used to draw the samples from a known probability distribution act as the comparative truth. Whereas the subsequent approach of design-based simulation is most suitable where real-life HAR data structures are of interests i.e. in our case. Such an approach is, however, also suitable in such a situation where the probability distribution is not available. Such models with sampling variations are considered to be most computationally convenient where a single complete dataset is under consideration during simulations.

Quantitative and Qualitative Evaluations

The conclusions for the imputations methods can also be drawn and carefully qualified on the basis of population parameters. For this, the performance of an imputation technique can be validated using quantitative evaluations with the support of ground-truth population parameters. However, the qualitative evaluation using the simulated conditions for imputed values in comparison with ground-truth becomes more challenging when the imputation performance deviates from normality. In such circumstances, the qualitative evaluations become highly dependent on the simulation conditions. The comparative results by avoiding the performance drop, however, can be improved through non-parametric models. Though these models may perform badly but their results would still outperform any other approach having more practical relevance [207].

Checking the quality of imputation, which has a very little chance to introduce errors in data due to observed complete data already modelled into an ontology. So, there is no or little chance of introducing new patterns as these violate the ontology (T-BOX & A-BOX) consistency properties. Semantic rules, based on an observed data (Consistent) and a reasoner determines inconsistent or any invalid data for Test or data with missingness. Any inconsistency in the data is handled by reasoner resulting in "Unknown", which is catered through the updation / new Semantic rules representing the respective new activity.

Internal Consistency

An internal consistency check is one of the most commonly adapted graphical diagnostics method for performance evaluation over the ground-truth, observed data and imputed data. This method performs comparisons as an internal check for data to be imputed against the available observed data. These comparisons are drawn over the ground-truth, missed values and imputed values using boxplots, histograms, plots for cumulative distribution, quantile-quantile plots, strip plots and especially histograms. These plots mostly compare ground-truth and imputed data values, which demonstrates whether the observed values are positively skewed, negatively skewed or symmetrical distribution for imputed values [208].

In this thesis, SemImput, however, ensures all tuples meet the semantic rules, which uses *Pellet* reasoner to check logical consistencies, which segregates strong conflicts (those cannot be resolved) and weak resolvable conflicts. It also provides improvement of the ontology model to cover all cases and strengthening the weak conflicts by removing structural clash. Semantic rules are sufficient to handle all type of missingness required for SemImput Ontology representing activity recognition datasets.

5.1.2 Error Metrics

The performance of imputation methods can be measured using error metrics, which compare observed and imputed data concentrations across several levels of missingness such as 20%, 40%, 60%, 80% missing. HAR based studies are mostly concerned with daily activity data concentration associated with several sensors to measure the health effect. Absolute Bias (AB), Mean Absolute Percentage Error (MAPE), Coefficient of Determination (CoD), Root Means Square Error (RMSE), and Mean Absolute Error (MAE) are some of the metrics to evaluate error between observed and imputed HAR data within smart-home environment [35, 209].

In this thesis, the SemImput model also compares observed and semantically imputed concentrations across five levels of aforementioned missingness. It is also observed that anything less than 10% may be of little influence on the ground truth data. The experimental results also suggested that augmentation performance for under 50% percent of simulated missingness has a severe impact.

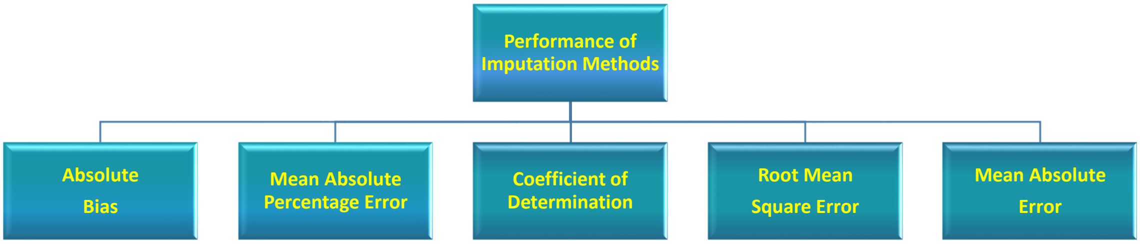

Performance of Imputation Methods
Absolute Bias
Mean Absolute Percentage Error
Coefficient of Determination
Root Mean Square Error
Mean Absolute Error

Figure 5.1: Error Metrics used for Performance Evaluations.

Absolute Bias

The results obtained through the proposed SemImput method has to be unbiased, as any bias can influence to misleading results and induce misinterpretation. So a negligible amount of bias for any parameter may yield infinite bias while calculating relative bias. The bias calculation as an error metric has to be performed carefully and a suitable method has to be adapted [207].

$$AB = |\ddot{x} - \hat{x}| \tag{5.1}$$

Mean Absolute Percentage Error

Similarly, MAPE error metrics provide the percent difference between the ground-truth HAR values and semantically imputed values. MAPE is again an easily computed and interpretable metric [35].

$$\text{MAPE} = \left| \frac{\ddot{x} - \hat{x}}{\vec{x}} \right| \cdot 100 \tag{5.2}$$

Coefficient of Determination

Most of the models are evaluated for their goodness of fit measure using the coefficient of determination, which is also the most common metric. As mentioned in Equation 5.3, it is calculated by squaring the correlation coefficient between ground-truth variables and imputed variables for their concentrations [35].

$$R^2 = \left(\frac{\sum_{i=1}^{n} (x_i - \hat{x}) * (\dot{x}_i - \ddot{x})}{\sqrt{\sum_{i=1}^{n} (x_i - \hat{x})^2} + \sqrt{\sum_{i=1}^{n} (\dot{x}_i - \ddot{x})^2}} \right)^2 . \tag{5.3}$$

Here x_i and $\dot{x}_i$ denotes the i_{th} observation for the ground-truth and imputed datasets, whereas, $\hat{x}_i$ and $\ddot{x}_i$ represent the means for the semantically imputed and ground-truth dataset. Such an error metric is widely used across many comparative studies and is an efficient comparable metric. However, it is subject to the limitation with increase size in the dataset where the imbalance of data exists between ground-truth and imputed values.

Root Mean Square Error

Another metrics, Root mean square error (RMSE) is widely used for determining the error between ground-truth and imputed values, which is computed through as mentioned [35]:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \dot{x}_i)^2} \tag{5.4}$$

Such an error metric again suffers the same issue as of R^2 when the dataset size become bigger and more differences exist between ground-truth and imputed variables. In such a case, these metrics may not produce appropriate results [35].

Mean Absolute Error

Lastly, mean absolute error (MAE) is another most commonly used metric for error estimation and evaluation between imputed and observed values [35].

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |x_i - \dot{x}_i| \tag{5.5}$$

MAE, however, is also less affected where large differences amongst the datasets are involved.

5.2 Results and Discussion for Multi-strategy Data Imputation

The performance evaluation for *SemImput* framework is measured using non-imputed and semantically imputed HAR datasets. The results are compared with other popular methods, which were investigated using the same datasets.

5.2.1 Data Description

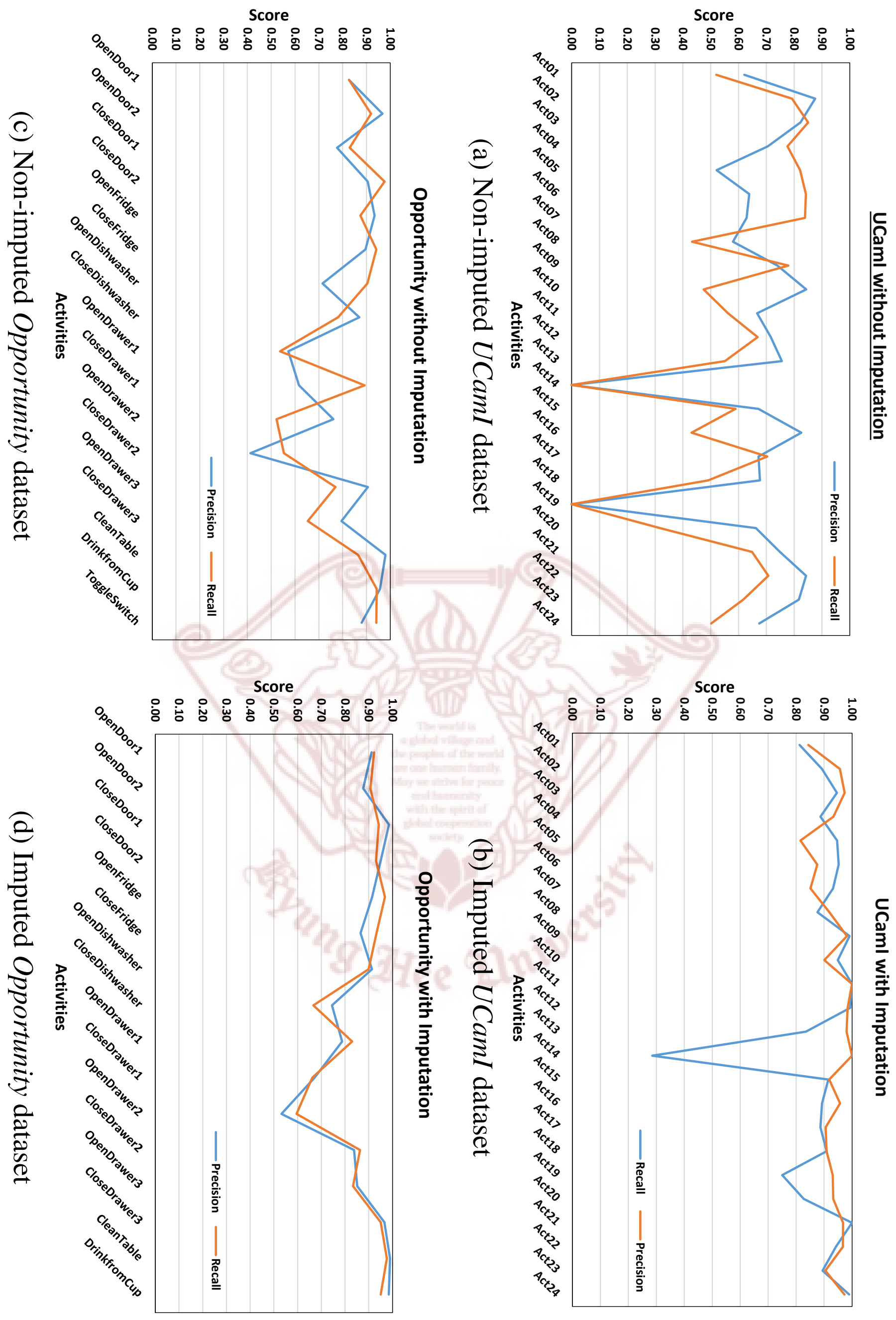
To compare the HAR performance of the proposed *SemImput* framework, firstly, the experiments were performed on the *UCamI* dataset. It offers recognition of 24 set of activities for non-imputed and imputed datasets. Secondly, the *Opportunity* dataset contains manipulative gestures of short duration such as *opening* and *closing*, of *Doors*, *Dishwasher*, and *Drawers*. These were collected for four subjects who were equipped with five different body attached sensors for the tracking of static and dynamic activities [210]. Due to the involvement of several sensors, data transmission problems among wireless sensors lead to segments of data being missed represented by *Null*. For this reason, we analyzed the data and performed the required imputation in order to complement the missing segments of data [77, 192]. Lastly, we tested *SemImput* framework on the UCI-ADL dataset, which was collected while monitoring 10 different ADLs [211] using *passive infrared*, *reed switches*, and *float sensors*. These sensors were used to detect *motion*, *opening* and *closing* binary states of the objects and activities such as *toileting*, *sleeping*, *Showering*.

5.2.2 Performance Metrics

We measured the impact of imputation against the non-imputed datasets using commonly used metrics, such as accuracy, precision, and f-measure. The *SemDeep-ANN* models were validated by splitting the datasets independently into train and test sets using a *leave one day out* approach. During the evaluation process, we retained one full day from each of the dataset for testing, whereas the remaining samples are used as a training set. This process is repeated for each day, with the overall average accuracy obtained as a performance measure.

5.2.3 Discussion

This study examines and evaluates the *SemImput* framework for HAR classification results for which the precision and recall curves are shown in Figure 5.2 a–d and Figure 5.3 e–h. The framework achieved an overall accuracy of 71.03% for set of activities recognized from non-imputed *UCamI* dataset as mentioned in Table 5.1 and Table 5.2. The activity *Prepare breakfast* (Act02) yielded the highest precision of 87.55%, but it was also misclassified with the activities *Breakfast* (Act05) and *Dressing* (Act22) respectively. Similarly, the activity *Enter the Smartlab* (Act10) was also classified with the highest precision, it was, however, misclassified as the activity *Put waste in the bin* (Act15). The activity *Breakfast* (Act05) with the lowest precision 52.14% was mostly misclassified as activities *Prepare breakfast* (Act02) and *Wake up* (Act24). Furthermore, the activity *Eat a Snack* (Act08) with lower precision of 57.95% was misclassified as the activity *Prepare Lunch* (Act03) due to the involvement of similar sensors and floor area. The activity *Visit in the SmartLab* (Act14) and *Wash dishes* (Act19) was hard to detect as they have lessor number of annotated examples. The experimental results indicate an increased recognition accuracy to 92.62% after modeling the *UCamI* dataset into ontology-based complex activity structures and by performing the semantic imputation as shown in Figure 5.2b. The plot for these illustrates that the activity *Breakfast* (Act05) having the lowest recognition precision of 81.54% was most often classified as the activity *Prepare breakfast* (Act02). The activities *Play a videogame* (Act11) and *Visit in the SmartLab* (Act14) were recognized with 100% accuracy, which were having lower accuracies with the non-imputed data. Similarly, the activity *Relax on the sofa* (Act12) was also recognized with the highest precision rate of 98.44% as shown in Table 5.2. This suggests that semantic data imputation provided positive data values, which resulted in the increase of classification accuracies for individual activities.



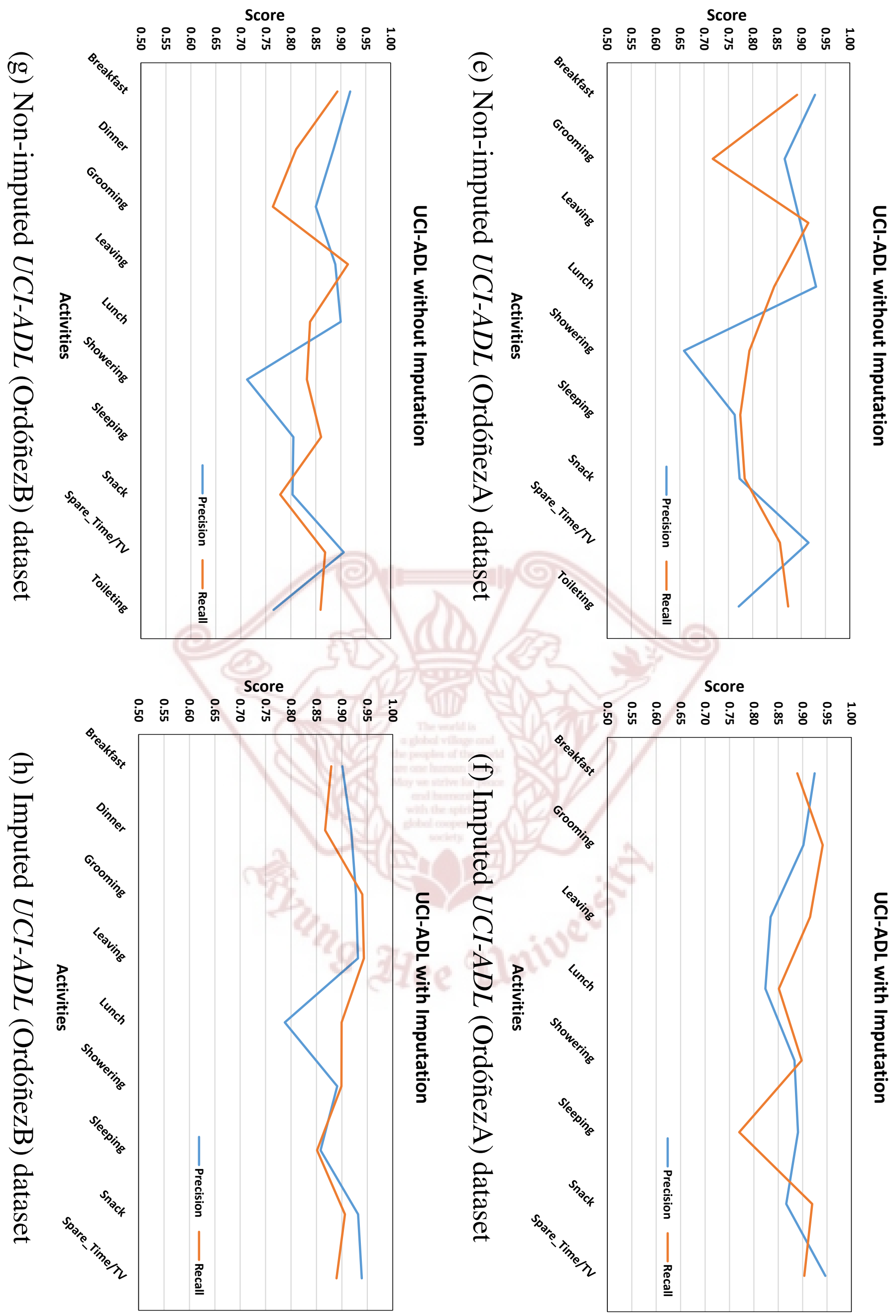
(a) Non-imputed *UCamI* dataset

(b) Imputed *UCamI* dataset

(c) Non-imputed *Opportunity* dataset

(d) Imputed *Opportunity* dataset

Figure 5.2: Classification performance of SemImput framework (UcamI & Opporutunity): Precision & Recall.



(e) Non-imputed *UCI-ADL* (OrdóñezA) dataset

(f) Imputed *UCI-ADL* (OrdóñezA) dataset

(g) Non-imputed *UCI-ADL* (OrdóñezB) dataset

(h) Imputed *UCI-ADL* (OrdóñezB) dataset

Figure 5.3: Classification performance of SemImput framework (UCI-ADL): Precision & Recall.

Table 5.1: Confusion matrix for per-class HAR using non-imputed UCamI dataset

(**Non-imputed**)

Ground Truth Activities

Predicted Activities

	Act_1	Act_2	Act_3	Act_4	Act_5	Act_6	Act_7	Act_8	Act_9	Act_{10}	Act_{11}	Act_{12}	Act_{13}	Act_{14}	Act_{15}	Act_{16}	Act_{17}	Act_{18}	Act_{19}	Act_{20}	Act_{21}	Act_{22}	Act_{23}	Act_{24}
Act_1	62.07	0	0.69	4.48	0	0	5.52	0	0	0.69	0	0	0	0	13.79	0	2.07	0	1.38	2.41	0	6.90	0	0
Act_2	0	87.55	0	0	4.06	0	0	0	0	0	0	0	0.14	0	0	0.14	0.70	0.14	0	2.10	0.42	4.48	0	0.28
Act_3	0.67	0	82.23	0	0	3.04	0	3.88	1.07	1.69	0	0.67	0	0	2.81	0	0.73	0.90	0.39	1.80	0	0.11	0	0
Act_4	5.70	0	0	70.50	0	0	7.43	0	0	3.76	0.61	0	0	0	6.71	0	2.75	0.51	0.61	0	0	1.42	0	0
Act_5	0	10.25	0	0	52.14	0	0	0	0	0	0	0	0	0	0	2.29	8.66	3.08	0	1.00	9.05	2.29	0	11.24
Act_6	0	0	9.43	0	0	63.85	0	1.38	9.92	1.28	0	5.89	0	0	2.95	0	2.46	0.49	0.59	1.38	0	0.39	0	0
Act_7	5.89	0	0	13.72	0	0	62.89	0	0	2.12	4.24	0	0	0	4.51	0	2.12	0	0.18	0	0	2.95	1.38	0
Act_8	1.54	0	11.28	0	0	4.62	0	57.95	4.62	0	0	0	0	0	3.59	0	1.03	3.08	0	7.69	0	4.62	0	0
Act_9	0.31	0	7.28	0	0	2.29	0	1.87	74.01	1.77	0	2.81	0.94	0.31	2.91	0	3.01	1.35	0.10	0.42	0	0.62	0	0
Act_{10}	0	0	2.23	0	0	0	0	0	0	84.36	0	0	0	2.23	11.17	0	0	0	0	0	0	0	0	0
Act_{11}	0	0	0	2.33	0	0	10.08	0	0	0.78	66.67	0	0	0	6.20	0	6.20	0.78	1.55	0	0	4.65	0.78	0
Act_{12}	0.34	0	10.14	0	0	0.34	0	0	10.14	1.69	0	71.62	0	0	3.72	0	1.01	1.01	0	0	0	0	0	0
Act_{13}	0	9.30	0	0	0	1.16	0	0	1.16	0	0	0	75.58	1.16	8.14	0	1.16	0	0	0	1.16	1.16	0	0
Act_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Act_{15}	2.33	0	2.20	3.76	0	2.33	2.20	2.20	1.55	3.63	0.26	0.52	5.18	0.78	67.10	0	1.94	0.39	0.39	0.39	0	2.72	0.13	0
Act_{16}	0	0.76	0	0	0	0	0	0	0	0	0	0	0	0	0	82.58	9.85	1.52	0	0	0	2.27	0	3.03
Act_{17}	0.51	0.68	0.51	0.51	1.70	1.10	0.59	1.44	2.55	0.34	1.02	0.17	0	0	2.97	8.50	67.20	3.74	0.08	0.17	0.34	4.93	0.34	0.59
Act_{18}	0	0	2.49	0	0.50	1.49	0	1.99	0	2.99	0	0	0	0	0.50	1.49	11.94	67.66	1.00	5.47	0	1.99	0	0.50
Act_{19}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Act_{20}	1.25	1.25	6.25	0	2.50	1.25	0	11.25	0	0	0	0	0	0	2.50	0	3.75	0	2.50	66.25	0	1.25	0	0
Act_{21}	0	8.49	0	0	9.27	0	0	0	0	0	0	0	0	0	0	1.93	1.54	0.39	0	0	74.90	1.16	0	2.32
Act_{22}	0.27	2.01	0	0	1.34	0	0.40	0	0	0.13	0.13	0	0.40	0	0.81	0.40	4.83	0.81	0	0.13	0.27	84.30	2.01	1.74
Act_{23}	0	0	0	0	0	0	2.82	0	0	0	0	0	0	0	1.41	0	2.82	1.41	0	0	0	9.86	81.69	0
Act_{24}	0	2.75	0	0	12.84	0	0	0	0	0	0	0	0	0	0	4.13	3.21	1.38	0	0	1.83	6.42	0	67.43

Table 5.2: Confusion matrix for per-class HAR using imputed UCamI dataset

	(**Non-imputed**) Ground Truth Activities																							
Predicted Activities	Act_1	Act_2	Act_3	Act_4	Act_5	Act_6	Act_7	Act_8	Act_9	Act_{10}	Act_{11}	Act_{12}	Act_{13}	Act_{14}	Act_{15}	Act_{16}	Act_{17}	Act_{18}	Act_{19}	Act_{20}	Act_{21}	Act_{22}	Act_{23}	Act_{24}
Act_1	62.07	0	0.69	4.48	0	0	5.52	0	0	0.69	0	0	0	0	13.79	0	2.07	0	1.38	2.41	0	6.90	0	0
Act_2	0	87.55	0	0	4.06	0	0	0	0	0	0	0	0.14	0	0	0.14	0.70	0.14	0	2.10	0.42	4.48	0	0.28
Act_3	0.67	0	82.23	0	0	3.04	0	3.88	1.07	1.69	0	0.67	0	0	2.81	0	0.73	0.90	0.39	1.80	0	0.11	0	0
Act_4	5.70	0	0	70.50	0	0	7.43	0	0	3.76	0.61	0	0	0	6.71	0	2.75	0.51	0.61	0	0	1.42	0	0
Act_5	0	10.25	0	0	52.14	0	0	0	0	0	0	0	0	0	0	2.29	8.66	3.08	0	1.00	9.05	2.29	0	11.24
Act_6	0	0	9.43	0	0	63.85	0	1.38	9.92	1.28	0	5.89	0	0	2.95	0	2.46	0.49	0.59	1.38	0	0.39	0	0
Act_7	5.89	0	0	13.72	0	0	62.89	0	0	2.12	4.24	0	0	0	4.51	0	2.12	0	0.18	0	0	2.95	1.38	0
Act_8	1.54	0	11.28	0	0	4.62	0	57.95	4.62	0	0	0	0	0	3.59	0	1.03	3.08	0	7.69	0	4.62	0	0
Act_9	0.31	0	7.28	0	0	2.29	0	1.87	74.01	1.77	0	2.81	0.94	0.31	2.91	0	3.01	1.35	0.10	0.42	0	0.62	0	0
Act_{10}	0	0	2.23	0	0	0	0	0	0	84.36	0	0	0	2.23	11.17	0	0	0	0	0	0	0	0	0
Act_{11}	0	0	0	2.33	0	0	10.08	0	0	0.78	66.67	0	0	0	6.20	0	6.20	0.78	1.55	0	0	4.65	0.78	0
Act_{12}	0.34	0	10.14	0	0	0.34	0	0	10.14	1.69	0	71.62	0	0	3.72	0	1.01	1.01	0	0	0	0	0	0
Act_{13}	0	9.30	0	0	0	1.16	0	0	1.16	0	0	0	75.58	1.16	8.14	0	1.16	0	0	0	1.16	1.16	0	0
Act_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Act_{15}	2.33	0	2.20	3.76	0	2.33	2.20	2.20	1.55	3.63	0.26	0.52	5.18	0.78	67.10	0	1.94	0.39	0.39	0.39	0	2.72	0.13	0
Act_{16}	0	0.76	0	0	0	0	0	0	0	0	0	0	0	0	0	82.58	9.85	1.52	0	0	0	2.27	0	3.03
Act_{17}	0.51	0.68	0.51	0.51	1.70	1.10	0.59	1.44	2.55	0.34	1.02	0.17	0	0	2.97	8.50	67.20	3.74	0.08	0.17	0.34	4.93	0.34	0.59
Act_{18}	0	0	2.49	0	0.50	1.49	0	1.99	0	2.99	0	0	0	0	0.50	1.49	11.94	67.66	1.00	5.47	0	1.99	0	0.50
Act_{19}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Act_{20}	1.25	1.25	6.25	0	2.50	1.25	0	11.25	0	0	0	0	0	0	2.50	0	3.75	0	2.50	66.25	0	1.25	0	0
Act_{21}	0	8.49	0	0	9.27	0	0	0	0	0	0	0	0	0	0	1.93	1.54	0.39	0	0	74.90	1.16	0	2.32
Act_{22}	0.27	2.01	0	0	1.34	0	0.40	0	0	0.13	0.13	0	0.40	0	0.81	0.40	4.83	0.81	0	0.13	0.27	84.30	2.01	1.74
Act_{23}	0	0	0	0	0	0	2.82	0	0	0	0	0	0	0	1.41	0	2.82	1.41	0	0	0	9.86	81.69	0
Act_{24}	0	2.75	0	0	12.84	0	0	0	0	0	0	0	0	0	0	4.13	3.21	1.38	0	0	1.83	6.42	0	67.43

The *Opportunity* dataset represents 17 ADLs and is of complex nature by having missing samples labeled as *Null* due to sensor disconnections. Figure 5.2c–d shows the per class precision and recall for recognized ADLs with the *Opportunity* dataset. The presented framework evaluates the *Opportunity* dataset without the 'Null' class by obtaining an overall accuracy of 86.57%, and an increased accuracy with the imputed dataset by 91.71%. The comparisons for both confusion matrices are shown in Table 5.3 and Table 5.4.

Table 5.3: Confusion matrix for per-class HAR using non-imputed Opportunity dataset

	(**Imputed**) Ground Truth Activities																
Predicted Activities	*OpenDoor1*	*OpenDoor2*	*CloseDoor1*	*CloseDoor2*	*OpenFridge*	*CloseFridge*	*OpenDishwasher*	*CloseDishwasher*	*OpenDrawer1*	*CloseDrawer1*	*OpenDrawer2*	*CloseDrawer2*	*OpenDrawer3*	*CloseDrawer3*	*CleanTable*	*DrinkfromCup*	*ToggleSwitch*
OpenDoor1	90.00	0	10.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OpenDoor2	0	92.26	0	1.19	0	0	0	0	0.60	1.79	0	1.19	2.98	0	0	0	0
CloseDoor1	1.16	0	90.70	0	1.16	1.16	5.81	0	0	0	0	0	0	0	0	0	0
CloseDoor2	0	3.62	0	94.20	0	0	0	0.72	0.72	0.72	0	0	0	0	0	0	0
OpenFridge	0.52	0	0	0	93.04	5.67	0.52	0	0	0	0	0	0	0	0	0	0.26
CloseFridge	0	0.79	0	0	1.98	96.84	0	0.40	0	0	0	0	0	0	0	0	0
OpenDishwasher	0.56	0	0	0	3.35	0	93.30	2.23	0.56	0	0	0	0	0	0	0	0
CloseDishwasher	0	0	0	0	1.55	0	7.75	89.92	0	0	0	0.78	0	0	0	0	0
OpenDrawer1	0	1.75	0	0	0	0	5.26	1.75	66.67	10.53	3.51	1.75	5.26	1.75	0	0	1.75
CloseDrawer1	1.06	1.06	1.06	0	1.06	0	5.32	0	1.06	82.98	0	0	0	1.06	3.19	2.13	0
OpenDrawer2	0	1.41	0	0	5.63	0	0	1.41	8.45	2.82	66.20	8.45	1.41	0	0	2.82	1.41
CloseDrawer2	0	4.76	0	0	0	0	0	4.76	2.38	11.90	9.52	59.52	0	7.14	0	0	0
OpenDrawer3	0	2.94	0	0	0	0	0	0	0	0	7.84	0	86.27	2.94	0	0	0
CloseDrawer3	0	0	0	0	0	0	0	0	0	0	2.08	9.38	5.21	83.33	0	0	0
CleanTable	0	0	0	0	0	0	0	0	0	0	2.22	1.67	0	0	95.00	1.11	0
DrinkfromCup	0.17	0	0	0	0	0	0.17	0.17	0	0.34	0.51	0	0	0.51	0.51	97.62	0
ToggleSwitch	0	0	0	0	0	0	0	0	1.00	1.00	0	0	1.49	1.49	0	0	95.02

Table 5.4: Confusion matrix for per-class HAR using imputed Opportunity dataset

Predicted Activities

Ground Truth Activities (Non-imputed)	OpenDoor1	OpenDoor2	CloseDoor1	CloseDoor2	OpenFridge	CloseFridge	OpenDishwasher	CloseDishwasher	OpenDrawer1	CloseDrawer1	OpenDrawer2	CloseDrawer2	OpenDrawer3	CloseDrawer3	CleanTable	DrinkfromCup	ToggleSwitch
OpenDoor1	82.65	0	17.05	0	0.26	0	0	0	0	0	0	0	0	0	0	0.18	0
OpenDoor2	0	91.98	0	1.57	0.26	0	0	0	0	1.35	0	0	0	0	0	0.18	0
CloseDoor1	16.33	0.62	82.95	0.79	0	0	0	0	0	0	0	0	0	0	0	0.37	0.58
CloseDoor2	0	7.41	0	97.64	0	0	0	0	0	0	0	0	0	0	0	0.18	0
OpenFridge	0	0	0	0	87.47	3.65	2.40	2.92	0	1.35	1.49	2.13	0	0	1.18	0	0.58
CloseFridge	0	0	0	0	7.42	94.16	0	0	0	0	0	0	0	0	0.59	0	0
OpenDishwasher	0	0	0	0	2.81	0	90.42	10.95	1.49	0	1.49	0	4.42	0	1.76	4.40	0
CloseDishwasher	0	0	0	0	0.51	0.36	2.99	78.10	2.99	0	0	8.51	0	0.99	0	0.18	0
OpenDrawer1	0	0	0	0	0	0.36	0.60	0	53.73	6.76	19.40	4.26	1.77	0	0	0	1.75
CloseDrawer1	0	0	0	0	0.26	0.73	0	0	25.37	89.19	11.94	21.28	0	0	0	0	1.75
OpenDrawer2	0	0	0	0	0	0	0	0	0	1.35	52.24	6.38	6.19	0	0	0	0
CloseDrawer2	0	0	0	0	0	0	0	1.46	1.49	0	4.48	55.32	3.54	26.73	0	0	0
OpenDrawer3	0	0	0	0	0	0	0	0	0	0	1.49	2.13	76.99	6.93	0	0	0
CloseDrawer3	0	0	0	0	0	0	1.20	4.38	2.99	0	0	0	6.19	65.35	0	0	0
CleanTable	0	0	0	0	0.26	0	0	0	0	0	0	0	0	0	86.47	0.18	0.58
DrinkfromCup	0	0	0	0	0.26	0.36	1.20	0.73	0	0	0	0	0	0	10.00	94.32	0.58
ToggleSwitch	1.02	0	0	0	0.51	0.36	1.20	1.46	11.94	0	7.46	0	0.88	0	0	0	94.15

As shown in Figure 5.3e–f for the *UCI-ADL* Ordóñez-A raw dataset, an overall classification result with 82.27% accuracy was obtained. It included activities like *Grooming*, *Spare_Time/TV*, and *Toileting* having the most number of instances and the activity *Lunch* with minimum number of instances. However, the classification results as mentioned in Table 5.5 show that the activities *Leaving* and *Breakfast* have the highest recognition accuracy as compared to the activity *Grooming* with the lower classification accuracy. In order to verify the proposed *SemImput* framework, it was also tested on the semantically imputed *UCI-ADL* Ordóñez-A dataset. This resulted in an increased recognition accuracy for activities such as *Breakfast*, *Lunch*, and *Leaving* significantly as shown in Figure 5.3f. It was due to the introduction of the semantic structure understanding of events with respect to morning, afternoon, and generalization of semantic rules for such activities for imputing missing values. The improvement in statistical quality through imputation raised the recognition accuracy significantly up to 89.20%. Similarly, an increased performance is also observed for the *UCI-ADL* Ordóñez-B dataset for the overall activities with imputed data, especially for the *Dinner* and *Showering* as shown in Table 5.6. The global accuracy for *UCI-ADL* Ordóñez-B dataset was improved from 84.0% to 90.34%, which also proves the significance of proposed framework as shown in Table 5.7.

Table 5.5: Confusion matrix for per-class HAR using non-imputed & imputed *UCI-ADL* (OrdóñezA) dataset

(**Non-imputed**)

Ground Truth Activities

		Breakfast	*Grooming*	*Leaving*	*Lunch*	*Showering*	*Sleeping*	*Snack*	*Spare_Time/TV*	*Toileting*
Predicted Activities	*Breakfast*	89.12	0.20	2.20	0.74	3.21	3.92	0	0.51	0.10
	Grooming	2.46	71.00	9.21	2.30	5.26	6.67	2.01	0.62	0.47
	Leaving	0.23	0.02	91.12	0.23	0.32	2.23	2.73	3.02	0.10
	Lunch	3.20	0.09	0.10	84.41	3.62	4.61	3.26	0.12	0.59
	Showering	0.01	5.30	0	0	79.12	4.33	8.64	0.32	2.28
	Sleeping	0	4.49	0.01	0	5.21	77.15	6.23	0.20	6.71
	Snack	0.12	0.02	0.04	3.32	12.54	0.02	77.50	3.20	3.24
	Spare_Time/TV	0.05	0.10	0	0.02	4.32	0.88	0	85.10	9.54
	Toileting	1.20	1.05	0.18	0.12	8.12	1.20	0	0	88.13

(**Imputed**)

Ground Truth Activities

		Breakfast	*Grooming*	*Leaving*	*Lunch*	*Showering*	*Sleeping*	*Snack*	*Spare_Time/TV*	*Toileting*
Predicted Activities	*Breakfast*	96.51	0.21	0.35	1.21	0.32	0.01	0.04	0.02	1.33
	Grooming	0.12	88.01	7.20	3.39	0.83	0.04	0.18	0	0.23
	Leaving	0.14	0.21	94.02	0	0.50	1.79	3.10	0.13	0.11
	Lunch	1.21	4.45	1.65	91.12	0.75	0.41	0.03	0.38	0
	Showering	0	0.62	0.75	2.10	85.23	2.20	6.11	2.98	0.01
	Sleeping	0.87	0.55	0.02	5.45	4.39	88.69	0.01	0.02	0
	Snack	0.01	0.74	0	5.64	2.21	6.88	77.02	7.01	0.49
	Spare_Time/TV	0	0.43	0	1.20	1.35	0.35	0.40	92.45	3.82
	Toileting	1.01	0	0	0	8.60	0	0	1.32	89.07

Table 5.6: Confusion matrix for per-class HAR using non-imputed & imputed *UCI-ADL* (OrdóñezB) dataset

(**Non-imputed**)

Ground Truth Activities

Predicted Activities	*Breakfast*	*Dinner*	*Grooming*	*Leaving*	*Lunch*	*Showering*	*Sleeping*	*Snack*	*Spare$_T$ime/TV*	*Toileting*
Breakfast	88.95	1.65	0	0.01	3.45	0.06	0.08	4.21	1.36	0.23
Dinner	0.64	81.06	1.23	0.98	4.55	0.16	1.88	7.26	2.24	0
Grooming	0.35	0.12	76.43	0.21	0	1.18	16.23	0.45	4.61	0.42
Leaving	0	0.02	0.29	91.49	0.36	0.12	0	0.30	3.10	4.32
Lunch	2.01	1.71	2.36	0.92	83.90	0.26	0.08	4.68	3.45	0.63
Showering	0.83	1.65	1.65	3.70	0	83.17	0.61	0.84	0.12	7.43
Sleeping	2.34	4.65	3.87	4.62	0.15	2.64	81.73	0	0	0
Snack	1.60	0.65	0.23	0	0.45	0	0	77.92	5.75	13.40
Spare$_T$ime/TV	0	0	2.89	0.54	0.03	6.25	0	0	90.29	0
Toileting	0.03	0	1.01	0.50	0.32	8.16	0.90	0	3.08	86.00

(**Imputed**)

Ground Truth Activities

Predicted Activities	*Breakfast*	*Dinner*	*Grooming*	*Leaving*	*Lunch*	*Showering*	*Sleeping*	*Snack*	*Spare$_T$ime/TV*	*Toileting*
Breakfast	97.10	0.01	0.20	0.12	0	0.65	0.32	1.41	0	0.19
Dinner	0.80	87.90	1.65	0.65	1.32	0.53	2.64	3.21	1.21	0.09
Grooming	0.32	2.40	86.69	0.07	2.45	2.89	0.05	4.08	0.10	0.95
Leaving	0.01	1.12	0.98	94.01	0.51	0.03	1.32	0.85	0.05	1.12
Lunch	0.45	0.06	0	1.32	94.30	0.12	1.24	1.19	1.32	0
Showering	0.78	1.36	1.40	0.45	0.09	94.47	0.08	0.01	0.24	1.12
Sleeping	0.63	1.89	0	1.61	0.65	3.99	89.87	0.31	0.08	0.97
Snack	1.10	2.45	1.77	1.74	1.11	0.99	2.10	85.10	2.49	1.15
Spare$_T$ime/TV	0	0.32	0.65	0.89	0.47	1.54	3.19	2.15	90.58	0.21
Toileting	0.01	0.09	1.05	0.45	0.25	7.14	0.06	0.89	1.11	88.95

Table 5.7: Recognition accuracy gain using the proposed *SemImput* framework. (Unit:%)

Method	**Datasets**	**Number of Activities**	**(Mean Recognition Accuracy) Non-Imputed**	**Imputed**	**Standard Deviation**
Proposed *SemImput*	*Opportunity* [26]	17	86.57	91.71	±2.57
	UCI-ADL OrdóñezA [211]	9	82.27	89.20	± 3.47
	UCI-ADL OrdóñezB [211]	10	84.0	**90.34**	± 3.17
	UCamI [1]	24	71.03	**92.62**	±10.80

As shown in Table 5.8, the proposed *SemImput* framework along with *SemDeep-ANN* model not only improved the recognition rate for individual activities within the datasets but also improved the global accuracy over each dataset. We also compared the activity classification performance of our framework with a different state-of-the-art methods. The presented results show the potential of *SemImput* framework with significant accuracy gain. Although for the *UCI-ADL* Ordóñez-A and *Opportunity* datasets, our methodology was worse, it still achieved significant recognition performance score of 89.20% and 91.71%, respectively. These findings show that combining the ADLs classification with semantic imputation can lead to comparatively better HAR performance.

Table 5.8: Comparison results of the proposed *SemImput* framework with state-of-the-art HAR methods.

State-of-the-Art methods	Datasets	Number of Activities	Mean Recognition Accuracy(%)	SemImput Gain
Razzaq et al. [115]	*UCamI* [1]	24	47.01	**+45.61**
Salomón et al. [212]	*UCamI* [1]	24	90.65	**+1.97**
Li et al. [192]	*Opportunity* [26]	17	**92.21**	-0.50
Salguero et al. [77, 152]	*UCI-ADL* OrdóñezA [211]	9	**95.78**	-6.58
	UCI-ADL OrdóñezB [211]	10	86.51	**+3.83**

5.3 Results and Discussion for Vision-based Multioccupant State Imputation

In literature there exist several performance measures to deal with single-target and multi-target tracking, however, none of them proved to be a defacto standard. In our experiments, we adopted some of the effective multi-occupant detection and tracking evaluation strategies to: a) detect and track the multi-occupants and b) classify multi-occupant activities in TVS_F_{seq}. For this, we investigated frame properties in the sequences to identify the influence of different parameters such as variable thresholds and overlap measures on the overall performance. Moreover, conformity of evaluation measures to any other application and sequence have been proven by the *uMoDT* framework on VOT-TIR2016 sequences other than TVS_F_{seq}.

5.3.1 Multi-occupant detection and tracking evaluation

Objectively quantitative assessment of multi-occupant detection and tracking is not a straight forward task. Most of the evaluation techniques require a ground-truth $\mathrm{G_i}$, which serves as a reference to measure the performance quantitatively. We adopted such evaluation methods, which rely on frame based spatial overlap between $\mathrm{G_i}$ and bounding rectangles $\mathrm{BR_n}$ [213].

Evaluation metrics

The object detection in benchmark sequences and multi-occupant detection in TVS_F_{seq} uses standard *Pascal*, *Intersection over Union* (IoU) criterion, a natural bounding box evaluation measure for comparing spatial overlap and localization accuracy [202]:

$$IoU(BR_\mathrm{n}, G_\mathrm{i}) = \frac{BR_\mathrm{n} \cap G_\mathrm{i}}{BR_\mathrm{n} \cup G_\mathrm{i}} \tag{5.6}$$

Performance evaluation and analysis

We take the advantage of the *Counting* algorithm to estimate number of occupants against $\mathrm{G_i}$ frame-wise in each of the sequence [214]. In our experiments, we considered count detection as true positive (TP) for *IoU* greater than 0.5 otherwise as false positive (FP). For *IoU*<0.5, however, we also considered rotated $\mathrm{BR_n}$ locations for each object obtained from KF in the frame to see if updated object state has any spatial overlap relation with ground-truth. Fig. 4.5(a-f) present results for $\mathrm{G_i}$, detected, and KF predicted $\mathrm{BR_n}$ frame-wise in each sequence. The best counting success rate is achieved by using improved frame pre-processing algorithms *TVS-MoFV* (5) and *TVS-MoDT* (6) for the *Soccer* sequence with around 94.76% whereas TVS_F_{seq} achieved a counting accuracy of 88.46%. Results by the counting algorithm using KF predicted occupants exhibit an excellent performance for each sequence where occupants are well separated in the frames as compared to the sequences in which they are occluded by each other.

Table 5.9: Evaluation comparison of the uMoDT framework for benchmark sequences and TVS_F_{seq}

	Name	**FP↓**	**FN↓**	**MOTA↑**	**IDS↓**	**Precision↑**	**Recall↑**	**MSE↓**
Dataset	ETHZ-CLA	441	414	5.58	210	0.61	**0.44**	1.04
	Soccer	311	1540	**74.42**	246	**0.94**	0.39	5.19
	Crouching	163	428	57.17	243	0.80	0.29	1.08
	Depthwise	456	408	53.03	180	0.72	0.38	0.96
	Crowd	110	211	57.40	110	0.81	0.41	12.27
	TVS_F_{seq}	52	469	**64.26**	72	**0.87**	0.42	**0.84**

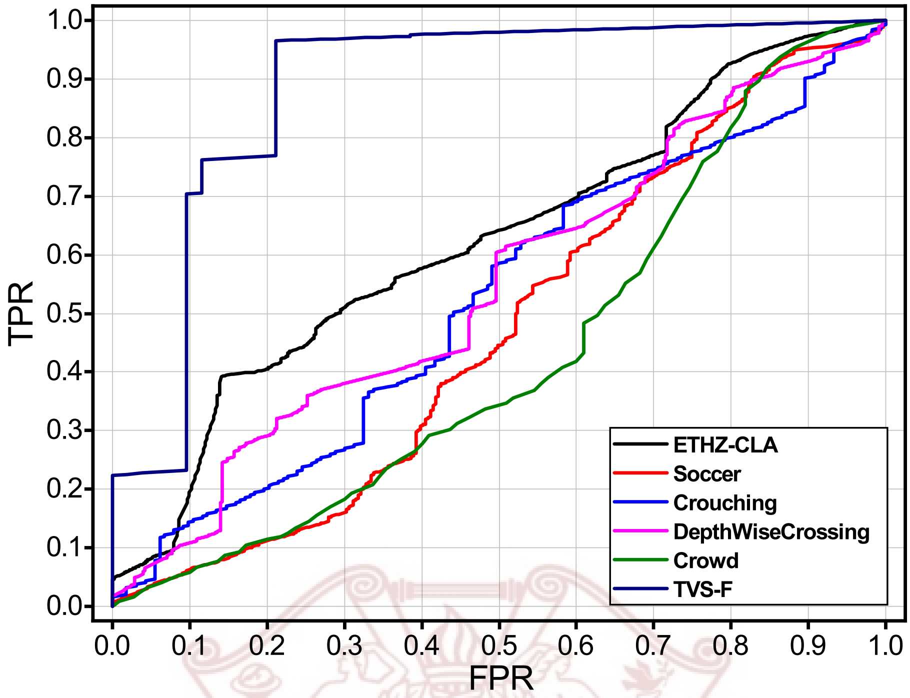


Figure 5.4: ROC curves for benchmark sequences and TVS_F_{seq}

To evaluate multi-occupant detection and tracking performance, it is not suitable to use only one single metrics, therefore, we extend the frame-wise *IoU* overlap measure for performance evaluation by estimating *Multiple Object Tracking Accuracy* (MOTA), an accepted evaluation measure [220]. MOTA measure also takes into account the impact of erroneous responses such as: false negatives (FN_t), false positives (FP_t), number of identity switches $\mathrm{IDS_t}$, and $\mathrm{G_t}$ at time *t*. By combining these sources of error, MOTA is defined as:

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDS_t)}{\sum_t G_t} \tag{5.7}$$

We report quantitative evaluations and comparative analysis through the experiments over a set of test sequences for frame-based detection and tracking in Tables 5.9 and 5.10 respectively. It is evident that the *uMoDT* framework demonstrated better performance in terms of MOTA for benchmark sequences and TVS_F_{seq}. It outperformed other techniques on all sequences especially for *Soccer* sequence and TVS_F_{seq} with MOTA scores of *74.42%* and *64.26%* respectively.

Table 5.10: Evaluation comparison for the uMoDT framework against other techniques

	Name	**FP↓**	**FN↓**	**MOTA↑**	**IDS↓**
Method	Bochinski *et al.* [215]	5702	70278	57.1	2167
	Wan *et al.* [216]	10604	56182	62.6	1389
	Bewley *et al.* [217]	7318	32615	33.4	1001
	Murray *et al.* [218]	3130	76202	27.4	786
	Chen *et al.* [219]	9253	85431	47.6	792
	Gade *et al.* [214]	9.8%	18.8%	**70.36**	219
	uMoDT (TVS_F_{seq})	52	469	**64.26**	72

Additionally, the Mean Squared Error (MSE) between the localization of predicted $\mathrm{BR_n}$ and $\mathrm{G_i}$ was also computed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(BR_{\mathrm{n}} - G_{\mathrm{i}})^2 \tag{5.8}$$

The error rates showed lowest MSE value of 0.84, which was achieved for TVS_F_{seq} and a highest MSE value of 12.27 for *Crowd* sequence. The tabulated results, however, showed a higher number of $\mathrm{IDS_t}$, an increased MSE, and a decreased MOTA, which appeared to be from occlusions and deforming blobs.

The performance of *uMoDT* is also compared by constructing ROC curves for accumulated true detection rates and false positive rates using $\mathrm{G_i}$ and predicted $\mathrm{BR_n}$ with IoU>0.5 as shown in Fig. 5.4. The ROC curve produced by TVS_F_{seq} has shown a larger area under the curve than other sequences. This suggests and validates the robustness of the proposed algorithm for occupant detection. TVS_F_{seq} has lessor FPR, which is due to minimal occlusion as compared to other sequences especially in *Crowd* sequence, which has maximum occlusion. Fig. 5.5 shows the resulting precision-recall curves based on overlap metric. Such a quantitative analysis proves as how successfully the $\mathrm{BR_n}$ are predicted for $\mathrm{G_i}$ in the benchmark sequences and TVS_F_{seq}. The *uMoDT* framework achieved a highest area under the curves with an average 97.16% precision rate for TVS_F_{seq} and the lowest one with around 72.04% for *ETHZ-CLA* sequence.

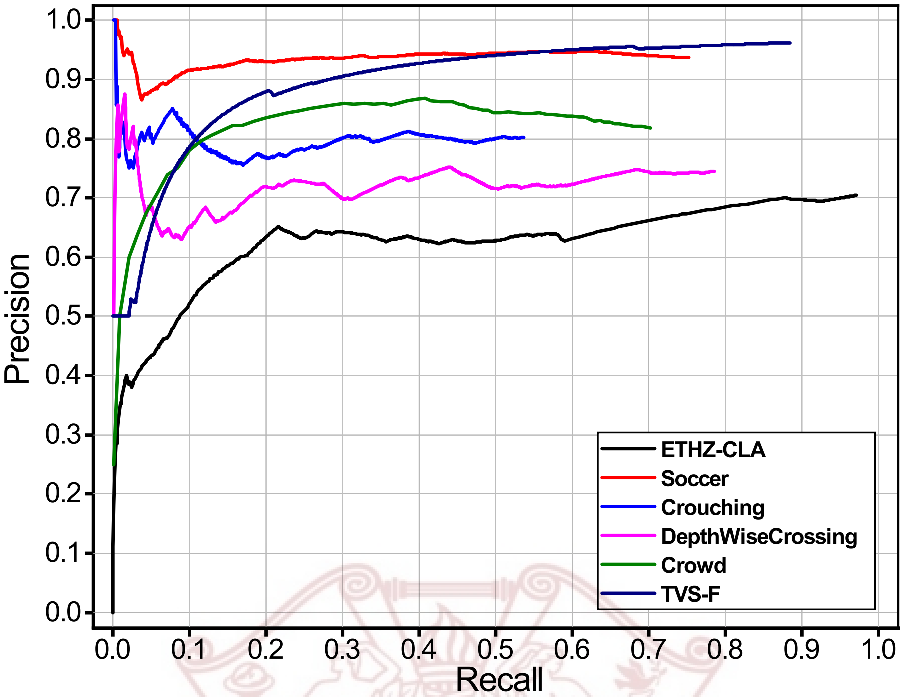


Figure 5.5: Precision-recall curves for benchmark sequences and TVS_F_{seq}

uMoDT robustness

To assess the ability of the *uMoDT* framework as how it deals with the tracking failure, we further quantify it for robustness measure correlated with accuracy. Robustness refers to the *uMoDT* failures whenever the overlap *IoU* measure becomes equal to zero. To measure the average overlap areas and complete failures, these measures are intuitively computed for benchmark sequences with *IoU* threshold value equal to zero. We also assumed each occupant in a frame as a separate entity, represented by an independent motion trajectory to evaluate tracking performance [221]. The resulting robustness, however, in some cases does not have an upper bound so it was interpreted as a reliability, defined by $e^{-S(F_0/N)}$ for visualizing purpose [222, 223]. Here N denotes number of frames for an individual sequence, S represents the number of frames since the last failure, and F_0 is a failure rate, which is set as *IoU* equal to zero. We executed the *uMoDT* framework separately for each sequence to record their average scores, failure rate and unsupervised re-initialization for multi-occupants.

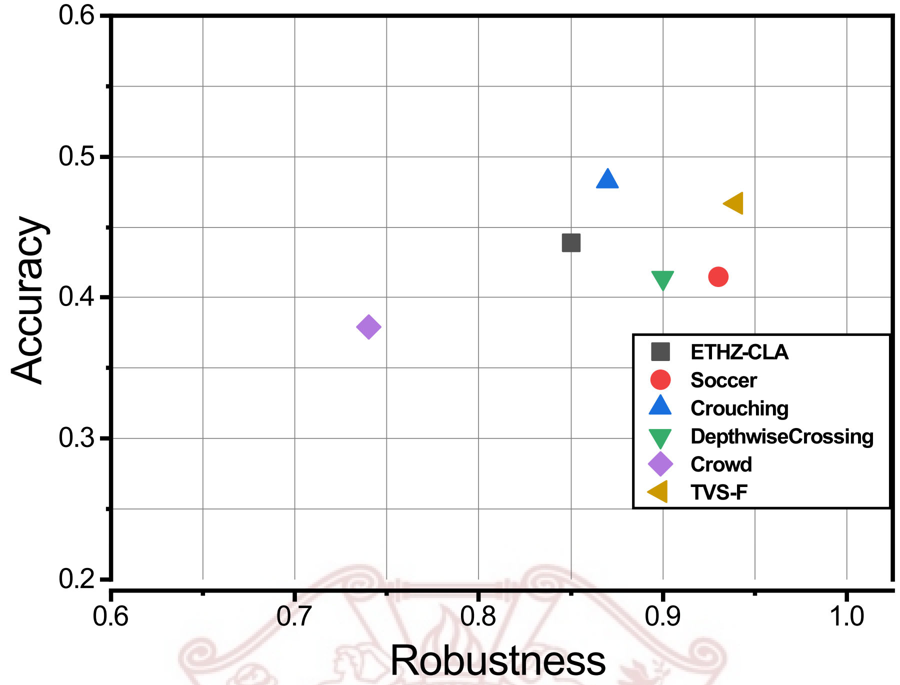


Figure 5.6: Accuracy-robustness plot for the *uMoDT* with benchmarks and TVS_F_{seq}

Fig. 5.6 demonstrates the effectiveness of the *uMoDT* framework, which proved to be most robust on *TVS-F* sequence (positioned most right) but it was surpassed by *Crouching* sequence, which appeared to be more accurate (positioned higher). The observed high robustness for TVS_F_{seq} is because of no occlusion, static distinguishable background and quality of multi-occupant estimates using KF. On the other hand, high average accuracy for *Crouching* sequence is observed, which is due to frequent re-initialization as occupant's appearance is challenging which matches with background. The *uMoDT* framework performed differently between the benchmark sequences depending on their frame properties, however, it achieved an overall best performance except for the *Crowd* sequence (positioned lowered). At a closer look, we see that in terms of accuracy it is challenging as occupants are not well distinguishable from background and also frequent *uMoDT* failures occur due to occlusions. It still, however, has achieved satisfactory robustness.

5.3.2 Multi-occupant activity recognition

In the following subsection, to show the generality of the TVS-AR method, we describe and evaluate the proposed CNN-based model using the TVS_F_{seq} for AR. We present the classification results to prove the performance and suitability of the presented approach using low-resolution TVS_F_{seq} in terms of accuracy [224]. We used frame-based approach for recognizing 16 different activities showing the efficacy of a model by demonstrating it for a high HAR accuracy score of approximately 90.99$\%$.

Activity recognition evaluation metrics

The performance metric that is most widely used to evaluate a classifier in the context of multiclass classification is overall accuracy [196]. The recognition accuracy is linear to the number of training frames. The training frames were used to fit in the parameters such as weights, validation set to fine tune the parameters and CNN architecture. The performance of the customized CNN was evaluated on validation split as a test data to validate the generalization and prediction power of the classifier. Additionally, the other most common performance evaluation metrics such as precision, recall, F-measure also provided an essential information required to assess the classification model [75].

Performance evaluation of activities

For each experiment, we followed the data splits and cross-validation evaluation technique for TVS_F_{seq}. We divided TVS_F_{seq} into three splits: training split TVS_F_{Train} to train CNN model, validation split to tune the hyper-parameters such as learning rate, epoch size on unseen data, and finally test split to evaluate the classification performance. An average accuracy of 97.34$\%$ was achieved with a learning rate of 0.01 for 28,485 TVS_F_{seq}. A drop in accuracy, however, was observed with a decrease in the learning rate. The test split contained 1,920 TVS-F for validating 16 activities as mentioned in the confusion matrix illustrated through Table 5.11. It is observed that the TVS-AR method accurately classified most of single-occupant and multi-occupant activities. Nevertheless, some confusion has been observed for multi-occupant activities such as *StandingWalking* (Act_{10}) and *StandingStretching* (Act_9) have been confused due to similar

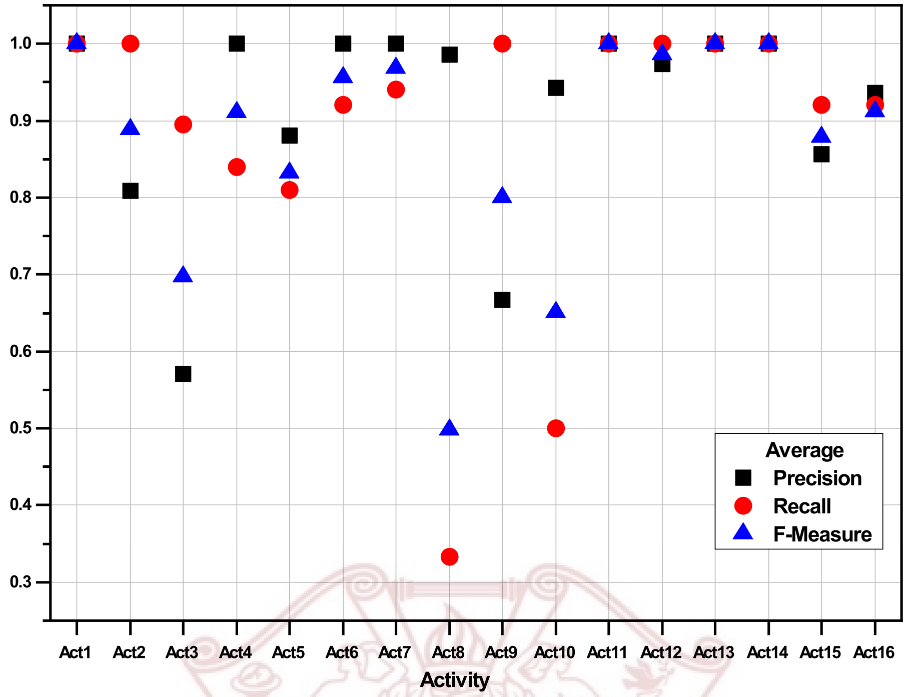


Figure 5.7: Classification accuracy using CNN for test TVS-F

motion patterns for *Walking* and *Stretching*. This is due to the activity *Stretching*, which involves extension of arms and returning to their original position, again sharing motion patterns to the activity *Standing* in a TVS_F_{seq}. Similarly, static multi-occupant activities *SittingSitting* (Act_3) and *StandingStanding* (Act_8) share similar occupant appearances in the TVS_F_{seq}. For these, the activities *Standing* and *Sitting* were confused due to similar heat maps in the frames. Furthermore, Fig. 5.7 shows the evaluation metrics in terms of Precision, Recall and F-Measure. By visualizing these, it can be concluded that multi-occupant activity i.e. (Act_8) with both occupants *Standing* and (Act_{10}) with one occupant *Standing* and other one *Walking* has shown the lowest performance for the test split of TVS_F_{seq}.

Table 5.11: Average accuracy confusion matrix for multi-occupant HAR

		Ground Truth Activities															
		Act_1	Act_2	Act_3	Act_4	Act_5	Act_6	Act_7	Act_8	Act_9	Act_{10}	Act_{11}	Act_{12}	Act_{13}	Act_{14}	Act_{15}	Act_{16}
Predicted Activities	Act_1	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Act_2	0	120	0	0	6	0	0	0	0	0	0	0	0	0	0	0
	Act_3	0	0	117	0	0	0	0	80	0	0	0	0	0	0	0	0
	Act_4	0	0	0	96	0	0	0	0	0	0	0	0	0	0	0	0
	Act_5	0	0	0	24	114	0	0	0	0	0	0	0	0	0	0	0
	Act_6	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0
	Act_7	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0
	Act_8	0	0	3	0	0	0	0	40	0	0	0	0	0	0	0	0
	Act_9	0	0	0	0	0	0	0	0	120	60	0	0	0	0	0	0
	Act_{10}	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0
	Act_{11}	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0
	Act_{12}	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0
	Act_{13}	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0
	Act_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0
	Act_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0
	Act_{16}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120

Chapter 6

Conclusion and Future Direction

This chapter finally draws the dissertation conclusions, provides evidence for future directions and also discloses potential applications where the proposed methodology can be applied for better outcomes.

6.1 Conclusion

This thesis proposed a novel *SemImput* framework to perform *Semantic Imputation* for missing data using public datasets for offline recognition of ADLs. It leverages the strengths of both structure-based and instance-based similarities while performing semantic data imputation. By using ontological model *SemImputOnt*, it uses SPARQL queries executed over the ABox data for semantic data expansion, conjunction separation, identification of missing attributes, and their instances leading towards semantic imputation. In order to further increase the quality of the data, we also utilized time-series longitudinal imputation. The obtained results and presented analysis suggest that gain in recognition accuracy varies with the nature and quality of dataset through the *SemImput*. We validated it, over *UCamI*, *Opportunity*, and UCI-ADL datasets. It achieves the highest accuracy of 92.62% for *UCamI* dataset using a SemDeep-ANN pre-trained model. A substantial, comprehensive, and comparative analysis with state-of-the-art methodologies for these three datasets were also performed and presented in this paper. Based on the empirical evaluation, it was shown that *DeepSem-ANN* consistently performed well on semantically imputed data by achieving an improved overall classification accuracy. Such a technique can be applied for HAR based systems, which generate data from obtrusive and unobtrusive sources in a smart environment. In this work, we proposed and demonstrated an *unobtrusive Multi-occupant Detection and Tracking* (*uMoDT*) framework for HAR based on low-resolution TVS. In this study, by using a binarization technique with Gaussian filter for smoothing, a morphological improvement with inversion and dilation process, an individual occupant in the form of the blob was detected over a sequence of frames. This blob was further tracked by using a KF with location improvement and evaluated with Intersection over Union (IoU). The above methods achieved detection and tracking accuracy of 88.46% for Thermal Vision Sensor frame sequence (TVS_F_{seq}). Additionally, a CNN-based multi-occupant HAR method was evaluated, achieving a validation accuracy of 97.34% and an accuracy of 90.99% for classification tasks. This experimentation demonstrates improvements in occupant detection and, activity association using TVS. The experimental evaluation using state-of-the-art benchmark datasets also revealed the robustness and effectiveness of the proposed framework. Further improvements may be achieved by introducing multiple TVS(s) for HAR. These settings may include movable TVS to recognize ADLs for more complex scenarios at different indoor locations.

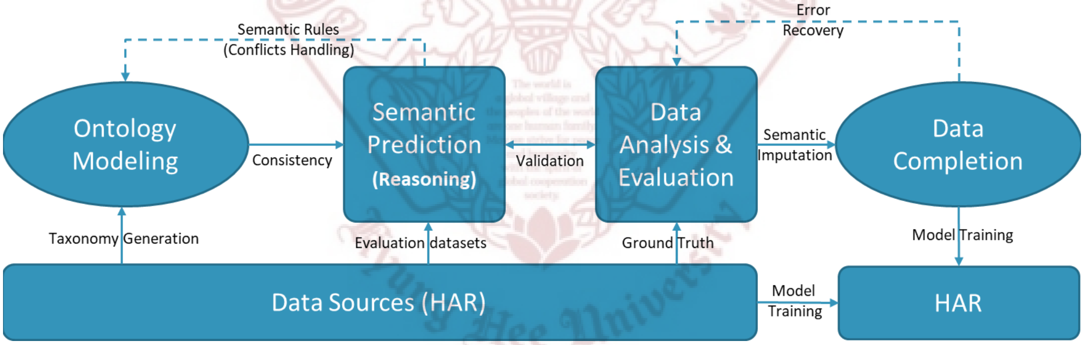


Figure 6.1: Semantic Data Imputation Evaluation & Prediction Conceptual model

6.2 Future Direction

The performed experimental study also shows that there is no universally best imputation method and the mean imputation is shown to be the least beneficial [146]. In the future, we plan to explore, execute, and enhance the *SemImput* framework for real-time HAR systems Fig. 6.1. The presented conceptual model is a complex framework in which every subcomponent interacts with others. The idea here is to use the Semantic Imputation methodology as a prediction, which is thoroughly analyzed and evaluated based on the ground-truth before the *Data Completion* task. Furthermore, we plan to extend our methodology for improving longitudinal imputation as some accuracy degradation is observed while recognizing HAR. We believe that our proposed conceptualization of methodology will help in increasing the quality of smart-home data by performing missing data imputation and will increase the recognition accuracy. On the negative side, the *SemImput* framework requires an ontology modeling effort for any activity inclusion or an introduction of a new dataset. For this, we plan to explore a scheme for unified activity modeling ontology for representing the same activities and investigate it further for HAR performance.

6.3 Potential Applications

The presented research can be directly applied to healthcare applications, entertainment and games, home and office automation, industrial applications, security and surveillance involving human movement. However, applying this technique to the aforementioned domains would require domain-specific ontologies to handle data preprocessing challenges.

There exist several knowledge-based medical systems under health discipline, which uses multiple imputation methods required to estimate the missing data are widely acknowledged. Similarly, data imputation methodologies are also supporting missing data identification in environmental domains. where multiple strategies are adopted to predict missing data for environmental sensors. In the research and real-world, imputation techniques are also applied over the survey data and industrial databases where data mining methods are used to identify and extract patterns over the big-data. These patterns are used to extract useful information using machine learning-based statistical algorithms to replace the missing values where necessary. As discussed previously, various applications opt for multiple imputations, which is believed to be more superior over single imputation methods where the missing data mechanism is MAR. However, in such a situation, a suitable method to perform multiple imputation has to be adopted to deal with the imputation battles. These challenges become even more complex where several ML-based application software is used to perform perfect prediction [225].

As far as vision-based activity recognition is concerned, it has been a research focus since long due to its unavoidable importance in the field of human-computer interaction, robot industry, user-interface/user-experience, and surveillance systems. In these application areas, research has widely deployed a variety of modalities, RGB single camera, infrared devices or thermal cameras to capture human contexts. Using these methodologies various application scenarios have been investigated and yet to be explored fully, which include single object tracking, group tracking, crow sensing or recognitions. Additionally, vision-based systems have also proved to be an important area for surveillance, suspicious activity monitoring or robot learning.

Bibliography

[1] "UCAmI Cup 2018," http://mamilab.esi.uclm.es/ucami2018/UCAmICup.html, (accessed on Feb, 2021). [Online]. Available: http://mamilab.esi.uclm.es/ucami2018/UCAmICup.html

[2] G. Mokhtari, N. Bashi, Q. Zhang, and G. Nourbakhsh, "Non-wearable human identification sensors for smart home environment: a review," *Sensor Review*, vol. 38, no. 3, pp. 391–404, 2018.

[3] A. Ghosh, D. Chakraborty, D. Prasad, M. Saha, and S. Saha, "Can we recognize multiple human group activities using ultrasonic sensors?" in *Communication Systems & Networks (COMSNETS), 2018 10th International Conference on*. IEEE, 2018, pp. 557–560.

[4] M. Safyan, Z. U. Qayyum, S. Sarwar, R. García-Castro, and M. Ahmed, "Ontology-driven semantic unified modelling for concurrent activity recognition (oscar)," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 2073–2104, 2019.

[5] C. Curley, R. M. Krause, R. Feiock, and C. V. Hawkins, "Dealing with missing data: A comparative exploration of approaches using the integrated city sustainability database," *Urban affairs review*, vol. 55, no. 2, pp. 591–615, 2019.

[6] A. Benmansour, A. Bouchachia, and M. Feham, "Multioccupant activity recognition in pervasive smart home environments," *ACM Computing Surveys (CSUR)*, vol. 48, no. 3, p. 34, 2016.

[7] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, “Sensor-based activity recognition,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 790–808, 2012.

[8] G. Singla, D. J. Cook, and M. Schmitter-Edgecombe, “Recognizing independent and joint activities among multiple residents in smart environments,” *Journal of ambient intelligence and humanized computing*, vol. 1, no. 1, pp. 57–63, 2010.

[9] R. Gade and T. B. Moeslund, “Constrained multi-target tracking for team sports activities,” *IPSJ Transactions on Computer Vision and Applications*, vol. 10, no. 1, p. 2, 2018.

[10] J. Synnott, J. Rafferty, and C. D. Nugent, “Detection of workplace sedentary behavior using thermal sensors,” in *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*. IEEE, 2016, pp. 5413–5416.

[11] M. Fiaz, A. Mahmood, and S. K. Jung, “Tracking noisy targets: A review of recent object tracking approaches,” *arXiv preprint arXiv:1802.03098*, 2018.

[12] S. N. Tran, Q. Zhang, and M. Karunanithi, “On multi-resident activity recognition in ambient smart-homes,” *arXiv preprint arXiv:1806.06611*, 2018.

[13] R. Gade, T. B. Moeslund, S. Z. Nielsen, H. Skov-Petersen, H. J. Andersen, K. Basselbjerg, H. T. Dam, O. B. Jensen, A. Jørgensen, H. Lahrmann *et al.*, “Thermal imaging systems for real-time applications in smart cities,” *International Journal of Computer Applications in Technology*, vol. 53, no. 4, pp. 291–308, 2016.

[14] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, “A survey of appearance models in visual object tracking,” *ACM transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 4, p. 58, 2013.

[15] J. Shen, Z. Liang, J. Liu, H. Sun, L. Shao, and D. Tao, “Multiobject tracking by submodular optimization,” *IEEE Transactions on Cybernetics*, 2018.

[16] S. García, J. Luengo, and F. Herrera, *Data preprocessing in data mining*. Springer, 2015.

[17] T.-C. Chiang, B. Bruno, R. Menicatti, C. T. Recchiuto, and A. Sgorbissa, “Culture as a sensor? a novel perspective on human activity recognition,” *International Journal of Social Robotics*, vol. 11, no. 5, pp. 797–814, 2019.

[18] V. Borges and W. Jeberson, “Survey of context information fusion for sensor networks based ubiquitous systems,” *arXiv preprint arXiv:1309.0598*, 2013.

[19] M. Wang, C. Perera, P. P. Jayaraman, M. Zhang, P. Strazdins, and R. Ranjan, “City data fusion: Sensor data fusion in the internet of things,” *arXiv preprint arXiv:1506.09118*, 2015.

[20] I. Ekerete, C. Nugent, O. M. Giggins, and J. McLaughlin, “Unobtrusive sensing solution for post-stroke rehabilitation,” in *Smart Assisted Living*. Springer, 2020, pp. 43–62.

[21] M. A. Razzaq and S. Lee, “Mmou-ar: Multimodal obtrusive and unobtrusive activity recognition through supervised ontology-based reasoning,” in *International Conference on Ubiquitous Information Management and Communication*. Springer, 2019, pp. 963–974.

[22] A. Scalmato, A. Sgorbissa, and R. Zaccaria, “Describing and recognizing patterns of events in smart environments with description logic,” *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1882–1897, 2013.

[23] K. Tschumitschew and F. Klawonn, “Effects of drift and noise on the optimal sliding window size for data stream regression models,” *Communications in Statistics-Theory and Methods*, vol. 46, no. 10, pp. 5109–5132, 2017.

[24] M. A. Razzaq, C. Villalonga, S. Lee, U. Akhtar, M. Ali, E.-S. Kim, A. M. Khattak, H. Seung, T. Hur, J. Bang *et al.*, “mlcaf: Multi-level cross-domain semantic context fusioning for behavior identification,” *Sensors*, vol. 17, no. 10, p. 2433, 2017.

[25] D. R. Faria, M. Vieira, C. Premebida, and U. Nunes, “Probabilistic human daily activity recognition towards robot-assisted living,” in *2015 24th IEEE international symposium on robot and human interactive communication (RO-MAN)*. IEEE, 2015, pp. 582–587.

[26] "Opportunity Dataset," http://www.opportunity-project.eu/challengeDownload.html, (accessed on Feb, 2021). [Online]. Available: http://www.opportunity-project.eu/challengeDownload.html

[27] "ADLs Recognition Using Binary Sensors Dataset," https://archive.ics.uci.edu/ml/datasets/Activities+of+Daily+Living+%28ADLs%29+Recognition+Using+Binary+Sensors, (accessed on Feb, 2021). [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Activities+of+Daily+Living+%28ADLs%29+Recognition+Using+Binary+Sensors

[28] C.-H. Liu, C.-F. Tsai, K.-L. Sue, and M.-W. Huang, "The feature selection effect on missing value imputation of medical datasets," *Applied Sciences*, vol. 10, no. 7, p. 2344, 2020.

[29] M. Templ, A. Kowarik, and P. Filzmoser, "Iterative stepwise regression imputation using standard and robust methods," *Computational Statistics & Data Analysis*, vol. 55, no. 10, pp. 2793–2806, 2011.

[30] Y. Liu, T. Dillon, W. Yu, W. Rahayu, and F. Mostafa, "Missing value imputation for industrial iot sensor data with large gaps," *IEEE Internet of Things Journal*, 2020.

[31] M. S. Osman, A. M. Abu-Mahfouz, and P. R. Page, "A survey on data imputation techniques: Water distribution system as a use case," *IEEE Access*, vol. 6, pp. 63 279–63 291, 2018.

[32] N. Kamkhad, K. Jampachaisri, P. Siriyasatien, and K. Kesorn, "Toward semantic data imputation for a dengue dataset," *Knowledge-Based Systems*, p. 105803, 2020.

[33] J. L. Schafer, *Analysis of incomplete multivariate data*. CRC press, 1997.

[34] V. Tresp, M. Bundschus, A. Rettinger, and Y. Huang, "Towards machine learning on the semantic web," in *Uncertainty reasoning for the Semantic Web I*. Springer, 2008, pp. 282–314.

[35] S. J. Hadeed, M. K. O'Rourke, J. L. Burgess, R. B. Harris, and R. A. Canales, "Imputation methods for addressing missing data in short-term monitoring of air pollutants," *Science of The Total Environment*, p. 139140, 2020.

[36] M. Kim, H. Kang, S. Kwon, Y. Lee, K. Kim, and C. S. Pyo, “Augmented ontology by handshaking with machine learning,” in *Advanced Communication Technology (ICACT), 2017 19th International Conference on*. IEEE, 2017, pp. 740–743.

[37] R. Nishihara, P. Moritz, S. Wang, A. Tumanov, W. Paul, J. Schleier-Smith, R. Liaw, M. I. Jordan, and I. Stoica, “Real-time machine learning: The missing pieces,” *arXiv preprint arXiv:1703.03924*, 2017.

[38] N. Moran, S. Nieland, B. Kleinschmit *et al.*, “Combining machine learning and ontological data handling for multi-source classification of nature conservation areas,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 54, pp. 124–133, 2017.

[39] A. Danylenko, C. Kessler, and W. Löwe, “Comparing machine learning approaches for context-aware composition,” in *International Conference on Software Composition*. Springer, 2011, pp. 18–33.

[40] M. Weeks, “Methods of imputation for missing data,” mimeo, Faculty of Economics and Politics, University of Cambridge, Tech. Rep., 2001.

[41] N. Fazakis, G. Kostopoulos, S. Kotsiantis, and I. Mporas, “Iterative robust semi-supervised missing data imputation,” *IEEE Access*, vol. 8, pp. 90 555–90 569, 2020.

[42] X.-H. Zhou, “Challenges and strategies in analysis of missing data,” *Biostatistics & Epidemiology*, vol. 4, no. 1, pp. 15–23, 2020.

[43] Z. Guo, Y. Wan, and H. Ye, “A data imputation method for multivariate time series based on generative adversarial network,” *Neurocomputing*, vol. 360, pp. 185–197, 2019.

[44] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*. John Wiley & Sons, 2019, vol. 793.

[45] K. J. Nishanth and V. Ravi, “Probabilistic neural network based categorical data imputation,” *Neurocomputing*, vol. 218, pp. 17–25, 2016.

[46] S. Laaksonen, “Regression-based nearest neighbour hot decking,” *Computational Statistics*, vol. 15, no. 1, pp. 65–71, 2000.

[47] J. Laurikkala, E. Kentala, M. Juhola, I. Pyykkö, and S. Lammi, "Usefulness of imputation for the analysis of incomplete otoneurologic data," *International Journal of Medical Informatics*, vol. 58, pp. 235–242, 2000.

[48] C. B. Do and S. Batzoglou, "What is the expectation maximization algorithm?" *Nature biotechnology*, vol. 26, no. 8, pp. 897–899, 2008.

[49] Q. Wang, J. Rao *et al.*, "Empirical likelihood-based inference under imputation for missing response data," *The Annals of Statistics*, vol. 30, no. 3, pp. 896–924, 2002.

[50] I. B. Aydilek and A. Arslan, "A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm," *Information Sciences*, vol. 233, pp. 25–35, 2013.

[51] M. Krishna and V. Ravi, "Particle swarm optimization and covariance matrix based data imputation," in *2013 IEEE International Conference on Computational Intelligence and Computing Research*. IEEE, 2013, pp. 1–6.

[52] L. Folguera, J. Zupan, D. Cicerone, and J. F. Magallanes, "Self-organizing maps for imputation of missing data in incomplete data matrices," *Chemometrics and Intelligent Laboratory Systems*, vol. 143, pp. 146–151, 2015.

[53] S. v. Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in r," *Journal of statistical software*, pp. 1–68, 2010.

[54] K. Lakshminarayan, S. A. Harp, R. P. Goldman, T. Samad *et al.*, "Imputation of missing data using machine learning techniques." in *KDD*, 1996, pp. 140–145.

[55] F. A. Dahl, "Convergence of random k-nearest-neighbour imputation," *Computational Statistics & Data Analysis*, vol. 51, no. 12, pp. 5913–5917, 2007.

[56] S. Zhang, J. Zhang, X. Zhu, Y. Qin, and C. Zhang, "Missing value imputation based on data clustering," in *Transactions on computational science I*. Springer, 2008, pp. 128–138.

[57] L. A. Belanche, V. Kobayashi, and T. Aluja, "Handling missing values in kernel methods with application to microbiology data," *Neurocomputing*, vol. 141, pp. 110–116, 2014.

[58] A. Aleryani, W. Wang, and B. De La Iglesia, "Multiple imputation ensembles (mie) for dealing with missing data," *SN Computer Science*, vol. 1, pp. 1–20, 2020.

[59] J. Josse, J. Pagès, and F. Husson, "Multiple imputation in principal component analysis," *Advances in data analysis and classification*, vol. 5, no. 3, pp. 231–246, 2011.

[60] H. Tan, Z. Yang, G. Feng, W. Wang, and B. Ran, "Correlation analysis for tensor-based traffic data imputation method," *Procedia-Social and Behavioral Sciences*, vol. 96, pp. 2611–2620, 2013.

[61] F. Imani, C. Cheng, R. Chen, and H. Yang, "Nested gaussian process modeling for high-dimensional data imputation in healthcare systems," in *IISE 2018 Conference & Expo, Orlando, FL, May*, 2018, pp. 19–22.

[62] S. J. Choudhury and N. R. Pal, "Imputation of missing data with neural networks for classification," *Knowledge-Based Systems*, vol. 182, p. 104838, 2019.

[63] H. Turabieh, M. Mafarja, and S. Mirjalili, "Dynamic adaptive network-based fuzzy inference system (d-anfis) for the imputation of missing data for internet of medical things applications," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9316–9325, 2019.

[64] S. Moritz, A. Sardá, T. Bartz-Beielstein, M. Zaefferer, and J. Stork, "Comparison of different methods for univariate time series imputation in r," *arXiv preprint arXiv:1510.03924*, 2015.

[65] R. Gravina, P. Alinia, H. Ghasemzadeh, and G. Fortino, "Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges," *Information Fusion*, vol. 35, pp. 68–80, 2017.

[66] M. Gochoo, T.-H. Tan, S.-H. Liu, F.-R. Jean, F. Alnajjar, and S.-C. Huang, "Unobtrusive activity recognition of elderly people living alone using anonymous binary sensors and dcnn," *IEEE Journal of Biomedical and Health Informatics*, 2018.

[67] "UJAmI," http://ceatic.ujaen.es/ujami/sites/default/files/2018-07/UCAmI%20Cup.zip, (accessed on Feb, 2021). [Online]. Available: http://ceatic.ujaen.es/ujami/sites/default/files/2018-07/UCAmI%20Cup.zip

[68] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[69] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep learning models for real-time human activity recognition with smartphones," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 743–755, 2020.

[70] Z. Hussain, Q. Z. Sheng, and W. E. Zhang, "A review and categorization of techniques on device-free human activity recognition," *Journal of Network and Computer Applications*, p. 102738, 2020.

[71] uMoDT, "Source code," https://github.com/masifrazzaq/TVS-DTC/, accessed: 2019-04-10.

[72] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: overview, challenges and opportunities," *arXiv preprint arXiv:2001.07416*, 2020.

[73] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, "Innohar: a deep neural network for complex human activity recognition," *Ieee Access*, vol. 7, pp. 9893–9902, 2019.

[74] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition." in *Ijcai*, vol. 15, 2015, pp. 3995–4001.

[75] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[76] B. K. Iwana and S. Uchida, "Dynamic weight alignment for temporal convolutional neural networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3827–3831.

[77] A. G. Salguero, P. Delatorre, J. Medina, M. Espinilla, and A. J. Tomeu, "Ontology-based framework for the automatic recognition of activities of daily living using class expression learning techniques," *Scientific Programming*, vol. 2019, 2019.

[78] P.-L. Vaudry and G. Lapalme, "Narrative generation from extracted associations," in *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, 2015, pp. 136–145.

[79] J. D. Cerón, D. M. López, and B. M. Eskofier, "Human activity recognition using binary sensors, ble beacons, an intelligent floor and acceleration data: A machine learning approach," in *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 2, no. 19, 2018, p. 1265.

[80] L. Meng, C. Miao, and C. Leung, "Towards online and personalized daily activity recognition, habit modeling, and anomaly detection for the solitary elderly through unobtrusive sensing," *Multimedia Tools and Applications*, vol. 76, no. 8, pp. 10 779–10 799, 2017.

[81] D. Riboni, T. Sztyler, G. Civitarese, and H. Stuckenschmidt, "Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 1–12.

[82] L. Chen, C. D. Nugent, and G. Okeyo, "An ontology-based hybrid approach to activity modeling for smart homes." *IEEE Trans. Human-Machine Systems*, vol. 44, no. 1, pp. 92–105, 2014.

[83] R. Singh, A. Sonawane, and R. Srivastava, "Recent evolution of modern datasets for human activity recognition: a deep survey," *Multimedia Systems*, pp. 1–24, 2019.

[84] L. Chen and C. D. Nugent, *Human Activity Recognition and Behaviour Analysis*. Springer, 2019.

[85] S. Nigam, R. Singh, and A. Misra, "A review of computational approaches for human behavior detection," *Archives of Computational Methods in Engineering*, vol. 26, no. 4, pp. 831–863, 2019.

[86] B. Fu, N. Damer, F. Kirchbuchner, and A. Kuijper, "Sensing technology for human activity recognition: A comprehensive survey," *IEEE Access*, vol. 8, pp. 83 791–83 820, 2020.

[87] T. Kawashima, Y. Kawanishi, I. Ide, H. Murase, D. Deguchi, T. Aizawa, and M. Kawade, "Action recognition from extremely low-resolution thermal image sequence," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017, pp. 1–6.

[88] L. Chen and C. D. Nugent, "Sensor-based activity recognition review," in *Human Activity Recognition and Behaviour Analysis*. Springer, 2019, pp. 23–47.

[89] S. Murakami, T. Kimura, and I. E. Yairi, "Privacy-preserving resident monitoring system with ultra low-resolution imaging and the examination of its ease of installation," in *Annual Conference of the Japanese Society for Artificial Intelligence*. Springer, 2019, pp. 278–290.

[90] S. Shelke and B. Aksanli, "Static and dynamic activity detection with ambient sensors in smart spaces," *Sensors*, vol. 19, no. 4, p. 804, 2019.

[91] N. M. Nasrabadi, "Deeptarget: An automatic target recognition using deep convolutional neural networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 2687–2697, 2019.

[92] M. Tiwari and R. Singhai, "A review of detection and tracking of object from image and video sequences," *International Journal of Computational Intelligence Research*, vol. 13, no. 5, pp. 745–765, 2017.

[93] W.-C. Lin and C.-F. Tsai, "Missing value imputation: a review and analysis of the literature (2006–2017)," *Artificial Intelligence Review*, vol. 53, no. 2, pp. 1487–1509, 2020.

[94] X. Chen, Z. Wei, Z. Li, J. Liang, Y. Cai, and B. Zhang, "Ensemble correlation-based low-rank matrix completion with applications to traffic data imputation," *Knowledge-Based Systems*, vol. 132, pp. 249–262, 2017.

[95] D. P. Mesquita, J. P. Gomes, A. H. S. Junior, and J. S. Nobre, “Euclidean distance estimation in incomplete datasets,” *Neurocomputing*, vol. 248, pp. 11–18, 2017.

[96] S. Ghorbani and M. C. Desmarais, “Performance comparison of recent imputation methods for classification tasks over binary data,” *Applied Artificial Intelligence*, vol. 31, no. 1, pp. 1–22, 2017.

[97] R. Pan, T. Yang, J. Cao, K. Lu, and Z. Zhang, “Missing data imputation by k nearest neighbours based on grey relational structure and mutual information,” *Applied Intelligence*, vol. 43, no. 3, pp. 614–632, 2015.

[98] L. F. Burgette and J. P. Reiter, “Multiple imputation for missing data via sequential regression trees,” *American journal of epidemiology*, vol. 172, no. 9, pp. 1070–1076, 2010.

[99] J. Xia, S. Zhang, G. Cai, L. Li, Q. Pan, J. Yan, and G. Ning, “Adjusted weight voting algorithm for random forests in handling missing values,” *Pattern Recognition*, vol. 69, pp. 52–60, 2017.

[100] J. Luengo, S. García, and F. Herrera, “On the choice of the best imputation methods for missing values considering three groups of classification methods,” *Knowledge and information systems*, vol. 32, no. 1, pp. 77–108, 2012.

[101] B. Saha, S. Gupta, D. Phung, and S. Venkatesh, “Effective sparse imputation of patient conditions in electronic medical records for emergency risk predictions,” *Knowledge and Information Systems*, vol. 53, no. 1, pp. 179–206, 2017.

[102] I. B. Aydilek and A. Arslan, “A novel hybrid approach to estimating missing values in databases using k-nearest neighbors and neural networks,” *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 8, pp. 4705–4717, 2012.

[103] S. Zhang, “Nearest neighbor selection for iteratively knn imputation,” *Journal of Systems and Software*, vol. 85, no. 11, pp. 2541–2552, 2012.

[104] J. Shao, W. Meng, and G. Sun, “Evaluation of missing value imputation methods for wireless soil datasets,” *Personal and Ubiquitous Computing*, vol. 21, no. 1, pp. 113–123, 2017.

[105] A. Paul, J. Sil, and C. D. Mukhopadhyay, “Gene selection for designing optimal fuzzy rule base classifier by estimating missing value,” *Applied Soft Computing*, vol. 55, pp. 276–288, 2017.

[106] A. K. Wong and D. K. Chiu, “Synthesizing statistical knowledge from incomplete mixed-mode data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 796–805, 1987.

[107] X. Zhang, X. Song, H. Wang, and H. Zhang, “Sequential local least squares imputation estimating missing value of microarray data,” *Computers in biology and medicine*, vol. 38, no. 10, pp. 1112–1120, 2008.

[108] D. B. Rubin, *Multiple imputation for nonresponse in surveys*. John Wiley & Sons, 2004, vol. 81.

[109] U. Garciarena and R. Santana, “An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers,” *Expert Systems with Applications*, vol. 89, pp. 52–65, 2017.

[110] T. E. Raghunathan, J. M. Lepkowski, J. Van Hoewyk, P. Solenberger *et al.*, “A multivariate technique for multiply imputing missing values using a sequence of regression models,” *Survey methodology*, vol. 27, no. 1, pp. 85–96, 2001.

[111] H. C. Valdiviezo and S. Van Aelst, “Tree-based prediction on incomplete data using imputation or surrogate decisions,” *Information Sciences*, vol. 311, pp. 163–181, 2015.

[112] P. Royston, “Multiple imputation of missing values,” *The Stata Journal*, vol. 4, no. 3, pp. 227–241, 2004.

[113] Y. Liu and V. Gopalakrishnan, “An overview and evaluation of recent machine learning imputation methods using cardiac imaging data,” *Data*, vol. 2, no. 1, p. 8, 2017.

[114] S. Ramasamy Ramamurthy and N. Roy, “Recent trends in machine learning for human activity recognition—a survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1254, 2018.

[115] M. A. Razzaq, I. Cleland, C. Nugent, and S. Lee, “Multimodal sensor data fusion for activity recognition using filtered classifier,” *Proceedings*, vol. 2, no. 19, 2018. [Online]. Available: https://www.mdpi.com/2504-3900/2/19/1262

[116] Z. Cai, Z. Gu, Z. L. Yu, H. Liu, and K. Zhang, “A real-time visual object tracking system based on kalman filter and mb-lbp feature matching,” *Multimedia Tools and Applications*, vol. 75, no. 4, pp. 2393–2409, 2016.

[117] R. E. Kalman, “A new approach to linear filtering and prediction problems,” 1960.

[118] Q. Li, R. Li, K. Ji, and W. Dai, “Kalman filter and its application,” in *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*. IEEE, 2015, pp. 74–77.

[119] C. Huang, B. Wu, and R. Nevatia, “Robust object tracking by hierarchical association of detection responses,” in *European Conference on Computer Vision*. Springer, 2008, pp. 788–801.

[120] P. D. O’MALLEY, “Use of a kalman filter to improve realtime video stream image processing: An example,” 2000.

[121] T. Cipra and R. Romera, “Kalman filter with outliers and missing observations,” *Test*, vol. 6, no. 2, pp. 379–395, 1997.

[122] A. A. Vijay and A. K. Johnson, “An integrated system for tracking and recognition using kalman filter,” in *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*. IEEE, 2014, pp. 1065–1069.

[123] G. Khan, Z. Tariq, and M. U. G. Khan, “Multi-person tracking based on faster r-cnn and deep appearance features,” in *Visual Object Tracking in the Deep Neural Networks Era*. IntechOpen, 2019.

[124] M. Dimitrievski, D. Van Hamme, P. Veelaert, and W. Philips, “Cooperative multi-sensor tracking of vulnerable road users in the presence of missing detections,” *Sensors*, vol. 20, no. 17, p. 4817, 2020.

[125] J. Chen and Q. Zhang, "Distinct sampling on streaming data with near-duplicates," in *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. ACM, 2018, pp. 369–382.

[126] M. Espinilla, J. Medina, J. Hallberg, and C. Nugent, "A new approach based on temporal sub-windows for online sensor-based activity recognition," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2018.

[127] M. Fahim, T. Baker, A. M. Khattak, B. Shah, S. Aleem, and F. Chow, "Context mining of sedentary behaviour for promoting self-awareness using a smartphone," *Sensors*, vol. 18, no. 3, p. 874, 2018.

[128] G. Song, W. Qu, X. Liu, and X. Wang, "Approximate calculation of window aggregate functions via global random sample," *Data Science and Engineering*, vol. 3, no. 1, pp. 40–51, 2018.

[129] P. J. Haas, "Data-stream sampling: basic techniques and results," in *Data Stream Management*. Springer, 2016, pp. 13–44.

[130] B. Yamansavaşçılar and M. A. Güvensan, "Activity recognition on smartphones: Efficient sampling rates and window sizes," in *Pervasive Computing and Communication Workshops (PerCom Workshops), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[131] V. Braverman, R. Ostrovsky, and C. Zaniolo, "Optimal sampling from sliding windows," *Journal of Computer and System Sciences*, vol. 78, no. 1, pp. 260–272, 2012.

[132] K.-L. Wu and Y. Xia, "Adaptive sampling schemes for clustering streaming graphs," Feb. 6 2018, uS Patent 9,886,521.

[133] B. Hentschel, P. J. Haas, and Y. Tian, "Temporally-biased sampling for online model management," *arXiv preprint arXiv:1801.09709*, 2018.

[134] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive and mobile computing*, vol. 10, pp. 138–154, 2014.

[135] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. Cardoso, “Preprocessing techniques for context recognition from accelerometer data,” *Personal and Ubiquitous Computing*, vol. 14, no. 7, pp. 645–662, 2010.

[136] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, “Window size impact in human activity recognition,” *Sensors*, vol. 14, no. 4, pp. 6474–6499, 2014.

[137] X. Lai, Q. Liu, X. Wei, W. Wang, G. Zhou, and G. Han, “A survey of body sensor networks,” *Sensors*, vol. 13, no. 5, pp. 5406–5447, 2013.

[138] J. Wang, Y. Chen, L. Hu, X. Peng, and S. Y. Philip, “Stratified transfer learning for cross-domain activity recognition,” in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2018, pp. 1–10.

[139] L. Wang, T. Gu, X. Tao, H. Chen, and J. Lu, “Recognizing multi-user activities using wearable sensors in a smart home,” *Pervasive and Mobile Computing*, vol. 7, no. 3, pp. 287–298, 2011.

[140] H. N. Rafsanjani, C. R. Ahn, and M. Alahmad, “A review of approaches for sensing, understanding, and improving occupancy-related energy-use behaviors in commercial buildings,” *Energies*, vol. 8, no. 10, pp. 10 996–11 029, 2015.

[141] P. Hevesi, S. Wille, G. Pirkl, N. Wehn, and P. Lukowicz, “Monitoring household activities and user location with a cheap, unobtrusive thermal sensor array,” in *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*. ACM, 2014, pp. 141–145.

[142] S. S. Sengar and S. Mukhopadhyay, “Moving object detection based on frame difference and w4,” *Signal, Image and Video Processing*, vol. 11, no. 7, pp. 1357–1364, 2017.

[143] N. A. Mandellos, I. Keramitsoglou, and C. T. Kiranoudis, “A background subtraction algorithm for detecting and tracking vehicles,” *Expert Systems with Applications*, vol. 38, no. 3, pp. 1619–1631, 2011.

[144] J. Xing, H. Ai, and S. Lao, "Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1200–1207.

[145] H. S. Parekh, D. G. Thakore, and U. K. Jaliya, "A survey on object detection and tracking methods," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 2, pp. 2970–2979, 2014.

[146] A. Farhangfar, L. Kurgan, and J. Dy, "Impact of imputation of missing values on classification error for discrete data," *Pattern Recognition*, vol. 41, no. 12, pp. 3692–3705, 2008.

[147] T. Kautz, B. H. Groh, J. Hannink, U. Jensen, H. Strubberg, and B. M. Eskofier, "Activity recognition in beach volleyball using a deep convolutional neural network," *Data Mining and Knowledge Discovery*, vol. 31, no. 6, pp. 1678–1705, 2017.

[148] A. Farhangfar, L. A. Kurgan, and W. Pedrycz, "A novel framework for imputation of missing values in databases," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 5, pp. 692–709, 2007.

[149] Q. Ni, T. Patterson, I. Cleland, and C. Nugent, "Dynamic detection of window starting positions and its implementation within an activity recognition framework," *Journal of biomedical informatics*, vol. 62, pp. 171–180, 2016.

[150] S. Chernbumroong, S. Cang, and H. Yu, "A practical multi-sensor activity recognition system for home-based care," *decision support systems*, vol. 66, pp. 61–70, 2014.

[151] I.-H. Bae, "An ontology-based approach to adl recognition in smart homes," *Future Generation Computer Systems*, vol. 33, pp. 32–41, 2014.

[152] A. Salguero, M. Espinilla, P. Delatorre, and J. Medina, "Using ontologies for the online recognition of activities of daily living," *Sensors*, vol. 18, no. 4, p. 1202, 2018.

[153] M. K. Sarker, N. Xie, D. Doran, M. Raymer, and P. Hitzler, "Explaining trained neural networks with semantic web technologies: First steps," 2017.

[154] S. Demri, R. Fervari, and A. Mansutti, “Axiomatising logics with separating conjunction and modalities,” in *European Conference on Logics in Artificial Intelligence*. Springer, 2019, pp. 692–708.

[155] G. Meditskos, S. Dasiopoulou, and I. Kompatsiaris, “Metaq: A knowledge-driven framework for context-aware activity recognition combining sparql and owl 2 activity patterns,” *Pervasive and Mobile Computing*, vol. 25, pp. 104–124, 2016.

[156] E. Amador-Domínguez, P. Hohenecker, T. Lukasiewicz, D. Manrique, and E. Serrano, “An ontology-based deep learning approach for knowledge graph completion with fresh entities,” in *International Symposium on Distributed Computing and Artificial Intelligence*. Springer, 2019, pp. 125–133.

[157] R. Socher, D. Chen, C. D. Manning, and A. Ng, “Reasoning with neural tensor networks for knowledge base completion,” in *Advances in neural information processing systems*, 2013, pp. 926–934.

[158] Y. Zhu and J. Ferreira, “Data integration to create large-scale spatially detailed synthetic populations,” in *Planning support systems and smart cities*. Springer, 2015, pp. 121–141.

[159] W. Luo, J. Xing, X. Zhang, X. Zhao, and T.-K. Kim, “Multiple object tracking: A literature review,” *arXiv preprint arXiv:1409.7618*, 2014.

[160] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *Acm computing surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.

[161] X. Luo, Q. Guan, H. Tan, L. Gao, Z. Wang, and X. Luo, “Simultaneous indoor tracking and activity recognition using pyroelectric infrared sensors,” *Sensors*, vol. 17, no. 8, p. 1738, 2017.

[162] W.-C. Hu, C.-H. Chen, T.-Y. Chen, D.-Y. Huang, and Z.-C. Wu, “Moving object detection and tracking from video captured by moving camera,” *Journal of Visual Communication and Image Representation*, vol. 30, pp. 164–180, 2015.

[163] L. Hou, W. Wan, J.-N. Hwang, R. Muhammad, M. Yang, and K. Han, “Human tracking over camera networks: a review,” *EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 1, p. 43, 2017.

[164] B. Zhang, Z. Li, A. Perina, A. Del Bue, V. Murino, and J. Liu, “Adaptive local movement modeling for robust object tracking,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 7, pp. 1515–1526, 2016.

[165] W. Choi and S. Savarese, “A unified framework for multi-target tracking and collective activity recognition,” in *European Conference on Computer Vision*. Springer, 2012, pp. 215–230.

[166] J. Shen, D. Yu, L. Deng, and X. Dong, “Fast online tracking with detection refinement,” *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[167] T. Zebin, P. J. Scully, and K. B. Ozanyan, “Human activity recognition with inertial sensors using a deep learning approach,” in *2016 IEEE SENSORS*. IEEE, 2016, pp. 1–3.

[168] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[169] J. K. Dhillon, A. K. S. Kushwaha *et al.*, “A recent survey for human activity recoginition based on deep learning approach,” in *Image Information Processing (ICIIP), 2017 Fourth International Conference on*. IEEE, 2017, pp. 1–6.

[170] T. Dobhal, V. Shitole, G. Thomas, and G. Navada, “Human activity recognition using binary motion image and deep learning,” *Procedia computer science*, vol. 58, pp. 178–185, 2015.

[171] S. Ji, W. Xu, M. Yang, and K. Yu, “3d convolutional neural networks for human action recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[172] K. S. Ray and S. Chakraborty, “An efficient approach for object detection and tracking of objects in a video with variable background,” *arXiv preprint arXiv:1706.02672*, 2017.

[173] F. S. Leira, T. A. Johansen, and T. I. Fossen, “Automatic detection, classification and tracking of objects in the ocean surface from uavs using a thermal camera,” in *Aerospace Conference, 2015 IEEE*. IEEE, 2015, pp. 1–10.

[174] F. G. De Natale and G. Boato, “Detecting morphological filtering of binary images,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1207–1217, 2017.

[175] Y. Wang, X. Luo, S. Fu, and S. Hu, “Context multi-task visual object tracking via guided filter,” *Signal Processing: Image Communication*, 2018.

[176] A. Dehghan and M. Shah, “Binary quadratic programing for online tracking of hundreds of people in extremely crowded scenes,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 568–581, 2018.

[177] H. Ning, F. Shi, T. Zhu, Q. Li, and L. Chen, “A novel ontology consistent with acknowledged standards in smart homes,” *Computer Networks*, vol. 148, pp. 101–107, 2019.

[178] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, p. 33, 2014.

[179] S. Dernbach, B. Das, N. C. Krishnan, B. L. Thomas, and D. J. Cook, “Simple and complex activity recognition through smart phones,” in *Intelligent Environments (IE), 2012 8th International Conference on*. IEEE, 2012, pp. 214–221.

[180] G. Okeyo, L. Chen, H. Wang, and R. Sterritt, “Dynamic sensor data segmentation for real-time knowledge-driven activity recognition,” *Pervasive and Mobile Computing*, vol. 10, pp. 155–172, 2014.

[181] J. Wan, M. J. O’grady, and G. M. O’hare, “Dynamic sensor event segmentation for real-time activity recognition in a smart home context,” *Personal and Ubiquitous Computing*, vol. 19, no. 2, pp. 287–301, 2015.

[182] D. Triboan, L. Chen, F. Chen, and Z. Wang, “A semantics-based approach to sensor data segmentation in real-time activity recognition,” *Future Generation Computer Systems*, vol. 93, pp. 224–236, 2019.

[183] R. Chen and Y. Tong, "A two-stage method for solving multi-resident activity recognition in smart environments," *Entropy*, vol. 16, no. 4, pp. 2184–2203, 2014.

[184] J. Zhou and Z. Huang, "Recover missing sensor data with iterative imputing network," in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[185] J. Liu, Y. Li, X. Tian, A. K. Sangaiah, and J. Wang, "Towards semantic sensor data: An ontology approach," *Sensors*, vol. 19, no. 5, p. 1193, 2019.

[186] A. C. Yang, H.-H. Hsu, and M.-D. Lu, "Imputing missing values in microarray data with ontology information," in *2010 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*. IEEE, 2010, pp. 535–540.

[187] S. Song, A. Zhang, L. Chen, and J. Wang, "Enriching data imputation with extensive similarity neighbors," *Proceedings of the VLDB Endowment*, vol. 8, no. 11, pp. 1286–1297, 2015.

[188] H. Stuckenschmidt, "A semantic similarity measure for ontology-based information," in *International Conference on Flexible Query Answering Systems*. Springer, 2009, pp. 406–417.

[189] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Systems with Applications*, 2018.

[190] P. Cerda, G. Varoquaux, and B. Kégl, "Similarity encoding for learning with dirty categorical variables," *Machine Learning*, vol. 107, no. 8-10, pp. 1477–1494, 2018.

[191] F. Moya Rueda, R. Grzeszick, G. Fink, S. Feldhorst, and M. ten Hompel, "Convolutional neural networks for human activity recognition using body-worn sensors," in *Informatics*, vol. 5, no. 2. Multidisciplinary Digital Publishing Institute, 2018, p. 26.

[192] F. Li, K. Shirahama, M. Nisar, L. Köping, and M. Grzegorzek, "Comparison of feature learning methods for human activity recognition using wearable sensors," *Sensors*, vol. 18, no. 2, p. 679, 2018.

[193] B. Sahbani and W. Adiprawita, “Kalman filter and iterative-hungarian algorithm implementation for low complexity point tracking as part of fast multiple object tracking system,” in *System Engineering and Technology (ICSET), 2016 6th International Conference on*. IEEE, 2016, pp. 109–115.

[194] “Heimanntvs,” http://www.heimannsensor.com/productsimaging.php, accessed: 2019-03-3.

[195] J. e. a. Medina-Quero and C. Nugent, “Computer vision-based gait velocity from non-obtrusive thermal vision sensors,” *To be Sumitted*, vol. 0, no. 0, pp. 1–6, 2018.

[196] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, “Convolutional neural networks for human activity recognition using mobile sensors,” in *Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on*. IEEE, 2014, pp. 197–205.

[197] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[198] S. Albelwi and A. Mahmood, “A framework for designing the architectures of deep convolutional neural networks,” *Entropy*, vol. 19, no. 6, p. 242, 2017.

[199] Z. Gao, “Object-based image classification and retrieval with deep feature representations,” *Doctor of Philosophy Thesis, School of Computing and Information Technology, University of Wollongong*, 2018.

[200] M. Y. Teow, “Understanding convolutional neural networks using a minimal model for handwritten digit recognition,” in *Automatic Control and Intelligent Systems (I2CACIS), 2017 IEEE 2nd International Conference on*. IEEE, 2017, pp. 167–172.

[201] J. Portmann, S. Lynen, M. Chli, and R. Siegwart, “People detection and tracking from aerial thermal views,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1794–1800.

[202] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2137–2155, Nov 2016.

[203] Tzutalin, "Labelimg: Image annotation tool," https://github.com/tzutalin/labelImg, accessed: 2018-12-25.

[204] VOT2016, "Vot2016 benchmark," http://www.votchallenge.net/vot2016/, accessed: 2018-12-25.

[205] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[206] D. Mishkin, N. Sergievskiy, and J. Matas, "Systematic evaluation of convolution neural network advances on the imagenet," *Computer Vision and Image Understanding*, vol. 161, pp. 11–19, 2017.

[207] G. Vink, "Towards a standardized evaluation of multiple imputation routines," 2016.

[208] C. D. Nguyen, J. B. Carlin, and K. J. Lee, "Model checking in multiple imputation: an overview and case study," *Emerging themes in epidemiology*, vol. 14, no. 1, p. 8, 2017.

[209] A. Sundararajan and A. I. Sarwat, "Evaluation of missing data imputation methods for an enhanced distributed pv generation prediction," in *Proceedings of the Future Technologies Conference*. Springer, 2019, pp. 590–609.

[210] L. Peng, L. Chen, Z. Ye, and Y. Zhang, "Aroma: A deep multi-task learning based simple and complex human activity recognition method using wearable sensors," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 2, p. 74, 2018.

[211] F. Ordóñez, P. de Toledo, A. Sanchis *et al.*, "Activity recognition using hybrid generative/discriminative models on home environments using binary sensors," *Sensors*, vol. 13, no. 5, pp. 5460–5477, 2013.

[212] S. Salomón and C. Tîrnăucă, “Human activity recognition through weighted finite automata,” in *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 2, no. 19, 2018, p. 1263.

[213] V. Manohar, P. Soundararajan, H. Raju, D. Goldgof, R. Kasturi, and J. Garofolo, “Performance evaluation of object detection and tracking in video,” in *Asian Conference on Computer Vision*. Springer, 2006, pp. 151–161.

[214] R. Gade and T. Moeslund, “Thermal tracking of sports players,” *Sensors*, vol. 14, no. 8, pp. 13 679–13 691, 2014.

[215] E. Bochinski, V. Eiselein, and T. Sikora, “High-speed tracking-by-detection without using image information,” in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017, pp. 1–6.

[216] X. Wan, J. Wang, and S. Zhou, “An online and flexible multi-object tracking framework using long short-term memory,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1230–1238.

[217] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3464–3468.

[218] S. Murray, “Real-time multiple object tracking-a study on the importance of speed,” *arXiv preprint arXiv:1709.03572*, 2017.

[219] L. Chen, H. Ai, Z. Zhuang, and C. Shang, “Real-time multiple people tracking with deeply learned candidate selection and person re-identification.” *arXiv preprint arXiv:1809.04427v1*, 2018.

[220] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: the clear mot metrics,” *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

[221] Y. Wu, J. Lim, and M.-H. Yang, “Object tracking benchmark,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[222] L. Čehovin, M. Kristan, and A. Leonardis, “Is my new tracker really better than yours?” in *IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2014, pp. 540–547.

[223] L. Čehovin, A. Leonardis, and M. Kristan, “Visual object tracking performance measures revisited,” *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1261–1274, 2016.

[224] Q. Wang, D. Gong, M. Qi, Y. Shen, and Y. Lei, “Temporal sparse feature auto-combination deep network for video action recognition,” *Concurrency and Computation: Practice and Experience*, p. e4487, 2018.

[225] M. N. Norazian, A. Shukri, P. Yahaya, N. Azam, P. Ramli, N. F. Fitri, M. Yusof, A. Mohd Mustafa Al Bakri *et al.*, “Roles of imputation methods for filling the missing values: A review,” 2013.

Appendix A

List of Acronyms

Acronyms

In alphabetical order:

ADL Activities of Daily Living

BLE Bluetooth Low Energy

BR Bounding Rectangle

CNN Convolutional Neural Network

CV Computer Vision

HAR Human Activity Recognition

IoU Intersection over Union

KF Kalman fltering

LOCF Last Observation Carried Forward

MAR Missing at Random

MCAR Missing Completely at Random

MNAR Missing not at Random

MOT Multi-object Tracking

MOTA Multiple Object Tracking Accuracy

MSE Mean Squared Error

NOCB Next Observation Carried Backward

OWL Web Ontology Language

ReLU Rectifed Linear Unit

ROI Region of Interest

SemDeep ANN Semantic Deep Artificial Neural Network

SemImput Semantic Imputation

SemImputOnt Semantic Imputation Ontology

SGDM Stochastic Gradient Descent with Momentum

TIR Thermal Infrared

TVS Thermal Vision Sensor

TVS-F Thermal Vision Sensor Frame

TVS-F$_{seq}$ TVS frame sequence

TVS-MoFV Thermal Vision Sensor Multi-occupant feature vector

uMoDT unobtrusive Multi-occupant Detection and Tracking

Appendix B

List of Publications

B.1 International Journal Papers [12]

1 **Muhammad Asif Razzaq**, Javier Medina Quero, Ian Cleland, Chris Nugent, Usman Akhtar, Hafiz Syed Bilal Ali, Ubaid Ur Rehman, and Sungyoung Lee, "uMoDT: An unobtrusive Multi-occupant Detection and Tracking using robust Kalman filter for real-time activity recognition", *Multimedia Systems* (SCI, IF:1.956), Accepted May 2020.

2 **Muhammad Asif Razzaq**, Ian Cleland, Chris Nugent, and Sungyoung Lee, "SemImput: Bridging Semantic Imputation with Deep Learning for Complex Human Activity Recognition". *Sensors*, (SCIE, IF:3.031), doi:10.3390/s20102771, 2020.

3 **Muhammad Asif Razzaq**, Claudia Villalonga, Sungyoung Lee, Usman Akhtar, Maqbool Ali, Eun-Soo Kim, Asad Masood Khattak, Hyonwoo Seung, Taeho Hur, Jaehun Bang, Dohyeong Kim and Wajahat Ali Khan, "mlCAF: Multi-Level Cross-Domain Semantic Context Fusioning for Behavior Identification", *Sensors* (SCIE, IF:2.677), Doi: 10.3390/s17102433., 2017.

4 Bilal, Hafiz Syed Muhammad, Muhammad Bilal Amin, Jamil Hussain, Syed Imran Ali, **Muhammad Asif Razzaq**, Musarrat Hussain, Asim Abbas Turi, Gwang Hoon Park, Sun Moo Kang, and Sungyoung Lee, "Towards User-Centric Intervention Adaptiveness: Influencing Behavior-Context based Healthy Lifestyle Interventions.", *IEEE Access* (SCIE, IF: 4.098), Accepted, 2020 DOI: 10.1109/ACCESS.2020.3026688.

5 Bilal Hafiz Syed Muhammad, Muhammad Bilal Amin, Jamil Hussain, Syed Imran Ali, Shujaat Hussain, Muhammad Sadiq, **Muhammad Asif Razzaq**, Asim Abbas, Chunho Choi and

Sungyoung Lee, "On computing critical factors based healthy behavior index for behavior assessment", *International Journal of Medical Informatics* (2020): 104181.(SCI, IF:2.731), Accepted, 2020.

6 Usman Akhtar, Anita Sant'Anna, Chang-Ho Jihn, **Muhammad Asif Razzaq**, Jaehun Bang, and Sungyoung Lee, "A cache-based method to improve query performance of linked Open Data cloud", *Computing* (SCI, IF:2.063), Doi:doi.org/10.1007/s00607-020-00814-9, 2020.

7 Ubaid Ur Rehman, Dong Jin Chang, Younhea Jung, Usman Akhtar, **Muhammad Asif Razzaq** and Sungyoung Lee, "Medical Instructed Real-time Assistant for Patient with Glaucoma and Diabetic Conditions", *Applied Sciences* (SCIE, IF: 2.217), doi:10.3390/app10072216, 2020.

8 Usman Akhtar, **Muhammad Asif Razzaq**, Ubaid Ur Rehman, Muhammad Bilal Amin, Wajahat Ali Khan, Eui-Nam Huh, and Sungyoung Lee, "Change-Aware Scheduling for Effectively Updating Linked Open Data Caches", *IEEE Access* (SCIE, IF: 3.557), DOI:10.1109/ACCESS.2018.2871511, 2018.

9 Maqbool Ali, Soyeon Caren Han, Hafiz Syed Muhammad Bilal, Sungyoung Lee, Matthew Jee Yun Kang, Byeong Ho Kang, **Muhammad Asif Razzaq**, and Muhammad Bilal Amin, "iCBLS: An Interactive Case-Based Learning System for Medical Education", *International Journal of Medical Informatics* (SCI, IF:3.287), Vol.109, pp.55–69, 2018.

10 **Maqbool Ali**, Hafiz Syed Muhammad Bilal, **Muhammad Asif Razzaq**, Jawad Khan, Sungyoung Lee, Muhammad Idris, Mohammad Aazam, Taebong Choi, Soyeon Caren Han, and Byeong Ho Kang, "IoTFLiP: IoT-based Flip Learning Platform for Medical Education", *Digital Communications and Networks* (ESCI, Elsevier), Vol.3, pp.188–194, 2017.

11 Claudia Villalonga, **Muhammad Asif Razzaq**, Wajahat Ali Khan, Hector Pomares, Ignacio Rojas, Sungyoung Lee and Oresti Banos, "Ontology-Based High-Level Context Inference for Human Behavior Identification", *Sensors* (SCIE, IF: 2.033), Vol.16, Issue 10, doi:10.3390/s16101617, 2016.

12 Oresti Banos, Claudia Villalonga, Jaehun Bang, Taeho Hur, Donguk Kang, Sangbeom Park, Thien Huynh-The, Vui Le-Ba, Muhammad Bilal Amin, **Muhammad Asif Razzaq**, Wahajat Ali Khan, Choong Seon Hong and Sungyoung Lee, "Human Behavior Analysis by Means of Multimodal Context Mining", *Sensors* (SCIE, IF: 2.033), Vol.16, Issue 8, doi:10.3390/s16081264, 2016

B.2 International Conference Papers [12]

1 **Muhammad Asif Razzaq**, Jaehun Bang, Sunmoo Svenna Kang and Sungyoung Lee, "UnSkEm: Unobtrusive Skeletal-based Emotion Recognition for User Experience", *The 34th International Conference on Information Networking* (ICOIN 2020), Barcelona, Spain , Jan 7-10, 2020

2 **Muhammad Asif Razzaq**, and Sungyoung Lee, "MMOU-AR: Multimodal Obtrusive and Unobtrusive Activity Recognition Through Supervised Ontology-Based Reasoning", *The 13th International Conference on Ubiquitous Information Management and Communication* (IMCOM 2019), Phuket, Thailand, Jan 4-6, 2019.

3 **Muhammad Asif Razzaq**, Ian Cleland, Chris Nugent and Sugyoung Lee, "Multimodal Sensor Data Fusion for Activity Recognition Using Filtered Classifier", *International Conference on Ubiquitous Computing and Ambient Intelligence* (UCAmi 2018), Vol.2, Dec. 4-7, 2018.

4 **Muhammad Asif Razzaq**, Muhammad Bilal Amin and Sungyoung Lee, "An Ontology-based Hybrid Approach for Accurate Context Reasoning", *The 19th Asia-Pacific Network Operations and Management Symposium* (APNOMS 2017), Seoul, Korea, Sep 27-29, 2017.

5 **Muhammad Asif Razzaq**, Wajahat Ali Khan and Sungyoung Lee, "Intent-Context Fusioning in Healthcare Dialogue-based Systems using JDL Model", *15th International Conference On Smart homes and health Telematics* (ICOST 2017), Paris, France, Aug 29-31, 2017.

6 **Muhammad Asif Razzaq**, and Sungyoung Lee, "Context-aware Dialogue Management Framework for Healthcare", *39th Annual International Conference of the IEEE Engineering*

in Medicine and Biology Society (EMBC 2017), Jeju island, Korea, July 11-15, 2017.

7 Hafiz Syed Muhammad Bilal, **Muhammad Asif Razzaq**, Muhammad Bilal Amin and Sungyoung Lee, "Context-Based Lifelog Monitoring for Just-in-Time Wellness Intervention", *16th International Conference On Smart homes and health Telematics* (ICOST 2018), Singapore, Singapore, July 10-12, 2018.

8 Quero, Javier Medina, Matthew Burns, **Muhammad Asif Razzaq**, Chris Nugent, and Macarena Espinilla. "Detection of falls from non-invasive thermal vision sensors using convolutional neural networks." In *Multidisciplinary Digital Publishing Institute*, Proceedings, vol. 2, no. 19, p. 1236. 2018.

9 Hernandez, Netzahualcoyotl, **Muhammad Asif Razzaq**, Chris Nugent, Ian McChesney, and Shuai Zhang. "Transfer Learning and Data Fusion Approach to Recognize Activities of Daily Life." In Proceedings of the *12th EAI International Conference on Pervasive Computing Technologies for Healthcare*, pp. 227-231. 2018.

10 Jaehun Bang, Dong Uk Kang, Tae Ho Hur, Thien Hyun The, **Muhammad Asif Razzaq**, Wajahat Ali Khan, Oresti Banos and Sungyoung Lee, "IoT based Human Centric Context Awareness Framework for Healthcare and Wellness Platform", *2016 International Symposium on Perception, Action and Cognitive Systems* (PACS), Seoul, Korea, Oct 27-28, 2016.

11 Wajahat Ali Khan, Muhammad Bilal Amin, Oresti Banos, Taqdir Ali, Maqbool Hussain, Muhammad Afzal, Shujaat Hussain, Jamil Hussain, Rahman Ali, Maqbool Ali, Dongwook Kang, Jaehun Bang, Tae Ho Hur, Bilal Ali, Muhammad Idris, **Muhammad Asif Razzaq**, Sungyoung Lee and Byeong Ho Kang, "Mining Minds: Journey of Evolutionary Platform for Ubiquitous Wellness", *12th International Conference on Ubiquitous Healthcare* (u-Healthcare 2015), Osaka, Japan, Nov 30- Dec 2, 2015.

12 Claudia Villalonga, Oresti Banos, Wajahat Ali Khan, Taqdir Ali, **Muhammad Asif Razzaq**, Sungyoung Lee, Hector Pomares and Ignacio Rojas, "High-Level Context Inference for Human Behavior Identification", *7th International Work-conference on Ambient Assisted Living* (IWAAL 2015), Puerto Varas, Patagonia, Chile, Dec 2-3, 2015

B.3 Domestic Conference Papers [7]

1 **Muhammad Asif Razzaq** and Sungyoung Lee, "Quarantine-induced Adaptive Cardiac Rehabilitation During COVID-19: A Road Map", *The Korean Society of Heart Failure* (KSHF) 2020.

2 **Muhammad Asif Razzaq** and Sungyoung Lee, "A Survey of Current Datasets for Cardiovascular Disease Research", *The Korean Society of Heart Failure* (KSHF) 2019.

3 **Muhammad Asif Razzaq** and Sungyoung Lee, "Context Recognition using Non-wearable Sensors in Smart-Home Environment", *The Korean Society of Medical Informatics* (KOSMI), 1, 3, 2018.

4 **Muhammad Asif Razzaq** and Sungyoung Lee, "A Stream Reasoning Model for Continuous Query Answering", *Korea Computer Congress*, 328-330, 2016.

5 Jin Hyuk Gong, Dong Uk Kang, **Muhammad Asif Razzaq**, and Sungyoung Lee, "Implementation of Comprehensive User Context Recognition and Inference System", Proceedings of the *Korean Information Science Society* 6 (2016), 413-415, 2016.

6 **Muhammad Asif Razzaq** and Sungyoung Lee, "A Semantic Reasoning Model for Fortifying Decisions in Smart CDSS", *Korea Computer Congress*, 995-997, 2016.

7 **Muhammad Asif Razzaq** and Sungyoung Lee, "Diet Ontology for Healthier Life", *Korea Computer Congress*, 708-709 (2 pages), 2015.

B.4 Patents [2]

1 Sungyoung Lee and **Muhammad Asif Razzaq**, "Method for intent-context fusioning in healthcare systems for effective dialogue management", *Korean Intellectual Property Office*, Registered: 1020180039520 (2020.09).

2 Sungyoung Lee and **Muhammad Asif Razzaq**, "A semantic imputation framework bridged with deep learning using unobtrusive and obtrusive sensor data for human activity recognition", *Korean Intellectual Property Office*, Apply: (2020.08.27).

B.5 Standards [4]

1 Sungyoung Lee, Jee-In Kim, Jamil Hussain, Fahad Ahmed Satti, Muhmmad Asif Razzq, Cam-Hao Hua, "A Framework For Evaluating Usability Of Natural User Interactions", Asia-Pacific Telecommunity Standardization Program (ASTAP), The 31th APT Standardization Program Forum (ASTAP-31), Submission No. ASTAP-31/INP-48 , June 11, 2019.

2 Sungyoung Lee, Jee-In Kim, Hyun-Woo Nam, Muhammad Asif Razzaq, Jamil Hussain, JaeHun Bang, Cam-Hao Hua, Wajahat Ali Khan, Fahad Satti, Tae-Ho Hur, "Framework for evaluating the usability of natural user interfaces ", Information and Communication Organization Standard (TTAS), ICT Convergence Technical Committee PG415-Intelligent Device, Approval No.: 2019-1320, November 22, 2019

3 Sungyoung Lee, Jee-In Kim, Hyun-Woo Nam, Muhammad Asif Razzaq, Jamil Hussain, JaeHun Bang, Cam-Hao Hua, Wajahat Ali Khan, Fahad Satti, Tae-Ho Hur, " Adaptive Emotion Convergence Model in Various Forms for User Experience Evaluation ", Information and Communication Organization Standards (TTAS), Software/Content Technology Committee PG610-Digital Contents, Approval No.: 2019-1341, November 22, 2019

4 Sungyoung Lee, Jee-In Kim, Hyun-Woo Nam, Muhammad Asif Razzaq, Jamil Hussain, JaeHun Bang, Cam-Hao Hua, Wajahat Ali Khan, Fahad Satti, Tae-Ho Hur, "Multimodal Adaptive Emotion Fusion Model for UX Evaluation ", TTAS , Software/Contents Technology Committee PG610-Digital Contents, Application No.:2019-P1389, October 11, 2019