

HUMAN ACTION RECOGNITION IN AMBIENT ASSISTED LIVING USING  
CONTINUOUS INERTIAL DATA



by  
Gamze Uslu

Submitted to Graduate School of Natural and Applied Sciences  
in Partial Fulfillment of the Requirements  
for the Degree of Doctor of Philosophy in  
Computer Engineering

Yeditepe University

2022

HUMAN ACTION RECOGNITION IN AMBIENT ASSISTED LIVING USING  
CONTINUOUS INERTIAL DATA

APPROVED BY:

Prof. Dr. Şebnem Baydere

(Thesis Supervisor)

(Yeditepe University)

.....

Prof. Dr. Linet Özdamar

(Yeditepe University)

.....

Prof. Dr. Yusuf Sinan Akgül

(Gebze Technical University)

.....

Assist. Prof. Dr. Funda Yıldırım

(Yeditepe University)

.....

Assist. Prof. Dr. Yaşar Safkan

(Özyeğin University)

.....

DATE OF APPROVAL: .... /.... /20....

I hereby declare that this thesis is my own work and that all information in this thesis has been obtained and presented in accordance with academic rules and ethical conduct. I have fully cited and referenced all material and results as required by these rules and conduct, and this thesis study does not contain any plagiarism. If any material used in the thesis requires copyright, the necessary permissions have been obtained. No material from this thesis has been used for the award of another degree.

I accept all kinds of legal liability that may arise in case contrary to these situations.

Name, Last Name: Gamze Uslu

Signature .....

## **ACKNOWLEDGEMENTS**

I am grateful to my supervisor Prof. Dr. Sebnem Baydere for her immense guidance and support. Studying with her is absolutely a mind-broadening experience. She devotes her time to me without any hesitation whenever I need.

Also, I would like to thank my examination committee for their valuable feedback. The perspective they shared with me greatly enriched my point of view.

Finally, I would like to thank my family for their endless love and support. I am extremely lucky to have them. Their existence has always been irreplaceable for me.

## **ABSTRACT**

### **HUMAN ACTION RECOGNITION IN AMBIENT ASSISTED LIVING USING CONTINUOUS INERTIAL DATA**

Ambient intelligence systems open up numerous prospects for providing assistance to elderly and disabled people as well as healthy individuals. Wearable computing devices embroidered with data communication capabilities participating in IoT applications boost coverage of those systems, integrating intelligent modules to daily lives of people. Assistance offered by the intelligent systems can emerge as various use cases including smart healthcare embedded in daily living environments. Such systems incorporate activity recognition units as an integral part to determine the corresponding actuation. For activity detection units to get blended in the course of daily routines of people, detection of individual activities is not sufficient where additional constraints should also be considered alongside activity detection problem. These constraints necessitate dynamic segmentation of the streaming activity data during prediction stage in addition to activity detection. This thesis presents a spotting method considering a group of constraints within the scope of intelligent healthcare domain in daily living settings utilizing wearable sensor technology. This thesis contributes to improving the commonly used segmentation approaches in the sense that the spotting technique introduced in this thesis addresses several constraints which have not been tackled in previous work.

Spotting the target activity in a sequence of activities and transitions without applying a pre-determined window size is a challenging task. The sliding window method, which is the typical approach in segmenting the continuous data stream, operates with optimal segment sizes chosen considering the type and duration of the activities. Nevertheless, activity type and duration are usually unknown to the detection unit in practice. In this thesis, we proposed Non-predetermined Size Windowing (NSW) scheme to spot the target activity performed in a sequence of unseen activities. NSW is built on classifying progress based features in multi-layer training and prediction stages where time domain progress is expressed in terms of polynomials. Thus, it operates without incorporating the information regarding duration and type of the activities. We verified our method with a use case where

data are acquired by a single wrist-worn 3D accelerometer. We compared our method against fixed size windowing performed with varying window sizes and feature extraction schemes; *windowed energy*, *peak frequency*, *Shannon entropy* and *wavelet entropy*.

Additionally, we propose an Artificial Data Generation (ADG) scheme and further investigate its contribution in monitoring the adherence of the patients to the elbow flexion and extension physiotherapy exercise routines in their daily environments. Performing detection of an activity in the existence of unseen classes is a case of classification where multiclass classification schemes are inadequate since in some use cases acquisition of only target activity data is possible before prediction stage. One-class classification (OCC) is a strategy to deal with such use cases. As an alternative to adapting the architecture of an existing classifier to design an OCC scheme, ADG approach emerges as a practical way of performing OCC. The ADG solution utilized in this thesis is evaluated in a setting with a single wrist-worn accelerometer where training data are collected for the target class only and data that represent all other classes are produced artificially from the target activity data. We analyzed the ADG method coupling with four different classifiers; KNN, SVM, logistic regression and Naïve Bayes classifiers.

Finally, we quantified the relation between activity duration, size of activity set and success ratio to provide an insight for selecting the methods which segmentation schemes are built on top of. In a real time execution environment, utilization of a segmentation algorithm leads to partial availability of activity data until data corresponding to a particular activity or segment are completely acquired, which brings about the problem of incompleteness of data. Hence, a segmentation scheme should be resilient to such incompleteness, which is incompleteness mitigation (IM) property in a knowledge discovery scheme. Quantification of relation between activity duration, size of activity set and success ratio is an indication of degree of IM. We developed an Intraclass Correlation Coefficient (ICC) based metric to measure degree of IM. We analyzed peak frequency and wavelet entropy feature extraction schemes with KNN in an iterative setting to observe their capability in mitigating incompleteness of the acceleration data in several hand oriented activities.

## ÖZET

### ORTAM DESTEKLİ YAŞAM İÇİN KESİNTİSİZ ATALET VERİSİ KULLANARAK İNSAN AKTİVİTELERİNİN TANINMASI

İleri yaş veya fiziksel engel sebebiyle hayatı zorlaşan bireylerin ihtiyaçlarını gidermekle görevli ortamsal bakım sistemlerinin vereceği uygun tepkinin belirlenmesinde aktivite tanıma ünitelerinin (ATÜ) önemli bir yeri vardır. ATÜ'nün günlük yaşam akışıyla bütünleşebilmesi için, yalnız aktivitelerin tanınması yetmemekte, aktivite tanıma probleminin yanında dikkate alınması gereken kısıtlar bulunmaktadır. Bu kısıtlar, tahmin fazı sürmekteyken kesintisiz akış halinde olan aktivite verisinin dinamik bölümlenmesini gerektirir. Bu sebeple, yalnız aktivitenin türünün algılanmasının yeterli olacağı durumlardan farklı olarak, aktivitenin hem türünün hem de kesintisiz veri akışı içinde hangi bölümde bulunduğu algılanması (spotting) ihtiyacı doğmuştur. Bu tezde, günlük yaşam koşulları altında kullanılacak akıllı sağlık uygulamalarının gerektireceği bir kısıt grubunun göz önünde bulundurulduğu ve giyilebilir sensör teknolojisi ile birleştirilmiş bir spotting yöntemi sunulmaktadır. Bu tez, önerilen spotting tekniğinin önceki çalışmalarda değinilmemiş çeşitli kısıtları gözetmesi bakımından, yaygın olarak kullanılan bölümleme yaklaşımlarını iyileştirmeye katkıda bulunmaktadır.

Pencere boyutu önceden belirlenmeksizin, art arda gelen aktiviteler ve bu aktiviteler arasındaki geçişlerden oluşmuş sinyal içinden hedef aktivitenin türünün ve ait olduğu bölümün belirlenmesi zorlu bir işlemdir. Kesintisiz şekilde akan veriyi bölümlendirmede karakteristik yaklaşım olan kayan pencere yöntemi, aktivitelerin türü ve süresi göz önünde bulundurularak seçilen en uygun bölüt büyüklüğünün bilinmesiyle çalışmaktadır. Böyle olmakla beraber, aktivite türü ve süresi, uygulama açısından bakıldığında genellikle algılama ünitesinde tanımlı değildir. Bu tezde, tahmin fazından önce karşılaşılmamış aktiviteler ve bu aktiviteler arasındaki geçişlerden oluşmuş bir dizi içinden hedef aktivitenin türü ve ait olduğu bölütün belirlendiği Öntanımsız Boyutlu Pencereleme (ÖBP) yöntemini önermekteyiz. ÖBP, özniteliklerin zaman kümesindeki ilerleyişinin polinomlar cinsinden ifadesiyle, bu ilerlemenin çok katmanlı eğitim ve tahmin aşamalarıyla sınıflandırılması üzerine kurulmuştur. Böylelikle, ÖBP yöntemi aktivitelerin türü ve süresi ile ilgili bilgiler

dahil edilmeksizin işlemektedir. ÖBP'yi, verinin bileğe giyilen tek bir 3B ivme sensörü ile toplandığı bir kullanım örneği üzerinde gerçekledik. ÖBP'yi, değişen pencere boyutları ile öznitelik çıkarma (ÖÇ) yöntemleri (pencereli enerji, tepe frekansı, Shannon entropi ve dalgacık entropi) ile gerçekleşmiş sabit boyutlu pencereleme ile karşılaştırdık.

Ayrıca, yapay veri üretimi (YVÜ) yaklaşımının, hastaların günlük yaşam ortamlarında dirsek bükme ve uzatma egzersiz programına uyumunun izlenmesine katkısını inceledik. Bazı kullanım örneklerinde, tahmin aşamasından önce sadece hedef aktiviteye ait verinin toplanabilmesi mümkündür. Bu sebeple, tahmin aşamasında daha önce karşılaşılmamış sınıfların da bulunduğu durumda bir aktivitenin algılanması, çok sınıflı sınıflandırma yöntemlerinin yetersiz kaldığı bir problemdir. Tek-sınıf sınıflandırma (TSS) yaklaşımı, bu tür kullanım örneklerini ele almak için uygun bir yöntemdir. Bir TSS yöntemi tasarlamada, sınıflandırıcının mimarisini uyarlamamanın bir alternatifi olarak YVÜ yaklaşımı, kullanışlı bir yöntem olarak ortaya çıkmaktadır. Bu tezde YVÜ çözümü, eğitim verisinin sadece hedef sınıf için toplanması, diğer tüm sınıfları temsil eden verinin ise hedef aktivitenin verisinden yapay olarak üretilmesiyle, bileğe giyilen tek bir ivme sensörü kullanılarak uygulandı. YVÜ yöntemini dört farklı sınıflandırıcı (K-En Yakın Komşu, Destek Vektör Makinaları, lojistik regresyon ve Naïve Bayes) ile birleştirerek analiz ettik.

Son olarak, aktivite süresi, aktivite kümesinin büyüklüğü ve başarı yüzdesi arasındaki ilişkiyi ölçmek üzere, bölümlleme algoritmalarının (BA) bileşenlerini seçme yöntemi geliştirdik. BA'nın gerçek zamanlı işlemesi durumu, belirli bir aktivite ya da bölüme karşılık gelen verinin tamamına erişilene kadar aktivite verisinin sadece bir kısmına erişilebiliyor olmasına sebep olarak verinin bütünlüğünün sağlanamaması problemini (VB) ortaya çıkarır. Bu sebepten, bir BA'nın VB'ye karşı dayanıklı olması gerekmektedir. Aktivitenin süresi, aktivite kümesinin büyüklüğü ve başarı yüzdesi arasındaki ilişkinin ölçüsü, VB durumuna karşı dayanıklılık derecesinin bir göstergesidir. VB'ye karşı dayanıklılığı ölçmek için Sınıf İçi Korelasyon temelli bir ölçü geliştirdik. ÖÇ yöntemleri olarak tepe frekansı ve dalgacık entropi ile K-En Yakın Komşu sınıflandırıcısının birleşiminin tekrarlı uygulanmasıyla, ivme verisinde VB'ye karşı dayanıklılığını çeşitli el odaklı aktivitelerde gösterdik.

## TABLE OF CONTENTS

|                                                                                                            |      |
|------------------------------------------------------------------------------------------------------------|------|
| ACKNOWLEDGEMENTS .....                                                                                     | iv   |
| ABSTRACT .....                                                                                             | v    |
| ÖZET .....                                                                                                 | vii  |
| LIST OF FIGURES .....                                                                                      | xi   |
| LIST OF TABLES .....                                                                                       | xii  |
| LIST OF SYMBOLS/ABBREVIATIONS .....                                                                        | xiii |
| 1. INTRODUCTION .....                                                                                      | 1    |
| 1.1. MOTIVATION AND AIMES .....                                                                            | 7    |
| 1.2. OUTLINE .....                                                                                         | 11   |
| 2. BACKGROUND AND RELATED WORK .....                                                                       | 13   |
| 2.1. FIXED SIZED AND DYNAMIC WINDOWING APPROACHES .....                                                    | 15   |
| 2.2. ONE-CLASS CLASSIFICATION .....                                                                        | 20   |
| 2.3. KNOWLEDGE DISCOVERY IN EXERCISE MONITORING .....                                                      | 21   |
| 3. PROPOSED APPROACH .....                                                                                 | 23   |
| 3.1. ACTIVITY SPOTTING .....                                                                               | 23   |
| 3.1.1. Training Phase .....                                                                                | 26   |
| 3.1.2. Processing Continuous Sensor Data with Iterative Prediction .....                                   | 29   |
| 3.2. ARTIFICIAL DATA GENERATION FOR REPRESENTATION OF<br>UNSEEN CLASSES WITH RESPECT TO TARGET CLASS ..... | 39   |
| 3.3. INCOMPLETENESS MITIGATION METRIC .....                                                                | 41   |
| 3.3.1. Intraclass Correlation Coefficient .....                                                            | 43   |
| 4. EXPERIMENTAL SETUP .....                                                                                | 45   |
| 4.1. COMPARED METHODS IN SPOTTING .....                                                                    | 47   |
| 4.1.1. Activity Annotation and Sensor Data .....                                                           | 51   |

|                                                                                |    |
|--------------------------------------------------------------------------------|----|
| 4.2. EXISTENCE OF UNSEEN CLASSES IN THE PREDICTION STAGE .....                 | 54 |
| 4.3. PARTIAL AVAILABILITY (INCOMPLETENESS) OF DATA WITHIN A<br>SEGMENT .....   | 59 |
| 5. RESULTS .....                                                               | 64 |
| 5.1. PARTITIONING CONTINUOUS SENSOR DATA AND CLASSIFYING<br>SEGMENTS .....     | 64 |
| 5.1.1. Evaluation Methodology .....                                            | 64 |
| 5.1.1.1. Determining Degree of Polynomial.....                                 | 70 |
| 5.1.2. Statistical Analysis .....                                              | 71 |
| 5.2. REPRESENTATION OF UNSEEN CLASSES WITH ARTIFICIAL DATA<br>GENERATION ..... | 74 |
| 5.3. INCOMPLETENESS MITIGATION RATE .....                                      | 78 |
| 6. CONCLUSIONS .....                                                           | 82 |
| GLOSSARY .....                                                                 | 85 |
| REFERENCES .....                                                               | 87 |

## LIST OF FIGURES

|             |                                                                                                    |    |
|-------------|----------------------------------------------------------------------------------------------------|----|
| Figure 2.1. | Sliding window approach commonly applied in processing continuous sensor data .....                | 15 |
| Figure 3.1. | Overview of the two-pass training stage .....                                                      | 23 |
| Figure 3.2. | Flowchart showing segment search in continuous data stream (Algorithm 3.6 and Algorithm 3.7) ..... | 25 |
| Figure 4.1. | Sensor data for training instances of target activity .....                                        | 53 |
| Figure 4.2. | Simple actions forming the medication intake composite activity performed by a subject .....       | 54 |
| Figure 4.3. | System architecture for monitoring adherence to physiotherapy exercises                            | 55 |
| Figure 4.4. | Acceleration signals corresponding to training data for the target class ..                        | 62 |
| Figure 4.5. | Elbow flexion and extension activities performed by a subject .....                                | 63 |
| Figure 5.1. | Trained on subject 1 and predicted on subject 1 .....                                              | 66 |
| Figure 5.2. | Trained on subject 2 and predicted on subject 2 .....                                              | 67 |
| Figure 5.3. | Trained on subject 1 and predicted on subject 2 .....                                              | 68 |
| Figure 5.4. | Trained on subject 2 and predicted on subject 1 .....                                              | 69 |
| Figure 5.5. | Trained on subject 3 and predicted on subject 3 .....                                              | 71 |
| Figure 5.6. | Trained on subject 4 and predicted on subject 4 .....                                              | 72 |
| Figure 5.7. | Successful detection rate of ADG on a per-subject basis .....                                      | 76 |
| Figure 5.8. | Successful detection rate of ADG across 2 subjects .....                                           | 77 |
| Figure 5.9. | <i>imr</i> for simple activities in subsets .....                                                  | 78 |

## LIST OF TABLES

|             |                                                                                  |    |
|-------------|----------------------------------------------------------------------------------|----|
| Table 4.1.  | Activity sequence including the exercise protocol and <i>other</i> actions ..... | 59 |
| Table 5.1.  | Abbreviations for the compared method names .....                                | 65 |
| Table 5.2.  | Methods surpassed by those in Fig. 5.1 .....                                     | 66 |
| Table 5.3.  | Methods surpassed by those in Fig. 5.2 .....                                     | 67 |
| Table 5.4.  | Methods surpassed by those in Fig. 5.3 .....                                     | 68 |
| Table 5.5.  | Methods surpassed by those in Fig. 5.4 .....                                     | 69 |
| Table 5.6.  | Methods surpassed by those in Fig. 5.5 .....                                     | 71 |
| Table 5.7.  | Methods surpassed by those in Fig. 5.6 .....                                     | 72 |
| Table 5.8.  | Degrees outperformed by optimal degree .....                                     | 73 |
| Table 5.9.  | Kolmogorov-Smirnov test results .....                                            | 73 |
| Table 5.10. | Mid-p-value McNemar test results .....                                           | 74 |
| Table 5.11. | Expected and actual TPR of compared methods .....                                | 74 |
| Table 5.12. | ICC for compared methods .....                                                   | 78 |
| Table 5.13. | <i>imr</i> ratings for peak frequency+KNN .....                                  | 79 |
| Table 5.14. | <i>imr</i> ratings for wavelet entropy+KNN .....                                 | 80 |
| Table 5.15. | Simple actions coexisting in the subsets .....                                   | 81 |

## LIST OF SYMBOLS/ABBREVIATIONS

|            |                                    |
|------------|------------------------------------|
| <i>AAL</i> | Ambient assisted living            |
| <i>ADG</i> | Artificial data generation         |
| <i>HAR</i> | Human activity recognition         |
| <i>ICC</i> | Intraclass correlation coefficient |
| <i>KNN</i> | K-nearest-neighbors                |
| <i>imr</i> | Incompleteness mitigation rate     |
| <i>NSW</i> | Non-predetermined size windowing   |
| <i>OCC</i> | One-class classification           |
| <i>SVM</i> | Support vector machines            |

## 1. INTRODUCTION

Healthcare applications can be enriched with intelligent systems to accentuate their capabilities [1–7]. Ambient assisted living (AAL) applications which emanate as a subdomain of intelligent systems in healthcare reside at the intersection of several technologies such as intelligent systems, data communication, sensors, and health informatics. AAL applications can serve as a procedure which complements the medication protocol prescribed by doctors. Other functions of AAL include preserving or improving health status of individuals supporting them in their daily routines. Hence, AAL can be regarded as a healthcare application.

AAL applications operate deploying ambient care systems in daily living environments. These systems are mainly composed of detection and actuation facilities. Detection modules include Human Activity Recognition (HAR) modules, which monitor the individuals so that actuation facilities can be triggered accordingly to deliver the intended service. The detection task in HAR is realized thanks to the cooperation between intelligent units, sensors, and data communication components.

HAR is extensively studied as a tool for improving health related services. Drivers can be monitored to detect heart attacks and avoid road accidents [8]. Patients who undergo heart attack cannot call the healthcare center due to unconsciousness. For such patients IoT enabled remote monitoring facilities can convey the emergency to healthcare center [9] so that they can be helped immediately. Elderly people whose health status tends to deteriorate due to aging can be assisted with HAR [10]. Remote physical therapy in home environment can be possible benefitting from a smartwatch and machine learning for limb rehabilitation [11]. COVID-19 outbreak increases the demand for remotely operating HAR systems to carry out in-home exercise tracking since people need to stay at home as a result of health related precautions. Elderly people with weakened health conditions or patients may have difficulty in performing domestic tasks at home. Service robots supported with HAR modules can provide assistance [12] with these tasks so that health problems they experience do not prevent them from leading an independent life. HAR can operate as an assistive

technology tool in facilitating interaction between elderly people and robots [13]. HAR has been studied for monitoring people who suffer from various diseases. [14] identifies dementia wandering as an early symptom. [15], [16] and [17] monitored patients with Parkinson's disease, diabetes and asthma respectively. [18] studies on detection of epileptic seizure. Sleep quality [19] and sleep position [20] are among other health related parameters which can be assessed with HAR. In addition to elderly people, activities of youth [21], children [22], newborns and infants [23] are studied as well. As opposed to the use cases of intelligent systems and HAR in healthcare illustrated above, a knowledge discovery module which is responsible for HAR task in AAL should provide a larger set of functionalities.

Intelligent modules collaborate with data communication infrastructures so that AAL and other healthcare applications can operate remotely. Mobile and wireless computing technologies empower patient monitoring facilities [24]. Wireless sensor networks are among data communication facilities enabling intelligent modules to be reachable in daily living settings in healthcare context as in Wireless Medical Sensor Networks [25]. Intelligent systems engaged in healthcare for undertaking HAR related tasks cooperate with sensing units for collecting human activity data. Wearable sensing devices coupled with IoT infrastructures [26] [27] increase the coverage of intelligent modules serving in healthcare. Wearable sensor units can collect and transmit human activity data in the form of several modalities as in accelerometer and gyroscope. Sensing devices alternative to wearable sensors such as environmental sensors also exist. Environmental sensing devices including passive infrared sensors and pressure mats may introduce limitations since they necessitate the users' interaction with the sensor. Cameras offer another option for data collection, however, their usage can raise privacy issues. Hence, wearable sensing units can be more preferable by the users which can be exemplified with adoption of watches, shoes and jewellery in data collection.

Human activity data collected by the sensors can be composed of a wide variety of activities. The types of the activities differ depending on age, health status, context of the people, and application domain. In human activity recognition studies, different types of activities have been analyzed including daily living activities (walking, sitting, standing, lying, eating, drinking, toothbrushing), domain specific activities as in sports [28], healthcare

and security [29]. The taxonomy of the activities includes different categorizations of the activities besides activity types mentioned above. Activities are also analyzed considering their position relative to each other in temporal domain, i.e. whether they are sequential, interleaving or concurrent, which is another categorization in the activity glossary. Activity composition is another facet further categorizing activities with respect to their inclusion of one another, regarding whether an activity is a part of another activity. According to composition point of view, activities can be referred as lower level or higher level with respect to each other.

The category of an activity influences activity recognition system architecture. If activities to recognize in a specific use case are low level in terms of composition, a less complex activity recognition module can be sufficient. Higher level activities, which compose of lower level activities make the recognition problem much more challenging since a higher level activity actually corresponds to a *sequence* of lower level activities, which should be detected individually. Moreover, the use case can also incorporate additional constraints in the recognition problem. If the time when a low level activity will be performed during the day is known beforehand as a specification of the use case, then the underlying IoT infrastructure can cooperate with the intelligent module in a synergistic manner, simplifying the recognition problem. The IoT architectures are capable of generating a timely notification which triggers the start of monitoring of a low level activity. In that case, the recognition task does not necessitate knowing what activities can precede or succeed the activity to detect. On the contrary, there exist use cases where the detection unit does not know when the activity will be performed. The recognition method which endeavors to recognize the activities within the framework of such use cases has to consider a larger set of constraints to tackle with the challenges added to the problem. Periodicity of activities introduces another activity categorization as well affecting the choice of activity recognition system architecture. As a result of the fact that a periodic activity has segments repeatedly occurring, probability of correctly detecting that subsegment is higher for HAR module. Even if a subsegment is misclassified, the activity of interest can be correctly identified in later subsegments of the same type, which is a factor relaxing the challenge introduced by constraints. If the activity is instantaneous in addition to being non-periodic, the chance of accurately identifying the activity becomes even lower.

Thanks to the variety of sensors which can be integrated to IoT architectures, numerous activity detection use cases can be defined attempting to solve daily life challenges. A significant portion of these use cases demands the recognition unit to be able to perform recognition task despite the lack of knowledge of when the activity will be performed. In elderly and patient care or providing ambient assistance to healthy people within the course of daily life, monitoring of adherence to medication intake or physiotherapy exercise routines opens up essential use cases. The related application scenarios lack in providing the information of when the activities will be performed during the day. In some applications, IoT infrastructure is desired to be in the monitoring state all day long to facilitate long-term continuous monitoring. Though this specification elevates energy consumption, in some monitoring tasks including fall detection continuous monitoring is a must. Some detection tasks can benefit from timely notifications received from IoT facilities in terms of shortening the period during which monitoring is necessary instead of monitoring for the whole day. However, the exact time when the people monitored will perform the activities upon receipt of the notification is not known although monitoring period is shorter than the whole day. Thus, absence of knowledge about when an activity will be performed is a significant constraint in IoT based daily human activity recognition. This constraint leads to a bottleneck for knowledge discovery schemes cooperating with IoT modules since the activities preceding and succeeding the activity of interest are not known in the context of a daily living environment.

Whether the monitoring scheme lasts for all day long or shorter intervals, continuous monitoring requires processing sensor data as a stream, which contains a sequence of activities. Detecting activities residing in a sequence embedded in streaming sensor data enforces the knowledge discovery scheme to undertake segmentation task as well. With the incorporation of this constraint, the recognition task turns into an *activity spotting* problem rather than an *activity detection* problem only where segmentation is not necessary. In activity detection tasks, the activity types become activity classes for which data are collected corresponding to training and prediction stages of the classification method. When the set of activities which can be encountered in the prediction stage are known before the training stage, training data are collected for that *known* set of activities. Considering applications in this profile, multi-class classification schemes have been extensively used

in activity recognition [30]. With the inclusion of the two constraints mentioned above, namely the lack of knowledge about when the activities will be performed and the necessity of undertaking segmentation task in addition to detection, the continuous data stream contains *unseen* activities in the prediction stage. Therefore, widely used multi-class classification architecture in the activity recognition domain is not adequate for the contexts where unseen activities are encountered in the prediction stage. In order to deal with this challenge, expressing multi-class classification problem in terms of a combination of one-class classifiers can be an option. Because in one-class classification (OCC), differentiating between a *target* and an *other* class is possible collecting training data for the target class only. Considering this aspect of OCC, recognizing an activity of interest existing in a sequence of activities where the sequence contains unseen activity classes encountered in the prediction stage becomes a problem which can be approached modelling the activity of interest as the *target* class where all the unseen classes are categorized as *other* class in OCC context.

Continuously streaming activity data processing algorithms in activity spotting domain face further constraints requiring pre-defined knowledge. In order to recognize the activities in a sequence, the sensor data containing the sequence are segmented so that activity labels corresponding to those segments are predicted. However, the optimal segment size should be known before applying segmentation. Since the sequence is formed of multiple activity types whose duration is different, determining a single segment size to utilize in segmentation is not sufficient. Moreover, same activity can be performed with different duration as well. Sensor sampling rate is another factor influencing the amount of data existing in a segment. For this reason, sampling rate should also be considered while determining the partitioning point between two adjacent segments as a parameter related to activity duration. The variety of possible duration values which the activities last for leads to even more complication due to the existence of unseen classes in the prediction stage. Without knowing what activity classes can be encountered in the prediction stage, the duration values corresponding to those activities cannot be known either, hence, the activity duration cannot be obtained as an information which can be fed to the knowledge discovery scheme. This constraint enforces to come up with a segmentation scheme which can operate without a pre-determined window size.

In real time classification of streaming sensor data, the contents of a segment are not available to the knowledge discovery unit in its entirety for some amount of time during which classification task should be in progress. Hence, decisions generated by a classification algorithm in a real time setting are based on partially available data, which are actually incomplete data. This problem demands incompleteness mitigation capability in a knowledge discovery scheme. A second source of incompleteness results from networking. Real time stream processing units are executed on data transmitted over a communication network, which could be a lossy transmission medium introducing another reason for incomplete data. In this type of incompleteness, unavailability of data is permanent as opposed to the other source mentioned above. In network related incompleteness, data can be lost as blocks or individual sensor readings. *Iterative* processing of continuous sensor data evaluates the influence of most recently transmitted sensor reading on detection result in consecutive iterations in spotting task. This approach enables recognition without embedding sensor sampling rate as an input to the knowledge discovery scheme. Furthermore, iterative processing of sensor data helps mitigation of incompleteness. If a knowledge discovery scheme can generate decisions based on such iterative processing, then partial availability problem can be eliminated mitigating incompleteness. Assessing incompleteness mitigation capability of knowledge discovery schemes in an iterative setting can provide insight for selecting the methods on top of which segmentation schemes are built.

Monitoring adherence to medication intake and physiotherapy exercise routines are among use cases of knowledge discovery in healthcare. Physical therapy includes various types of exercises which target strengthening different muscles. Similarly, medication intake activity could be defined differently depending on the type of drug consumed. For instance, the drug can be in the form of liquid, solid or cream as in syrup, pill or topical ointment respectively. Consumption of medication is performed in different ways for different forms of the drug. These different consumption styles bring about emergence of different versions of medication intake. Though all the versions semantically indicate the same activity, which is medication intake, they actually correspond to different activities from signal pattern point of view, i.e. "take pill", "take syrup" or "apply ointment".

## 1.1. MOTIVATION AND AIMES

HAR realizes knowledge discovery task thanks to the cooperation between segmentation, feature extraction and activity detection units. A continuous data stream is typically segmented utilizing a sliding window approach so that the type of activity embedded in that segment can be identified. Each segment is processed with feature extraction schemes and the generated features are input to activity detection approaches such as classification algorithms to determine the type of the activity.

Sliding window based methods necessitate determining an optimal segment size. The optimal segment size is chosen considering the type and duration of the activities, which results in demanding a specific sampling rate. However, activity type and duration are usually unknown to the detection unit in practice. In the context of daily living environments, people tend to pursue different daily routines. Therefore, the potential combinations of the activities, diversification in activity type and duration are hard to foresee. The sensor sampling rate is already influenced by hardware related constraints. If the detection unit requests a particular sensor sampling rate, the overall system design becomes even more constrained.

In addition to the drawback raised by imposing the optimal window size, the feature extraction schemes employed in sliding window schemes do not consider the progress of the features. Instead, they utilize time domain or frequency domain features without contribution of the progress information. However, it is known that, even if the activities are different, most of the time domain and frequency domain feature extraction schemes can yield the same features, such as in the case of mean, variance and energy [31].

In our work, we propose *Non-predetermined Size Windowing (NSW)* method which is a continuous activity data stream processing scheme without undertaking a pre-determined window size computation procedure. Our scheme is designed as a general framework to spot the target activity performed among several other activities and transitions in the sequence. The novelty of our approach is that we come up with a window accumulation method built on top of progress based features where time domain progress of polynomial

based features are classified in multi-layer training and prediction stages. Thus, the proposed scheme operates without incorporating the information regarding duration and type of the activities as well as sensor sampling rate.

The definitions of progress based features and polynomial based features are as follows: Since raw sensor data introduce variation along time dimension, the variation shows how raw sensor data progress in time. Hence, how data change in time can yield time domain progress. We embed time domain progress information into feature computation and call the newly generated features *progress based features*. We model the time domain progress information embedded in the features using polynomials mapping the time value to the feature value. Time value is represented by the sequence of the feature in the series of the features corresponding to raw data. We call the position of the features on the space created by the polynomial functions *polynomial based features*. Time domain progress is determined using distance relative to the position, hence, *progress based features* are expressed in terms of *polynomial based features*.

We attack the problem of locating the occurrence of target activity and categorizing the other activities and transitions between them as *other*. The variety of activities can be extensive in the environments where HAR are deployed, hence, the necessity of collecting training data for all possible activities and transitions introduces impracticality. In this regard, we collect training data for only the target activity in the sequence and we design our method injecting one-class classification capability. Since training data acquisition is performed for only the target activity, the only *seen* class in our model is the target activity which is in the form of a simple activity. Therefore, training dataset does not contain any sequence of simple activities or transitions between them. Instead, training dataset is composed of only the seen class. However, all of the simple activities and transitions in the sequence forming the prediction dataset except the target class are *unseen*. Consequently, training data acquisition is not performed for the unseen classes and the unseen classes are not known to the system.

Operating without a pre-determined window size is particularly crucial when unseen classes exist in the prediction sequence. Because the issue of pre-determining the optimal window size requires considering the unseen classes as well as the target class. However, optimal

window size computation should be achieved when training data corresponding to the unseen classes do not exist. Furthermore, optimum window size can be different for various unseen classes, which makes the issue even more difficult to deal with.

We designed NSW as an iterative scheme where the arrival of a new sample of sensor data introduces a new iteration. Since the samples arrive conforming to a temporal sequence, sensor data are processed as time series. In the prediction stage, the most recently arriving sample is accumulated in the window at each iteration and the window accumulation is terminated by the classification decision generated by our algorithm. If the layer-1 of prediction classified the window into the unstable state based on matching the time domain progress patterns, window accumulation stops. The window is exposed to a backwards traversal procedure in the successive layers of prediction to search for the segment which yields the label of the activity to spot. In the training stage, the patterns are generated with the help of polynomials representing time domain progress of the data, based on the information that data advance along time axis. In the one-class classification model we utilize, we generate artificial data to serve as the outlier class. In our continuous data stream processing scheme, outlier class represents all the simple activities and transitions in the set of activities under consideration except the target activity.

We verified our method with a proof-of-concept use case where data are acquired by a single wrist-worn 3D accelerometer. Studies demonstrating that increasing number of the sensors as well as employing different sensor modalities improves the success [32] [33] [34] [35] [36] [37]. [32] combines accelerometer and sEMG. [33] utilizes a combination of accelerometer, gyroscope, and magnetometer sensors to benefit from diversification of modalities in successful detection of activities. [34] evaluates activity recognition studies to reach a standard basis to compare studies in a common framework, mentioning studies utilizing a mixture of modalities. [35] indicates the use of multiple sensors in several studies as well. However, considering that incorporating more sensors deteriorates the practicality of the solution from the users' point of view, we present our solution with minimum number of sensors.

The contributions of this thesis are the following:

- We focused on segmentation problem with the following contents and our related article entitled "A segmentation scheme for knowledge discovery in human activity spotting" is accepted in *IEEE Transactions on Cybernetics* [38].
  - We propose NSW method, which is a continuous activity data stream processing scheme, for spotting target activity in a sequence of simple activities and transitions. NSW has the advantage of operating without a pre-determined window size and our method can substitute the sliding window approach. Elimination of a pre-determined window size is particularly important in the sense that a sliding window approach functions with the optimal window size. Also, the optimal window size is determined considering type and duration of the activities and transitions in the sequence as well as sampling rate of the sensor.
  - We design our NSW method as a one-class classification approach to increase the usability of windowing schemes in real conditions for spotting the target activity while unseen simple activities and transitions exist in the prediction sequence.
  - We propose progress based features to alleviate the potential insufficiency of time domain and frequency domain features in benefiting from the differences in the shape of signal while performing the spotting task mentioned above.
- We developed a practicable artificial data generation scheme (ADG) for recognition of an activity of interest in the existence of unseen classes in the prediction stage. With our ADG method, we integrate OCC mechanism to the knowledge discovery approach we put forward. Our proposed approach is published in our work entitled "Accelerometer based classification of elbow flexion and extension exercises" in 2020 *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* [39]. The contents of our work can be summarized as follows:
  - We propose a practical and non-invasive HAR solution which can be deployed in a daily environment for monitoring elbow flexion and extension exercises performed among a **sequence** of other daily activities that are unknown to the learning model.
  - We propose a practicable artificial data generation scheme (ADG) for our OCC approach. We train the system with the target exercise, where all the other daily activities in the sequence become **unseen classes**. Hence, recognition is

performed when **unseen classes** exist in the prediction stage.

- We developed an Intra-Class-Correlation (ICC) based metric to measure degree of incompleteness mitigation. The performance of a knowledge discovery scheme in iterative execution results in varying resilience to incompleteness. Developing a spotting algorithm based on accumulating windows benefits from iterative processing, necessitating increased incompleteness mitigation capability. Hence, our metric provides insights which aid development of a spotting scheme. Our proposed metric is published in our work entitled "*On the activity detection with incomplete acceleration data using iterative KNN classifier*" in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* [40]. The contents of our work can be summarized as follows:
  - We quantified the relation between activity duration, size of activity set and success ratio, where such quantification is an indication of degree of incompleteness mitigation so that an insight can be provided for selecting the methods which segmentation schemes are built on top of.
  - We investigated incompleteness mitigation capability on the acceleration data with an iterative setting in several hand oriented activities which constitute a medication intake use case.

We developed our knowledge discovery method using a single 3D accelerometer worn on the wrist. Though many HAR studies are based on data acquired with cameras, we did not include cameras in our study. Because monitoring people with cameras can raise privacy issues in daily settings for AAL applications.

## 1.2. OUTLINE

The rest of the thesis is organized as follows: Chapter 2 summarizes the related studies. Section 3.1 explains our spotting method and progress based features. Section 3.1.1 and Section 3.1.2 elaborates on the training phase and how continuous sensor data are processed with iterative prediction respectively. Section 3.2 mentions our ADG method for representing unseen classes based on target class. Section 3.3 presents our proposed incompleteness mitigation metric. Chapter 4 and 5 illustrate our experimental setup and

results, respectively. Finally Chapter 6 gives conclusions and future work.



## 2. BACKGROUND AND RELATED WORK

AAL applications assist individuals in daily living settings. These applications function benefitting from HAR modules to monitor assisted people. Such a monitoring procedure requires collecting data from sensors continuously and performing knowledge discovery on the acquired continuous data stream. A sequence of logically indivisible activities exists in the gathered sensor data. In the sequence, transitions reside between those activities as well. In terms of activity description, logically indivisible activities can be viewed as *simple activities* where combinations of simple activities and transitions build a composite activity. One of the main needs in knowledge discovery on continuous data stream is detecting defined target actions among the set of simple activities and transitions residing in the sequence. In this chapter, we mention components of knowledge discovery schemes utilized in accelerometer based HAR, namely classification, feature extraction and segmentation modules, giving the state of the art studies as well. In addition, we point out the distinction between fixed sized and dynamic windowing schemes as generally followed approaches for segmentation stage. We also indicate the role of different sensor modalities in HAR. Then, we explain OCC approach and incompleteness mitigation concept.

Knowledge discovery schemes in HAR are a combination of feature extraction, classification and segmentation methods. In typical knowledge discovery architecture utilized in HAR, raw sensor data are partitioned into segments. The segments are processed with feature extraction methods so that the contents of a segment are summarized in the specified feature. In classification stage, the features, which represent instances in machine learning point of view, are fed to a classifier to figure out the label corresponding to the feature tested.

The sensor type used is important in HAR in the sense that different sensor modalities can enrich the variety of application domains. However, accelerometers are the most dominantly used type of sensor, which emerges as an alternative to cameras considering privacy issue in HAR. [41] merges data obtained from accelerometer and gyroscope of smartphone with beacon support to also include indoor localization facet so that complex activities

can be recognized within AAL context. [42] studies on combination of accelerometer and gyroscope in HAR. [43] fuses wearable accelerometer data and ambient sensors within a smart kitchen environment. The number of sensors and sensor location are determined considering the type of activity.

Classification of activities embedded in acceleration data has been studied with various supervised or unsupervised approaches. [44] enhances classification with a distance based rejection mechanism on top of SVM and MLP. They merge acceleration with pressure data using a shoe tracker device. [45] processes acceleration data in an unsupervised fashion using Hidden Markov Model coupled with Expectation-Maximization. Data acquired with a single 3D accelerometer are classified with an extreme learning machine based method in [46]. [47] proposes an evolutionary classification approach for accelerometer and gyroscope data. [48] benefits from a probabilistic fusion of classifiers to process accelerometer data. In processing data acquired with accelerometers worn on the wrist, [49] performs classifier update based on uncertainty assessment.

Feature extraction in HAR with acceleration data has been performed utilizing individual or augmented features. Discrete Wavelet Transform [50], autoregressive coefficients [51], merging time-domain and frequency-domain features [52] and power spectral density [53] are among such approaches. [54] applies signal magnitude area and Kernel Discriminant Analysis in addition to autoregressive coefficients. Another study with combined features is [55] where the combination includes autoregressive coefficients in a hierarchy.

Continuous sensor data are first partitioned into segments so that feature extraction and classification methods can be applied on the segments. Generation of the segments is generally performed using a sliding windowing approach. In a window sliding scheme, the next segment is obtained with a window slide. There are two main categories of sliding windowing utilized in HAR for AAL which are fixed sized or dynamic windowing. These methods exhibit diversification about window size determination procedure utilized. The selected window size influences the location of start and end of a window. The main source of diversification results from the parameters used in window size determination schemes.

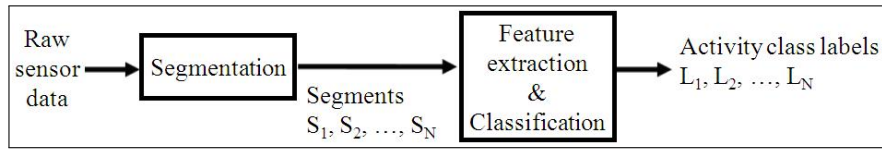


Figure 2.1: Sliding window approach commonly applied in processing continuous sensor data

## 2.1. FIXED SIZED AND DYNAMIC WINDOWING APPROACHES

In the commonly utilized approach for streaming data processing, a sliding window scheme is applied where the continuous data are first partitioned into windows and then these windows are classified into activity classes as given in Fig. 2.1. Several studies exist on fixed sized sliding windowing based activity detection [56–68]. The selection of window size considerably influences the recognition performance [69], which requires considering the type and duration of the activities as well as the sensor sampling rate [40] [69–72]. Thus, finding the correct segment size emerges as a bottleneck in segmentation algorithms while processing streaming data [70] [73–75].

Though some recent studies propose several dynamic windowing approaches, these approaches do not include contributions of our work. We categorize dynamic windowing approaches according to the following criteria given with their description:

- Transition:** The compared method embodies *transition* property if the *transitions between the activities* are fed to the knowledge discovery scheme as an input. In employing the transitions as input, the continuous data stream is first segmented and the resulting windows are utilized in training stage as well as prediction stage, hence the patterns generated in the learning scheme are influenced by the transitions between the activities. With such an approach, recognition success might benefit from the fact that the order of actions is the same in the data stream utilized in both training and prediction stages. In that case, the order of actions, which will be performed in real deployment conditions, should also be known while training the module.
- Fully dynamic:** Utilizing a pre-determined window size, feeding the knowledge

discovery scheme with the *activity duration* as input, or not providing dynamic reduction of window size prevents a method from being *fully dynamic*.

- **One-class:** The method satisfies *OCC (one-class classification)* property if the segmentation strategy facilitates spotting the target activity where unseen classes only exist in the prediction stage with training data collected for only the target class.
- **Progress based features:** *Progress based features* facet exists in a method if the features are extracted considering how the data corresponding to an activity progress during the period in which the activity is performed.

Some recent studies also addressed the window size selection issue in their models. [76] performs online data stream processing for anomaly detection with a sliding window based approach and probability density estimation. The work in [77] benefits from the usage of Multivariate Gaussian Distribution for determining the size of the window corresponding to an activity. In [78], the performance of the proposed model is analyzed using different fixed window size values based on the acceleration data. In [79], the authors search for the transition points based on different window sizes. In [80], the authors present a transition detection method based on standard deviation trend. The work in [81] studies how overlapping sliding windows affect the success of activity detection. In [82], the authors use overlapped sliding windowing to detect instantaneous activities with convolutional neural network and fuzzy logic on smartphone data acquired from six sensors and [83] puts forward a change point detection strategy using Information Gain. The aforementioned studies on window size selection problem lack the fully dynamic property in their segmentation approach.

Studies in [84–86] use a single wearable 3D accelerometer sensor in their setup similar to our work, whereas the study in [87] incorporates the use of multiple sensors.

[84] incorporate an adaptive windowing scheme on condition that the activity is cyclic. For acyclic activities, they perform fixed time windowing. In their adaptive windowing scheme, they need to determine the period of cyclic activities performing autocorrelation over a segment. They determine the size of these segments knowing the period of the activities existing in the activity set. Hence, their method requires activity duration to be an input for

adaptive windowing. They compute mean, energy and frequency domain entropy along axes and Pearson product-moment correlation as the features. They feed the partitioned sensor data as the training data, hence patterns generated in their method are not free of transitions and the sequence of simple actions.

[85] process continuous sensor data with an adaptive sliding window for transitional activities and fixed sliding window for non-transitional activities. The features they extracted are average, standard deviation, skewness, signal magnitude area, slope, absolute slope, spectral energy, mean trend, windowed mean difference, maximum and minimum. In their adaptive sliding window strategy, they determine the initial window size based on activity type and activity duration information. The amount of window overlap and expansion are a function of initial window size. Hence, activity type and activity duration are input in their windowing strategy. Their system is trained with segments obtained from continuous data, thus patterns generated in their method are not free of transitions and the sequence of simple actions. Dynamic reduction of window size does not exist in their method.

Time-domain and frequency-domain features such as mean, root mean squared, standard deviation, median absolute deviation, range, signal magnitude area, correlation coefficient, tilt angle, spectral energy, spectral entropy, skewness, kurtosis, largest frequency component, and frequency signal weighted average are used in [86]. Their segmentation strategy is not free of applying a pre-determined window size, since the subsegments they process are assigned a pre-determined size of 256 samples. Hence, their windowing strategy is not fully dynamic. Instead, they identify change points dynamically and repartition the data between two consecutive change points into windows of a pre-determined size.

The dense labelling approach proposed in [87] attacks the continuous data processing problem considering each sample individually as we do in our study, however there are major differences between their work and ours. Firstly, their proposed approach does not employ a window accumulation scheme, instead they process subsequences with size  $T$  and utilize a fixed window overlap, which forces their method to still operate with the help of a pre-determined window size. Hence, their method is not fully dynamic. They test their

method with subsequences of size 24, 100, 500, 1000 and 3000 to show that their method is not sensitive to the subsequence size. Nevertheless, in their experiments they train the system with a single subsequence length. They do not present an analysis related to the performance of their method when the subsequence length utilized in the training stage changes. A pre-determined subsequence length employed in the training stage is also a pre-determined window size. According to their tests, their method is not sensitive to the varying overlapping rate, either. However, their experiments do not include an analysis related to the effect of overlapping rate as the subsequence size in training stage changes. Also, their proposed segmentation strategy does not attack the problem within OCC context and the features they utilize are not progress based features. In terms of the *transition* property, their method employs the transitions as input.

[88] proposes a windowing approach where sensor activations arriving after the moment where decision should be generated also participate in determining the decision. They applied their method on binary sensor data using Convolutional Neural Network and Long Short-Term Memory. Their proposed approach benefits from associating binary sensors with objects in the daily living environment and they utilize the window size of 60 s on the data they acquired with 12 sensors, hence their approach is not equivalent to our method.

[89] puts forward a spotting method to detect eating and drinking based on data acquired with accelerometer and gyroscope. In another study carried out on accelerometer and gyroscope data, [90] recognized walking, walking downstairs, walking upstairs, standing, sitting and lying down employing long short-term memory neural networks. [91] also studies on recognition of eating and drinking with accelerometer and gyroscope data. The approaches proposed in [89] - [91] do not address the contributions of our work either.

In the approach utilized by [92], there are two prediction stages, the first of which determines the window boundaries and the second of which determines the activities corresponding to those windows. On the other hand in our method, determination of the boundaries of a segment and identification of the label corresponding to that segment take place in a coalesced manner in the same prediction stage, which is the single prediction stage. Their method fixes all the window sizes prior to moving to the prediction stage where activity

classification occurs, hence their method is also a variant of fixed sized sliding windowing, enforcing a predetermined window size. Their window size determination procedure is not as dynamic as ours because in their method, after detection of activities corresponding to windows starts, the window size values which are already computed before do not change anymore. When prediction of activities needs to be carried out in real time, their window size determination procedure has to be carried out in an offline prediction stage instead of a real time prediction stage as opposed to our approach. Hence, their approach is not fully dynamic. Their work does not include *transition*, *one-class* and *progress based features* properties we described above either, lacking in providing the contributions of our study.

The following activity recognition studies have a segmentation focus on skeleton data. In [93], the authors propose the Multi-stage Adaptive Regression approach to target the problem of partial observation of activities which leads to difficulties in online activity recognition based on video streams. In [94], the authors employ a novel segmentation approach for partitioning sequences of unlabelled activities on skeleton data using a Long Short-Term Memory Recurrent Neural Network. Similarly, [95] and [96] also study on activity recognition with segmentation focus. The method given in [97] processes the sequence using graphical models on skeleton data while [98] detects complex activities with a fusion method merging grayscale and depth data. In [99], the authors use skeleton data to detect postures while also collecting RGB data to help define temporal information. The work in [100] and [101] also uses skeleton data whereas in [102] human action recognition techniques concentrating on RGB-D data are analyzed. In [103] and [104], the authors focus on spotting hand gestures. In [103], they tackle the problem using nearest neighbour with maximum cosine similarity. They focus on data with digits and letters in terms of trajectory classification. [104] utilizes EMG and acceleration data. Neither of these studies address transition property as well as fully dynamic properties.

In overall, HAR studies utilized an extensive list of time-domain and frequency-domain features in their models; such as mean, energy, frequency domain entropy along axes, Pearson product-moment correlation, root mean squared, standard deviation, median absolute deviation, range, signal magnitude area, correlation coefficient, tilt angle, spectral energy, spectral entropy, skewness, kurtosis, largest frequency component and frequency

signal weighted average. Nevertheless, these features are not extracted considering how the data corresponding to an activity progress during the period in which the activity is performed differing from our novel feature extraction model.

## 2.2. ONE-CLASS CLASSIFICATION

When the training data for all of the classes which can be encountered in the prediction phase are not available, the classification problem can be attacked with one-class classification approach [105]. The capability to mitigate scarcity of data makes OCC a suitable option for monitoring of physical activities. Tracking the activities in daily living environments introduces a wide variety of activity classes which can be encountered throughout the course of daily routines. Extensive diversification of activity types is an unavoidable constraint in AAL.

In the literature, OCC has been vastly studied utilizing SVM based architectures. A taxonomy of one-class classification methods considering whether they are built on top of Support Vector Machines [106] [107] or not is given in [108]. [109] develops an approach on top of one-class SVM put forward by Schölkopf [110] with tuned parameters for recognizing erroneous actions where normal and erroneous samples are utilized for training and testing respectively. In [111] one-class Support Vector Machine processes features where autoencoders participate in dimension reduction. [112] combines one-class Support Vector Machine and Convolutional Autoencoder in increasing compactness of clusters. [113] comes up with a KNN based one-class classification approach considering average distance. [114] puts forward a one-class classification scheme derived from Extreme Learning Machines, where they focused on minimization of discrepancy in training. Generative adversarial nets [115] and convolutional neural network [116] [117] [118] [119] [120] are among other methods integrated to OCC approach. [121] evaluates the performance of several methods built as one-class classifiers where parameters of these approaches are determined with Particle Swarm Optimization. [122] proposes a one-class classification algorithm utilizing convex hull to generate target class boundary.

As an alternative to adapting the architecture of a classifier for injecting one-class detection

capability, producing artificial data to represent the negative class is a potential approach, which enables employing any classification method to perform one-class classification [105]. Synthetic data generation can be performed utilizing distribution properties of data. In [123], probability distribution is determined based on real data using generative adversarial networks to obtain synthetic data. In [124], negative data are produced with an autoencoder based distribution learning scheme. [125] presents a method for hyperparameter selection of one-class SVM in an automated way.

Ensemble learning is preferred in several other studies as well for applying OCC. Emerging as another solution to mitigate the absence of negative examples in OCC, [126] proposes a method to build one-class ensembles. In [127], the classification problem is conceptualized as a combination of multiple OCC models each of which is responsible for a subgroup of samples with similar density. A weight assigning procedure is used in [128] to obtain an improved group of OCC models. [129] employs multi-objective optimization in finding the group of models. Normalization of model output for combination procedure is studied in [130].

One-class classification approaches where architecture of an existing classifier is adapted are known to usually necessitate manual selection of values corresponding to a parameter in the training phase [131], which is a demanding problem even in the current status. Considering that different contexts to monitor might already require customization on classification architectures, we chose generating artificial data to represent the negative class in our one-class approach instead of choosing an already existing one-class classifier so that we can come up with a more generic monitoring framework where the classification algorithm to incorporate can be selected more freely as the context-dependant variations arise.

### **2.3. KNOWLEDGE DISCOVERY IN EXERCISE MONITORING**

Supporting rehabilitation related tasks with recognition systems including sensing units and machine learning is an active research field. [132] measures elbow spasticity with machine learning. [133] studies on recognition of exercises with a convolutional neural network based method. [134] utilizes a pair of IMU sensors for rehabilitation. [135] presents a

convolutional neural network based method for sports rehabilitation. [136] processes sEMG data in detecting upper limb motion utilizing convolution neural network.

Monitoring elbow flexion and extension is frequently studied utilizing EMG and sEMG data. In [137] Wavelet Packet Transform (WPT) is used as the feature extraction scheme to obtain final activity mappings with the least square method. In [138], utilization of multi-scale entropy to detect elbow flexion and extension based on sEMG signals is investigated. In this study, sEMG data collected with a sampling rate of 1000 Hz is processed using Artificial Neural Network (ANN). [139] determines kinematic properties regarding the elbow flexion and extension actions processing EMG data. They investigate mean absolute value, mean value, entropy, harmonic mean and mean frequency as time and frequency domain features. [140] studies on elbow flexion activity using smart piezo-resistive textile sensor attached on the sweater and deep neural network (DNN). Their method recognize the joint angle of the elbow results in an accuracy in the range 76.9%-83.66%. Neither of the aforementioned studies incorporates the ability to detect elbow flexion and extension when unseen daily activities exist in the prediction stage. Besides, data acquisition performed with EMG or sEMG requires high sampling rate deteriorating practicability in the context of the network transmission.

### 3. PROPOSED APPROACH

The knowledge discovery scheme we propose for the activity spotting task in this thesis is composed of a segmentation module accompanied by an ADG module to fulfil OCC mechanism. In this chapter, we elaborate on these modules. We also explain our proposed incompleteness mitigation metric in more detail.

#### 3.1. ACTIVITY SPOTTING

In our spotting method, continuous sensor data are composed of a sequence of simple actions, which introduce a logically indivisible unit in an application scenario. While the prediction stage is performed on continuous sensor data, training data are formed of only simple actions. Neither the sequence of simple activities nor the transitions between them exist in training data to reduce the dependency of the performance on the activity types studied. Since the raw sensor data stream comprises the time axis, we process the activity data as time series. In both training and prediction phases, sensor data are represented as a matrix where row number embeds time domain information into the model. Each row indicates a sensor event and the row number designates the coordinate of the related sensor event in time domain such that data at row  $i$  arrive right before data at row  $i + 1$ .

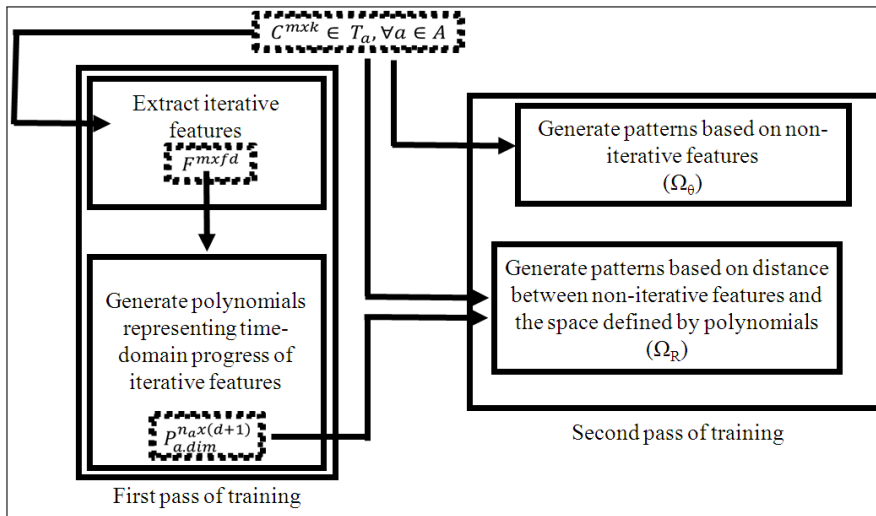


Figure 3.1: Overview of the two-pass training stage

Classification of features with time-domain progress information embedded is performed thanks to a two-pass training phase, which is summarized in the block diagram given in Fig. 3.1. In Fig. 3.1, the matrix  $C^{mxk}$  represents the time series data with  $m$  rows and  $k$  columns. Training data in the form of  $C^{mxk}$  for all simple actions are fed to the modules specified in Fig. 3.1. If the features are extracted utilizing part of a sample  $C^{mxk}$ , the features are iterative. Conversely, if the feature extraction is performed on the entirety of  $C^{mxk}$ , then the features are non-iterative. In the first pass of training, iterative features are extracted given the raw training data, yielding the feature matrix  $F^{mxf^d}$ . Polynomials representing time domain progress of iterative features are generated based on  $F^{mxf^d}$ , outputting  $P_{a.dim}^{n_{ax}(d+1)}$ , which finalizes first pass of training. In the second pass of training, patterns  $\Omega_\Theta$  and  $\Omega_R$  are generated. Raw training data are processed to yield  $\Omega_\Theta$  based on non-iterative features. Distance between non-iterative features, which are computed based on raw training data, and the space defined by polynomials  $P_{a.dim}^{n_{ax}(d+1)}$  is found to generate  $\Omega_R$  patterns. The continuous sensor data are processed to determine a sequence of segments using a 3-layer prediction stage with a window accumulation scheme. Each new sensor reading  $C^{mxk}(i, :)$  introduces a new iteration  $i$  of accumulation. The window accumulation scheme is coalesced with a layered decision generation mechanism. With the cooperation of the decisions in layers, the continuous sensor data are partitioned into the segments and activity classes corresponding to those segments are identified. The extracted segments and the classes assigned to the segments are the outputs of our proposed spotting method. The value of  $\gamma$  signals the point of partitioning when the activity class belonging to  $\Phi^{\gamma xk}$  is labelled. The accumulation and accompanying layered decision generation schemes proceed until the entire continuous sensor data signal  $C^{mxk}$  is processed, which is marked as the final iteration  $m$ . Due to the sequential relationship between iterations, the accumulation procedure advances along time axis  $t$ . The output segments have an intersection with the actual activities existing in the continuous data, enabling activity spotting.

Fig. 3.2 summarizes the 3-layer prediction stage where the output segments are generated as well as their labels. Data are accumulated in window  $\Phi^{\gamma xk}$  until the unstable state is reached. When to stop data accumulation in  $\Phi^{\gamma xk}$  is determined as the continuous data processing is in progress, hence we eliminate determination of a pre-determined optimal window size. As the next sensor events are accumulated in  $\Phi^{\gamma xk}$ , feature vector  $F^{fd}$  is extracted. The distance

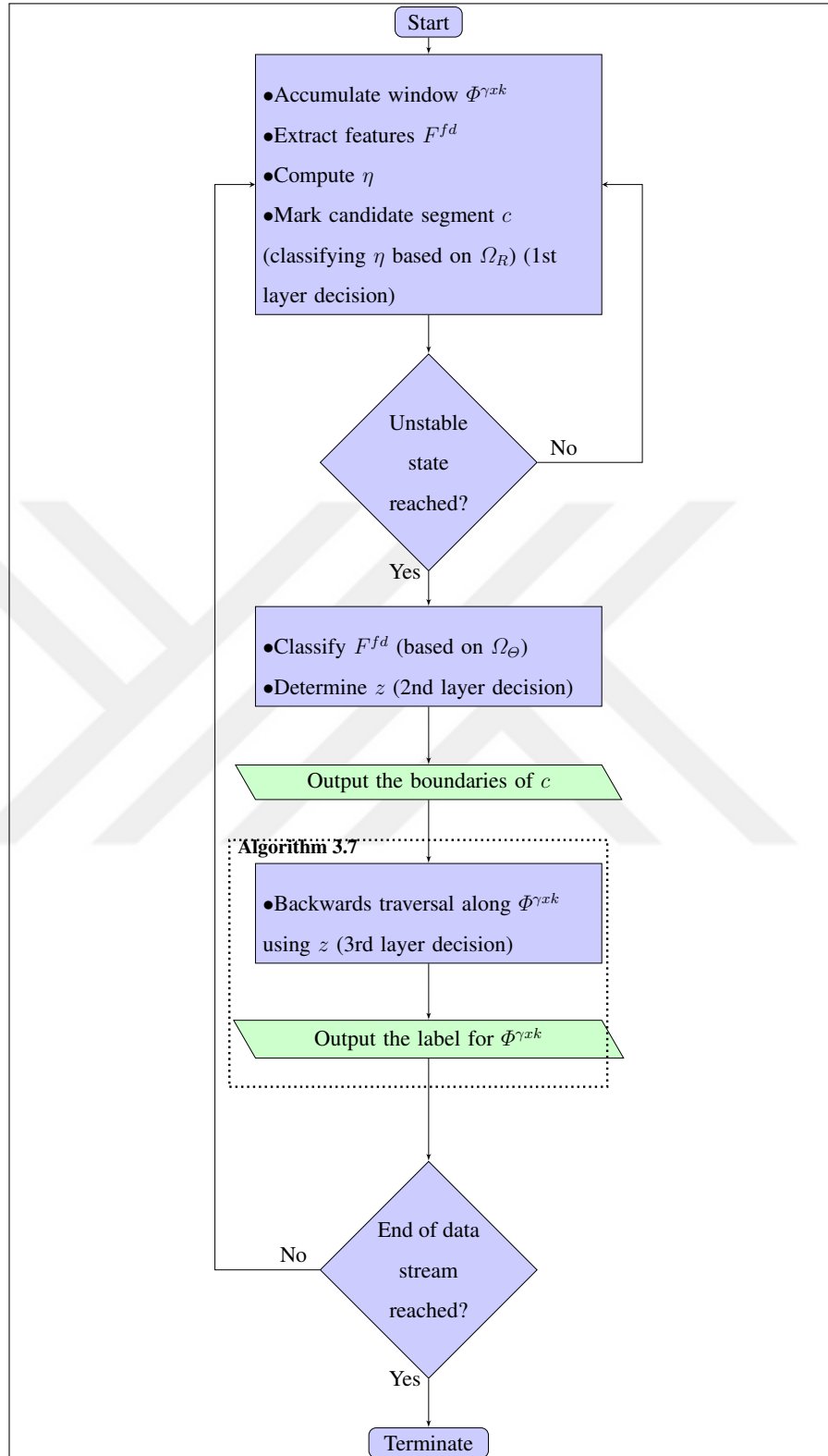


Figure 3.2: Flowchart showing segment search in continuous data stream (Algorithm 3.6 and Algorithm 3.7)

$\eta$  between  $F^{fd}$  and  $P_{a.dim}^{n_a x(d+1)}$ , which are the polynomials output in the training stage, is computed. Candidate segment  $c$  is marked classifying  $\eta$  based on  $\Omega_R$  in the prediction stage of layer 1. If the unstable state is reached, the boundaries of the candidate segment are determined. In prediction of layer 2,  $F^{fd}$  is classified based on  $\Omega_\theta$  to generate the decision  $z$ . Subsegments of  $\Phi^{\gamma xk}$  are searched utilizing  $z$  based on  $\Omega_\theta$  as the 3rd layer of prediction, generating the label for the segment, which corresponds to  $\Phi^{\gamma xk}$ . The algorithm goes on with accumulating a new window, following the same procedure until continuous sensor data are exhausted. Detailed explanations of the components in Fig. 3.1 and 3.2 are given in Algorithm 3.1- 3.5 and Eq 3.3- 3.6. The notations used in these algorithms and equations are described in the glossary.

$$C^{m \times n} \cup B^n = E^{(m+1) \times n} \text{ such that}$$

$$E(i, :) = \begin{cases} C(i, :), \forall 1 \leq i \leq m \\ B^n, i = m + 1 \end{cases} \quad (3.1)$$

### 3.1.1. Training Phase

Algorithm 3.1 elaborates on how polynomial coefficient matrix is generated for each attribute of all simple actions given set of training samples.  $C^{m \times k}$ , the raw acceleration data matrix for a training sample belonging to a simple action is exposed to *NPFE* iteratively. The feature matrix  $F^{m \times fd}$  is populated, such that a separate row is computed at each iteration applying *NPFE* on the partial data matrix  $M$ . Prior to extracting features for current iteration with *NPFE*,  $M$  is constructed applying  $\cup$  operator on  $C^{m \times k}(i, :)$ . After feature matrix formation is complete, polynomials of degree  $d$  are fit to generate the mapping between the feature vectors, which constitute the rows of the feature matrix and their sequence number. A separate polynomial is fit for each column of the feature matrix, denoted by  $F^{m \times fd}(:, dim)$ . Finally, each polynomial coefficient vector generated with the *fit* operation to represent a  $C^{m \times k}$  is stored in the corresponding row of  $P_{a.dim}^{n_a x(d+1)}$ .  $F^{m \times fd}(i, :)$  is found extracting features from  $M$  and  $M$  contains the contents of  $C^{m \times k}$ .

up to iteration  $i$ . Therefore,  $F^{mxfd}(i+1, :)$  repeatedly encodes the information stored in  $F^{mxfd}(i, :)$  and we call the features accumulated in  $F^{mxfd}$  *iterative features*. Consequently,  $P_{a.dim}^{nax(d+1)}$  models time domain progress of iterative features.

Algorithm 3.1. firstPassOfTraining()

```

for  $\forall a \in A$  do
  for  $C^{mxk} \in T_a$  do // iterates  $n_a$  times
     $M \leftarrow \emptyset, j \leftarrow 1$ 
    for  $\forall 1 \leq i \leq m$  do
       $M \leftarrow M \cup C^{mxk}(i, :)$ 
       $F^{mxfd}(i, :) \leftarrow NPFE(M)$ 
    end for
     $P_{a.dim}^{nax(d+1)}(j, :) \leftarrow \text{fit}([1:m], F^{mxfd}(:, dim), d), \forall 1 \leq dim \leq fd$ 
     $j \leftarrow j + 1$ 
  end for
end for

```

Eq 3.2 shows how the representative polynomial  $PV^{(d+1)}$  is evaluated for the sequence number  $seq$  and degree  $d$ .

$$eval(PV^{(d+1)}, seq, d) = \sum_{i=1}^{d+1} PV_{i-1}^{(d+1)} seq^{i-1} \quad (3.2)$$

Distance computation is explained in Algorithm 3.2. For each attribute,  $DS_i^{fd}.coef$  is evaluated with the function *eval* as in Eq 3.2.  $DS_i^{fd}.coef$  represents a polynomial which maps the sequence number  $seq$  to  $B_i^{fd}$ . For each attribute, the distance between  $B_i^{fd}$  and  $V_i^{fd}$  is computed and stored in  $I_i^{fd}$  using modulo operation. Prior to modulo operation computed for non-zero values of  $V_i^{fd}$ , both  $B_i^{fd}$  and  $V_i^{fd}$  are truncated as designated by the *trunc* operation, then their absolute values are calculated. According to the formal definition of modular arithmetic, modulo, which is  $|trunc(V_i^{fd})|$  in Algorithm 3.2, should be a positive

natural number. When  $V_i^{fd}$  is zero, the modulo  $|trunc(V_i^{fd})|$  also becomes zero, which leads to an undefined case for modular arithmetic. In order to remove this undefined case,  $I_i^{fd}$  is set to zero when  $V_i^{fd}$  is zero. After distance values for all attributes are computed and stored in  $I^{fd}$ ,  $|\vec{I}^{fd}|$  is computed summing the square of the distance values for all attributes, then finding the squareroot of the sum.  $|\vec{I}^{fd}|$  is returned as the distance between  $B^{fd}$  and  $V^{fd}$ .

$$\forall \alpha, \beta, (r(\alpha) = r(\beta) \implies \alpha = \beta)$$

(3.3)

$$r(a) = Q_i^{|A|}, \forall a \in A$$

Algorithm 3.2.  $dist(V^{fd}, seq, DS^{fd})$

```

for  $\forall 1 \leq i \leq fd$  do
   $B_i^{fd} \leftarrow eval(DS_i^{fd}.coef, seq, DS_i^{fd}.deg)$ 
  if  $V_i^{fd} \neq 0$  then
     $I_i^{fd} \leftarrow |trunc(B_i^{fd})| \% |trunc(V_i^{fd})|$ 
  else
     $I_i^{fd} \leftarrow 0$ 
  end if
end for
 $|\vec{I}^{fd}| \leftarrow \sqrt{\sum_{i=1}^{fd} (I_i^{fd})^2}$ 
return  $|\vec{I}^{fd}|$ 

```

The operation  $r$  given in Eq 3.3, being a one-to-one function, maps each simple action  $a$  to the unique code  $Q_i^{|A|}$ .

In Algorithm 3.3, distance between polynomial coefficient matrix and set of training samples is computed. Each polynomial stored in a row of a polynomial coefficient matrix is fetched

to populate the map structure of polynomials. Total number of training samples collected for each action is computed and assigned to  $\beta$ . Then, for each simple action in the set, the unique code corresponding to simple activity  $a$  is obtained with the  $r$  function explained in Eq 3.3 and stored in  $B_2^2$ , which will be the label of the instance stored in  $B_1^2$ . For each element of  $T_a$ , features  $F^{fd}$  are computed. Utilizing the operation  $dist$  given in Algorithm 3.2,  $m$  is mapped to the space determined by the polynomials stored in  $MS_{a,j}^{n_a}$ . The vector which represents  $m$  in this space is compared against  $F^{fd}$  to compute the distance between these two vectors. The resulting distance value is stored in the corresponding element of  $B^2$ . Then,  $R^{\beta x^2}$  is appended by  $B^2$ . For each element of  $T_a$ , distance computation and expansion of  $R^{\beta x^2}$  are repeated for all  $MS_{a,j}^{n_a}$ . Polynomials stored in  $MS_{a,j}^{n_a}$  represent time domain progress of iterative features as explained in Algorithm 3.1.  $F^{fd}$  used in distance computation with  $dist$  in Algorithm 3.3 represents the features corresponding to the entirety of  $C^{mxk}$  instead of the contents of  $C^{mxk}$  up to some iteration  $i < m$ . The vector to which  $m$  is mapped using  $MS_{a,j}^{n_a}$  in  $dist$  algorithm indicates the virtual features corresponding to the ending point expected by  $MS_{a,j}^{n_a}$  whereas  $F^{fd}$  symbolizes the actual ending point of  $C^{mxk}$ . Thus,  $R^{\beta x^2}$  is a measure of the variations between actual and virtual ending points of the training data. The other meta-training data matrix output by the second pass of training stage is  $\Theta^{\beta x(fd+1)}$  and the feature vectors forming  $\Theta^{\beta x(fd+1)}$  represent actual ending points of the training data. In Algorithm 3.3, after formation of  $R^{\beta x^2}$  is complete,  $\Theta^{\beta x(fd+1)}$  is populated. Features  $F^{fd}$  are extracted for each  $C^{mxk}$ , then, the vector  $\Pi^{(fd+1)}$  is constructed such that  $\Pi^{(fd+1)}$  contains the elements of  $F^{fd}$  as well as the label identifying the instance stored in  $F^{fd}$ . The label assigned to  $\Pi_{fd+1}^{(fd+1)}$  is computed using the  $r$  function as performed in the population of  $R^{\beta x^2}$  previously. Then,  $\Theta^{\beta x(fd+1)}$  is appended by  $\Pi^{(fd+1)}$ . After expansion of  $\Theta^{\beta x(fd+1)}$  is repeated for  $C^{mxk} \in T_a, \forall a \in A$ ,  $cls\_training$  is applied on  $R^{\beta x^2}$  and  $\Theta^{\beta x(fd+1)}$  to output patterns  $\Omega_R$  and  $\Omega_\Theta$  respectively, finalizing second pass of training stage.

### 3.1.2. Processing Continuous Sensor Data with Iterative Prediction

Our method outputs a series of segments where each segment  $\Phi^{\gamma xk}$  is assigned a class label  $l$ . The output segments indicate that time domain progress of the signal within  $\Phi^{\gamma xk}$  is observed to be similar to that of the signals with label  $l$  utilized in the training stage.

## Algorithm 3.3. secondPassOfTraining()

```

//Populate  $MS_a^{n_a}$ :
for  $\forall a \in A$  do
  for  $\forall 1 \leq j \leq n_a$  do
    for  $\forall 1 \leq dim \leq fd$  do
       $DS_{dim}^{fd}.coef \leftarrow P_{a.dim}^{n_a x (d+1)}(j, :), DS_{dim}^{fd}.deg \leftarrow d$ 
    end for
     $MS_{a.j}^{n_a} \leftarrow DS^{fd}$ 
  end for
end for
 $\beta \leftarrow \sum_{i=1}^{|A|} (n_a)_i, R^{\beta x 2} \leftarrow \emptyset$ 
for  $\forall a \in A$  do
   $B_2^2 \leftarrow r(a)$ 
  for  $C^{mxk} \in T_a$  do
     $F^{fd} \leftarrow NPFE(C^{mxk})$ 
    for  $\forall 1 \leq j \leq n_a$  do
       $B_1^2 \leftarrow dist(F^{fd}, m, MS_{a.j}^{n_a}), R^{\beta x 2} \leftarrow R^{\beta x 2} \cup B^2$ 
    end for
  end for
end for
 $\Theta^{\beta x (fd+1)} \leftarrow \emptyset$ 
for  $\forall a \in A$  do
   $\Pi_{fd+1}^{1x(fd+1)} \leftarrow r(a)$ 
  for  $C^{mxk} \in T_a$  do
     $F^{fd} \leftarrow NPFE(C^{mxk})$ 
     $\Pi_i^{(fd+1)} \leftarrow F_i^{fd}, \forall 1 \leq i \leq fd$ 
     $\Theta^{\beta x (fd+1)} \leftarrow \Theta^{\beta x (fd+1)} \cup \Pi^{(fd+1)}$ 
  end for
end for
 $\Omega_R \leftarrow cls\_training(R^{\beta x 2})$ 
 $\Omega_\Theta \leftarrow cls\_training(\Theta^{\beta x (fd+1)})$ 

```

Detected similarity between time domain progress patterns yield that an output segment has an intersection with the activity of label  $l$ . The intersection is the portion where the activity of class  $l$  occurred in composite data coincides with the patterns of activity  $l$  generated in the training stage. Utilizing patterns  $\Omega_R$  and  $\Omega_\Theta$  obtained in Algorithm 3.3, continuous composite activity signal  $C^{mxk}$  is processed in *searchSegments* operation (Algorithm 3.6) with iterative prediction. In Algorithm 3.6,  $\lambda$  and  $\rho$ , which represent start and end of a segment located by the algorithm respectively, are initialized to zero.  $L^{\tau x4}$  is the matrix storing all the segments located, where  $L^{\tau x4}(\tau, :)$  represents the segment located after the segment  $L^{(\tau-1)x4}(\tau-1, :)$ .  $L^{\tau x4}(\tau, :)$  is the vector storing the information related to start, end, detected label and duration for the segment no.  $\tau$ . *searchSegments* continuously marks a candidate segment and labels it as  $c$ , then traverses within  $c$  backwards to see whether an opposing decision is generated or not as the later iterations are left out. Traversal within  $c$  is initiated considering the candidate segment labels  $prev$  and  $c$ , generated for two consecutive iterations numbered  $i$  and  $i+1$  respectively. At the beginning of Algorithm 3.6,  $prev$  is initialized to *null*, indicating that no candidate segment is marked yet. Raw sensor data accumulated in segments are designated by  $\Phi^{\gamma xk}$ . When sufficient data are accumulated in  $\Phi^{\gamma xk}$ , the segment stored in  $\Phi^{\gamma xk}$  will be marked as a candidate segment. In case *searchSegments* cannot derive a candidate segment based on a segment  $\Phi^{\gamma xk}$  for a long time, the system might be unresponsive, hence we incorporated a watchdog parameter  $\varepsilon$  to preserve real time responsiveness.  $\Phi^{\gamma xk}$  and  $\gamma$  are initialized as well prior to executing *loop-i*, where  $C^{mxk}$  is processed in  $m$  iterations. In *loop-i*,  $\gamma$  is compared against  $\varepsilon$ . If they are equal,  $\Phi^{\gamma xk}$  is transferred to its initial state to contain a new segment. All elements of  $B^4$  are set to  $\infty$  to signal that the current segment is misclassified due to entering watchdog state.  $L^{\tau x4}$  is appended by  $B^4$ , indicating the detection result for the related segment and Algorithm 3.6 skips to next iteration of *loop-i* in order to restart accumulating data in  $\Phi^{\gamma xk}$ . If  $\gamma$  is less than  $\varepsilon$ ,  $\Phi^{\gamma xk}$  is appended by  $C^{mxk}(i, :)$ . Whenever  $\Phi^{\gamma xk}$  is appended by a vector,  $\gamma$  is incremented by 1. The increment operation is not shown to simplify the notations. Algorithm 3.6 enforces data to be accumulated in  $\Phi^{\gamma xk}$  until  $\gamma$  becomes equal to  $d+1$ . Recalling that, in the training stage, polynomial fitting is performed on at least  $d+1$  data points,  $\Phi^{\gamma xk}$  is made to contain at least  $d+1$  rows in Algorithm 3.6 as well. After ensuring that  $\gamma$  is at least  $d+1$ , feature vector  $F^{1xkd}$  is extracted given  $\Phi^{\gamma xk}$ . Distance

between  $F^{fd}$  and the vector mapping  $\gamma$  to the space defined by the polynomials in  $MS_{a,j}^{n_a}$  is computed with  $dist$  and stored in  $\eta$ . The activity label corresponding to the test instance  $\eta$  is predicted based on  $\Omega_R$ . Generated label is stored in  $\Lambda_j^{n_a}$ . Classification of the distance  $\eta$  is repeated for  $MS_{a,j}^{n_a}, \forall 1 \leq j \leq n_a \wedge \forall a \in A$ . *majorityVoting* –  $\Lambda$  in Algorithm 3.4 determines the majority of the decisions stored in  $\Lambda^{n_a}$  for action  $a$  and the result is stored in  $D_{r(a)}^{|A|}, \forall a \in A$ . Then, candidate segment  $c$  is marked applying *majorityVoting* –  $D$  given in Algorithm 3.5 on  $D^{|A|}$ . When candidate segment  $c$ , spanning up to current iteration  $i$ , is labelled as -1, if  $prev$  is not equal to *null*, this implies that signal reached an unstable state, having started to contain irrelevant iterations recently compared to former iterations. This state is followed by splitting data to extract the segment stored in  $\Phi^{\gamma xk}$ . Backwards traversal is initiated to check whether a label with increased similarity is captured or not. If  $prev$  is labelled as *target*,  $F^{fd}$  is classified using  $\Omega_\theta$ . The classification result  $z$  and threshold  $h$  are utilized while evaluating subsegments of  $\Phi^{\gamma xk}$  searched in *searchSubsegments* algorithm.  $\lambda$  and  $\rho$  are updated to show the boundaries of the candidate segment, where  $\rho$  is set to  $i$ , the current iteration number, whereas  $\lambda$  takes the value of  $\rho - \gamma$ . Start, end and the duration of  $c$  is stored in the related elements of  $B^4$ . *searchSubsegments* sets  $B_3^4$  to the label assigned to  $\Phi^{\gamma xk}$  and  $L^{\tau x4}$  is appended by  $B^4$ , which represents the boundaries and the label of the newly identified segment  $\Phi^{\gamma xk}$ . Before moving to the next iteration, the contents of  $\Phi^{\gamma xk}$  are removed to restart accumulating the data of new iterations. The only row of  $\Phi^{\gamma xk}$  retained is the last row, which is  $\Phi^{\gamma xk}(\gamma, :)$ , since the contents of the segments with labels  $prev$  and  $c$  differ in this iteration. After handling the unstable state,  $prev$  is set to  $c$  and Algorithm 3.6 goes on with the next iteration. Tasks performed in the unstable state are repeated when  $i == m$  as well because this condition indicates that there are no more iterations and the last segment is being extracted. Signal stored in  $\Phi^{\gamma xk}$  up to iteration  $i$ , might be deteriorated as later iterations are accumulated such that the actual type of  $\Phi^{\gamma xk}$  might be missed. Since the only action for which training data are obtained naturally is the target action and training data for other actions are synthetically obtained based on target class data, the model might introduce bias favouring target class where signal belonging to other classes might be degenerated by later iterations to output a label of target. Thus, labelling the extracted segment using *searchSubsegments* procedure is undertaken in the unstable state when  $prev == target$  with an attempt to attenuate the

bias against other classes. By doing that, Algorithm 3.6 enforces more criteria to satisfy for target class patterns to be able to classify an instance segment as target, when considered together with the contents of the *searchSubsegments* procedure.

Algorithm 3.4. majorityVoting- $\Lambda(\Lambda^{n_a})$

```

 $H_i^{|A|} \leftarrow 0, \forall 1 \leq i \leq |A|$ 
for  $\forall 1 \leq i \leq n_a$  do
    //loop-j:
    for  $\forall 1 \leq j \leq |A|$  do
        if  $\Lambda_i^{n_a} = Q_j^{|A|}$  then
             $H_j^{|A|} \leftarrow H_j^{|A|} + 1$ 
            Quit loop-j
        end if
    end for
end for

 $i\_max \leftarrow \operatorname{argmax}_{i \in X} f(i)$ 
 $\omega \leftarrow \sum_{i=1}^{|A|} [f(i) = f(i\_max)]$ 
if  $\omega > 1$  then
    return -1
else if  $\omega == 1 \wedge i\_max \neq r(a)$  then
    return -1
else
    return  $i\_max$ 
end if

```

Algorithm 3.4 explains *majorityVoting* -  $\Lambda$  operation used in Algorithm 3.6. *majorityVoting* -  $\Lambda$  processes  $\Lambda^{n_a}$ , which is the vector of candidate decisions to find out the simple action  $a$  obtaining most of the votes. In Algorithm 3.4, the elements of the vector  $H^{|A|}$  storing the number of votes for each action are set to zero initially. Each element of  $\Lambda^{n_a}$  is compared against the unique code values stored in  $Q^{|A|}$  to determine the

activity type located in  $\Lambda_i^{n_a}$ . In the case of a match, the element of  $H^{|A|}$  corresponding to the action whose unique code is  $Q_j^{|A|}$  is incremented by one, indicating that the action represented by  $Q_j^{|A|}$  gains one more vote. Then, *loop\_j*, which iterates as many times as  $|A|$ , is quit. Let  $X = \{1, 2, \dots, |A|\}$ ,  $Y = \{H_1^{|A|}, H_2^{|A|}, \dots, H_{|A|}^{|A|}\}$  and  $f : X \Rightarrow Y$  such that  $f(i) = H_i^{|A|}$ . After all elements in  $H^{|A|}$  are computed, the element  $i \in X$  where  $f(i)$  takes its maximum is computed and stored in  $i\_max$ . The number of occurrences of the simple action whose unique code corresponds to  $i\_max$  is assigned to  $\omega$ . Depending on  $\omega$ , the ultimate decision generated by *majorityVoting* –  $\Lambda$  is found. If  $\omega$  is greater than 1, then there is more than one candidate simple action which obtains maximum number of votes. In this case, *majorityVoting* –  $\Lambda$  returns -1, indicating a decision cannot be made. If  $\omega$  is 1 but  $i\_max$  is not equal to  $r(a)$ , *majorityVoting* –  $\Lambda$  similarly returns -1.  $i\_max$  differing from  $r(a)$  demonstrates that the simple action obtaining maximum number of votes is different from the simple action whose training samples yield the polynomials used to generate the decisions stored in  $\Lambda^{n_a}$ . If neither of the conditions leading to the *cannot decide* cases mentioned above,  $i\_max$  is returned signalling the simple action whose unique code corresponds to  $i\_max$  obtains most of the votes.

Algorithm 3.5. majorityVoting-D( $D^{|A|}$ )

```

 $H_i^{|A|} \leftarrow 0, \forall 1 \leq i \leq |A|$ 
for  $\forall 1 \leq i \leq |A|$  do
    //loop-j:
    for  $\forall a \in A$  do
        if  $D_i^{|A|} = Q_{r(a)}^{|A|}$  then
             $H_{r(a)}^{|A|} \leftarrow H_{r(a)}^{|A|} + 1$ 
            Quit loop-j
        end if
    end for
end for

 $i\_max \leftarrow \operatorname{argmax}_{i \in X} f(i)$ 
 $\omega \leftarrow \sum_{i=1}^{|A|} [f(i) = f(i\_max)]$ 
 $\zeta \leftarrow \sum_{i=1}^{|A|} [f(i) = -1]$ 
if  $\omega > 1$  then
    return -1
else
    if  $H_{i\_max}^{1 \times |A|} < \zeta$  then
        return -1
    else
        return  $i\_max$ 
    end if
end if

```

*majorityVoting* –  $D$  operation shown in Algorithm 3.5 is similar to Algorithm 3.4, however, the structure of the vectors on which they operate are different from each other, consequently, the majority voting scheme presented in *majorityVoting* –  $D$  exhibits some differences from that in *majorityVoting* –  $\Lambda$ . While *majorityVoting* –  $\Lambda$  targets finding the most frequently occurring label in  $\Lambda^{n_a}$ , *majorityVoting* –  $D$  performs the same task

on  $D^{|A|}$ . Each element of  $\Lambda^{n_a}$  corresponds to a decision output by *cls\_prediction* when applied on  $\Omega_R$  and  $\eta$ , hence  $\Lambda_i^{n_a} \in \{Q_1^{|A|}, Q_2^{|A|}, \dots, Q_{|A|}^{|A|}\}, \forall 1 \leq i \leq n_a$ . On the other hand, each element of  $D^{|A|}$  corresponds to the decision returned by *majorityVoting* –  $\Lambda$  applied on  $\Lambda^{n_a}$ , therefore,  $D_i^{|A|} \in \{Q_1^{|A|}, Q_2^{|A|}, \dots, Q_{|A|}^{|A|}\} \vee D_i^{|A|} \in \{-1\}$ , where -1 signals that  $\Lambda^{n_a}$  does not yield a decision while being processed by *majorityVoting* –  $\Lambda$  as mentioned in Algorithm 3.4. In Algorithm 3.5,  $\omega$  is computed as in Algorithm 3.4, additionally,  $\zeta$ , which is the number of occurrences of -1 in  $H^{|A|}$ . For the unique code  $i\_max$  of an action  $a \in A$  to be returned as the decision, the maximum number of votes obtained by  $a$ , which is  $H_{i\_max}^{|A|}$ , should surpass  $\zeta$  as well as satisfying the criterion of  $\omega$  being greater than 1.

Algorithm 3.7 explains *searchSubsegments* operation where backwards traversal is performed on candidate segment  $\Phi^{\gamma xk}$  to test the agreement between the labels assigned to subsegments of  $\Phi^{\gamma xk}$  designated as  $\Gamma^{\chi xk}$  and the label  $z$ , which is determined prior to invoking *searchSubsegments* in Algorithm 3.6. *searchSubsegments* algorithm determines the label identifying  $\Phi^{\gamma xk}$ , consulting the decisions generated based on subsegments.  $\Gamma^{\chi xk}$  is populated with rows of  $\Phi^{\gamma xk}$  up to  $\Phi^{\gamma xk}(v, :)$ . *loop* –  $s$  iterates with initial value of  $s$  being equal to  $\gamma$ , where  $s$  is decremented by 1 until being equal to 1. features  $F^{fd}$  belonging to  $\Gamma^{\chi xk}$  are extracted and the label to assign to  $\Gamma^{\chi xk}$  is predicted based on  $\Omega_\theta$ . If the resulting classification decision  $c\_sub$  is equal to  $z\_inv$ , this indicates that the subsegment  $\Gamma^{\chi xk}$  disagrees with the decision  $z$ , where  $z\_inv$  stands for the inverse of  $z$ . Since the activity labels are *target* and *negative*, if  $z$  is *target*,  $z\_inv$  symbolizes the negative class and vice versa. In the case of disagreement between  $c\_sub$  and  $z$ , whether  $\Gamma^{\chi xk}$  contains sufficient amount of data is checked comparing  $\psi$  and  $h$ .  $\psi$  is the ratio between the size of  $\Gamma^{\chi xk}$  and  $\Phi^{\gamma xk}$ . If  $\psi$  is greater than or equal to the threshold  $h$ , it means that the amount of data stored in  $\Gamma^{\chi xk}$  is enough for the classification decision  $c\_sub$  to identify  $\Phi^{\gamma xk}$ . Disagreement between the decisions generated based on  $\Gamma^{\chi xk}$  and  $\Phi^{\gamma xk}$  signals that as  $\Gamma^{\chi xk}$  is appended by data of later iterations, signal starts degenerating, which is the reason why  $\Gamma^{\chi xk}$  has a priority over  $\Phi^{\gamma xk}$  in terms of determining the label in the case of disagreement. If  $\psi$  is greater than or equal to  $h$ ,  $z\_inv$  is returned as the label identifying the segment  $\Phi^{\gamma xk}$ , terminating *searchSubsegments* algorithm. If the label identifying  $\Phi^{\gamma xk}$  is not found during backwards traversal in *loop* –  $s$ ,  $z$  is returned as the label.

Algorithm 3.6.  $searchSegments(C^{mxk})$

```

 $\lambda \leftarrow 0, \rho \leftarrow 0, L^{\tau x4} \leftarrow \emptyset, prev \leftarrow null, \Phi^{\gamma xk} \leftarrow \emptyset, \gamma \leftarrow 0$ 
//loop - i
for  $\forall 1 \leq i \leq m$  do
  if  $\gamma == \varepsilon$  then
     $\Phi^{\gamma xk} \leftarrow \emptyset, \gamma \leftarrow 0$ 
     $B_p^4 \leftarrow \infty, \forall 1 \leq p \leq 4$ 
     $L^{\tau x4} \leftarrow L^{\tau x4} \cup B^{1x4}$ 
    Skip to next iteration of loop - i
  end if
   $\Phi^{\gamma xk} \leftarrow \Phi^{\gamma xk} \cup C^{mxk}(i, :)$ 
  if  $\gamma < d + 1$  then
    Skip to next iteration of loop - i
  end if
   $F^{fd} \leftarrow NPFE(\Phi^{\gamma xk})$ 
  for  $\forall a \in A$  do
    for  $\forall 1 \leq j \leq n_a$  do
       $\eta \leftarrow dist(F^{fd}, \gamma, MS_{a,j}^{n_a}), \Lambda_j^{n_a} \leftarrow cls\_prediction(\Omega_R, \eta)$ 
    end for
     $D_{r(a)}^{|A|} \leftarrow majorityVoting - \Lambda(\Lambda^{n_a})$ 
  end for
   $c \leftarrow majorityVoting - D(D^{|A|})$ 
  if  $(c == -1 \wedge prev \neq null) \vee (i == m)$  then
    if  $prev == target$  then
       $z \leftarrow cls\_prediction(\Omega_\Theta, F^{fd}), \rho \leftarrow i, \lambda \leftarrow \rho - \gamma, B_1^4 \leftarrow \lambda, B_2^4 \leftarrow$ 
 $\rho - 1, B_4^4 \leftarrow \gamma$ 
       $B_3^4 \leftarrow searchSubsegments(\Phi^{\gamma xk}, z, h), L^{\tau x4} \leftarrow L^{\tau x4} \cup B^4$ 
    end if
     $B^k \leftarrow \Phi^{\gamma xk}(\gamma, :), \Phi^{\gamma xk} \leftarrow \emptyset, \Phi^{\gamma xk} \leftarrow \Phi^{\gamma xk} \cup B^k$ 
  end if
   $prev \leftarrow c$ 
end for

```

Algorithm 3.7.  $searchSubsegments(\Phi^{\gamma^{xk}}, z, h)$ 

```

//loop - s
for  $\forall \gamma \geq s \geq 1$  do
   $\Gamma^{\chi^{xk}} \leftarrow \emptyset$ 
  for  $\forall 1 \leq v \leq s$  do
     $\Gamma^{\chi^{xk}} \leftarrow \Gamma^{\chi^{xk}} \cup \Phi^{\gamma^{xk}}(v, :)$ 
  end for
   $F^{1x^{fd}} \leftarrow NPFE(\Gamma^{\chi^{xk}})$ 
   $c_{sub} \leftarrow cls\_prediction(\Omega_{\Theta}, F^{1x^{fd}})$ 
  if  $c_{sub} == z_{inv}$  then
     $\psi \leftarrow (\chi/\gamma)x100$ 
    if  $\psi \geq h$  then
      return  $z_{inv}$ 
    end if
  end if
end for
return  $z$ 

```

### 3.2. ARTIFICIAL DATA GENERATION FOR REPRESENTATION OF UNSEEN CLASSES WITH RESPECT TO TARGET CLASS

Outlier class data are artificially obtained as given in Algorithm 3.8 with our ADG method, where normal distribution is fit to target class data to generate random numbers representing the distribution fit. In our work, while utilizing the normal distribution to represent the negative class, the classes which will be categorized as negative are not assumed to exhibit normal distribution properties all the time. Instead, our assumption is that target class instances in the test phase will be more similar to target class patterns compared to the patterns generated based on the artificially generated data as well as the classes which will be categorized as negative will be more similar to the patterns of artificially generated data compared to the target class patterns. In Algorithm 3.8, each of the instances in the set of training data  $T$  is in the form of a matrix  $I^{d \times m}$ , indicating the input matrix having the number of rows and columns as  $d$  and  $m$  respectively.  $d$  implies duration of activity stored in the instance  $I^{d \times m}$  since it is affected by the sampling frequency of the sensor and the amount of time consumed to perform the activity.  $m$  is the number of axes in the sensor. For each vector represented as a column of  $I^{d \times m}$ , namely  $I^{d \times m}(:, i)$ , a normal distribution, which is expressed as  $P$ , is fit. Output matrix denoted by  $O^{d \times m}$ , which is of the same size as  $I^{d \times m}$ , specifies an instance in the set of outlier class,  $A$ , where an entry at row  $j$  and column  $i$  of  $O^{d \times m}$  is denoted by  $O_{ji}^{d \times m}$ . Each  $O_{ji}^{d \times m}$  entry is set to a random number drawn from a normal distribution with the mean  $P_\mu$  and standard deviation  $P_\sigma$  of  $P$ , after applying  $rni$  function, which indicates the operation of rounding to the nearest integer as shown in *line 7* of Algorithm 3.8. In *line 7*,  $r(D)$  designates the operation of generating a random number from standard normal distribution. After computation of all  $O_{ji}^{d \times m}$  values in  $O^{d \times m}$  is complete,  $A$  is set to the union of  $O^{d \times m}$  and  $A$ . Finally, all of the artificial data instances are returned in  $A$ .

Algorithm 3.8. generateArtificialData()

```

1:  $A \leftarrow \emptyset$ 
2:  $D \leftarrow normalDistribution$ 
3: for  $\forall I^{d_{xm}} \in T$  do
4:   for  $\forall 1 \leq i \leq m$  do
5:      $P \leftarrow fit(I^{d_{xm}}(:, i), D)$ 
6:     for  $\forall 1 \leq j \leq d$  do
7:        $O_{ji}^{d_{xm}} \leftarrow rni(P_{\sigma} * r(D) + P_{\mu})$ 
8:     end for
9:   end for
10:   $A \leftarrow A \cup O^{d_{xm}}$ 
11: end for
12: return  $A$ 

```

Instances of target and outlier classes are processed with Wavelet entropy as in [40] [141], where Coiflet 4 wavelet with decomposition level 3 is applied on  $I^{d_{xm}}(:, i)$  and  $O^{d_{xm}}(:, i)$ ,  $\forall 1 \leq i \leq m$ . Resulting coefficient vectors are processed with Shannon entropy scheme as given in Eq 4.6 so that the feature vectors  $F^{1xm}$  are obtained for both target and outlier classes. In Eq 4.6, each element of the feature vector,  $F_j^{1xm} \forall 1 \leq j \leq m$ , is computed based on the coefficient vector  $C^{1xtj}$  belonging to axis  $j$ . For simplifying calculation,  $\log((C_i^{1xtj})^2)$  is equated to zero when  $(C_i^{1xtj})^2$  takes the value of zero because  $\log(0)$  is undefined,  $\log$  expressing natural logarithm.

The continuous data stream is partitioned into segments utilizing explicit segmentation such that each segment contains exactly one occurrence of the target and other class activities. For classifying the feature vectors, we used KNN, SVM, logistic regression and Naïve Bayes classifiers.

$$F_j^{1xm} = \sum_{i=1}^{t_j} (C_i^{1xtj})^2 * \log((C_i^{1xtj})^2) \quad (3.4)$$

### 3.3. INCOMPLETENESS MITIGATION METRIC

Accuracy of detection for an action depends on the types of other actions in the set, hence detection accuracy should be evaluated considering all possible combinations of actions in the set. For this reason, we group the actions into *subsets*, where a *subset*, in our context, indicates each one of the subsets of the whole simple action set  $A$ , with at least two elements. Hence, the number of *subsets*, designated as  $|S|$ , becomes equal to  $2^{n_{sa}} - n_{sa} - 1$ .

We express the capability of incompleteness mitigation in terms of percentage of correctly identified iterations, which we name *incompleteness mitigation rate (imr)*. Eq 3.5 defines *imr*, where  $|S_i|$  is the number of activities in subset  $S_i$  and  $c$  is the number of correctly identified iterations. *imr* computes percentage of correct iterations where a simple action sample  $C^{mk}$  introduces  $m$  iterations in total. Thanks to *imr*, incompleteness mitigation is evaluated for each action as many times as the number of subsets in which they appear, including the effect of types of other activities against which the activity  $a$  is compared.

$$imr_{i,a} = \frac{\sum_{j=1}^n \frac{c_j}{m_j} x 100}{n}, \quad (3.5)$$

$$\forall 1 \leq i \leq |S| \wedge 1 \leq a \leq |S_i|$$

In order to measure incompleteness resilience of evaluated methods across all subsets, we utilize ICC (Intraclass Correlation Coefficient), which is used to assess reliability of ratio data [142] as inter-rater reliability [143].

In the iterative scheme we carried out, the training data is the whole action and iterations, on which prediction is performed, contain only a part of the action. Hence, decisions taken at each iteration are made based on incomplete data, which raises incompleteness problem.

Algorithm 3.9- 3.10 summarize how these methods are utilized iteratively, where the

following legend clarifies the notations used in algorithms:

- $C^{m \times n}$ : Matrix  $\mathbf{C}$  with  $\mathbf{m}$  rows and  $\mathbf{n}$  columns. When  $\mathbf{m}=1$ ,  $C^{m \times n}$  designates a vector with  $\mathbf{n}$  elements. When  $C^{m \times n}$  denotes a raw acceleration data matrix,  $\mathbf{m}$  indicates number of sensor events that form a window of size  $\mathbf{m}$  whereas  $\mathbf{n}$  indicates the number of axes in the sensor.

- $T_a$ : set of training samples in the form of raw sensor data, belonging to action  $a \in A$ .

- $\mathbf{C}(\mathbf{i}, :)$ :  $i$ th row of matrix  $\mathbf{C}$ .

$\mathbf{k}$ : Number of axes for the accelerometer.

- $\mathbf{n}$ : number of training samples per action.

- $\mathbf{nsa}$ : number of simple actions in the set  $A$ .

- $ldv$ : Labelled distance vector construction given in Eq 3.7. The function  $ldv$  takes the distance vector  $I^{fd}$  as the parameter and outputs a new vector  $V_i^{1+fd}$ . The resulting vector designates the label, which corresponds to a simple action, in its first element. The rest of the elements in  $V_i^{1+fd}$  are the same as the contents of  $I^{fd}$ .

$$C^{m \times n} \cup B^n = E^{(m+1) \times n} \text{ such that}$$

$$E(i, :) = \begin{cases} C(i, :), \forall 1 \leq i \leq m \\ B^n, i = m + 1 \end{cases} \quad (3.6)$$

Algorithm 3.9 elaborates on training phase of compared methods. For each simple action sample  $C^{m \times k}$  in the training set, features are computed with the feature extraction schemes of the compared methods,  $FE_{compared}$ , generating the feature vector  $F^{fd}$ . Each labelled feature vector is accumulated in  $t_a$ , which is then processed with training phase of the selected classifier,  $cls\_training$ . In our setup,  $cls$  corresponds to KNN with 5 neighbours.

$$ldv(I^{fd}, A, a) = V_i^{1+fd} = \begin{cases} r, A_r = a \wedge i = 1 \\ I_{i-1}^{fd}, \text{ otherwise} \end{cases} \quad (3.7)$$

Iterative prediction in the compared methods is explained in Algorithm 3.10. In *ppi\_compared* scheme, where *ppi* is used as an acronym for *prediction per iteration*, test sample  $C^{mxk}$  is processed in  $m$  iterations. At each iteration, current row in the matrix  $C^{mxk}$  is appended in  $M$ , from which the features of that iteration are extracted with  $FE_{compared}$ . Generated feature vector  $F^{fd}$  is processed with the prediction module of the selected classifier *cls\_prediction*. At the end of the current iteration, output of *cls\_prediction* is stored in the vector  $IR$ , which keeps resulting labels for all iterations. After all of the iterations are classified,  $IR$  is returned, completing the *ppi\_compared* algorithm.

Algorithm 3.9. training\_compared()

```

for  $\forall a \in A$  do
   $t_a \leftarrow \emptyset$ 
  for  $C^{mxk} \in T_a$  do
     $F^{fd} \leftarrow FE_{compared}(C^{mxk})$ 
     $t_a \leftarrow t_a \cup ldv(F^{fd}, A, a)$ 
  end for
end for
 $cls\_training(t_a), \forall a \in A$ 

```

The training set is formed of the features extracted from the whole sample  $C^{mxk}$  instead of a part of the sample whose number of rows is less than  $m$ , as expressed in Algorithm 3.9. On the other hand, in prediction phase, features are extracted from part of the sample, as given in Algorithm 3.10, which injects iterative manner into compared schemes.

### 3.3.1. Intraclass Correlation Coefficient

Inter-rater reliability evaluation is performed in ICC considering targets rated by judges [144], measuring how reliable the ratings are across all raters [145]. In our model, subsets being the targets and raters being the simple actions, we evaluate how repeatable the *imr* values across all subsets and all simple actions. The fact that our evaluation is designed to measure variation as the activity type changes makes ICC a suitable metric. Another reason

Algorithm 3.10.  $ppi\_compared(C^{mxk})$

```

 $M \leftarrow \emptyset$ 
for  $\forall 1 \leq i \leq m$  do
     $M \leftarrow M \cup C(i, :)$ 
     $F^{fd} \leftarrow FE_{compared}(M)$ 
     $IR_i^m \leftarrow cls\_prediction(F^{fd})$ 
end for
return  $IR$ 

```

for utilizing ICC is that our ratings are *imr* values, which are of ratio type. *imr*, which is a function of the percentage of correct iterations, exhibits the properties of ratio data [142], which are having equal increments and absolute zero (the case where all iterations are misclassified).

## 4. EXPERIMENTAL SETUP

In this chapter, we depict the experimental setting utilized to realize the segmentation and ADG modules of our spotting scheme. In addition, we give the experimental setup corresponding to our incompleteness mitigation metric.

We focused on application scenarios with the following constraints:

- The activity of interest is preceded and succeeded by several unknown activities in streaming data.
- The activity of interest is performed within the course of daily routines instead of a predefined time interval during the day.
- The start and end of activities in the sequence are not known by the knowledge discovery unit.

We applied our proposed approach on use cases which exemplify the constraints given above. These constraints inject nondeterminism to the tracking problem. The application scenarios we worked on are gathered from daily living settings for providing the people assistance with healthcare. These application scenarios are based on contexts in which the people are monitored to evaluate their adherence to prescribed medication and physiotherapy exercises in daily settings utilizing a wrist-worn sensing device. The physiotherapy exercise protocol we focused on is centered around elbow based rehabilitation for enhancing upper extremity functionality. We selected the use cases we studied from healthcare domain because health related services necessitate integration of HAR modules for increased coverage.

Similar to monitoring adherence to physiotherapy exercises, medication intake tracking is also a use case where monitoring patients' adherence is useful. Failing to adhere to medication intake results in deviating from the prescribed therapeutical protocol. Elderly people might forget to take medication due to cognitive impairment. Also, fast pace of daily responsibilities might prevent individuals from remembering to take medication even if they

do not suffer from any cognitive problems. In order to facilitate adherence to medication protocol, medication should be taken at the prescribed time during the day. Hence, the monitoring scheme should be triggered at the moment determined by the prescribed medical procedure so that monitoring begins. Such a monitoring scheme is needed instead of merely generating notifications which remind people of taking medication. Despite being notified, people might again forget to take medication because of the factors mentioned above. People without cognitive problems can be distracted by daily affairs leading to forgetfulness after being notified. For example, when a person is on the verge of taking medication, his/her phone might ring. His/her attention might be directed towards the topic of conversation on the phone. After the phone call ends, the person might forget that s/he has not taken the medication until then. Consequently, s/he might go on with his/her next daily activity without having taken the medication. Forgetting to take medication at the prescribed time can occur even without being interrupted by an event. The person might happen to think about something else when s/he is about to take medication which can lead to forgetting to take the drug. A monitoring scheme triggered at the prescribed time of medication intake can recognize the activities to decide on whether the drug is taken or not. Upon generating this decision, appropriate actuation mechanisms can participate in the solution to complete the application scenario. Cognitive deterioration might lead to confusing the correct time of medication intake with an arbitrary instance during the day. Such a case might result in taking the drug earlier than the prescribed instant, which can have equivalently serious consequences. Consuming more medication than prescribed amount is another forgetfulness related misapplication of medication protocol. In order to address challenges exemplified above, smart devices with integrated HAR modules can be developed. An example of such smart devices is a smart medicine box with a medication-holding container which is unlocked conditionally. The container is locked until the prescribed time of medication intake for avoiding earlier intake of the drug. After the container is unlocked at the prescribed moment, the container grants access to only prescribed amount of medication at a time, a single pill for instance. Meanwhile, monitoring scheme gets activated to allow the consequent actuation facilities to proceed if the person fails to perform "medication intake" activity.

#### 4.1. COMPARED METHODS IN SPOTTING

In this section, we elaborate on the knowledge discovery schemes against which we compared our method. Also, we describe how our method and the compared methods are executed in the given experimental setting.

In the compared schemes, the continuous sensor data are partitioned using fixed sized windowing, where the experiments are performed with varying feature extraction schemes and window sizes. The experimented feature extraction schemes are *windowed energy*, *peak frequency*, *Shannon entropy* and *wavelet entropy* considering ease of computation they provide. Also, wavelet based features [146] [147] [148], peak frequency [149] [150] [151], energy [152] and Shannon entropy [56] are commonly in use within the scope of acceleration based human activity recognition, which is another reason for our selection of compared features. We run each of these feature extraction schemes with the window size set to 10, 20, 30, 40 and 50 samples per window.

In both our method and compared schemes mentioned above one class classification is utilized where negative class data are generated artificially. The extracted features are classified applying KNN with 10 neighbours in all methods. Selecting the number of neighbours as the square root of the number of the training samples is an existing approach facilitating practicability [153], hence we also employed a similar strategy in choosing the number of neighbours  $h$  as in Eq 4.1, indicating that  $h$  is equal to the square root of  $n_a$ , which is the number of training instances per simple action  $a$ .

$$h = \sqrt{n_a} \quad (4.1)$$

We measured the prediction performance with an independent test set which is composed of 55 instances per subject. The instances in the test set, i.e. the test instances are different from the training instances in terms of structure. Each training instance contains only a simple action whereas each test instance contains a sequence of simple actions, which are

explained in Section 4.1.1.

The number of training samples on a per subject basis for our setup for the experiments are given as follows: The number of training instances acquired for target action is 55, which is equal to the number of training instances corresponding to artificially generated *other* class data, hence the total number of training instances is 110. Considering the number of training instances in the experiments mentioned above, the number of neighbours become equal to 10. We performed our evaluations based on data acquired from 4 subjects - 2 male and 2 female.

Algorithm 4.1- 4.2 and Eq 4.2- 4.5 summarize definition of the compared approaches. While explaining the compared methods, we retain the notations we adopted for clarifying our proposed approach in Section 3.1.

Algorithm 4.1. training\_compared()

```

 $\Theta^{\beta x(f d+1)} \leftarrow \emptyset$ 
for  $\forall a \in A$  do
   $\Pi_{f d+1}^{(f d+1)} \leftarrow r(a)$ 
  for  $C^{m x k} \in T_a$  do
     $F^{f d} \leftarrow F E C(C^{m x k})$ 
     $\Pi_i^{(f d+1)} \leftarrow F_i^{f d}, \forall 1 \leq i \leq f d$ 
     $\Theta^{\beta x(f d+1)} \leftarrow \Theta^{\beta x(f d+1)} \cup \Pi^{(f d+1)}$ 
  end for
end for
 $\Omega_{\Theta} \leftarrow cls\_training(\Theta^{\beta x(f d+1)})$ 

```

Algorithm 4.1 elaborates on training phase of compared methods. For each simple action instance  $C^{m x k}$  in the training set, features are computed with the feature extraction schemes of the compared methods,  $F E_{compared}$ , generating the feature vector  $F^{f d}$ . The vector  $\Pi^{(f d+1)}$  contains the elements of  $F^{f d}$  along with the label identifying the instance stored in  $F^{f d}$ . The  $r$  function computes  $\Pi_{f d+1}^{(f d+1)}$ , which indicates the label corresponding to the

instance designated by  $F^{fd}$ .  $\Theta^{\beta x(fd+1)}$  is appended by  $\Pi^{(fd+1)}$  for  $C^{mxk} \in T_a, \forall a \in A$ . Finally,  $\Theta^{\beta x(fd+1)}$  is processed with training phase of the selected classifier, *cls\_training*, to output the patterns  $\Omega_\Theta$ .

Algorithm 4.2. *prediction\_compared*( $C^{mxk}, W$ )

```

 $\Phi^{\gamma xk} \leftarrow \emptyset, \gamma \leftarrow 0, L^{\tau x4} \leftarrow \emptyset$ 
for  $\forall 1 \leq i \leq m$  do
     $\Phi^{\gamma xk} \leftarrow \Phi^{\gamma xk} \cup C^{mxk}(i, :)$ 
    if  $(\gamma == W) \vee (i == m)$  then
         $F^{fd} \leftarrow FEC(\Phi^{\gamma xk})$ 
         $B_1^4 \leftarrow i - W + 1, B_2^4 \leftarrow i, B_4^4 \leftarrow W$ 
         $B_3^4 \leftarrow cls\_prediction(\Omega_\Theta, F^{fd})$ 
         $L^{\tau x4} \leftarrow L^{\tau x4} \cup B^4$ 
         $\Phi^{\gamma xk} \leftarrow \emptyset, \gamma \leftarrow 0$ 
    end if
end for
return  $L^{\tau x4}$ 

```

How fixed sized windowing is applied on compared methods is explained in *prediction\_compared* operation given in Algorithm 4.2. The test instance  $C^{mxk}$ , which contains a sequence of simple actions in the form of continuous sensor data, is divided into windows, each of which is of size  $W$ . The portion of continuous sensor data corresponding to a window is stored in  $\Phi^{\gamma xk}$ , where  $\gamma$  indicates the number of samples currently stored in  $\Phi^{\gamma xk}$ . The label assigned to a window  $\Phi^{\gamma xk}$  by the algorithm as well as the starting and ending sample numbers reside in a row of matrix  $L^{\tau x4}$ . The contents of  $\Phi^{\gamma xk}$  and  $L^{\tau x4}$  are initially  $\emptyset$ , where  $\gamma$  is 0.  $\Phi^{\gamma xk}$  is appended by the currently processed sample  $C^{mxk}(i, :)$  until  $\gamma$  becomes equal to  $W$  or  $i$  becomes equal to  $m$ , signalling that  $C^{mxk}(i, :)$  is the last sample to process and there are no more samples left with which the window will be appended. Once the process of appending  $\Phi^{\gamma xk}$  is complete, feature vector  $F^{fd}$  corresponding to  $\Phi^{\gamma xk}$  are extracted with *FEC*. The vector  $B$  stores the index of starting and ending samples in the currently processed window  $\Phi^{\gamma xk}$ , which are represented as  $i - W + 1$  and  $i$  respectively,

as well as the label determined by the prediction module of the classifier  $cls$  and the window size  $W$ . Prediction is performed utilizing the patterns  $\Omega_\Theta$  generated in Algorithm 4.1 on the instance  $F^{fd}$ . Then,  $B$  becomes the current row of  $L^{\tau x4}$  and the  $\Phi^{\gamma xk}$  is reassigned to its initial value to go on with processing the next window. After all of the samples in  $C^{mxk}$  are processed, the class labels predicted for the activities in the sequence are returned as  $L^{\tau x4}$ .

$FEC$  indicates the feature extraction schemes included in the methods against which we compare our proposed scheme.  $FEC$  is referred in Algorithm 4.1 along with Algorithm 4.2 to stand for the separate formulations specific to every compared feature extraction scheme. We followed the formulations for compared methods used in [141] [40]. In the given descriptions of compared feature extraction schemes,  $j$ th element of  $F^{fd}$  is denoted by  $F_j^{fd}$  while the value at row  $i$  and column  $j$  of  $C^{mxk}$  is indicated by  $C_{ij}^{mxk}$ . The formulations corresponding to the compared methods, i.e. windowed energy, Shannon entropy, peak frequency and wavelet entropy are explained as follows:

Eq 4.2 and 4.6 elaborates on how  $F^{fd}$  is determined out of  $C^{mxk}$  with windowed energy and Shannon Entropy respectively. Considering that logarithm of 0 is undefined, the value of  $\log((C_{ij}^{mxk})^2)$  is calculated as zero in the case that  $(C_{ij}^{mxk})^2$  corresponds to zero in order to determine Shannon entropy feature with a reduced way.  $\log$  symbolizes natural logarithm.

$$F_j^{fd} = \sum_{i=1}^m (C_{ij}^{mxk})^2, \forall 1 \leq j \leq k \wedge fd = k \quad (4.2)$$

$$F_j^{fd} = \sum_{i=1}^m (C_{ij}^{mxk})^2 * \log((C_{ij}^{mxk})^2), \quad (4.3)$$

$$\forall 1 \leq j \leq k \wedge fd = k$$

$FFT$  (Fast Fourier Transform) is performed given  $C^{mxk}(:, j)$  to compute peak frequency as in Eq 4.5.  $F^{fd}$  is formed of the maximum frequency components acquired evaluating the

absolute value of the result of FFT operation. Before applying  $FFT$ ,  $C^{mxk}(:, j)$  is padded with zeros so that  $m$  turns into a power of 2.

Coiflet 4 wavelet with decomposition level 3 is employed for  $C^{mxk}(:, j)$ ,  $\forall 1 \leq j \leq k \wedge fd = k$  while evaluating wavelet entropy. Then, in order to determine  $F^{fd}$ , Shannon entropy is computed based on the coefficients.

$$F_j^{fd} = \max_f |FFT(C^{mxk}(:, j))|, \forall 1 \leq j \leq k \wedge fd = k \quad (4.4)$$

#### 4.1.1. Activity Annotation and Sensor Data

We studied on 9 different activities illustrated in Fig. 4.2. The simple activities ***grab the jug, take jug towards the glass, pour water, put jug back, move the hand towards the medicine box to pick pill, pick pill, put pill in mouth, drink water*** and ***rest*** given in Fig. 4.2 form the continuous data stream on which we verified our method and the compared methods. The given sequence of simple activities exemplifies a use case where medication intake recognition is performed in a daily living environment. The subject starts performing the activities in the sequence with ***grab the jug*** activity (Fig. 4.2b), moving the hand from a starting position as given in Fig. 4.2a. Then, the simple activities shown in Fig. 4.2c-4.2j are performed sequentially, which is followed by changing to the resting position illustrated in Fig. 4.2k, which finalizes the sequence. The sequence explained in Fig. 4.2 forms the contents of a test instance.

In all the experiments, the activities in the sequence except the target activity are labelled as *other* activity. ***Put pill in mouth*** given in Fig. 4.2h is the target activity in the experiments since it is the core element in the logical view of medication intake recognition use case. Target activity can be described as a simple activity of interest performed between other activities and transitions within the category of *other class* existing in a sequence. In activity recognition domain, mostly periodic activities are studied whereas recognizing non-periodic activities is more challenging, which is the reason for studying the given sequence.

Fig. 4.1 illustrates 3D acceleration sensor data corresponding to the set of target activity signals for the training stage. Each data point, which corresponds to an element in the 3D acceleration vector, is defined with a sequence no given on the horizontal axis as well as the associated acceleration value specified on the vertical axis. The subject wears a single 3D acceleration sensor on the dominant wrist, which is the tool used for data acquisition. In order to facilitate wearability of the sensor, the circuit board on which the sensor is located is fixed over a watch using adhesive tape as displayed in Fig. 4.2. The data transmitted by the sensor are uncalibrated. The sensor and the PC interconnect over USB. In acceleration measurements, a value of 8192 and -8192 stand for 1g and -1g respectively. Sensor parameters regarding calibration and axis incompatibility are not input, thus drifting is likely to occur with a rate of approximately 10%, which is a consequence of the inherent structure of MEMS. Acquisition rate for data utilized in training and prediction stages is 24 samples per second.

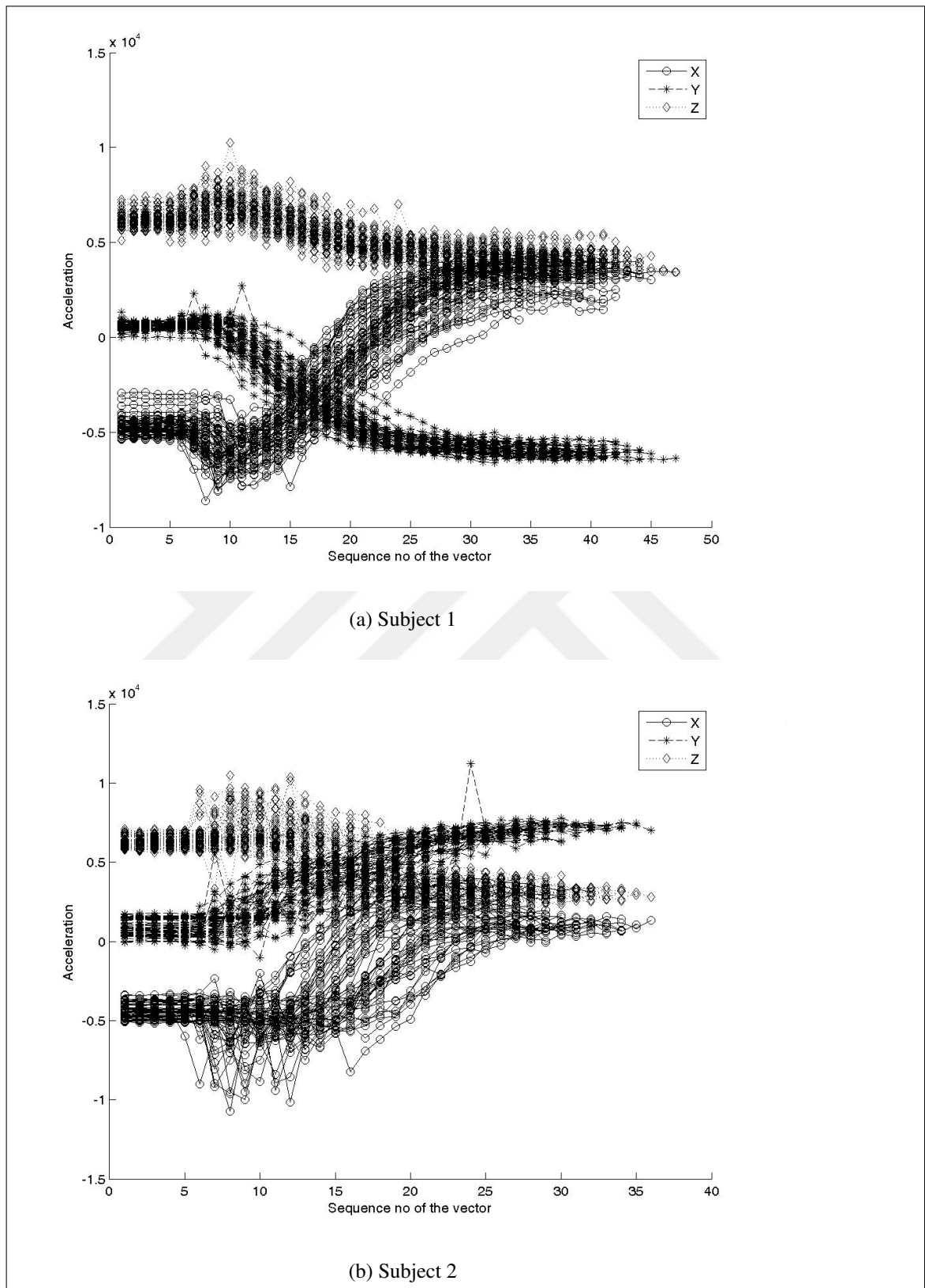


Figure 4.1: Sensor data for training instances of target activity



Figure 4.2: Simple actions forming the medication intake composite activity performed by a subject

#### 4.2. EXISTENCE OF UNSEEN CLASSES IN THE PREDICTION STAGE

Figure 4.3 illustrates a generic system architecture for monitoring adherence to physiotherapy exercises. In our application scenario, the time when the exercise is expected

to be performed is registered in the application beforehand. When it is time to exercise, the system starts monitoring and sensor data stream is initiated (Step 1). At this time, the patient starts exercising either as soon as s/he hears the notification generated by the system or after performing several other daily activities. After completing the exercises, s/he might go on with practicing daily activities until the timeout specified by the system expires. Step 2 depicts feature extraction, classification and segmentation stages by a data processing unit. Decisions generated as a result of sensor data processing (Step 3) incorporates whether the exercise is performed or whether the exercise protocol is performed correctly. Finally, as indicated in Step 4, health care staff or a health care center are informed of the decisions.

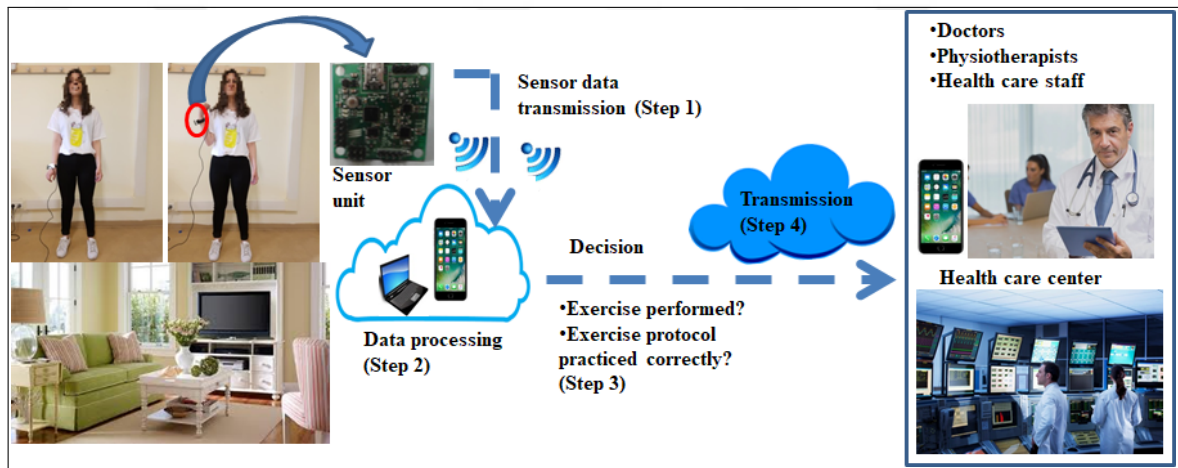


Figure 4.3: System architecture for monitoring adherence to physiotherapy exercises

In the context of physical therapy, adherence to prescribed exercises introduces great impact on the effectiveness of physiotherapy. Existing mechanisms of tracking adherence include percentage of attendance for in-clinic therapy and patient's feedback as well as utilization of electronic tools such as pedometers for in-home therapy. Monitoring adherence based on feedback of the patients is error-prone. Even if the patients are ready to adhere to physical therapy in home environment, other factors, which do not result from intentional behaviours of the patients, such as cognitive conditions might deteriorate adherence. In addition, the way patients perform the exercises might deviate from the prescribed scheme. Hence, e-health applications reminding people of exercising and monitoring whether the way they exercise conforms to what is prescribed has a promising role to facilitate adherence control. There are several reasons for choosing elbow flexion and extension exercise in our study.

Elbow related injuries influence the capability of performing daily living activities, which require upper extremity functionality. Standing up while sitting on a chair or lying on the bed exemplify daily activities which necessitate the contribution of elbow muscles to weight-bearing for healthy people. In addition, amputation of lower extremity and spinal cord injuries at S1 level and below compel patients to rely on upper extremity for weight bearing. People suffering from these health issues are prescribed to perform resistance training, which is in the form of elbow flexion and extension exercises, so that they can gain sufficient strength prior to using the elbow excessively. In human activity recognition systems (HAR), sensor data acquired from the subject and the environment are processed to identify the target activities. This processing involves feature extraction, segmentation and classification steps. Tasks expected of a HAR system designed for monitoring adherence to physical exercises can exhibit a wide variety. For example, it is possible to determine whether the exercises are practiced or not. Also, health care professionals can ensure the way the subject performs the exercises conforms to the instructions specified.

We studied on an activity sequence which contains a physiotherapy exercise protocol as well as several other activities. The exercise protocol is composed of  $k$  repetitions of the target activity, where a set of  $k$  repetitions is practiced twice. The activity sequence we study is summarized in Table 4.1, where each unit of activity is given along with the sequence number showing in which order the activities are performed. The activity sequence is determined considering how an elbow flexion and extension exercise routine is embedded in the daily routine of a patient in physiotherapy practice. The patient initiates the activity sequence, with the activity *leave phone onto the table* as given in Table 4.1, exemplifying how a patient switches to exercising from his/her daily routine. The second activity is *position yourself while standing*, where the patient gets ready to exercise attaining a straight posture. Then, the exercise protocol starts, where elbow flexion and extension exercise (the target activity) is repeated  $k$  times, as indicated by the activity with sequence number 3. After 1st set of  $k$  repetitions is complete, the subject performs *sit and position yourself on the chair* activity. *Spend time with phone*, the action no. 5, is where the subject spends some time with his/her phone in order to rest while sitting. The subject *gets up and positions himself/herself while standing* to get ready for the second set of  $k$  repetitions of the target activity in actions 6 and 7. Finally, s/he *sits and positions himself/herself on the chair* to

switch to a resting state with action 8. The actions given in the sequence except number 3 and 7 present 6 different classes which will be categorized as *other*, whereas each single repetition of elbow flexion and extension exercise indicates the target class. The selection of the classes in the category of *other* results from the fact that they can be performed in both home and office settings, to increase the practicability of the exercise routine.

The  $k$  parameter value is selected as 15 considering the categorization of physiotherapy exercises. In the context of physiotherapy, the category of the exercise protocol influences how many times an exercise is repeated. If the repetition count is 15, the exercise scheme contributes to both strength and endurance gains. It's worth noting that  $k$  is not a parameter which influences the success of detection performance.

In our activity description within the specified application scenario, target class is a single movement performed by the elbow, which is a single flexion followed by a single extension. The 3D acceleration signals composed of the target activity instances in the training set are shown in Figure 4.4, where horizontal axis indicates the sequence number of the signal vector whereas the vertical axis designates the acceleration value corresponding to the vector with given sequence number. Subjects can prefer wearing the sensor partly away from the wrist since they feel more comfortable. Thus, the location where the sensor resides can be away from the points which are the primary sources of motion and the signal produced by the wrist may be sensed by the device in a modified manner. This variation in the way the sensor is worn by the subjects causes the dataset to be more demanding.

In our experimental setup, we tested the system with KNN, SVM, logistic regression and Naïve Bayes classifiers. The 3D accelerometer is worn on the dominant wrist of the subject. The sensor, which yields uncalibrated data, is connected to the PC through a USB interface. Acceleration values of 8192 and -8192 correspond to 1g and -1g respectively. In the sensor configuration, parameters related to calibration and incompatibility on axes are not set, hence drifting is possible at a rate around 10% resulting from the nature of MEMS. Training and prediction data are acquired with a rate of 24 samples per second.

In data collection phase, two subjects, where subject 1 is female and subject 2 is male, are

involved. The profiles of the subjects are significantly different from each other. Besides the difference in the gender, subject 2 is actively engaged in sports, whereas subject 1 is not. Also dominant wrist is different for the subjects, where subject 1 wears the sensor on the right wrist while the sensor is located on the left wrist for subject 2. We evaluated the presented method with both intra-subject and inter-subject measures. If the training and testing dataset are acquired from the same subject, the evaluation is intra-subject, whereas if the method is trained with a subject and tested on another subject, then the evaluation is inter-subject. Training dataset is composed of 55 instances of collected data and 55 instances of artificially generated data per subject. Each instance of the collected data corresponds to single occurrence of target activity, whereas each instance of artificially generated data represents outlier class, which are generated based on the target class as explained in Section 3.2. For the prediction dataset, 10 and 53 instances are acquired from subject 1 and 2 corresponding to 300 and 60 instances of target and other classes respectively. In Figure 4.5, the exercise protocol we studied is demonstrated. Figure 4.5a-4.5c illustrate the sequence of movements from the front view, where the given sequence corresponds to the target class. The subject performs the elbow flexion and extension in full range where the sensor is worn on the dominant wrist. The subject starts performing the target activity initiating the flexion of the elbow as illustrated in Figure 4.5a upon hearing the "START" command. Another person observes the subject and operates the software interfacing to the sensor. The operating person terminates data recording when the subject completes the extension of the elbow as in Figure 4.5c to complete the recording of a single training instance corresponding to the target class. Each instance of the prediction data is composed of the activity sequence given in Table 4.1. In this sequence, 36 activities occur, since the set of 15 repetitions of target activity is practiced twice and there are 6 additional activities in the category of *other*. Boundaries between the activities forming the sequence in a prediction data instance are marked in the data annotation stage thanks to the sensor interface. Markings enable each adjacent boundary to have a single segment between them, which contains exactly one occurrence of a target or other class activity, applying explicit segmentation on the activity sequence. While recording an instance of prediction data, the operating person puts a marker in the data whenever the subject completes an activity in the category of *other* and each single occurrence of the target class activity to

mark the boundaries. Each instance in the training set belonging to target and outlier class corresponds to the matrices  $I^{d_{xm}}$  and  $O^{d_{xm}}$  mentioned in Section 3.2 respectively. Similarly, each segment between two adjacent boundaries is in the form of a matrix subjected to feature extraction and classification schemes.

Table 4.1: Activity sequence including the exercise protocol and *other* actions

| Sequence no. | Activity                                    |
|--------------|---------------------------------------------|
| 1            | Leave phone onto the table                  |
| 2            | Position yourself while standing            |
| 3            | $k$ repetitions of the target activity      |
| 4            | Sit and position yourself on the chair      |
| 5            | Spend time with phone                       |
| 6            | Get up and position yourself while standing |
| 7            | $k$ repetitions of the target activity      |
| 8            | Sit and position yourself on the chair      |

### 4.3. PARTIAL AVAILABILITY (INCOMPLETENESS) OF DATA WITHIN A SEGMENT

The knowledge discovery schemes we explored in incompleteness mitigation analysis contain wavelet entropy and peak frequency as feature extraction schemes, where KNN is used for feature classification. According to our findings, the evaluated knowledge discovery schemes exhibit low ICC score, indicating that more proficient iterative classification schemes are necessary.

We collect the activity signals with a single wearable 3D accelerometer to present our evaluation, conforming to a non-invasive solution.

We present two knowledge discovery schemes - *wavelet entropy* and *peak frequency*, combining KNN with 5 neighbours for feature classification. We executed these schemes iteratively so that we can observe their behaviour in incompleteness resilience.

$FE_{compared}$ , which is used in Algorithm 3.9 and 3.10, is a generic name representing the feature extraction scheme for the compared methods and it differs for each of those methods.  $F_j^{fd}$  showing  $j$ th element of the feature vector  $F^{fd}$  and  $C_{ij}^{mxk}$  designating the element at row  $i$  and column  $j$  of the matrix  $C^{mxk}$ ,  $FE_{compared}$  is defined specifically for wavelet entropy and peak frequency.

Peak frequency feature, shown in Eq 4.5 is obtained applying Fast Fourier Transform  $FFT$  on  $C^{mxk}(:, j)$ . The maximum of the frequency components found computing the absolute value of FFT output becomes the features in  $F^{fd}$ .  $C^{mxk}(:, j)$  is padded with zeros until  $m$  becomes power of 2, prior to being fed to  $FFT$ .

$$F_j^{fd} = \max_f |FFT(C^{mxk}(:, j))|, \forall 1 \leq j \leq k \wedge fd = k \quad (4.5)$$

In wavelet entropy computation, Coiflet 4 wavelet with decomposition level 3 is determined given  $C^{mxk}(:, j)$ ,  $\forall 1 \leq j \leq k \wedge fd = k$ , which is followed by applying Shannon entropy on the coefficients to yield the feature vector  $F^{fd}$ . Shannon entropy calculation on the coefficients is given in Eq 4.6, where  $W^{tj}$  denotes coefficient vector for axis  $j$ . While computing Shannon entropy, since  $\log(0)$  is undefined,  $\log((W_i^{tj})^2)$  is evaluated as zero where  $(W_i^{tj})^2$  becomes equal to zero, for ease of computation. The logarithm operation designated as  $\log$  indicates natural logarithm.

$$F_j^{fd} = \sum_{i=1}^{t_j} (W_i^{tj})^2 * \log((W_i^{tj})^2), \quad (4.6)$$

$$\forall 1 \leq j \leq k \wedge fd = k$$

We acquired activity signals in a real testbed, where activities in the dataset constituting *medication intake* action are practiced. These activities are *drink*, *open-pill-box*, *put-glass-back* and *put-pill-in-mouth*, lasting for 7.83 s, 2.71 s, 1.96 s and 3.70 s respectively. The

human subject is instructed to conform to the specified durations of activities with the help of audio alerts. However, the periods during which the subject performs the activities do not completely match the named values for timing. These values are found to be appropriate after performing the given activities, which is the reason why they are chosen for instructing the subject.

A wearable watch (TI Chronos eZ430) with a built-in 3D accelerometer is used for collecting the data, being worn on the right wrist. Wireless communication between the sensor and PC is established with an RF access point, plugged into the PC over USB. Incorporating a 3.07 GHz processor and 1.86 GB of RAM, the PC is the unit where acceleration signals along X, Y and Z axes are received from the sensor and classified. Sensor transmits the raw activity data at a frequency of 23.26 Hz, collecting 44 samples for each simple action, yielding a dataset of 176 samples in total. The results given in Section 5.3 are delivered using each of those 176 samples in testing thanks to leave-one-out cross validation.

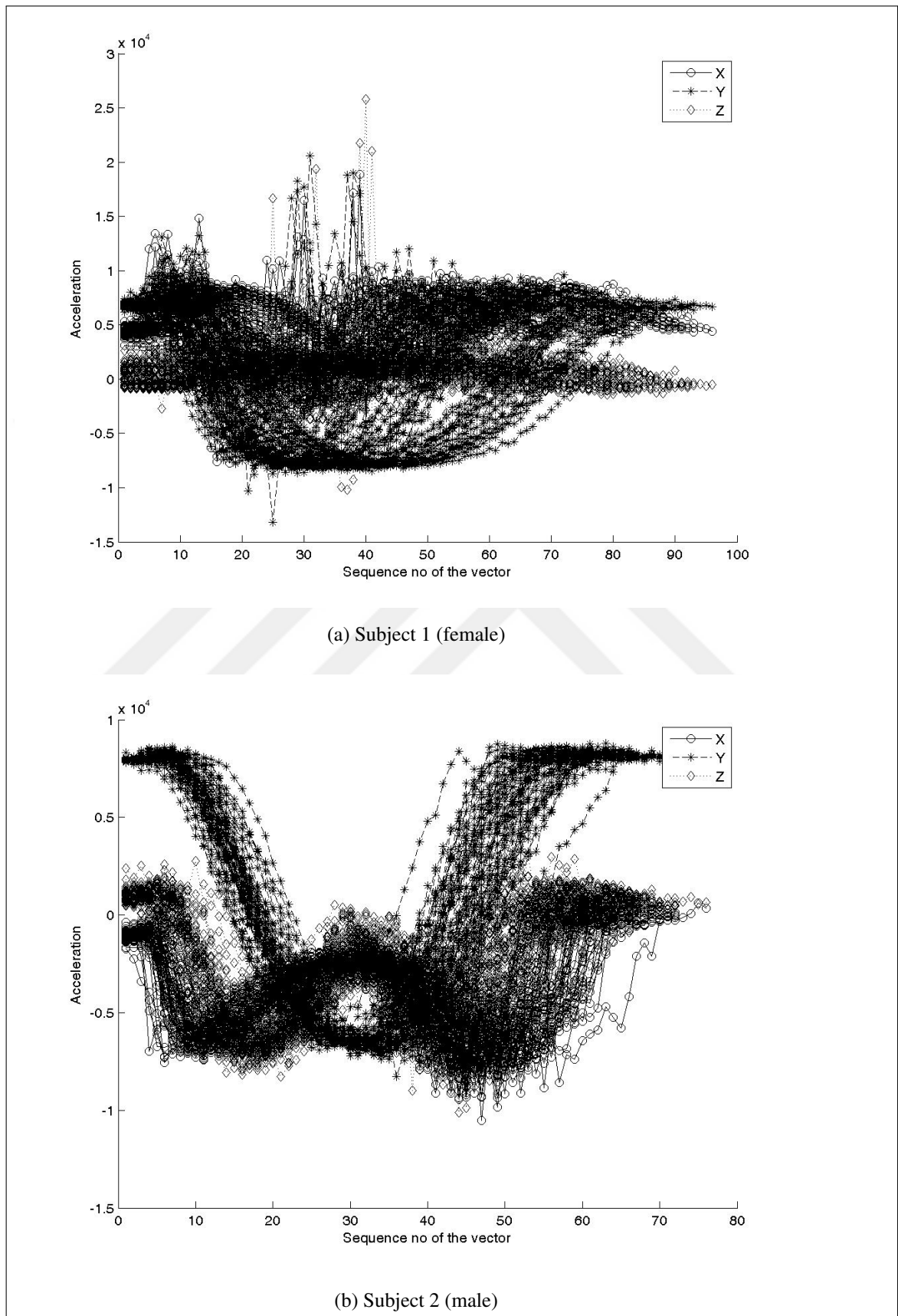


Figure 4.4: Acceleration signals corresponding to training data for the target class

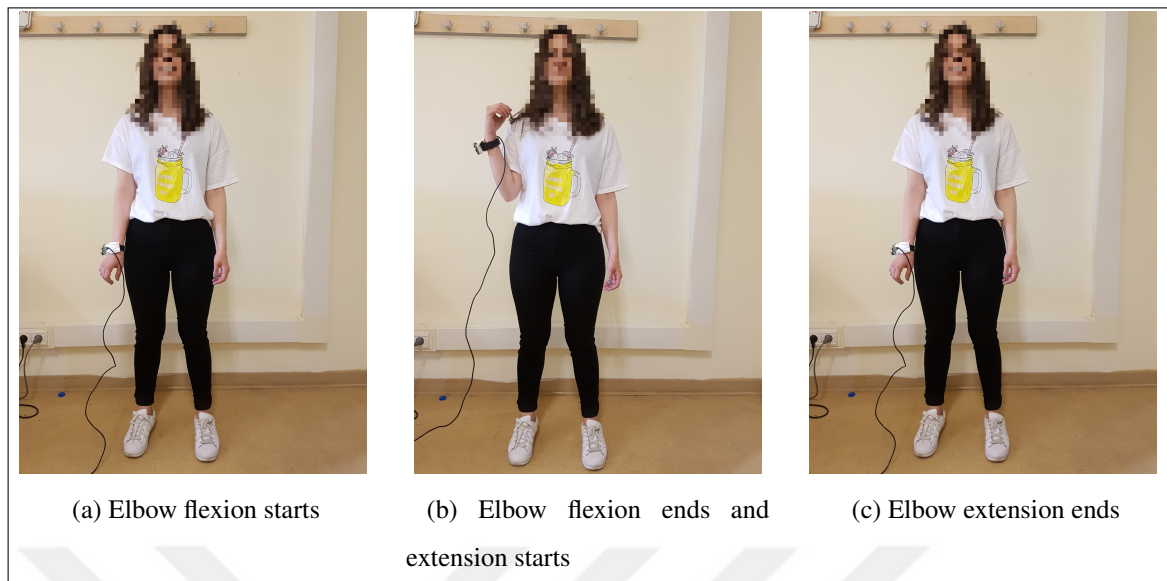


Figure 4.5: Elbow flexion and extension activities performed by a subject

## 5. RESULTS

In this chapter, we present our findings related to segmentation and ADG modules of our spotting method as well as our proposed incompleteness mitigation metric.

### 5.1. PARTITIONING CONTINUOUS SENSOR DATA AND CLASSIFYING SEGMENTS

#### 5.1.1. Evaluation Methodology

We evaluate our method and the compared methods in both intra-subject and inter-subject tests. In intra-subject evaluation, the training and test instances are acquired from the same subject whereas in inter-subject case, the training and test instances are acquired from different subjects. We abbreviate the intra-subject and inter-subject tests as Ti-Pj, which indicates that training and prediction instances are obtained from subject i and j respectively. We compute median accuracy (MA) and median true positive rate (TPR) for each case of Ti-Pj to evaluate the detection performance. MA is calculated as given in Eq 5.1, where  $acc_q$  designates accuracy for each of the  $n$  instances in the independent test set for a Ti-Pj case. In Eq 5.1,  $tp$ ,  $tn$ ,  $fp$  and  $fn$  indicate the number of true positives, true negatives, false positives and false negatives respectively. Median (MED) of  $acc_q$ ,  $\forall 1 \leq q \leq n$  equals  $MA$ . Similarly, TPR is found computing median of  $tp_q$ ,  $\forall 1 \leq q \leq n$ .

$$MA = MED(acc_q)_{q=1}^n, TPR = MED(tp_q)_{q=1}^n, \quad (5.1)$$

$$acc_q = (tp + tn) / (tp + fp + fn + tn)$$

Performance of our method NSW in comparison with other approaches in spotting the target activity within the sequence is given in Fig. 5.1 - 5.6, where subject 1 and 4 indicate the male subjects as well as subjects 2 and 3 designate female subjects. Fig. 5.1 - 5.2 and Fig. 5.5 - 5.6 present intra-subject case whereas Fig. 5.3 and Fig. 5.4 give inter-subject results.

According to our findings, our method performs best when degree is 8. The number of the samples which should exist in an accumulating window is  $d + 1$  for degree  $d$  due to general properties of polynomials. Hence, the observation that the best degree is 8 indicates the lowerbound for the number of samples which should exist in an accumulating window is 9 considering our experiments. Our method is given as N8 in short in Fig. 5.1 - 5.6, where the numerical encoding 8 next to N indicates the degree with which the method is executed. N8 outperforms the compared methods, reaching a median accuracy of 89% and a median true positive rate of 1 in both intra-subject and inter-subject cases as given in Fig. 5.1 and Fig. 5.3 respectively.

The results other than our method N8 given in Fig. 5.1-5.4 belong to compared methods whose median true positive rate is 1. Methods whose median true positive rate is 0 are excluded from Fig. 5.1-5.4 and given in Tables 5.2- 5.5 as resulting in a median true positive rate of 0 indicates that these methods cannot locate the target activity in the sequence, therefore whether these methods can locate *other* activities or not is not meaningful. The abbreviations corresponding to names of the compared methods in Fig. 5.1-5.6 and Tables 5.2- 5.5 are given in Table 5.1.

Table 5.1: Abbreviations for the compared method names

| Name of the compared method | Abbreviation |
|-----------------------------|--------------|
| Peak frequency              | P            |
| Wavelet entropy             | W            |
| Shannon entropy             | S            |
| Windowed energy             | D            |

The numerical encodings given next to the abbreviations of the compared method names in Fig. 5.1 - 5.6 and Tables 5.2- 5.5 symbolize the predetermined window size. The window sizes input to compared methods are 10, 20, 30, 40 and 50 as previously mentioned in Section 4.1, which are encoded as 1, 2, 3, 4 and 5 respectively in Fig. 5.1 - 5.6 and Tables 5.2- 5.5. It's worth reemphasizing that the meaning of numerical encoding given with our method abbreviated as N and the compared methods is different. The encoding

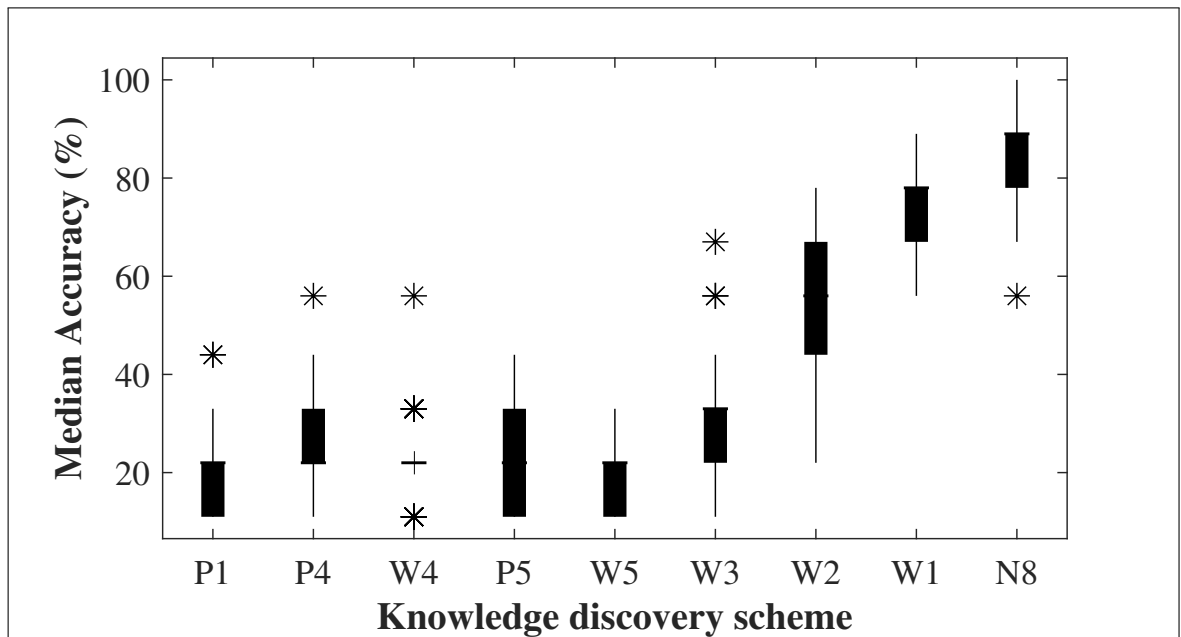


Figure 5.1: Trained on subject 1 and predicted on subject 1

Table 5.2: Methods surpassed by those in Fig. 5.1

| Method     | Accuracy (%) | TPR |
|------------|--------------|-----|
| P3         | 44           | 0   |
| P2         | 78           |     |
| S1-5, D1-5 | 89           |     |

given with our method indicate the polynomial degree whereas the encodings presented with the compared methods specify the predetermined window size.

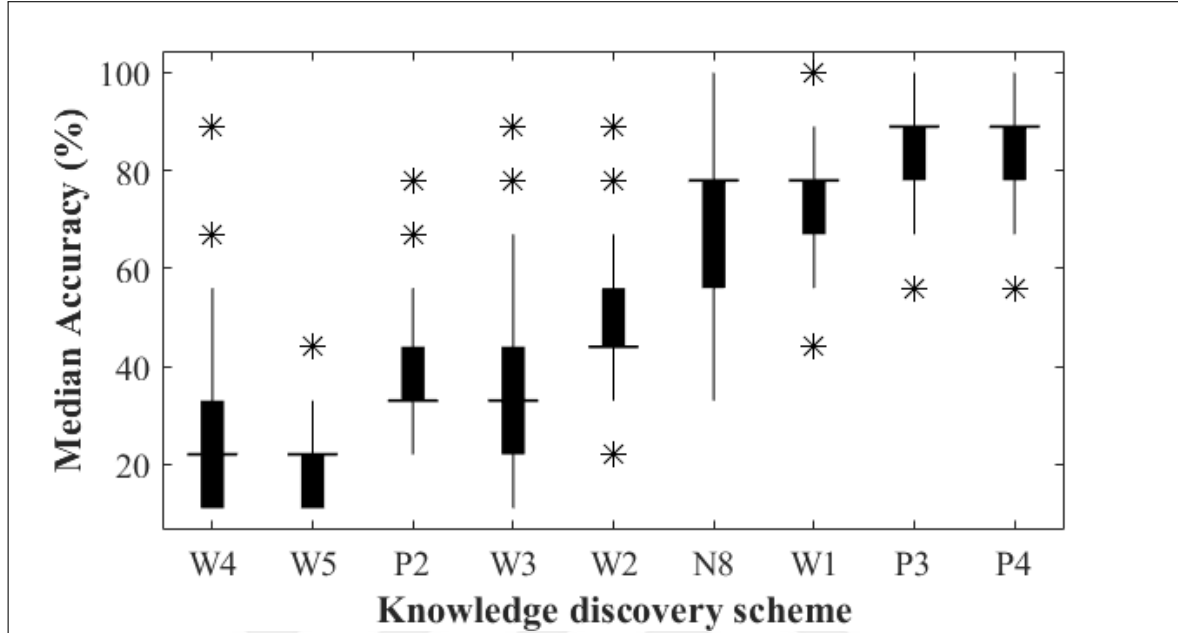


Figure 5.2: Trained on subject 2 and predicted on subject 2

Table 5.3: Methods surpassed by those in Fig. 5.2

| Method             | Accuracy (%) | TPR |
|--------------------|--------------|-----|
| P1, P5, S1-5, D1-5 | 89           | 0   |

Based on the results depicted in Fig. 5.1-5.4, peak frequency and wavelet entropy are the compared methods which manage to score a median true positive rate of 1 whereas windowed energy and Shannon entropy lead to a median true positive rate of 0.

Among the compared methods which obtain a median true positive rate of 1 and are shown in Fig. 5.1-5.4, P3 reaches a median accuracy of 89% and 78% in intra-subject and inter-subject cases respectively as illustrated in Fig. 5.2 and Fig. 5.4 as well as reaching a median true positive rate of 1 in both intra-subject and inter-subject cases. Considering these scores achieved by P3, we can conclude that P3 is the best performing method among the compared methods, however, P3 is still outperformed by our method N8. In addition, P3 cannot be an alternative of our method N8, since P3 does not even appear in Fig. 5.1 due to resulting in a

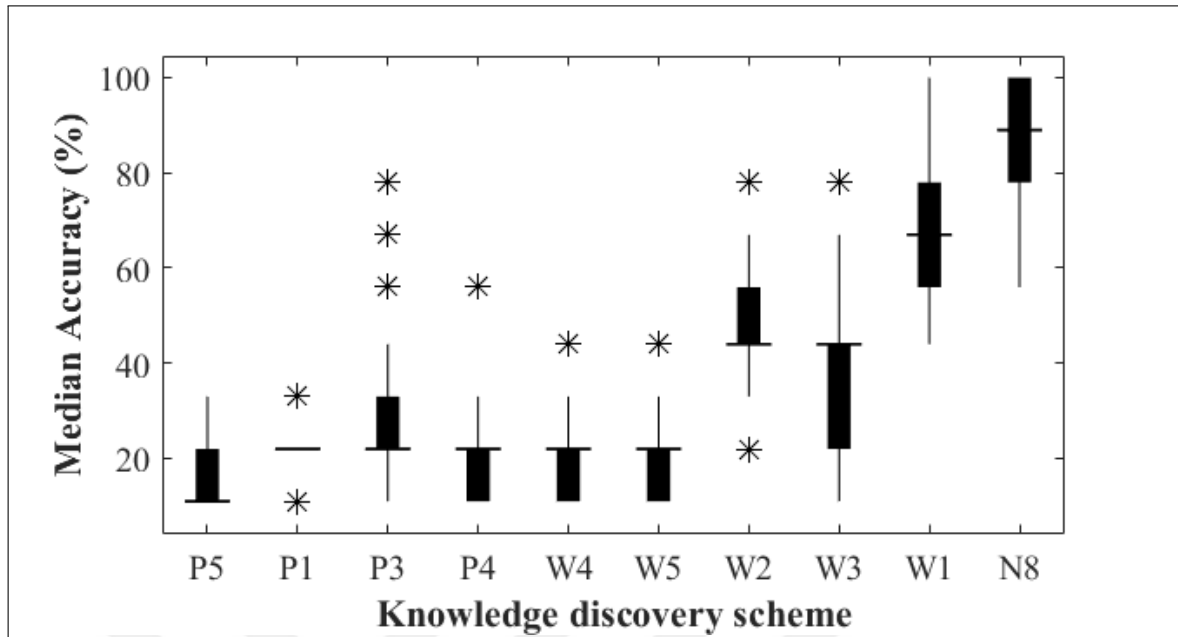


Figure 5.3: Trained on subject 1 and predicted on subject 2

Table 5.4: Methods surpassed by those in Fig. 5.3

| Method     | Accuracy (%) | TPR |
|------------|--------------|-----|
| P2         | 56           | 0   |
| S1-5, D1-5 | 89           |     |

median true positive rate of 0.

The performance of N8 in terms of the properties of the training samples are investigated as follows: As given in Fig. 5.3 median accuracy is 89% in inter-subject classification while spotting the test samples of subject 2 using training samples of subject 1. Considering that spotting the test samples of subject 2 based on training samples of subject 2 results in a lower median accuracy of 78% as presented in Fig. 5.2 despite being an instance of intra-subject case, we can conclude that training samples acquired from subject 1 are more proficient in classifying the test samples of subject 2. This observation might result from the difference in the data acquisition schedule of the subjects. Training samples of subject 1 are collected on the same day whereas acquisition of the training samples belonging to subject 2 spans across different days. Hence, the subject 2 might tend to perform the same

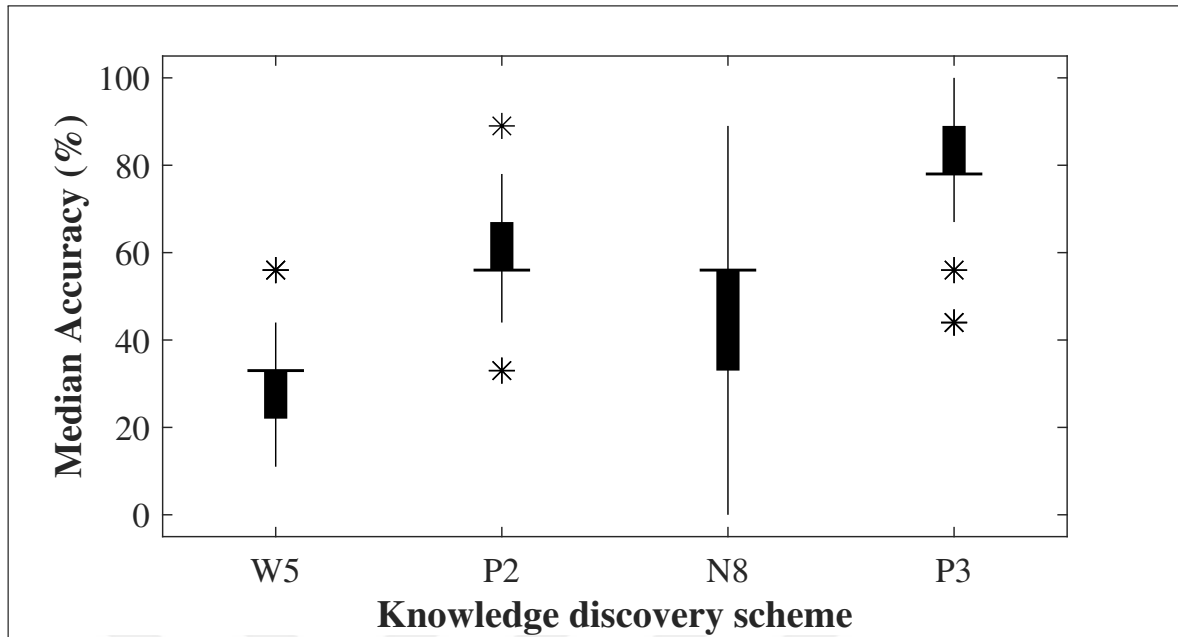


Figure 5.4: Trained on subject 2 and predicted on subject 1

Table 5.5: Methods surpassed by those in Fig. 5.4

| Method            | Accuracy (%) | TPR |
|-------------------|--------------|-----|
| W4                | 33           | 0   |
| W3                | 44           |     |
| W2                | 56           |     |
| W1                | 67           |     |
| P5                | 78           |     |
| P1,P4, S1-5, D1-5 | 89           |     |

activity differently compared to the other day. Also, the sensor drift might be a function of time, which can further deteriorate the signal belonging to the same activity, increasing the difference between the training samples belonging to the same activity when the samples are collected on different days. The difference between training samples of the same subject for the same activity influence the patterns generated, leading to the decrease observed in Fig. 5.3 compared to Fig. 5.1. Intra-subject diversification of the training signals is greater for subject 2 than subject 1, as can be seen in Fig. 4.1a and 4.1b. Spotting in the test samples for subject 1 based on training samples of subject 2 leads to a

median accuracy of 56% as designated in Fig. 5.4, which is lower than the result observed when classifying the test samples of subject 2 based on training samples of subject 1, which is 89% as given in Fig. 5.3. The decrease encountered in inter-subject performance when training samples of subject 2 are utilized might also result from the increased intra-subject diversification in the training signals of subject 2 compared to subject 1. In the cases where training samples of subject 2 are utilized, which correspond to Fig. 5.2 and 5.4, P3 reaches greater median accuracy compared to N8, however, this observation does not imply that N8 is outperformed by P3, since P3 does not even appear in Fig. 5.1 due to resulting in a median true positive rate of 0, as mentioned above.

Our method achieves a median accuracy of 67% and a median TPR of 1 in T3P3 case as given in Fig. 5.5 and Table 5.6. Though methods S2-3, D2-3 score a higher median accuracy value than our method for T3P3, they yield a median TPR of 0 in the other cases which are T1P1, T2P2, T4P4, T1P2 and T2P1 hence our method still outperforms other methods considering overall performance.

Results corresponding to the case T4P4 are shown in Fig. 5.6 and Table 5.7. Our method N8 outperforms all the other compared methods, scoring a median accuracy of 78% and a median TPR of 1. W1-5 and P1-5 reach a median TPR of 1 as well, however, their median accuracy is below 78% as shown in Fig. 5.6. The methods with worst performance are S1-5 and D1-5 in this case, since their median TPR is 0.

#### ***5.1.1.1. Determining Degree of Polynomial***

Degree of polynomial  $d$  is an input for the training and prediction stages for NSW. We experimented with the degrees 1-9 and observed that the optimal value for  $d$  is 8. Degrees which are higher than 9 are not included in our tests to avoid increasing computational complexity further though there is a chance for a degree higher than 9 could perform better. The results belonging to the degrees outperformed by degree 8 are given in Table 5.8. In Table 5.8,  $D$ ,  $A$  and  $T$  stand for degree, accuracy (%) and TPR respectively.

The number of the samples which should exist in an accumulating window is  $d+1$  for degree

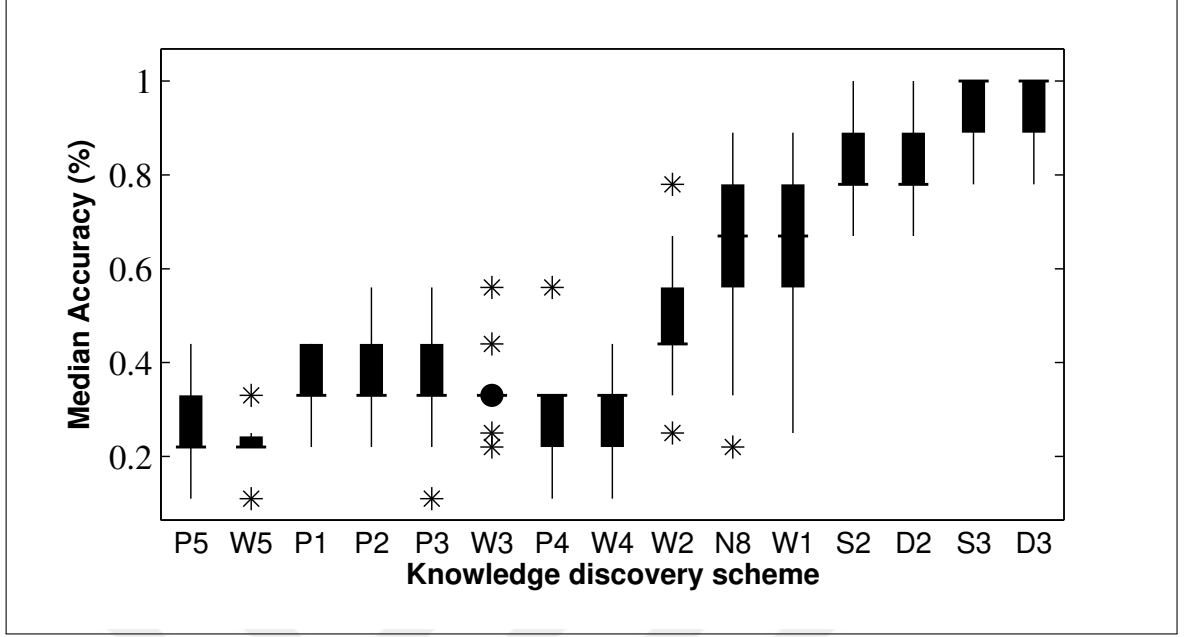


Figure 5.5: Trained on subject 3 and predicted on subject 3

Table 5.6: Methods surpassed by those in Fig. 5.5

| Method             | Accuracy (%) | TPR |
|--------------------|--------------|-----|
| S1, S4-5, D1, D4-5 | 89           | 0   |

$d$  due to general properties of polynomials. Hence, for degree  $d$ , the number of samples existing in a training instance, i.e.  $m$  for a  $C^{mxk} \in T_a$  should take  $d + 1$  as its minimum. Increasing  $d$  allows our method to process larger training instances. However, keeping the size of an individual training instance as small as possible reduces computational complexity in addition to the performance gain obtained by reducing the degree. In this regard,  $d$  being 9 indicates that the minimum number of samples existing in a training instance should be 10, which marks the point where we stop experimenting with a higher  $d$ .

### 5.1.2. Statistical Analysis

We applied two-sample Kolmogorov-Smirnov (K-S) test on the data vectors  $G^{1xn}$  and  $H^{1xn}$ , where the vectors represent the results corresponding to our method N8 and a compared method respectively. We have repeated the test for all combinations of Ti-Pj for 20 compared

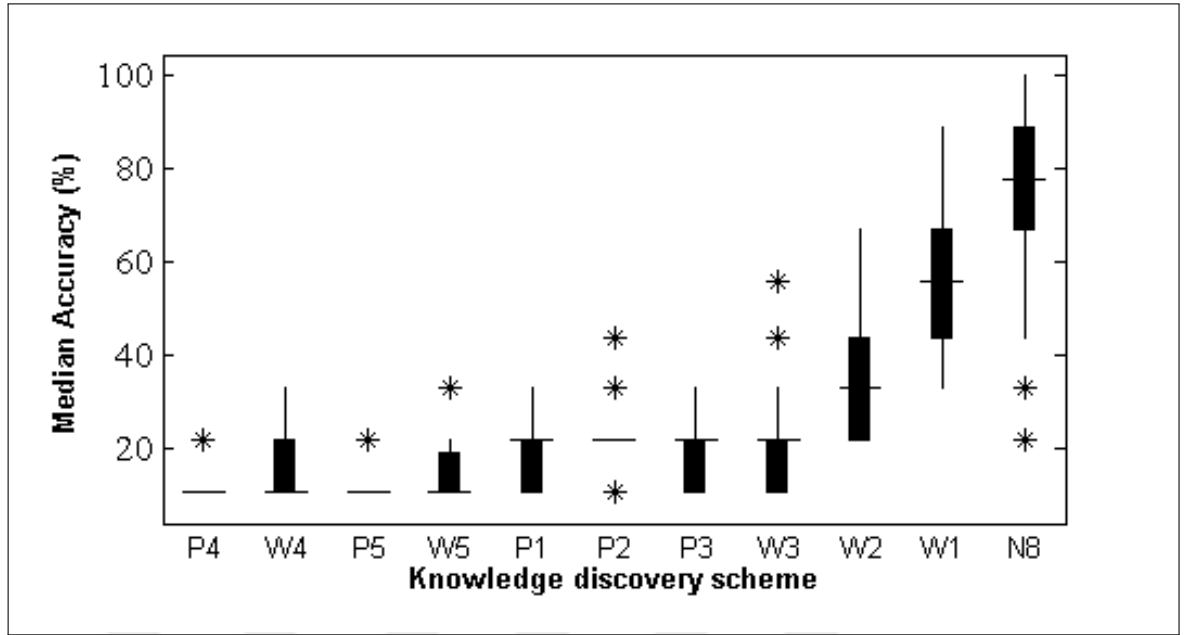


Figure 5.6: Trained on subject 4 and predicted on subject 4

Table 5.7: Methods surpassed by those in Fig. 5.6

| Method     | Accuracy (%) | TPR |
|------------|--------------|-----|
| S1-5, D1-5 | 89           | 0   |

methods yielding 120 tests in total. The motivation behind the test is to obtain a decision whether the data in the compared vectors are from the same continuous distribution or not. All tests reject the null hypothesis at the 5% significance level except the ones given in Table 5.9, verifying the significance of the results we presented. Table 5.9 presents the results of K-S test performed on the results with median accuracy.

For median true positive rate results, we perform mid-p-value McNemar test for testing the null hypothesis that  $G^{1xn}$  and  $H^{1xn}$  can equally predict at the 5% significance level. In Table 5.10, the tests for which the null hypothesis is accepted and rejected are shown. The compared method is expected to yield a median true positive rate of 1 and 0 in the tests where the null hypothesis is accepted and rejected respectively, since our method N8 resulted in a median true positive rate of 1. All tests satisfy this expectation except the tests belonging to the compared methods given in Table 5.11 where the expected and actual median true

Table 5.8: Degrees outperformed by optimal degree

| (a) T1-P1 |    |   | (b) T2-P2 |    |   | (c) T1-P2 |    |   | (d) T2-P1 |    |   |
|-----------|----|---|-----------|----|---|-----------|----|---|-----------|----|---|
| D         | A  | T | D         | A  | T | D         | A  | T | D         | A  | T |
| 6         | 11 | 1 | 1-4, 6-7  | 22 | 0 | 1-6       | 22 | 0 | 1-7       | 11 | 1 |
| 5         | 33 |   | 5, 9      | 56 |   | 9         | 78 |   | 9         | 33 |   |
| 9         | 78 |   |           |    |   | 7         | 89 | 1 |           |    |   |
| 7         | 89 |   |           |    |   |           |    |   |           |    |   |
| 1-4       | 11 | 0 |           |    |   |           |    |   |           |    |   |

Table 5.9: Kolmogorov-Smirnov test results

| Case  | All tests reject the null hypothesis? |
|-------|---------------------------------------|
| T1-P1 | Yes                                   |
| T2-P2 | Yes except with W1                    |
| T1-P2 | Yes                                   |
| T2-P1 | Yes except with W3                    |
| T3-P3 | Yes except with W1                    |
| T4-P4 | Yes                                   |

positive rate results differ. The actual values are the results appearing for the compared methods in Fig. 5.1-5.6 and Table 5.4 whereas the expected results are the ones obtained considering Table 5.10. The compared methods whose expected results are specified as 0 in Table 5.11 yield a lower median accuracy value compared to our method N8 as can be seen in Fig. 5.1-5.6, which indicates that our method outperforms these methods also in terms of median true positive rate in addition to median accuracy. D4, whose expected and actual median true positive rate is 1 and 0 as in Table 5.11, scores a median accuracy of 89% as in Table 5.4. Hence, we can observe that D4 in T1-P2 case, performs as successfully as our method in terms of median true positive rate and median accuracy considering the analysis with mid-p-value McNemar test results since our method achieves a median true positive rate of 1 and a median accuracy of 89% as shown in Fig. 5.3.

Considering McNemar test results given in Table 5.10, in terms of median TPR, our method performs similar to S3 and D3, which yields highest median accuracy in the case of T3-P3 as can be seen in Fig. 5.5, hence our method is among the best performing methods in terms of median TPR for the T3-P3 case. In T4-P4 case, our method outperforms the other methods in terms of median TPR in addition to median accuracy as can be seen in Table 5.10 and Fig. 5.6.

Table 5.10: Mid-p-value McNemar test results

| Case  | Accepted       | Rejected                           |
|-------|----------------|------------------------------------|
| T1-P1 | W1-3, P4       | P1-3, S1-5, D1-5, W4-5, P5         |
| T2-P2 | W1-3, P2-4     | P1, P5, S1-5, D1-5, W4-5           |
| T1-P2 | W1-4, P3-4, D4 | P1-2, P5, S1-5 D1-3, D5, W5        |
| T2-P1 | P2-3           | P1, P4-5, S1-5, W1-5, D1-5         |
| T3-P3 | S3, D3         | P1-5, W1-5, D1-2, D4-5, S1-2, S4-5 |
| T4-P4 | None           | All                                |

Table 5.11: Expected and actual TPR of compared methods

| Case  | Compared method    | Expected | Actual |
|-------|--------------------|----------|--------|
| T1-P1 | P1, P5, W4-5       | 0        | 1      |
| T2-P2 | W4-5               | 0        | 1      |
| T1-P2 | P1, P5, W5         | 0        | 1      |
| T1-P2 | D4                 | 1        | 0      |
| T2-P1 | W5                 | 0        | 1      |
| T3-P3 | P1-5, W1-5, S2, D2 | 0        | 1      |
| T4-P4 | P1-5, W1-5         | 0        | 1      |

## 5.2. REPRESENTATION OF UNSEEN CLASSES WITH ARTIFICIAL DATA GENERATION

We evaluated our method in terms of median success rate (MSR) as calculated in Eq 5.2. For each of the  $n$  instances in the prediction dataset, accuracy ( $acc$ ) is computed for the

activity  $a$ , which indicates *target* or *other*. Median (MED) of  $acc_{a_i}$ ,  $\forall 1 \leq i \leq n$  yields  $MSR$ . Accuracy of instance no.  $i$  belonging to action  $a$ , which is  $acc_{a_i}$ , is the percentage of correctly identified instances of class  $a$  in the sequence in instance  $i$ .  $|C|_{a_i}$  designates the number of correctly identified instances, which belong to class  $a$ , whereas  $|W|_{a_i}$  indicates the number of instances of class  $a$  in instance no.  $i$ . The value of  $n$  for subjects 1 and 2 are 10 and 53 respectively whereas  $|W|_{a_i}$  is 30 for target class and 6 for other class  $\forall 1 \leq i \leq n$ .

$$MSR = MED(acc_{a_i})_{i=1}^n,$$

(5.2)

$$acc_{a_i} = \frac{|C|_{a_i}}{|W|_{a_i}} \times 100, \forall a \in A, A = \{target, other\}$$

Figure 5.7 and 5.8 illustrate the successful detection rate of ADG when used with different classifiers. SVM emerges as the best performing classifier among the ones we study considering intra-subject and inter-subject evaluations on a per-subject basis as given in Figure 5.7. Knowing that one-class SVM is a method commonly utilized for its flexibility in the recognition task [125], our results also verify the success of SVM in an OCC problem. SVM achieves a median success rate of 100% and 83,3% in detection of target and other classes respectively in intra-subject evaluation (Figure 5.7a and 5.7c) as well as reaching a median success rate of 90% and 100% in detection of target and other classes respectively in inter-subject case (Figure 5.7b and 5.7d). Collecting training data from multiple subjects, whose profile match the subject yielding the data on which prediction is carried out is mentioned as a way for inter-subject prediction in [31]. Instead, in our work, we performed inter-subject classification where training data are collected from a single subject, whose profile significantly differs, to also address the scalability of the approach in terms of subjectwise diversification as well.

KNN is observed to slightly outperform SVM in intra-subject case, achieving a median success rate of 96,7% and 100% in recognizing target and other classes respectively. However, SVM performs much better than KNN in inter-subject evaluation. In inter-subject evaluation, we can see that in classification of the prediction instances belonging to subject 1

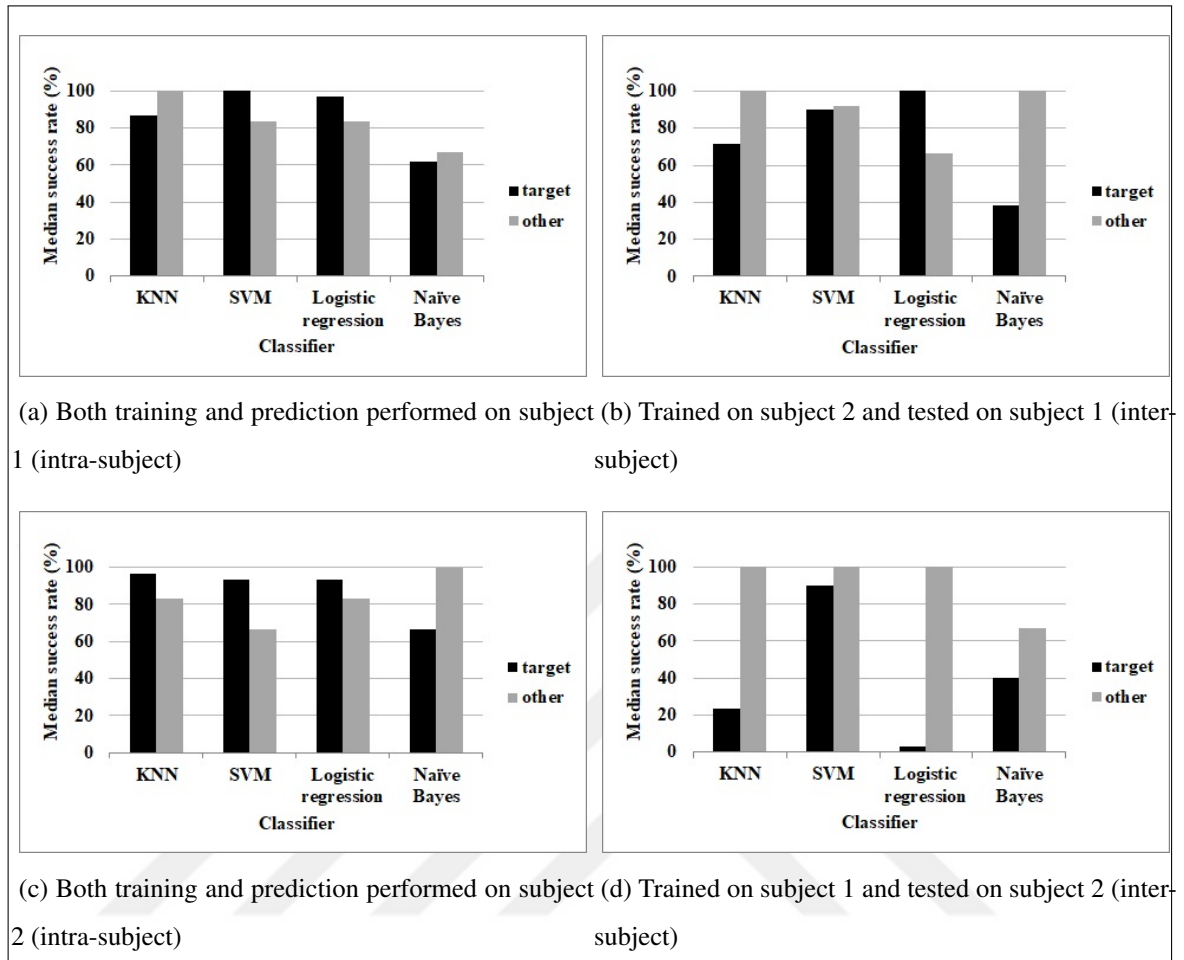


Figure 5.7: Successful detection rate of ADG on a per-subject basis

as given in Figure 5.7b, all classifiers detect both target and other classes well except Naive Bayes detecting only the other class successfully, which is reasonable since Naive Bayes is observed to be surpassed by other classifiers in intra-subject case for detecting the target class for both subjects as illustrated in Figure 5.7a and 5.7c. Inter-subject success rates achieved in classifying prediction instances of subject 1 and 2 highly differ in detection of target class with KNN and logistic regression as can be seen in Figure 5.7b and Figure 5.7d respectively, showing that SVM is more robust for handling differences related to profiles of the subjects than the compared classifiers.

The overall results across 2 subjects are given in Figure 5.8, where the artificial data generation method is validated based on the results obtained with the studied classifiers

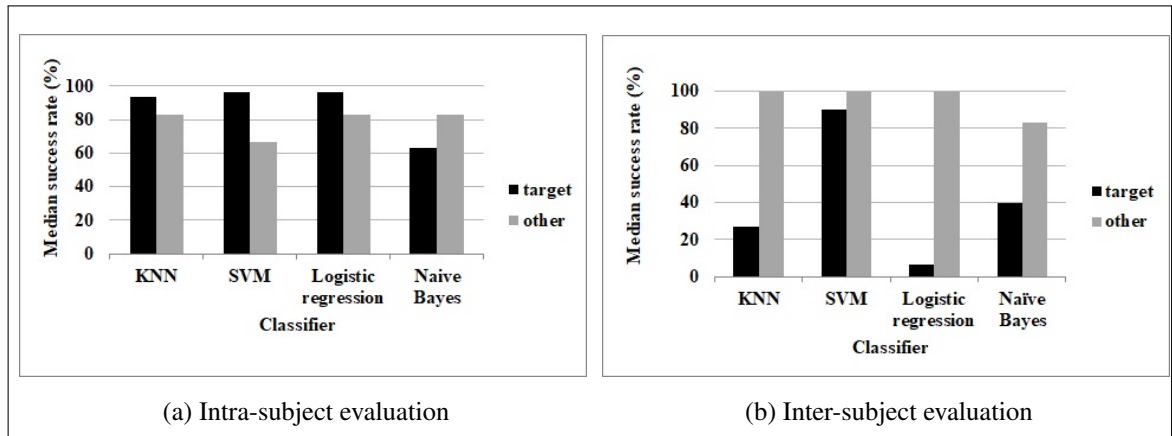


Figure 5.8: Successful detection rate of ADG across 2 subjects

KNN, SVM, logistic regression and Naïve Bayes on the features generated with Wavelet entropy. All classifiers exceed median success rate of 63,3% and 66,7% in detecting the target and other classes respectively with intra-subject evaluation considering both subjects (Figure 5.8a). In the overall results for inter-subject evaluation with both subjects as illustrated in Figure 5.8b, the classifiers except SVM are considerably influenced by differences related to profiles of the subjects in detecting target class, where SVM still achieves a median success rate of 90% and 100% in detection of target and other classes respectively. The overall results are based on all 63 instances of the activity sequence given in Table 4.1.

In addition to the results verifying our one-class classification approach with artificial negative data generation in Figure 5.7 and 5.8, we performed another analysis to observe the performance of a one-class classifier where architecture of an existing classification scheme is adapted to inject one-class capability. In this analysis, we run the one-class SVM implementation provided by LIBSVM [107] in Weka with default settings where the classifier is trained with only target class instances and tested with instances belonging to both target and other classes. The instances fed to the classifier in training and prediction phases are the same as those of the analysis with our artificial data generation scheme. According to the results obtained on one-class SVM of LIBSVM, all instances of target and other classes are predicted to belong to the other class. Since all target class

instances are detected as other class, predicting other class instances correctly cannot be regarded as a successful result. This experiment verifies that parameter selection in one-class classifiers where architecture of existing classifiers are adapted to inject one-class classification capability needs special consideration as also indicated in [131] [154], which severely deteriorates the practicability of the approach. In our one-class classification scheme with artificial data generation method, we did not undertake parameter selection procedures specific to one-class classification and verified our method based on default settings provided by Weka tool, which demonstrates the practicability of our approach.

### 5.3. INCOMPLETENESS MITIGATION RATE

ICC results are shown for compared methods in Table 5.12, verifying that a more proficient method in adaptation to activity type with incompleteness mitigation problem is necessary, as performance of these methods are inadequate.

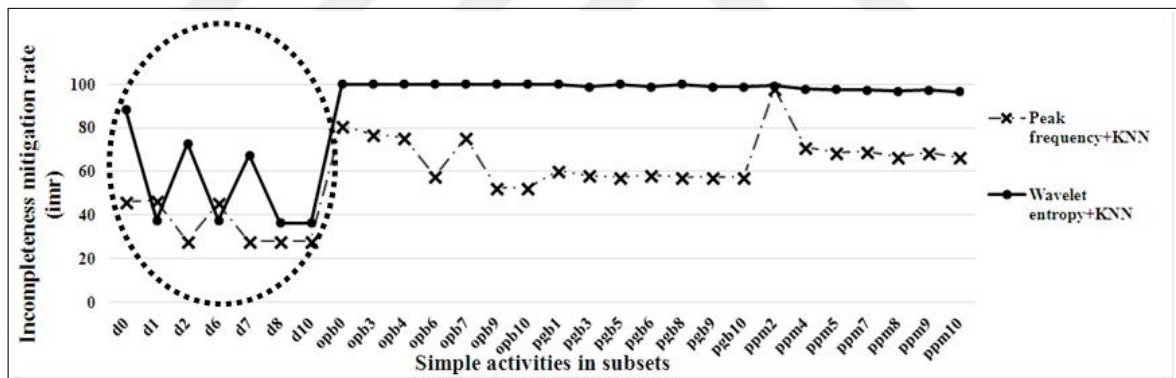


Figure 5.9: *imr* for simple activities in subsets

Table 5.12: ICC for compared methods

| Knowledge discovery scheme | ICC   |
|----------------------------|-------|
| PeakFrequency+KNN          | 0,098 |
| WaveletEntropy+KNN         | 0,007 |

Table 5.13 - 5.14 show *imr* ratings of peak frequency and wavelet entropy respectively, coupled with KNN, which are used to generate the ICC results depicted in Table 5.12. The

*Subset* column in Table 5.13 - 5.14 indicates  $S_i$ , while the ratings correspond to  $imr_{i,a}$  in Eq 3.5,  $a$  designating one of the simple actions *drink* ( $d$ ), *open-pill-box* ( $opb$ ), *put-glass-back* ( $pgb$ ) and *put-pill-in-mouth* ( $ppm$ ). Ratings show that *imr* for an action  $a$  in a subset varies when the types of actions in the subset changes. Evaluating the inter-action reliability with ICC on these ratings, where the raters are simple actions, we can observe that the configuration with peak frequency achieves better adaptation to varying activity types. However, still much improvement is necessary. Figure 5.9 shows *imr* for both compared knowledge discovery schemes. The labels in the axis named *Simple activities in subsets* stand for the simple action and the subset which contains it. For example,  $d6$  indicates that the specified *imr* result belongs to *drink* action when it is in subset 6.

Table 5.13: *imr* ratings for peak frequency+KNN

| Subset | d    | opb  | pgb  | ppm  |
|--------|------|------|------|------|
| 0      | 46,2 | 80,9 | -    | -    |
| 1      | 46,9 | -    | 60,6 | -    |
| 2      | 28,3 | -    | -    | 98,0 |
| 3      | -    | 77,2 | 58,2 | -    |
| 4      | -    | 75,4 | -    | 71,3 |
| 5      | -    | -    | 57,4 | 68,7 |
| 6      | 45,5 | 58,1 | 58,5 | -    |
| 7      | 28,3 | 75,4 | -    | 69,3 |
| 8      | 28,3 | -    | 57,4 | 66,6 |
| 9      | -    | 52,6 | 57,4 | 68,6 |
| 10     | 28,3 | 52,6 | 57,4 | 66,5 |

In Figure 5.9, the configuration with wavelet entropy yields very high *imr* in most of the ratings, however, as can be seen in the region enclosed within a dotted circle, the ratings regarding the *drink* action, *imr* drops, indicating that raters do not reliably agree. Though *imr* drops for both methods in the specified region, relying on the decision generated by the wavelet entropy is most dangerous, when it comes to building a segmentation-free method on top of this decision. Because the confidence that will be assigned to the method will be

high due to raters other than *drink* and for the *drink* action, the assigned high confidence will be catastrophic, which results in the configuration with wavelet entropy having lowest ICC. As another alternative, if different confidence values are assigned depending on the action type, then the success of the method will depend on the action type even more. The set of actions which coexist in each subset are summarized in Table 5.15.

Table 5.14: *imr* ratings for wavelet entropy+KNN

| Subset | d    | opb   | pgb   | ppm  |
|--------|------|-------|-------|------|
| 0      | 88,3 | 100,0 | -     | -    |
| 1      | 37,8 | -     | 100,0 | -    |
| 2      | 72,8 | -     | -     | 99,2 |
| 3      | -    | 100,0 | 98,8  | -    |
| 4      | -    | 100,0 | -     | 98,0 |
| 5      | -    | -     | 100,0 | 97,6 |
| 6      | 37,8 | 100,0 | 98,8  | -    |
| 7      | 67,4 | 100,0 | -     | 97,2 |
| 8      | 36,5 | -     | 100,0 | 96,7 |
| 9      | -    | 100,0 | 98,8  | 97,4 |
| 10     | 36,5 | 100,0 | 98,8  | 96,6 |

*imr* measures feature classification success iteration-wise. Utilizing an iterative activity detection scheme instead of the combination of a segmentation algorithm and a classifier necessitates knowing feature classification success for each iteration. So we need to assess how feasible the iterative classification schemes under consideration would be prior to build a method on top of them to process a continuous data stream. In this regard, ICC of feature classification results in terms of *imr* can guide us in selecting the method to create a strategy for detecting sequences of activities in a continuous data stream.

Our evaluation provides the insight that a method with higher ICC is a better candidate for being a module of an iterative activity sequence detection procedure, exhibiting the necessary robustness against varying activity types. We also evaluated maximum percentage

Table 5.15: Simple actions coexisting in the subsets

| Subset | Simple actions   |
|--------|------------------|
| 0      | d, opb           |
| 1      | d, pgb           |
| 2      | d, ppm           |
| 3      | opb, pgb         |
| 4      | opb, ppm         |
| 5      | pgb, ppm         |
| 6      | d, opb, pgb      |
| 7      | d, opb, ppm      |
| 8      | d, pgb, ppm      |
| 9      | opb, pgb, ppm    |
| 10     | d, opb, pgb, ppm |

of incomplete data ( $R$ ) for simple activities in subsets.  $R$  is a slightly modified version of  $imr$ , such that in computation of  $R$ ,  $c$  indicates the number of correctly identified iterations at the end of a chunk, where no incorrectly identified iterations intervene between two correctly identified iterations. A greater value of  $R$  is better since it expresses that the classifier starts to generate correct decisions constantly at earlier iterations, without intervention of incorrectly labelled iterations. According to our findings, the variation of  $R$  is similar to that of  $imr$  depending on the activity types. Hence,  $imr$  also reflects maximum percentage of incomplete data.

## 6. CONCLUSIONS

Adherence to prescribed healthcare routines can be challenging for patients because of unintentional reasons. HAR can yield solutions for monitoring adherence in AAL context. Keeping track of adherence to target activities performed within the course of everyday routines in daily living settings requires monitoring individuals continuously.

Continuous monitoring of human activities necessitates performing knowledge discovery on sensor data with a sequence of activities and transitions. A challenging task in such a knowledge discovery scheme is spotting the target activity in the sequence without applying a pre-determined window size. Sliding window based methods operate well with an optimal segment size chosen considering the type and duration of the activities. Nevertheless, activity type and duration are usually unknown to the detection unit in practice.

In our work, we propose a new method to spot the target activity performed in a sequence of activities. The novelty of our approach is that we come up with a window accumulation method built on progress based features where time domain progress of polynomial based features are classified in multi-layer training and prediction stages. The method is designed with one-class classification capability which enables spotting the target activity where unseen classes only exist in the prediction stage with training data collected for only the target class. We verified our method with a proof-of-concept use case where data are acquired by a single wrist-worn 3D accelerometer. The knowledge discovery schemes we compared against our method are sliding windowing with the feature extraction schemes *windowed energy*, *peak frequency*, *Shannon entropy* and *wavelet entropy*. We injected dynamic processing property to the windowing in the compared knowledge discovery schemes executing the sliding windowing module with different window size values. Our method outperformed the compared schemes, reaching a median accuracy of 89% and a median true positive rate of 1 in both intra-subject and inter-subject cases based on an independent test set where each test instance contains a sequence of 9 different activities.

In order to inject OCC capability to our knowledge discovery model, we proposed a

practicable ADG module to represent unseen classes. In our OCC approach, we generated the outlier class data artificially based on the target class data. We verify the feasibility of our artificial data generation method extracting the features with Wavelet entropy and classifying the features with KNN, SVM, logistic regression and Naïve Bayes classifiers. We verified our ADG model on a use case gathered from physiotherapy context where we recognized elbow flexion and extension exercises as the target activity. The studied activity sequence includes several unseen activities performed along with the repeated instances of target activity so that the exercise protocol can be integrated to daily routines of the patients. We verified our ADG method on a setup with a wearable 3D acceleration sensor. Thanks to the OCC characteristic, a classifier does not have to be trained for all activities encountered in a daily living environment and improved scalability is achieved.

Iterative processing contributes to spotting target activity without utilizing a predefined window size. Processing data streaming in real-time iteratively entails that a knowledge discovery scheme should be capable of producing a decision based on incomplete data, which is incompleteness mitigation. We proposed a metric which evaluates incompleteness mitigation so that procedure of choosing from iterative knowledge discovery schemes can be assisted. To create an iterative setup, we applied iterative processing on two KNN based knowledge discovery schemes and evaluated their performance in incompleteness problem raised by iterative processing. The compared knowledge discovery schemes are peak frequency and wavelet entropy, coupled with KNN for feature classification. While choosing the knowledge discovery solution to deploy, adaptation to varying activity types is a facet improving the design, so we evaluated incompleteness resilience inspecting adaptation to activity types also and developed our metric based on ICC. According to our findings, the configuration with peak frequency outperforms the one with wavelet entropy in adaptation to variation in activity type for incompleteness mitigation problem regarding their ICC scores. Despite exhibiting superior performance compared to wavelet entropy, peak frequency does not yield satisfactory results, indicating the need for a more proficient classification algorithm for the same problem.

As the future work, we plan to go on with the following tasks: We will investigate strategies for tuning the parameters for our method. We will explore the capabilities of the

method in different application domains. We will extend our dataset with more subjects to study on a larger population. We will investigate the performance of other artificial data generation methods in the same problem domain. We will explore incompleteness mitigation performance of knowledge discovery methods in iterative execution when unseen classes exist in the prediction stage using our metric.



## GLOSSARY

- $B_i^n$ :  $i$ th element in vector  $B^n$  having  $n$  elements.
- fd**: Number of elements in the feature vector  $F^{fd}$ . It is equal to number of attributes as well.
- $F^{m \times fd}$ : Feature matrix generated by applying NPFE.
- A**: set of simple actions
- $|A|$ : The number of simple actions in the set  $A$
- $Q^{|A|}$ : Activity vector such that  $Q_i^{|A|} = i$ , denotes the unique code assigned to the activity  $a, \forall a \in A$
- $T_a$ : set of training samples in the form of raw sensor data for action  $a \in A$ .
- $R^{\beta \times 2}$ : meta-training data matrix of distance values
- $\Theta^{\beta \times (fd+1)}$ : meta-training data matrix of feature vectors
- C(:,i)**:  $i$ th column of matrix **C**.
- C(i,:)**:  $i$ th row of matrix **C**.
- $PV^{d+1}$ : Vector of coefficients for a polynomial with degree  $d$
- $n_a$ : number of training instances for simple action  $a$ .
- $P_{a.dim}^{n_a \times (d+1)}$ : Polynomial coefficient matrix for dimension  $dim$ , simple action  $a$  and degree  $d$
- $\cup$ : Rowwise matrix expansion operator. Its left and right operands are a matrix and a vector respectively. The result of applying  $\cup$  operator is the input matrix appended by the input vector, as explained in Equation 3.6.
- fit**( $X^t, Y^t, \mathbf{d}$ ): Fits a polynomial with degree  $\mathbf{d}$  to the data point pairs  $(X_i^t, Y_i^t) \forall 1 \leq i \leq t$ . Then returns the vector of coefficients belonging to this polynomial. The *fit* operation requires  $t \geq d + 1$  since at least  $d + 1$  data points are necessary for fitting a polynomial of degree  $d$
- cls**: Auxiliary classifier. *cls\_training* and *cls\_prediction* are the training and prediction modules of the classifier *cls* respectively.
- $\Omega_X$ : Patterns corresponding to meta-training data matrix  $X$
- $DS_i^{fd}$ : Structure storing polynomial coefficients and degree for a single polynomial corresponding to attribute  $i$  where  $\forall 1 \leq i \leq fd$ . Coefficient vector and degree of a polynomial stored in a  $DS_i^{fd}$  structure is denoted by  $DS_i^{fd}.coef$  and  $DS_i^{fd}.deg$  respectively.

$DS^{fd}$  indicates polynomial coefficients for all attributes belonging to a single polynomial formed of the collection of  $DS_i^{fd}$  structures for  $\forall 1 \leq i \leq fd$ .

- $MS_a^{n_a}$ : Map structure storing polynomial coefficients and degree for  $n_a$  polynomials of simple action  $a$ .  $\forall 1 \leq i \leq n_a$ ,  $MS_{a,i}^{n_a}$  corresponds to  $DS^{fd}$ .

- $|\vec{I}^{fd}|$ : Magnitude of the vector  $I^{fd}$

- $NPFE$ : Feature extraction.  $NPFE$  is computed based on the feature definitions proposed in [155].



## REFERENCES

1. Mohanty R, Solanki SS, Mallick PK, Pani SK. In: Bhoi AK, Mallick PK, Liu CM, Balas VE, editors. A Classification Model Based on an Adaptive Neuro-fuzzy Inference System for Disease Prediction. Singapore: Springer Singapore; 2021. p. 131–149.
2. Gangavarapu T, S Krishnan G, Kamath S S, Jeganathan J. FarSight: Long-Term Disease Prediction Using Unstructured Clinical Nursing Notes. *IEEE Transactions on Emerging Topics in Computing*. 2020;1–1.
3. Sharma A, Garg N, Patidar S, San Tan R, Acharya UR. Automated pre-screening of arrhythmia using hybrid combination of Fourier–Bessel expansion and LSTM. *Computers in Biology and Medicine*. 2020;120:103753.
4. López-Martínez F, Núñez-Valdez ER, García-Díaz V, Bursac Z. A Case Study for a Big Data and Machine Learning Platform to Improve Medical Decision Support in Population Health Management. *Algorithms*. 2020;13(4).
5. Mulani J, Heda S, Tumdi K, Patel J, Chhinkaniwala H, Patel J. In: Dash S, Acharya BR, Mittal M, Abraham A, Kelemen A, editors. Deep Reinforcement Learning Based Personalized Health Recommendations. Cham: Springer International Publishing; 2020. p. 231–255.
6. Tragomalou A, Moschonis G, Manios Y, Kassari P, Ioakimidis I, Diou C, et al. Novel e-Health Applications for the Management of Cardiometabolic Risk Factors in Children and Adolescents in Greece. *Nutrients*. 2020;12(5).
7. Najafabadipour M, Zanin M, Rodríguez-González A, Torrente M, Nuñez García B, Cruz Bermudez JL, et al. Reconstructing the patient’s natural history from electronic health records. *Artificial Intelligence in Medicine*. 2020;105:101860.
8. Chowdhury MEH, Alzoubi K, Khandakar A, Khallifa R, Abouhasera R, Koubaa S,

- et al. Wearable Real-Time Heart Attack Detection and Warning System to Reduce Road Accidents. *Sensors*. 2019;19(12). Available from: <https://www.mdpi.com/1424-8220/19/12/2780>.
9. Li C, Hu X, Zhang L. The IoT-based heart disease monitoring system for pervasive healthcare service. *Procedia Computer Science*. 2017;112:2328–2334. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 21st International Conference, KES-20176-8 September 2017, Marseille, France.
  10. Schrader L, Vargas Toro A, Konietzny S, Rüping S, Schäpers B, Steinböck M, et al. Advanced Sensing and Human Activity Recognition in Early Intervention and Rehabilitation of Elderly People. *Journal of Population Ageing*. 2020 Jun;13(2):139–165. Available from: <https://doi.org/10.1007/s12062-020-09260-z>.
  11. Chae SH, Kim Y, Lee KS, Park HS. Development and Clinical Evaluation of a Web-Based Upper Limb Home Rehabilitation System Using a Smartwatch and Machine Learning Model for Chronic Stroke Survivors: Prospective Comparative Study. *JMIR Mhealth Uhealth*. 2020 Jul;8(7):e17216. Available from: <http://mhealth.jmir.org/2020/7/e17216/>.
  12. Piyathilaka L, Kodagoda S. In: Mejias L, Corke P, Roberts J, editors. Human Activity Recognition for Domestic Robots. Cham: Springer International Publishing; 2015. p. 395–408.
  13. Rodomagoulakis I, Kardaris N, Pitsikalis V, Mavroudi E, Katsamanis A, Tsiami A, et al. Multimodal human action recognition in assistive human-robot interaction. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; 2016. p. 2702–2706.
  14. Kim K, Hassan MM, Na S, Huh E. Dementia Wandering Detection and Activity Recognition Algorithm Using Tri-Axial Accelerometer Sensors. In: *Proceedings of the 4th International Conference on Ubiquitous Information Technologies*

*Applications*; 2009. p. 1–5.

15. Cheng W, Scotland A, Lipsmeier F, Kilchenmann T, Jin L, Schjodt-Eriksen J, et al. Human Activity Recognition from Sensor-Based Large-Scale Continuous Monitoring of Parkinson's Disease Patients. *In: 2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*; 2017. p. 249–250.
16. Cvetković B, Janko V, Romero AE, Kafalı Ö, Stathis K, Luštrek M. Activity Recognition for Diabetic Patients Using a Smartphone. *Journal of Medical Systems*. 2016 Oct;40(12):256. Available from: <https://doi.org/10.1007/s10916-016-0598-y>.
17. Morton RW, Elphick HE, Rigby AS, Daw WJ, King DA, Smith LJ, et al. STAAR: a randomised controlled trial of electronic adherence monitoring with reminder alarms and feedback to improve clinical outcomes for children with asthma. *Thorax*. 2017;72(4):347–354. Available from: <https://thorax.bmj.com/content/72/4/347>.
18. Karácsony T, Loesch-Biffar AM, Vollmar C, Noachtar S, Cunha JPS. A Deep Learning Architecture for Epileptic Seizure Classification Based on Object and Action Recognition. *In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; 2020. p. 4117–4121.
19. Sathyanarayana A, Ofli F, Fernandez-Luque L, Srivastava J, Elmagarmid A, Arora T, et al. Robust Automated Human Activity Recognition and Its Application to Sleep Research. *In: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*; 2016. p. 495–502.
20. Jeon S, Park T, Paul A, Lee Y, Son SH. A Wearable Sleep Position Tracking System Based on Dynamic State Transition Framework. *IEEE Access*. 2019;7:135742–135756.

21. Ren X, Ding W, Crouter SE, Mu Y, Xie R. Activity recognition and intensity estimation in youth from accelerometer data aided by machine learning. *Applied Intelligence*. 2016 Sep;45(2):512–529.
22. Hosseini A, Fazeli S, v Vliet E, Valencia L, Habre R, Sarrafzadeh M, et al. Children Activity Recognition: Challenges and Strategies. In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*; 2018. p. 4331–4334.
23. Jun K, Choi S. Unsupervised End-to-End Deep Model for Newborn and Infant Activity Recognition. *Sensors*. 2020;20(22). Available from: <https://www.mdpi.com/1424-8220/20/22/6467>.
24. Koceska N, Komadina R, Simjanoska M, Koteska B, Strahovnik A, Jošt A, et al. Mobile wireless monitoring system for prehospital emergency care. *European Journal of Trauma and Emergency Surgery*. 2020 Dec;46(6):1301–1308.
25. Li X, Peng J, Obaidat MS, Wu F, Khan MK, Chen C. A Secure Three-Factor User Authentication Protocol With Forward Secrecy for Wireless Medical Sensor Network Systems. *IEEE Systems Journal*. 2020 March;14(1):39–50.
26. Zhang Y, Cui J, Ma K, Chen H, Zhang J. A wristband device for detecting human pulse and motion based on the Internet of Things. *Measurement*. 2020;163:108036.
27. Zhou Z, Yu H, Shi H. Human Activity Recognition Based on Improved Bayesian Convolution Network to Analyze Health Care Data Using Wearable IoT Device. *IEEE Access*. 2020;8:86411–86418.
28. Zhuang Z, Xue Y. Sport-Related Human Activity Detection and Recognition Using a Smartwatch. *Sensors*. 2019;19(22). Available from: <https://www.mdpi.com/1424-8220/19/22/5001>.
29. Brusey J, Rednic R, Gaura EI, Kemp J, Poole N. Postural activity monitoring

- for increasing safety in bomb disposal missions. *Measurement Science and Technology*. 2009 jun;20(7):075204. Available from: <https://doi.org/10.1088/0957-0233/20/7/075204>.
30. Scheurer S, Tedesco S, Brown KN, O'Flynn B. Using Domain Knowledge for Interpretable and Competitive Multi-Class Human Activity Recognition. *Sensors*. 2020;20(4). Available from: <https://www.mdpi.com/1424-8220/20/4/1208>.
  31. Lara OD, Labrador MA. A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Communications Surveys Tutorials*. 2013 Third;15(3):1192–1209.
  32. Biagetti G, Crippa P, Falaschetti L, Turchetti C. Classifier Level Fusion of Accelerometer and sEMG Signals for Automatic Fitness Activity Diarization. *Sensors*. 2018;18(9).
  33. Chung S, Lim J, Noh KJ, Kim G, Jeong H. Sensor Data Acquisition and Multimodal Sensor Fusion for Human Activity Recognition Using Deep Learning. *Sensors*. 2019;19(7).
  34. Jordao A, Jr ACN, de Souza JS, Schwartz WR. Human Activity Recognition Based on Wearable Sensor Data: A Standardization of the State-of-the-Art. *CoRR*. 2018;abs/1806.05226. Available from: <http://arxiv.org/abs/1806.05226>.
  35. De-La-Hoz-Franco E, Ariza-Colpas P, Quero JM, Espinilla M. Sensor-Based Datasets for Human Activity Recognition – A Systematic Review of Literature. *IEEE Access*. 2018;6:59192–59210.
  36. Machot FA, Ranasinghe S, Plattner J, Jnoub N. Human Activity Recognition based on Real Life Scenarios. In: *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*; 2018. p. 3–8.

37. Zebin T, Sperrin M, Peek N, Casson AJ. Human activity recognition from inertial sensor time-series using batch normalized deep LSTM recurrent networks. *In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*; 2018. p. 1–4.
38. Uslu G, Baydere S. A segmentation scheme for knowledge discovery in human activity spotting. *IEEE Transactions on Cybernetics*; accepted, doi:10.1109/TCYB.2021.3137753.
39. Uslu G, Baydere S, Tekin A, Subaşı F. Accelerometer based classification of elbow flexion and extension exercises. *In: 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*; 2020. p. 2657–2663.
40. Uslu G, Baydere S. On the activity detection with incomplete acceleration data using iterative KNN classifier. *In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*; 2016. p. 003528–003533.
41. Vesa AV, Vlad S, Rus R, Antal M, Pop C, Anghel I, et al. Human Activity Recognition using Smartphone Sensors and Beacon-based Indoor Localization for Ambient Assisted Living Systems. *In: 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*; 2020. p. 205–212.
42. Wang A, Chen G, Yang J, Zhao S, Chang C. A Comparative Study on Human Activity Recognition Using Inertial Sensors in a Smartphone. *IEEE Sensors Journal*. 2016;16(11):4566–4578.
43. Garcia-Constantino M, Konios A, Mustafa MA, Nugent C, Morrison G. Ambient and Wearable Sensor Fusion for Abnormal Behaviour Detection in Activities of Daily Living. *In: 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*; 2020. p. 1–6.
44. Tang W, Sazonov ES. Highly Accurate Recognition of Human Postures and Activities Through Classification With Rejection. *IEEE Journal of Biomedical and*

*Health Informatics*. 2014;18(1):309–315.

45. Trabelsi D, Mohammed S, Chamroukhi F, Oukhellou L, Amirat Y. An Unsupervised Approach for Automatic Activity Recognition Based on Hidden Markov Model Regression. *IEEE Transactions on Automation Science and Engineering*. 2013;10(3):829–835.
46. Tian Y, Zhang J, Chen L, Geng Y, Wang X. Single Wearable Accelerometer-Based Human Activity Recognition via Kernel Discriminant Analysis and QPSO-KELM Classifier. *IEEE Access*. 2019;7:109216–109227.
47. Jansi R. Hierarchical Evolutionary Classification Framework for Human Action Recognition using Sparse Dictionary Optimization. *Swarm and Evolutionary Computation*. 2021:100873.
48. Catal C, Tufekci S, Pirmir E, Kocabag G. On the use of ensemble of classifiers for accelerometer-based activity recognition. *Applied Soft Computing*. 2015;37:1018–1022.
49. Mannini A, Intille SS. Classifier Personalization for Activity Recognition Using Wrist Accelerometers. *IEEE Journal of Biomedical and Health Informatics*. 2019;23(4):1585–1594.
50. Mitchell E, Monaghan D, O039;Connor NE. Classification of Sporting Activities Using Smartphone Accelerometers. *Sensors*. 2013;13(4):5317–5337. Available from: <https://www.mdpi.com/1424-8220/13/4/5317>.
51. Zhen-Yu He, Lian-Wen Jin. Activity recognition from acceleration data using AR model representation and SVM. In: *2008 International Conference on Machine Learning and Cybernetics*. vol. 4; 2008. p. 2245–2250.
52. Yazdansepar D, Niazi AH, Gay JL, Maier FW, Ramaswamy L, Rasheed K, et al. A Multi-featured Approach for Wearable Sensor-Based Human Activity Recognition.

- In: 2016 IEEE International Conference on Healthcare Informatics (ICHI); 2016. p. 423–431.*
53. Li X, Wang Y, Zhang B, Ma J. PSDRNN: An Efficient and Effective HAR Scheme Based on Feature Extraction and Deep Learning. *IEEE Transactions on Industrial Informatics*. 2020;16(10):6703–6713.
  54. Khan AM, Lee Y, Lee SY, Kim T. Human Activity Recognition via an Accelerometer-Enabled-Smartphone Using Kernel Discriminant Analysis. *In: 2010 5th International Conference on Future Information Technology; 2010. p. 1–6.*
  55. Khan AM, Lee Y, Lee SY, Kim T. A Triaxial Accelerometer-Based Physical-Activity Recognition via Augmented-Signal Features and a Hierarchical Recognizer. *IEEE Transactions on Information Technology in Biomedicine*. 2010;14(5):1166–1172.
  56. Szttyler T, Stuckenschmidt H, Petrich W. Position-aware activity recognition with wearable devices. *Pervasive and Mobile Computing*. 2017;38:281 – 295. Special Issue IEEE International Conference on Pervasive Computing and Communications (PerCom) 2016.
  57. Guo M, Wang Z. Segmentation and recognition of human motion sequences using wearable inertial sensors. *Multimedia Tools and Applications*. 2018 Aug;77(16):21201–21220.
  58. Alsheikh MA, Selim A, Niyato D, Doyle L, Lin S, Tan H. Deep Activity Recognition Models with Triaxial Accelerometers. *In: arXiv:1511.04664v2 [cs.LG] 25 Oct 2016; 2016. .*
  59. Zhang L, Wu X, Luo D. Recognizing Human Activities from Raw Accelerometer Data Using Deep Neural Networks. *In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA); 2015. p. 865–870.*
  60. Saha J, Chowdhury C, Roy Chowdhury I, Biswas S, Aslam N. An Ensemble

of Condition Based Classifiers for Device Independent Detailed Human Activity Recognition Using Smartphones †. *Information*. 2018;9(4).

61. Shoaib M, Bosch S, Incel OD, Scholten H, Havinga PJM. Complex Human Activity Recognition Using Smartphone and Wrist-Worn Motion Sensors. *Sensors*. 2016;16(4).
62. Vanrell S, Milone D, Rufiner H. Assessment of Homomorphic Analysis for Human Activity Recognition From Acceleration Signals. *IEEE Journal of Biomedical and Health Informatics*. 2017 07;PP:1–1.
63. Munoz-Organero M, Ruiz-Blazquez R. Time-Elastic Generative Model for Acceleration Time Series in Human Activity Recognition. *Sensors*. 2017;17(2).
64. Cao L, Wang Y, Zhang B, Jin Q, Vasilakos AV. GCHAR: An efficient Group-based Context—aware human activity recognition on smartphone. *Journal of Parallel and Distributed Computing*. 2018;118:67 – 80.
65. San-Segundo R, Blunck H, Moreno-Pimentel J, Stisen A, Gil-Martín M. Robust Human Activity Recognition using smartwatches and smartphones. *Engineering Applications of Artificial Intelligence*. 2018;72:190 – 202.
66. Wen J, Wang Z. Learning general model for activity recognition with limited labelled data. *Expert Systems with Applications*. 2017;74:19 – 28.
67. Wu D, Wang Z, Chen Y, Zhao H. Mixed-kernel based weighted extreme learning machine for inertial sensor based human activity recognition with imbalanced dataset. *Neurocomputing*. 2016;190:35 – 49.
68. Xu Y, Shen Z, Zhang X, Gao Y, Deng S, Wang Y, et al. Learning multi-level features for sensor-based human action recognition. *Pervasive and Mobile Computing*. 2017;40:324 – 338.

69. Baños O, Galvez JM, Damas M, Pomares H, Rojas I. Window Size Impact in Human Activity Recognition. *Sensors*. 2014;14(4):6474–6499.
70. Krishnan NC, Cook DJ. Activity recognition on streaming sensor data. *Pervasive and Mobile Computing*. 2014;10:138 – 154.
71. Fida B, Bernabucci I, Bibbo D, Conforto S, Schmid M. Varying behavior of different window sizes on the classification of static and dynamic physical activities from a single accelerometer. *Medical Engineering Physics*. 2015;37(7):705 – 711.
72. Niazi AH, Yazdansepas D, Gay JL, Maier FW, Ramaswamy L, Rasheed K, et al. Statistical Analysis of Window Sizes and Sampling Rates in Human Activity Recognition. In: *Proceedings of the 10th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2017)*; 2017. p. 319–325.
73. Wang N, Chen Y, Zhang X. Realtime recognition of multi-finger prehensile gestures. *Biomedical Signal Processing and Control*. 2014;13:262 – 269.
74. Wang L, Gu T, Tao X, Lu J. A hierarchical approach to real-time activity recognition in body sensor networks. *Pervasive and Mobile Computing*. 2012;8(1):115 – 130.
75. van Kasteren TLM, Englebienne G, Kröse BJA. An activity monitoring system for elderly care using generative and discriminative models. *Personal and Ubiquitous Computing*. 2010 Sep;14(6):489–498.
76. Zhang L, Zhao J, Li W. Online and Unsupervised Anomaly Detection for Streaming Data Using an Array of Sliding Windows and PDDs. *IEEE Transactions on Cybernetics*. 2019:1–6.
77. Ma C, Li W, Cao J, Du J, Li Q, Gravina R. Adaptive sliding window based activity recognition for assisted livings. *Information Fusion*. 2020;53:55 – 65.

78. Ahmadi MN, Pavey TG, Trost SG. Machine Learning Models for Classifying Physical Activity in Free-Living Preschool Children. *Sensors*. 2020;20(16).
79. Akbari A, Jafari R. Transition-Aware Detection of Modes of Locomotion and Transportation through Hierarchical Segmentation. *IEEE Sensors Journal*. 2020:1–1.
80. Shi J, Zuo D, Zhang Z. Transition Activity Recognition System Based on Standard Deviation Trend Analysis. *Sensors*. 2020;20(11).
81. Dehghani A, Sarbishei O, Glatard T, Shihab E. A Quantitative Comparison of Overlapping and Non-Overlapping Sliding Windows for Human Activity Recognition Using Inertial Sensors. *Sensors*. 2019;19(22).
82. Kang J, Kim J, Lee S, Sohn M. Transition activity recognition using fuzzy logic and overlapped sliding window-based convolutional neural networks. *The Journal of Supercomputing*. 2020 Oct;76(10):8003–8020.
83. Zamani M, Sadri A, Ghafoori Z, Moshtaghi M, Salim FD, Leckie C, et al. Unsupervised online change point detection in high-dimensional time series. *Knowledge and Information Systems*. 2020 Feb;62(2):719–750.
84. Taylor LE, Abdulla UA, Barlow MG, Taylor K. Detecting and exploiting periodicity in activity classification. In: *2017 IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH)*; 2017. p. 1–8.
85. Noor MHM, Salcic Z, Wang KIK. Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer. *Pervasive and Mobile Computing*. 2017;38(Part 1):41 – 59.
86. Ni Q, Patterson T, Cleland I, Nugent C. Dynamic detection of window starting positions and its implementation within an activity recognition framework. *Journal of Biomedical Informatics*. 2016;62:171 – 180.

87. Yao R, Lin G, Shi Q, Ranasinghe DC. Efficient dense labelling of human activity sequences from wearables using fully convolutional networks. *Pattern Recognition*. 2018;78:252 – 266.
88. Hamad RA, Salguero AG, Bouguelia M, Espinilla M, Quero JM. Efficient activity recognition in smart homes using delayed fuzzy temporal windows on binary sensors. *IEEE Journal of Biomedical and Health Informatics*. 2019:1–1.
89. Anderez DO, Lotfi A, Pourabdollah A. Eating and drinking gesture spotting and recognition using a novel adaptive segmentation technique and a gesture discrepancy measure. *Expert Systems with Applications*. 2020;140:112888.
90. Altuve M, Lizarazo P, Villamizar J. Human activity recognition using improved complete ensemble EMD with adaptive noise and long short-term memory neural networks. *Biocybernetics and Biomedical Engineering*. 2020.
91. Gomes D, Mendes-Moreira J, Sousa I, Silva J. Eating and Drinking Recognition in Free-Living Conditions for Triggering Smart Reminders. *Sensors*. 2019;19(12).
92. Li K, Habre R, Deng H, Urman R, Morrison J, Gilliland FD, et al. Applying Multivariate Segmentation Methods to Human Activity Recognition From Wearable Sensors' Data. *JMIR Mhealth Uhealth*. 2019 Feb;7(2):e11201.
93. Liu B, Cai H, Ju Z, Liu H. Multi-stage adaptive regression for online activity recognition. *Pattern Recognition*. 2020;98:107053.
94. Adama DA, Lotfi A, Ranson R. Adaptive Segmentation and Sequence Learning of human activities from skeleton data. *Expert Systems with Applications*. 2021;164:113836.
95. Aminikhanghahi S, Cook DJ. Enhancing activity recognition using CPD-based activity segmentation. *Pervasive and Mobile Computing*. 2019;53:75 – 89.

96. Li J, Tian L, Wang H, An Y, Wang K, Yu L. Segmentation and Recognition of Basic and Transitional Activities for Continuous Physical Human Activity. *IEEE Access*. 2019;7:42565–42576.
97. Arzani MM, Fathy M, Azirani AA, Adeli E. Switching Structured Prediction for Simple and Complex Human Activity Recognition. *IEEE Transactions on Cybernetics*. 2020:1–12.
98. Ni B, Pei Y, Moulin P, Yan S. Multilevel Depth and Image Fusion for Human Activity Detection. *IEEE Transactions on Cybernetics*. 2013 Oct;43(5):1383–1394.
99. Franco A, Magnani A, Maio D. A multimodal approach for human activity recognition based on skeleton and RGB data. *Pattern Recognition Letters*. 2020;131:293 – 299.
100. Jaouedi N, Perales FJ, Buades JM, Boujnah N, Bouhlef MS. Prediction of Human Activities Based on a New Structure of Skeleton Features and Deep Learning Model. *Sensors*. 2020;20(17).
101. Li J, Xie X, Pan Q, Cao Y, Zhao Z, Shi G. SGM-Net: Skeleton-guided multimodal network for action recognition. *Pattern Recognition*. 2020;104:107356.
102. Liu B, Cai H, Ju Z, Liu H. RGB-D sensing based human action and interaction analysis: A survey. *Pattern Recognition*. 2019;94:1 – 12.
103. Poularakis S, Katsavounidis I. Low-Complexity Hand Gesture Recognition System for Continuous Streams of Digits and Letters. *IEEE Transactions on Cybernetics*. 2016 Sep;46(9):2094–2108.
104. Pan T, Tsai W, Chang C, Yeh C, Hu M. A Hierarchical Hand Gesture Recognition Framework for Sports Referee Training-Based EMG and Accelerometer Sensors. *IEEE Transactions on Cybernetics*. 2020:1–12.

105. Witten IH, Frank E, Hall MA. In: *Data Mining-Practical Machine Learning Tools and Techniques*. MA, USA: Morgan Kaufmann; 2011. .
106. Matsubara Y, Nishimura H, Samura T, Yoshimoto H, Tanimoto R. Screen Unlocking by Spontaneous Flick Reactions with One-Class Classification Approaches. *In: 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*; 2016. p. 752–757.
107. Chang CC, Lin CJ. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*. 2011;2:27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
108. Khan SS, Madden MG. A Survey of Recent Trends in One Class Classification. *In: Proceedings of the 20th Irish Conference on Artificial Intelligence and Cognitive Science*. AICS'09. Berlin, Heidelberg: Springer-Verlag; 2010. p. 188–197.
109. Das B, Cook DJ, Krishnan NC, Schmitter-Edgecombe M. One-Class Classification-Based Real-Time Activity Error Detection in Smart Homes. *IEEE Journal of Selected Topics in Signal Processing*. 2016 Aug;10(5):914–923.
110. Schölkopf B, Platt JC, Smola AJ. Kernel Method for Percentile Feature Extraction. *Microsoft Research, Tech Rep 22*. 2000.
111. Guha A, Samanta D. Hybrid Approach to Document Anomaly Detection: An Application to Facilitate RPA in Title Insurance. *International Journal of Automation and Computing*. 2021 Feb;18(1):55–72.
112. Ribeiro M, Gutoski M, Lazzaretti AE, Lopes HS. One-Class Classification in Images and Videos Using a Convolutional Autoencoder With Compact Embedding. *IEEE Access*. 2020;8:86520–86535.
113. Girard N, Trullo R, Barrat S, Ragot N, Ramel JY. Interactive Definition and Tuning of One-Class Classifiers for Document Image Classification. *In: 2016 12th IAPR*

*Workshop on Document Analysis Systems (DAS)*; 2016. p. 358–363.

114. Mygdalis V, Iosifidis A, Tefas A, Pitas I. One class classification applied in facial image analysis. *In: 2016 IEEE International Conference on Image Processing (ICIP)*; 2016. p. 1644–1648.
115. Kim SH, Kim HJ, Kim JY. GAN-Based One-Class Classification for Personalized Image Retrieval. *In: 2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*; 2018. p. 771–774.
116. Perera P, Patel VM. Learning Deep Features for One-Class Classification. *IEEE Transactions on Image Processing*. 2019;28(11):5450–5463.
117. Chen D, Yan C, Wang M. A New One-Class Classification Method with Multiple Encoder-Decoder Pairs for Images. *In: 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. vol. 2; 2019. p. 116–121.
118. Chriki A, Touati H, Snoussi H, Kamoun F. Deep learning and handcrafted features for one-class anomaly detection in UAV video. *Multimedia Tools and Applications*. 2021 Jan;80(2):2599–2620.
119. Sun J, Shao J, He C. Abnormal event detection for video surveillance using deep one-class learning. *Multimedia Tools and Applications*. 2019 Feb;78(3):3633–3647.
120. Shabbir S, Malik MI, Siddiqi I. Offline Signature Verification Using Feature Learning and One-Class Classification. *In: Djeddi C, Kessentini Y, Siddiqi I, Jmaiel M, editors. Pattern Recognition and Artificial Intelligence*. Cham: Springer International Publishing; 2021. p. 242–254.
121. Cabral GG, de Oliveira ALI. One-class Classification for heart disease diagnosis. *In: 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*; 2014. p. 2551–2556.

122. Fernández-Francos D, Fontenla-Romero Alonso-Betanzos A. One-Class Convex Hull-Based Algorithm for Classification in Distributed Environments. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2017;PP(99):1–11.
123. Zhang C, Kuppannagari SR, Kannan R, Prasanna VK. Generative Adversarial Network for Synthetic Time Series Data Generation in Smart Grids. *In: 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*; 2018. p. 1–6.
124. Rivera AR, Khan A, Bekkouch IEI, Sheikh TS. Anomaly Detection Based on Zero-Shot Outlier Synthesis and Hierarchical Feature Distillation. *IEEE Transactions on Neural Networks and Learning Systems*. 2020:1–11.
125. Wang S, Liu Q, Zhu E, Porikli F, Yin J. Hyperparameter selection of one-class support vector machine by self-adaptive data shifting. *Pattern Recognition*. 2018;74(Supplement C):198 – 211.
126. Krawczyk B, Woźniak M. Dynamic classifier selection for one-class classification. *Knowledge-Based Systems*. 2016;107(Supplement C):43 – 53.
127. Liu J, Miao Q, Sun Y, Song J, Quan Y. Modular ensembles for one-class classification based on density analysis. *Neurocomputing*. 2016;171:262–276.
128. Krawczyk B. Forming Ensembles of Soft One-Class Classifiers with Weighted Bagging. *New Generation Computing*. 2015 Jul;33(4):449–466.
129. Krawczyk B, Woźniak M. Accuracy and diversity in classifier selection for one-class classification ensembles. *In: 2013 IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL)*; 2013. p. 46–51.
130. Fatemifar S, Awais M, Arashloo SR, Kittler J. Combining Multiple one-class Classifiers for Anomaly based Face Spoofing Attack Detection. *In: 2019 International Conference on Biometrics (ICB)*; 2019. p. 1–7.

131. Hempstalk K, Frank E, Witten IH. In: Daelemans W, Goethals B, Morik K, editors. One-Class Classification by Combining Density and Class Probability Estimation. Berlin, Heidelberg: Springer Berlin Heidelberg; 2008. p. 505–519.
132. Kim JY, Park G, Lee SA, Nam Y. Analysis of Machine Learning-Based Assessment for Elbow Spasticity Using Inertial Sensors. *Sensors*. 2020;20(6).
133. Zhu ZA, Lu YC, You CH, Chiang CK. Deep Learning for Sensor-Based Rehabilitation Exercise Recognition and Evaluation. *Sensors*. 2019;19(4).
134. Chen C, Lin Y, Lin C, Chen Y, Lee Y, Wang C. Rehabilitation System for Limbs using IMUs. In: *2020 21st International Symposium on Quality Electronic Design (ISQED)*; 2020. p. 285–291.
135. Tang D. Hybridized Hierarchical Deep Convolutional Neural Network for Sports Rehabilitation Exercises. *IEEE Access*. 2020;8:118969–118977.
136. Gao W, Guo S, Bu D. Convolution Neural Network (CNN)-based Upper Limb Action Recognition. In: *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*; 2020. p. 1576–1580.
137. Song Z, Wang Z, Guo S. SEMG-based posture recognition of elbow flexion and extension. In: *2013 ICME International Conference on Complex Medical Engineering*; 2013. p. 448–452.
138. Wang Z, Guo S, Gao B, Song X. Posture recognition of elbow flexion and extension using sEMG signal based on multi-scale entropy. In: *2014 IEEE International Conference on Mechatronics and Automation*; 2014. p. 1132–1136.
139. Rubiano A, Ramírez JL, Korso MNE, Jouandeau N, Gallimard L, Polit O. Features selection and classification to estimate elbow movements. *Journal of Physics: Conference Series*. 2015;657(1):012012.

140. Bobin M, Amroun H, Coquillart S, Bimbard F, Ammi M. DNN based Approach for the Assessment of Elbow Flexion with Smart Textile Sensor. *In: IEEE International Conference on Systems, Man, and Cybernetics SMC 2017, Banff, Canada. hal-01677679v2*; 2017. .
141. Olubanjo T, Ghovanloo M. Real-time swallowing detection based on tracheal acoustics. *In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference*; 2014. p. 4384–4388.
142. Reliability;. Accessed: 14.1.2016. <http://www.acsu.buffalo.edu/~wdmccall/os512d/reliabilityhtml.html>.
143. Questionnaire Development;. Accessed: 14.1.2016. [http://www.kcl.ac.uk/ioppn/depts/biostatistics/SAS/faqs9.aspx#a9\\_2](http://www.kcl.ac.uk/ioppn/depts/biostatistics/SAS/faqs9.aspx#a9_2).
144. Shrout PE, Fleiss JL. Intraclass Correlations : Uses in Assessing Rater Reliability. *Psychological Bulletin*. 1979;86(2):420–428.
145. Real Statistics Using Excel;. Accessed: 14.1.2016. <http://www.real-statistics.com/reliability/intraclass-correlation/>.
146. He Z. Activity recognition from accelerometer signals based on Wavelet-AR model. *In: 2010 IEEE International Conference on Progress in Informatics and Computing*. vol. 1; 2010. p. 499–502.
147. Liang Y, Zhou X, Yu Z, Guo B. Energy-Efficient Motion Related Activity Recognition on Mobile Devices for Pervasive Healthcare. *Mobile Networks and Applications*. 2014 Jun;19(3):303–317.
148. Mannini A, Sabatini AM. Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers. *Sensors*. 2010;10(2):1154–1175.
149. Parkka J, Ermes M, Korpiä P, Mantyjarvi J, Peltola J, Korhonen I. Activity

- classification using realistic data from wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*. 2006 Jan;10(1):119–128.
150. Attal F, Mohammed S, Dedabrishvili M, Chamroukhi F, Oukhellou L, Amirat Y. Physical Human Activity Recognition Using Wearable Sensors. *Sensors*. 2015;15(12):31314–31338.
  151. Hamäläinen W, Järvinen M, Martiskainen P, Mononen J. Jerk-based feature extraction for robust activity recognition from acceleration data. *In: 2011 11th International Conference on Intelligent Systems Design and Applications*; 2011. p. 831–836.
  152. Quiroz JC, Banerjee A, Dascalu SM, Lau SL. Feature Selection for Activity Recognition from Smartphone Accelerometer Data. *Intelligent Automation & Soft Computing*. 2017;0(0):1–9.
  153. Kiss C, Bichler M. Data Mining and Campaign Management in the Telecommunications Industry. *In: Systems PotMCoI, editor. MKWI 2004; Track: Coordination and Agent Technology in Value Networks*. GITO 2004; 2004. p. 61–74.
  154. Wang S, Liu Q, Zhu E, Porikli F, Yin J. Hyperparameter selection of one-class support vector machine by self-adaptive data shifting. *Pattern Recognition*. 2018;74(Supplement C):198 – 211.
  155. Uslu G, Baydere S. RAM: Real Time Activity Monitoring with feature extractive training. *Expert Systems with Applications*. 2015;42(21):8052 – 8063.