SMOKING RECOGNITION USING SMARTWATCH SENSORS

(AKILLI SAAT SENSÖRLERİ KULLANARAK SİGARA İÇMEYİ TANIMA)

by

SÜMEYYE AĞAÇ, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

JAN 2019

Approval of the thesis:

SMOKING RECOGNITION USING SMARTWATCH SENSORS

submitted by **SÜMEYYE AĞAÇ** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Galatasaray University** by,

Examining Committee Members:

Assoc. Prof. Özlem DURMAZ İNCEL
Supervisor, **Computer Engineering Department, GSU**

Prof. Cem ERSOY
Computer Engineering Department, BU

Assist. Prof. Günce Keziban ORMAN
Computer Engineering Department, GSU

Date:

ACKNOWLEDGMENTS

I would like to express my sincere appreciation and thanks to my supervisor, Özlem DURMAZ İNCEL, who was very attentive and available despite her many responsibilities. I would like to thank her for encouraging me on my thesis and for opening the door to be a good researcher since the last year of my BSc. With her, for the first time, I experienced to have a submitted paper and I got my first rejection. These moments are unforgettable and precious to me, I learned a lot about the academia from her. My interest to do the research was increasing day by day thanks to her. Her guidance and expert advice on both research as well as on my career are invaluable.

I have also benefited from comments and suggestions made by Muhammad SHOAIB in the development of this work and I take the opportunity to thank him.

I would also like to thank the University of Galatasaray, the Department of Computer Engineering and my professors for these years of teaching.

I would like to extend tremendous thanks to my parents, Perihan and İbrahim, my sister Esra and my brother İsmail who have given me all the facilities for my education, supported and encouraged me throughout my life. They are my inspiration and motivation. Thank you for all your sacrifices and for always believing in me.

Special thanks are due to my husband, Gökhan, for his continuous support and understanding. This thesis process could not be so pleased and meaningful without his constant faith, love, and patience. This accomplishment would not have been possible without him. Thank you.

This work is supported by Tubitak under Grant number 117E761.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Thanks to the technological advances, the use of smartwatches and other wearable devices is growing rapidly. They are equipped with various motion sensors and this makes them effective devices for human activity recognition. While smartwatches can be used to detect complex activities where hand-wrist movements play an important role, smartphones are more convenient to detect simpler locomotion activities. Moreover, an accelerometer is mostly sufficient to detect simple activities, such as walking, with good performance but a gyroscope can increase the recognition rate of more complex activities, such as smoking while walking. This also holds for other parameters, such as sensor sampling rate, feature set and window size, meaning that different activities require different settings for good identification. Besides, changing the parameters can cause higher and unnecessary resource consumption or vice versa on these resource limited devices. In this study, our main motivation is to explore the parameter space that may affect the recognition process in terms of accuracy and as well as resource usage on a large and complex dataset. For this purpose, we collected a dataset of 45 hours from 11 participants. The dataset includes ten different activities including smoking activities in four different postures, such as smoking while standing, some other activities which involve similar hand-wrist movements, such as drinking and some other simple activities, such as sitting.

Firstly, we analyze the impact of parameters using 4 different window sizes and overlaps, 63 different features extracted from each sensor, 4 different sensors, 2 different sensor combinations, 3 classifiers and 10 different activities. By changing the values of the mentioned parameters, in the datasets, we gather the recognition accuracies and find the best parameter sets to maximize the recognition performance. Additionally, we analyze the impact of participants' height on the recognition performance. The results show that simple time-domain features perform the best and while the combination of accelerometer and gyroscope sensors performs better for complex activities and accelerometer alone is sufficient for simple activities. When we consider the impact of height on the recognition performance, the results show that it does not have a

significant effect when all activities are considered, however, it does have an effect on smoking while standing, particularly for participants with a significant height difference than the others.

Secondly, we investigate context-aware activity recognition where parameters are selected on demand. We propose a dynamic parameter selection algorithm, which activates different sensors, sampling rates, window sizes and features on demand according to the type of the activity (simple or complex). This algorithm gets the type information from a state detection algorithm which identifies whether the user is performing a simple or a complex activity. We evaluate the performance of the algorithm both in terms of recognition rate and resource consumption and compare with using static and semi-dynamic parameters. We use feature sets of our first analysis, then we determine the high impact features in order to reduce the number of features and choose the most efficient ones, by applying feature selection algorithms. Results show that, both before and after feature selection, the dynamic parameter selection algorithm achieves 2 to 13% better recognition rate depending on the activity. Dynamic parameter selection, before applying feature selection, consumes 33% less energy and 20% less CPU time, compared to using static parameter selection. Additionally, using selected features in the dynamic parameter selection algorithm, we observe a decrease of 65% on the CPU and energy consumption over the last improvement.

Keywords : Human activity recognition, wearable computing, motion sensors.

ÖZET

Teknolojik gelişmeler sayesinde akıllı saat ve akıllı telefonların kullanımı hızla artmaktadır. Bu cihazlar çeşitli hareket sensörleri ile donatılmışlardır ve bu da onları insan aktivitelerini tanıma için etkili kılar. Akıllı saatler el-bilek hareketlerinin ön planda olduğu karmaşık aktiviteleri tespit etmek için kullanılırken, akıllı telefonlar daha basit hareketler içeren aktiveleri tespit etmek için daha uygundur. Buna ek olarak, yalnızca ivmeölçer kullanarak yürüme gibi basit aktiviteler iyi bir performansla tanımlanabilir, ancak yürürken sigara içmek gibi daha karmaşık aktiviteleri tanımlamada ivmeölçere ek olarak jiroskop kullanmak daha iyi sonuç verir. Bu durum, sensör örnekleme oranı, öznitelik seti ve pencere boyu gibi diğer parametreler için de geçerlidir. Dolayısıyla, iyi bir tanıma için farklı aktivitelerin farklı parametre ayarlarının olması gerekir. Ayrıca parametrelerin değiştirilmesi, bu kaynakları sınırlı cihazlarda kaynağın daha fazla ve gereksiz tüketilmesine ya da bunun tersine neden olabilir. Bu çalışmada bizim temel motivasyonumuz, karmaşık ve büyük bir veri seti kullanarak parametrelerin aktivite tanıma sürecine olan etkilerini doğruluk ve kaynak kullanımı açısından incelemektir. Bu amaçla, 11 katılımcıdan 45 saatlik bir veri seti toplandı. Veri seti, dört farklı duruşta (oturarak, ayakta, yürürken ve sohbet ortamında) sigara içme aktiviteleri, sigara içmeye benzer el-bilek hareketleri içeren (oturarak ve ayakta kahve içme ve yemek yeme) aktiviteleri ve bazı basit aktiviteleri (oturma, ayakta durma ve yürüme) içeren toplam on farklı aktivite içermektedir.

İlk olarak, 4 farklı pencere boyu, her sensör için ayrı ayrı hesaplanan 63 öznitelik, 4 farklı sensör, 2 farklı sensör kombinasyonu, 3 sınıflandırma algoritması ve 10 farklı aktivite için parametrelerin etkilerini analiz ettik. Bahsedilen parametreleri değiştirerek, veri setinde tanıma performansını en üst seviyede tutmak için en iyi parametre gruplarını bulduk. Buna ek olarak, katılımcıların boylarının tanıma performansına olan etkilerini de inceledik. Sonuçlar sadece zaman-alanlı özniteliklerin daha iyi performansa sahip olduğunu ve basit aktiviteleri tanımlamada sadece ivmeölçer yeterliyken karmaşık aktiviteler için ivmeölçer ve jiroskop birleşiminin daha iyi performans elde ettiğini gösterdi. Tüm aktiviteler dikkate alındığında boyun tanıma performansı üzerinde önemli bir etkiye sahip olmadığını görüldük. Ancak boyları birbirinden belirgin bir şekilde

farklı olan katılımcıların sadece ayakta sigara içme aktivitesine boyun etkisi farkedildi.

İkinci olarak, parametrelerin ihtiyaca göre seçildiği bağlam farklında bir aktivite tanıma üzerine çalıştık. Aktivitenin türüne göre (basit ya da karmaşık) farklı sensörleri, örnekleme aralıklarını, pencere boylarını ve öznitelikleri kullanan bir dinamik parametre seçimi algoritması önerdik. Kullanıcının basit veya karmaşık bir aktivite gerçekleştirip gerçekleştirmediğini belirleyen bir durum algılama algoritması sayesinde aktivitenin türünün ne olduğu bilgisini algoritmaya verdik. Algoritmanın performansını hem tanıma oranı hem de kaynak tüketimi açısından değerlendirdik. Dinamik parametre seçimi sonuçlarını, statik ve yarı dinamik parametre seçimi sonuçları ile karşılaştırdık. Önce, ilk analizimizde kullandığımız öznitelik gruplarını kullandık, ardından öznitelik seçimi algoritmalarını kullanarak hem öznitelik sayısını azalttığımız hem de en etkili öznitelikleri tespit ettiğimiz yüksek etkili öznitelikleri belirledik. Sonuçlar, hem öznitelik seçiminden önce hem de sonrası için, dinamik parametre seçimi algoritmasının aktiviteye bağlı olarak %2 ile %13 arasında bir performans artışı sağladığını gösterdi. Öznitelik seçiminini uygulamadan önce, dinamik parametre seçimi, statik parametre seçimine kıyasla %33 daha az enerji ve %20 daha az CPU tüketimi sağladı. Ayrıca, seçilen yüksek etkili öznitelikleri dinamik parametre seçimi algoritmasında kullanarak, son iyileştirmeye kıyasla CPU ve enerji tüketiminde %65'lik bir azalma elde ettik.

Anahtar Kelimeler : İnsan aktivitesini tanıma, giyilebilir hesaplama, hareket sensörleri.

1 INTRODUCTION

Human activity recognition on mobile and/or wearable devices, such as smartphones, smartwatches and smart glasses, is currently being used in different application areas, ranging from personal health care systems to entertainment (Lockhart et al., 2012). These devices integrated with various sensors are emerging as ideal platforms for human activity recognition and their ubiquity attract application developers and researchers (Lara and Labrador, 2013; Mukhopadhyay, 2015; Cornacchia et al., 2017) to build mobile activity recognition systems. Mostly, supervised machine learning algorithms are often utilized where sensor data is collected from these devices with the labels of the activities and a model is trained and built for the classification.

Compared to smartphones, smartwatches have advantages, such as ease of carrying, being attached to the wrist instead of being carried in a pocket or bag. Moreover, they make it easier to recognize more complex activities, especially those involving hand and arm movements, such as eating, typing, and drinking. Nowadays, the most common uses of smartwatches include getting the notifications on the watch rather than on the phone, watching the time and following the steps taken by the user, as a step-tracker. However, with the variety of the sensors included in the smartwatches, they can be used to recognize more complex activities and assist the user to track his or her routines and patterns.

One of the behavioral patterns that users may be interested to track is the smoking pattern, such as the number of cigarettes smoked, periods, and time of smoking. Such tracking can be useful for the user to get an insight into his or her smoking behavior and it can also be useful in an effective intervention for behavior change, such as quitting smoking or reducing the number of cigarettes smoked per day. Particularly, for smoking cessation programs, self-reporting puts a burden on the user, but smartwatches can enable automated self-reporting and provide more context to the smoking activity (Tang et al., 2014). However, it is challenging to detect smoking compared to simpler locomotion activities, such as walking, running, because it is not a periodic activity unlike these simple activities. Moreover, it can be performed in various postures (sitting, walking, standing) and in combination with different activities (in a group while chatting, alone, while drinking coffee). It can also be confused with similar activities, such as eating, drinking, that involve similar hand gestures. Height of the users may also impact the recognition performance.

While various studies have been presented in the domain of human activity recognition using smartphones and/or smartwatches (Lara and Labrador, 2013; Seneviratne et al., 2017), in some of the studies, recognition process is performed offline, not on the device, where the device is often used to log only the sensor data. This is mostly due to the limited resources of the devices, particularly the battery and it can be considered as a challenging task to implement different classifiers on these devices compared to offline recognition using machine learning tools on a PC, for instance. On the other hand, online activity recognition on the devices is necessary to enable practical applications, such as real-time fall detection. Moreover, it is necessary to verify the offline results and to investigate the resource consumption of the recognition process (Shoaib, Bosch, Incel, Scholten and Havinga, 2015).

Recently, examples of online recognition studies are also presented in the literature (Rawassizadeh et al., 2015; Shoaib, Bosch, Incel, Scholten and Havinga, 2015; Zheng et al., 2017; Shoaib et al., 2018). What we observe in these studies is that, there is a fixed set of parameters used in the recognition process. More precisely, the same set of sensors, sampling rates, features, window sizes are used for different types of activities. Based on an offline and online analysis in the previous studies (Shoaib et al., 2014; Shoaib, Bosch, Incel, Scholten and Havinga, 2016), they observed different behavior for different activities. For example, certain simple activities can be better recognized by an accelerometer alone, such as walking, jogging, and biking. On the other hand, other more complex activities require a gyroscope in addition to the accelerometer for better recognition, such as walking upstairs, smoking while in a conversation, eating while sitting. People, especially the knowledge workers, spend most of their time in an inactive state, such as sleeping, sitting, standing and this inactivity can be easily recognized without actually extracting a full set of features and running complex classifiers.

Moreover, selection of parameters, such as sensors, sampling rates, in a dynamic way according to the context of the user can lead to an energy-efficient online activity recognition, without compromising the recognition performance (Rachuri et al., 2011; Shoaib, 2017). Sampling more than one sensor at the same time consumes more resources compared to a single sensor. A better solution would be to turn on sensors on demand, such that if the activity cannot be recognized with a single sensor, then an additional sensor can be turned on. For this purpose, the activity recognition process should be context-aware such that the parameters like the sampling rates, features and sensors should be selected on demand.

In this thesis, we study the recognition of smoking activity with the motion sensors available on smartwatches. For this purpose, we collected a dataset from 11 participants including ten different activities. The data includes different smoking variations : smoking while sitting (smokeST), smoking while standing (smokeSD), smoking in a group (smokeGroup), and smoking while walking (smokeWalk), to address the challenge of smoking recognition in different postures. This dataset is not only composed of smoking but also includes activities with similar hand/arm gestures : eating while sitting (eat), drinking while sitting (drinkST), drinking while standing (drinkSD). Standing (stand), sitting (sit) and walking (walk) activities are also performed alone to differentiate these activities in combination with smoking.

In the first phase, our aim is to analyze the recognition of smoking activity in detail, with a focus on using different sensors, different classifiers, different and comprehensive set of features, different window sizes and overlap ratios. Seventeen features from four dimensions (x, y, z and magnitude) of accelerometer, gyroscope, and linear acceleration sensors are extracted. Another seventeen features from pitch and roll values computed from the accelerometer readings are also used in the analysis. We gather the recognition accuracies and find the best parameter sets to maximize the recognition performance. As the final parameter, we investigate the impact of height on smoking recognition performance. We analyze the impact of height both using regression analysis and by grouping the users with dissimilar heights and training on groups with different heights compared to similar heights.

We use three different classifiers, namely support vector machine, random forest and multilayer perceptron which are commonly used for activity recognition (Shoaib, Scholten, Havinga and Incel, 2016; Shoaib et al., 2014; Wang et al., 2005). Scikit-learn (Version 0.18.1) is used for the analysis of our dataset using these classifiers. Compared to previous works on smoking recognition (Tang et al., 2014; Scholl and Van Laerhoven, 2012), we do not study the recognition of puffs (hand-to-mouth gesture) to identify smoking periods. In the data collection phase, this makes it easy to label the start and end of smoking sessions instead of a fine-grain gesture labeling and enables continuous recognition.

In the second phase, we propose a dynamic parameter selection algorithm based on the results of the first analysis. We investigate the following research question : "What is the impact of dynamically changing the activity recognition parameters on the performance both in terms of recognition rate and resource consumption ?". We propose a context-aware algorithm which activates different sensors, sampling rates and features on demand according to the context of the user. The algorithm is preceeded by a state detection algorithm which detects whether the user is performing a simple or a complex activity, using accelerometer. If the activity is simple, then only the accelerometer is sampled at a lower rate, and using a small number of features the activity is classified. On the other hand, if the activity is complex, gyroscope is also turned on and both are sampled at a higher sampling rate and more features are extracted to classify the activity. For training the classifiers, we use the same dataset used in previous chapter, that includes eleven participants performing ten different activities. We evaluate the performance of the dynamic parameter selection algorithm in a scenario where users perform activities sequentially. We compare its performance with two cases where all parameters are fixed and where some of the parameters are dynamic, in other words semi-dynamic.

The main highlights and contributions of the thesis are listed as follows :

— We explore the whole parameter space that may affect the recognition process on a large and complex dataset in the context of smoking, considering 4 different window sizes and overlaps, 63 different features extracted from each sensor, 4 different sensors and 2 different sensor combinations and 3 classifiers.

— We investigate the impact of subject's height on the smoking recognition performance.

— We introduce the dynamic parameter selection algorithm based on our previous parameter analysis, which decides on the activity recognition parameters dynamically according to the activity type.

— We also propose a state detection algorithm to identify complex and simple states together with a state correction method.

— We evaluate the performance of the algorithm both in terms of recognition performance and resource consumption and compare with using static and semi-dynamic parameters. Test results show that the dynamic parameter selection algorithm achieves 2 to 13% better recognition rate depending on the type of activity and 22% less energy consumption, 50% less memory size and 11% less CPU time in terms of resource consumption, compared to using static parameters.

The organization of this thesis is as follows : In Section 2.1, we present the related studies that focus on smoking recognition with wearable devices. In Section 2.2, we summarize the related studies on dynamic activity recognition. In Section 3.1, we present our methodology for analysis, including the dataset, feature sets, classification methods. In Section 3.2, we present the results of our analysis in terms of feature set, classifier performance, sensor or sensor-combination performance and user height. In Section 3.3, we present the discussion. In Section 4.1 we present our methodology particularly the dynamic parameter selection algorithm. In Section 4.2, we present the performance analysis for static, semi-dynamic and dynamic activity recognition and finally in Section 5, we draw the conclusions.

2 RELATED WORK

In this section, we first present the related studies in the field of smoking recognition with wearable sensors and then we present the studies that focus on dynamic parameter selection for activity recognition.

2.1 Related Work : Smoking Recognition

Nowadays, the use of smartwatches is continuously increasing together with the utilization of smartphones. This common use of smartwatches provides an opportunity to realize human activity recognition, particularly for the activities which involve wrist, hand and arm movements. During activity recognition process, there are many parameters to explore, such as the types of sensors, sampling rates, window sizes, features, and classifiers.

In (Tang et al., 2014), the authors use a dataset of 11.8 hours where six participants performed complex smoking related activities, such as smoking while sitting, standing, eating, walking, using a phone, and talking in a group. To collect data for the smoking activities, they use two accelerometers at wrist position with a sampling rate of 40 Hz. They propose a two-layer model for automatic detection of puffs and smoking activities. First, they use a random forest classifier to calculate inter-puff intervals and puff frequency to detect puffing. Then, they recognize the smoking sessions using a threshold based method. They achieve a cross-validation f1-score of 70% for puffing and 79% for smoking detection for person-dependent evaluation.

The authors in (Scholl and Van Laerhoven, 2012) focused on the detection of only the smoking activity using a wrist-worn accelerometer-based device. Smoking sessions were performed by four participants. The authors reported a precision of 51.2% and a person-dependent recall of 70%. Similarly, the authors in (Ali et al., 2012) also studied the performance for the smoking activity. They use a dataset collected from 10 participants over 13 individual smoking sessions. There is only one sensor used for this study which is respiratory inductive plethysmography (RIP) sensor. This sensor collects respiration data with inhalation and exhalation of smoking and is worn around the chest area. They evaluate the performance with 10-fold cross-validation and with splitting data into train and test sets using SVM classifier. Detection results give an accuracy of

86.7% when there are concurrent activities, such as walking and speaking in collected data and 91% when there is not.

The authors in (Saleheen et al., 2015) use two wearable sensors, such as 6-axis initial sensors (3-axis accelerometers and 3-axis gyroscopes) for capturing hand gestures and RIP sensor for capturing breathing pattern. They apply 10-fold cross-validation on 40 hours of training data collected from 6 daily smokers. On this dataset, they reported a recall rate of 96.9%, and a false positive rate of 1.1%. Their dataset contains only four smoking related activities, which are smoking while standing, sitting, walking, and being in a conversation. But, they did not consider activities that could easily be confused with smoking activity in everyday life, such as eating and drinking.

In (Parate et al., 2014), the authors studied the smoking detection problem using a wristband containing three sensors : accelerometer, gyroscope and magnetometer. They use a dataset collected from 15 participants for a total of 17 smoking, 10 eating, 6 drinking sessions. The sessions included several activities, such as smoking while standing, smoking in a group, smoking while walking, eating, drinking and others. For smoking, they reported a precision of 91% and a recall of 81%.

In a recent study (Cole et al., 2017), the focus is on the successful detection of smoking events with accelerometer sensor using a smartwatch. 120 hours of data is collected from ten participants with a sampling rate of 20 Hz. The activities are smoking, drinking, walking, tying shoes and typing on a computer. Using artificial neural networks, they aim to classify the raw data into two groups : smoking and non-smoking. They achieve an accuracy of approximately 78% for the smoking activity.

In (Skinner et al., 2018), the authors evaluated a smartwatch-based system to detect smoking activity using accelerometer and gyroscope. Unlike other studies, this system does not require a connected smartphone and runs on a low-cost smartwatch. The system makes a passive detection as it does not require any input from the user. They performed a preliminary validation in a laboratory setting with 13 participants and free-living conditions. In this system, an instance of smoking is comprised of three movements which are hand raising to mouth, hand stationary at mouth, hand moving away from mouth. Data is sampled at 100 Hz and Decision Tree is used in the classification phase. They report 86% precision and 71% recall in free-living conditions.

In an earlier work (Shoaib, Bosch, Scholten, Havinga and Incel, 2015), they investigated the fusion of smartwatch and smartphone sensors for the recognition of thirteen daily human activities. They determine that certain complex activities cannot be well recognized only with a smartphone in the pocket. For that reason, they simulated a smartwatch using a smartphone on the wrist position. Furthermore, the study uses two different datasets : one for simple activities, such as walking, jogging, biking, sitting, standing, walking upstairs and walking downstairs and, one for complex activities, such as smoking, eating, typing, writing, drinking coffee and giving a talk. Two sensors (accelerometer and gyroscope) were used individually and together and only two time-domain features (mean and standard deviation) were calculated. They reported a higher accuracy for the complex activities, thanks to the fusion of smartwatch and smartphone sensors. For the simple activities, using two devices did not make a significant impact on the performance.

In (Shoaib, Scholten, Havinga and Incel, 2016), they proposed a two-layer hierarchical smoking detection algorithm (HLSDA) and analyzed its performance on the same dataset which is also used in this study. Their aim was to see the impact of using a lazy context rule-based correction method that utilizes neighboring data segments on the performance of activity recognition. In that study, they only used 6 features, including mean, standard deviation, minimum, maximum, kurtosis, and skewness. They showed that using HLSDA increases the performance up to 11% in terms of F1-measure. Compared to (Shoaib, Scholten, Havinga and Incel, 2016), in this study we explore all the parameters of the whole parameter space that may affect the recognition process, considering 4 different window sizes/overlaps, 68 different features extracted from each sensor, 4 different sensors and 2 different sensor combinations and 3 classifiers. Moreover, we investigate the impact of height on the smoking recognition performance.

In some studies, very few features have been used. For example, in (Scholl and Van Laerhoven, 2012) only mean and variance, in (Shoaib, Bosch, Scholten, Havinga and Incel, 2015) only mean and standard deviation are used as a feature set. However, in this study, we extract 17 features from each four dimensions (x, y, z and, magnitude) of each of sensors, as mentioned. There are also studies that investigate the use of more than 15 features extracted from each sensor (Tang et al., 2014; Ali et al., 2012; Saleheen et al., 2015). But these studies use all the features in a single feature set and they did not create different feature combinations to better observe their effect on the performance of activity recognition. A detailed comparison of studies that mainly use

Table 2.1: Comparative analysis of related work and our study

Ref.	**Sensors**	**Window Size -Overlap**	**Features**	**Classifiers**	**Activities**	**Evaluations**
(Tang et al., 2014)	A (2)	1, 3, 5, 7, 9, 15, 20, 25, 35 sec - 50%	mean, std, max, min, median, kurtosis, skewness, percentile, snr, rms, peak-peak amplitude, peak rate, corr, crossing rate between axes, slope, mse, r-squared	RF, TB	S_{SD} , S_{ST} , S_G, S_W, S_E, S_D, S_{UP}	PD-5 FCV, PID-LOSO
(Shoaib, Scholten, Havinga and Incel, 2016)	A, G	30 sec-–0%	max, min, kurtosis, skewness	HLSDA	S_{ST}, S_{SD}, D_{ST}, D_{SD}, E, ST, SD, S_G, S_W, W	PD-5 FCV, PID-LOSO
(Scholl and Van Laerhoven, 2012)	A	5.4 sec-NP	mean, variance	GMM	S_{SD}, $others$	NP
(Shoaib, Bosch, Scholten, Havinga and Incel, 2015)	A, G	2, 5, 10 sec—50%	mean, std	DT, KNN, SVM	S, D, E, T, WR, TY, ST, SD, J, B, W, W_{US}, W_{DS}	PD-10 FCV
(Parate et al., 2014)	A, G, M	20 sec-NP	duration, speed, distZ, distXY, dist, roll velocity, roll, pitch	RF, CRF	S_{SD}, S_{SG}, S_W, E, D, $others$	PD-10 FCV
(Bao and Intille, 2004)	A (5)	6.7 sec—50%	mean, energy, entropy, corr	DTa, KNN, DT, NB	W, ST, SD, WT, R, STR, SC, FL, BT, R_{EL}, R_{ES}, W_{CL}, WOC, ED, RD, B, STT, V, LD, CS	PD-NP, PID-LOSO
(Skinner et al., 2018)	A, G	NP	NP	DT	S	NP
(Cole et al., 2017)	A	5 sec-NP	NP	ANN, RBAI	S, E, D, W, TOC, TS	PD-TTVS
This study	A, G, LA	20, 30 sec—0%, 50%	mean, std, skewness, kurtosis, min, max, range, integration, corr, rms, absdiff, spectral energy, entropy, coefficient sum, erd, levens	SVM, RF, MLP	S_{ST}, S_{SD}, D_{ST}, D_{SD}, E, ST, SD, S_G, S_W, W	PD-10 FCV, PID-LOSO

wrist-worn sensors is summarized in terms of these parameters in Table 2.1 [1].

Compared to similar studies in the literature, we evaluated our study with four sensors (the accelerometer, the linear acceleration, the gyroscope and the pitch-roll) individually and some of them in combination. Smoking activities can be performed in various postures and in combination with other activities, and have similarities to other activities. Unlike most of the existing works, we studied various smoking variations with similar activities which also make it difficult to detect. Based on some studies showing that an overlap of 50% produces reasonable results (Preece et al., 2009; Bao and Intille, 2004), the sliding window approach with this overlap value is also investigated. In our study, we focus on the impact of sensors, features, window sizes (20 and 30 seconds), overlaps (50% and 0%) and classifiers. We have created seven feature sets to examine the effects of features in detail. We consider also multiple evaluation scenarios, as discussed in Section 3.2. Additionally, using statistical analysis, we analyze if the participants' heights have an impact on their activity recognition performance.

1. Sensors : A : Accelerometer, G : Gyroscope, LA : Linear Acceleration, M : Magnetometer. Features : max : maximum, min : minimum, std : standard deviation, snr : signal to noise ratio, rms : root mean square, corr : correlation coefficient, mse : mean squared error, absdiff : absolute difference, erd : euclidean related distance, levens : levenstein distance. Classifiers : HLSDA : Hierarchical Smoking Detection Algorithm, RF : Random Forest, TB : Threshold Based, GMM : Gaussian Mixture Model, DT : Decision Tree, KNN : K Nearest Neighbors, SVM : Support Vector Machine, CRF : Conditional Random Field, DTa : Decision Table, NB : Naïve Bayes, MLP : Multi-Layer Perceptron, ANN : Artificial Neural Networks, RBAI : Rule-Based Artificial Intelligence. Activities : S : smoking, S_{ST} : smoking while sitting, S_{SD} : while standing, S_G : while in a group, S_W : while walking, S_E : while eating, S_D : while drinking, S_{UP} : while using phone, D : drinking, D_{ST} : drinking while sitting, D_{SD} : while standing, E : eating, ST : sitting, SD : standing, W : walking, T : giving a talk, WR : writing, TY : typing, J : jogging, B : biking, W_U : walking upstairs, W_D : walking downstairs, R : running, STR : stretching, SC : scrubbing, FL : folding laundry, BT : brushing teeth, R_{EL} : riding elevator, R_{ES} : riding escalator, W_{CI} : walking carrying items, WOC : working on computer, ED : eating or drinking, RD : reading, STT : strength-training, V : vacuuming, LD : lying down, CS : climbing stairs, TOC : typing on computer, TS : tying shoes. Evaluation : PD : person dependent, PID : person independent, FCV : fold cross validation, LOSO : leave one subject out, TTVS : train/test/validation splitting, NP : not provided

2.2 Related Work : Dynamic Parameter Selection for Activity Recognition

In recent years, many studies have been made in the domain of human activity recognition using smartphones and/or smartwatches (Yan et al., 2012), (Konak et al., 2016) and (Khalifa et al., 2014). In most of these studies, recognition process is performed offline, not on the device, where the device is only used to log data. Due to the limited resources of smartwatches and smartphones, recently, researchers have been moving towards online activity recognition to verify the offline results and to investigate the resource consumption of the recognition process (Khalifa et al., 2014). Selection of parameters, such as sensors, feature sets, in a dynamic way according to the context of the user can lead to for energy-efficient online activity recognition, without compromising the recognition performance (Khalifa et al., 2014).

As we report in Section 3.2, fusion of accelerometer and gyroscope brings higher performance in terms of activity recognition process than using only the accelerometer. This is particularly true for more complex activities, such as smoking while walking, tough simpler activities, such as walking can be well recognized by using only an accelerometer. Sampling two sensors at the same time will consume more resources compared to a single sensor. A better solution would be to turn on sensors on demand, such that if the activity cannot be recognized with a single sensor, then an additional sensor can be turned on.

Among several studies which use smartwatch sensors for activity recognition, only (Ryder et al., 2009) and (Wang et al., 2009) utilize dynamic sensor selection. In (Ryder et al., 2009), the GPS sensor is used with the accelerometer to recognize the outdoor activities. When a user performs an indoor activity, the accelerometer achieves a good level of accuracy. More clearly, the GPS is turned off when the user is indoors and turned on if outdoors. In (Wang et al., 2009), researchers propose an energy efficient mobile sensing framework to recognize a user state. This framework turns on only necessary and energy efficient sensors in an adaptive way.

In (Seneviratne et al., 2017), researchers state that an imprudent selection of features and increasing feature set can result in high computational complexity in human activity recognition using wearable devices. This causes higher CPU usage and consequently higher power usage. In places, such as airports and shopping malls, (Khalifa et al., 2014) it would be of interest to detect if the person is standing in the elevator, on the escalator or standing, in order to navigate him/her. In such a case where real-time recognition is necessary, it is important to keep the number of features as low as possible. In (Khalifa et al., 2014), initially, they created their first feature set with x, y, and z axis of mean, standard deviation, skewness, kurtosis, and three other features. After the feature extraction phase, three feature selection algorithms (Information Gain (IG), Correlation Feature Selection (CFS), and Decision Tree Pruning (DTP)) are applied using Weka (Witten et al., 2016). They expect to get the best feature set as the output. Compared to others, DTP achieves the lowest number of features, which is five. After the classification phase, they observe that DTP's five features bring a higher accuracy compared to the first feature set with nineteen features. For example, after feature selection, it was found that only x axis of mean feature was sufficient when three axes of it was used in the beginning. Thus, it requires less computation, so it consumes less energy. Similarly, in Chapter 3, we used the three axes (even four with magnitude) of the features each time when choosing the feature. From this, it is understood that evaluating axes of features separately will provide a positive effect on energy and accuracy.

In (Zhang and Sawchuk, 2011), researchers use the off-the-shelf multimodal sensing platform equipped with an accelerometer, a gyroscope, and a magnetometer. However, data is sampled using the 3-axis accelerometer and 3-axis gyroscope in the study. They focus on nine types of activities, which are walking forward, walking left, walking right, going upstairs, going downstairs, jumping up, running, standing, and sitting. Their aim is to find the most important features to recognize activities. They use two types of features which are statistical and physical features. The first one includes the features, such as mean, median and skewness, the second one includes Normalized Signal Magnitude Area (SMA), energy and dominant frequency. Instead of a single layer feature selection and classification, they propose a multi-layer framework which is capable of using different features for different activity subsets. In this hierarchical structure, top layer distinguishes between static and dynamic activity classes, then, second layer walking-related activities and, jumping and running. In the last layers (three and four), walking related activities are differentiated. Thus, they use different classification models and feature sets at each step until reach activity.

In (Saeedi et al., 2014), researchers aim to detect the location of the wearable device on human bodies, such as right wrist and left ankle. They want to make choice of feature set and sensor depending on the location of device. Their sensor devices are equipped with 3-axis accelerometer and a 2-axis gyroscope. They want to compromise between accuracy and power consumption. Particularly, the aim is to minimize the consumed energy spent for sensing sensors data and computation power spent for feature sets while keeping a given accuracy. Additionally, power consumption for each feature are noted. Some of these features are common with features of this study which are median, mean, max, min, standard deviation (std) and root mean square (rms). In this list of features, mean, max and min have the minimum energy consumption which is approximately 8100 nJ and median has the maximum with 405159 nJ, as reported in the paper.

Sampling rate and window size have an important effect on the accuracy of activity recognition. In (Zheng et al., 2017), a user-independent and energy-efficient activity recognition system is proposed. The activities they are interested in are sitting, standing, walking, running, upstairs and downstairs. Three of them are common with our activities. Dataset is collected from 20 participants with the smartphone placed in the right-front pocket of the pants. They collect data comes from barometer and accelerometer sensor. Barometer is used for user's altitude or height information. At low sampling rate, it can detect if user is climbing star or not. Differently from Nyquist theorem, their system shows that it is possible to achieve high accuracies using low sampling rates, such as 1, 5 and 10 Hz. They use a window size of 1 second for 5 and 50 Hz, and 5 second for 1 Hz. While keeping a high accuracy (about 96% on average), they save 17.3% and 59.6% of the power using the sampling rate of 1 Hz compared with, respectively, the sampling rates of 5 Hz and 50 Hz.

In (Shoaib, 2017), the author states that the higher sampling rate can increase the accuracy, however, there is a trade-off between high battery consumption and accuracy. The most commonly used sampling rates in online activity recognition studies are 50, 20, 32, 100, 40, 10 Hz. For the simple activities, such as walking, running, he compared the accuracy of 10 Hz, 25 Hz, 50 Hz. Since 25 Hz and 50 Hz are very close to each other, he did not present it. At wrist position, the accuracy of 10 Hz had a slightly lower than 50 Hz. In (Shoaib, 2017) it is stated that it is better to use the only accelerometer with a not very high sampling rate for a small set of simple activities, such as sitting, standing and walking. He observes that changing window size have more impact on CPU consumption compared to changing sampling rate. He also emphasized the importance of using a context-aware algorithm that activates different sensors, features and classifiers when needed. However, he did not develop an algorithm and make an analysis. Taking this open issue into consideration, in Chapter 4, we develop a dynamic parameter selection algorithm and analyze it in detail.

3 SMOKING RECOGNITION WITH SMARTWATCH SENSORS

3.1 Methodology For Smoking Recognition

In this section our methodology for smoking recognition is explained, as well as the characteristics of the dataset utilized in this study.

3.1.1 Dataset and Preprocessing

We are collected a dataset from eleven participants including ten different human activities using a smartwatch and a smartphone application. The application logs timestamp, triaxial accelerometer, linear accelerometer, gyroscope and magnetometer readings from both smartwatch and smartphone, pressure and heart rate information from a smartwatch, latitude and longitude information of GPS from smartphone to a CSV file All data is sampled at 50 Hz. In this dataset, we focus on the recognition of smoking activity in different forms within the activities of the hand-wrist movements very similar to it, such as drinking and eating. More explicitly, the activities are smoking while standing (smokeSD), smoking while sitting (smokeST), eating (eat), drinking while standing (drinkSD), drinking while sitting (drinkST), standing (stand), sitting (sitting), smoking while walking (smokeWalk), walking (walk) and smoking in a group conversation (smokeGroup). Dataset contains approximately 45.09 hours of data : 16.84 hours for smoking activities, 15.57 hours for eating and drinking activities, 12.68 for simple activities. Each participant repeated the activities five times. All activities were performed for 5.23 hours except smoking while walking, walking and smoking in a group conversation which were performed 2.31, 2.31 and 4.17 hours, respectively. More details about the duration of activities can be found in (Shoaib, Scholten, Havinga and Incel, 2016).

During the collection of data, one smartphone and one smartwatch were used by every participant (see Figure 3.1). The smartphones (Samsung Galaxy S2 or S3) were placed in the pocket of their right pants and the smartwatches (LG Watch R, LG Watch Urbane or Sony Watch 3) on their right wrist position. During the treatment of this data, we used only the data collected from the smartwatch considering that purpose is to analyze the performance of smartwatch for activities where the hand movements are more significant. The sensors used during the collection of data are accelerometer, gyroscope and linear acceleration of smartwatches. The data was sampled at 50 Hz for all sensors. More details of the dataset are presented in Table 3.1.

Figure 3.1: Smartphones and smartwatches used during the collection of data

We segment raw data into different time windows, then we compute different features for each segment. For the feature extraction phase, we use a sliding window approach. Since the window size and the overlap are important factors on continuous activity recognition, we consider four different cases as follows :

— Case 1 : Window size of 20 seconds with a 0% overlap.

— Case 2 : Window size of 30 seconds with a 0% overlap.

— Case 3 : Window size of 20 seconds with a 50% overlap.

— Case 4 : Window size of 30 seconds with a 50% overlap.

We tested smoking recognition with smaller or larger window sizes as well, however, recognition success was better for these window sizes (20 and 30 seconds). Each of three sensors has three components which are x, y and z axis. We added a fourth dimension

Table 3.1: Participants and details of the collected dataset

Participant No	Activities Performed	Duration per activity (minutes)	Total duration (minutes)	Gender	Height (cm)	Age (years)	Cigarette Usage
1	S_{ST}, S_{SD}, D_{ST}, D_{SD}, E, ST, SD, S_G, S_W, W	43	430	male	180	25	8-10 per day
2	S_{ST}, S_{SD}, D_{ST}, D_{SD}, E, ST, SD, S_G, S_W, W	47	470	male	172	30	0-10 per week
3	S_{ST}, S_{SD}, D_{ST}, D_{SD}, E, ST, SD, S_G, S_W, W	48	480	male	175	25	2-6 per day
4	S_{ST}, S_{SD}, D_{ST}, D_{SD}, E, ST, SD, S_G	37	296	male	156	28	0-10 per week
5	S_{ST}, S_{SD}, D_{ST}, D_{SD}, E, ST, SD, S_G	18	144	male	174	23	18-20 per day
6	S_{ST}, S_{SD}, D_{ST}, D_{SD}, E, ST, SD, S_G	20	160	female	164	20	3-7 per week
7	S_{ST}, S_{SD}, D_{ST}, D_{SD}, E, ST, SD, S_G	16.8	134.4	male	181	20	9-11 per day
8	S_{ST}, S_{SD}, D_{ST}, D_{SD}, E, ST, SD, S_G	20	160	female	172	29	4-6 per day
9	S_{ST}, S_{SD}, D_{ST}, D_{SD}, E, ST, SD	24	168	male	167	35	0-10 per week
10	S_{ST}, S_{SD}, D_{ST}, D_{SD}, E, ST, SD	19	133	male	181	27	7-12 per day
11	S_{ST}, S_{SD}, D_{ST}, D_{SD}, E, ST, SD	18.6	130.2	male	170	45	15-20 per day

called magnitude which is the sum of the square root of related sensor's x, y and z axis. After this, we calculate features from the readings gathered from each sensor's four dimensions. In addition, to better observe the rotation information of smartwatches, we create a fourth sensor called pitch and roll. Raw accelerometer readings are used for computing pitch and roll values using the method in (Coskun et al., 2015).

3.1.2 Feature Sets

We calculate 17 features from each four dimensions (x, y, z and magnitude) of each of accelerometer (ACC), linear acceleration (LACC), gyroscope (GYR) and pitch and roll (PR) sensors from the segmented raw data. List of features that we compute is as follows :

- Mean : The average value of samples over a time window. It gives us a central value for a time window.
- Standard deviation (std) : The square root of the variance. It shows how much data sample is spread out around the mean and thus it gives an indication about the stability of sample (Figo et al., 2010).
- Median : Middle number of a sample. It divides the data sample in two parts : high half and lower half (Figo et al., 2010).
- Skewness : The measure calculated by lack of symmetry of data sample around its mean (Shoaib, Scholten, Havinga and Incel, 2016). A sample is not symmetric if its distribution does not same to the left and right of the mean.
- Kurtosis : The measure of whether the data in the sample has a lot of or less data in its tails compared to normal distribution (Shoaib, Scholten, Havinga and Incel, 2016).
- Min : The minimum value of samples over a time window.
- Max : The maximum value of samples over a time window.
- Range : The difference between the maximum and the minimum of samples over a time window.
- Integration : The measure used to estimate the speed and distance of the signal under the data curve and this is commonly applied to accelerometer data (Nambu, 2007).

- Correlation : Pearson's product-moment coefficient is the most commonly used correlation coefficient (Rodgers and Nicewander, 1988). It measures the relationship between each pair of axis and can be applied for accelerometer or gyroscope readings (Ortiz, 2015). It is effective to discriminate one dimensional activities, such as walking and climbing stairs (Yang et al., 2008).
- Root mean square (rms) : The square root of the mean of the squares of data over a time window.
- Absolute difference (absdiff) : The sum of the differences between each data sample and the average of sample divided by the number of data points (Incel, 2015).
- Spectral energy : Thehe squared sum of spectral coefficients of signal over the length of the sample window(Incel, 2015).
- Entropy : The entropy metric can be roughly considered as frequency distribution which is high if the distribution is flat and low if peaky (Lu et al., 2010). It helps to differentiate activities which have similar energy values but different activity patterns (Figo et al., 2010).
- Sum of coefficients (coeffsum) : The sum of the first five FFT coefficients.
- Euclidean related distance (erd) : The square root of the sum of the squares of the differences between corresponding data over a time window.
- Levenshtein distance : The measurement of similarity between two strings. It determines the smallest number of insertions, deletions, and substitutions needed to transform the first to the second (Fiscus et al., 2006).

Features are comprised of time, frequency and string domain features. These individual features have been reported to be suitable for running on mobile phones and wearables and have been used in previous studies (Figo et al., 2010). Using all features together may be inefficient in terms of computation. Instead of using all features, we divided features into seven different feature groups to better observe their effects. More details about feature sets and their domains are presented in Table 3.2.

3.1.3 Classifiers and Validation

There are several algorithms for classification that have been applied to activity recognition. Particularly, we used Support Vector Machine (SVM), Random Forest (RF) and Multilayer Perceptron (MLP) which are commonly used for activity recognition (Shoaib,

Table 3.2: Feature Sets

Feature Set	Features	Domain
F1	min, max, skewness, kurtosis	Time
F2	mean, std, min, max	Time
F3	median, std, min, max, range, mean	Time
F4	mean, std, integration, correlation, rms, absdiff	Time
F5	spectral energy, entropy, coefficient sum	Frequency
F6	erd	String
F7	levenshtein	String

Scholten, Havinga and Incel, 2016; Shoaib et al., 2014; Wang et al., 2005). Scikit-learn implementations of the classifiers are used with the default settings and parameters. As the parameters of the RF algorithm, we used 11 trees (large number may increase the memory consumption), gini split, maximum depth none and two splits. For SVM, rbf kernel, 1.0 penalty parameter, 3 as the degree of the kernel function are used. For the MLP classifier, 1 hidden layer, constant learning rate, 200 as the maximum number of iterations are used.

In the validation phase, we realize an evaluation with 10-fold cross-validation without shuffling. In this method, the mechanism consists of dividing the dataset into ten equal parts ; use nine of these parts for training and one part for testing. In each iteration, one part used for testing is different, thus, all data is used for testing and for training. We use stratified 10-fold cross-validation which means that every part has nearly the same length. We also evaluated with a person-independent method when we explore the impact of height on the recognition performance.

3.2 Performance Analysis

In this section, we present the results of our recognition analysis. As mentioned, we explore a large set of parameters : 4 different window sizes/overlaps, 7 feature sets extracted from each sensor, 4 different sensors and 2 different sensor combinations and 3 classifiers. First, in Section 3.2.1, we analyze the recognition performance when all activities are considered. We investigate the impact of window size and overlap, feature set and sensors. In Section 3.2.2, we exclude the activities of smoking while walking, walking and smoking in a group since they were not performed by all participants. Similarly, we investigate the effect of mentioned parameters. Moreover, we investigate the impact of users' height in the recognition phase by training with different user groups and using regression. As the performance metric, we report F-measure (f1-score) values which is the harmonic mean of precision and recall. We choose F-measure because it is considered as a balanced performance metric by taking into account both recall and precision.

3.2.1 Scenario 1 : All Activities

In this section, we present the results obtained by following the methodology explained in Section 3.1 using all ten activities. In the following tests, the aim is to analyze the effect of different classifiers for three approaches in detail. The f1-score values range between zero and one, but in the text, f1-score values are discussed in terms of percentages, for the ease of reading.

3.2.1.1 Impact of Window Size and Overlap

In this section, we explore the impact of window size on the performance of classifiers. We change the window sizes and the ratio of window overlaps. We present and discuss the results obtained using accelerometer only in this section, however, the results with other sensors are also presented in Table A.1.

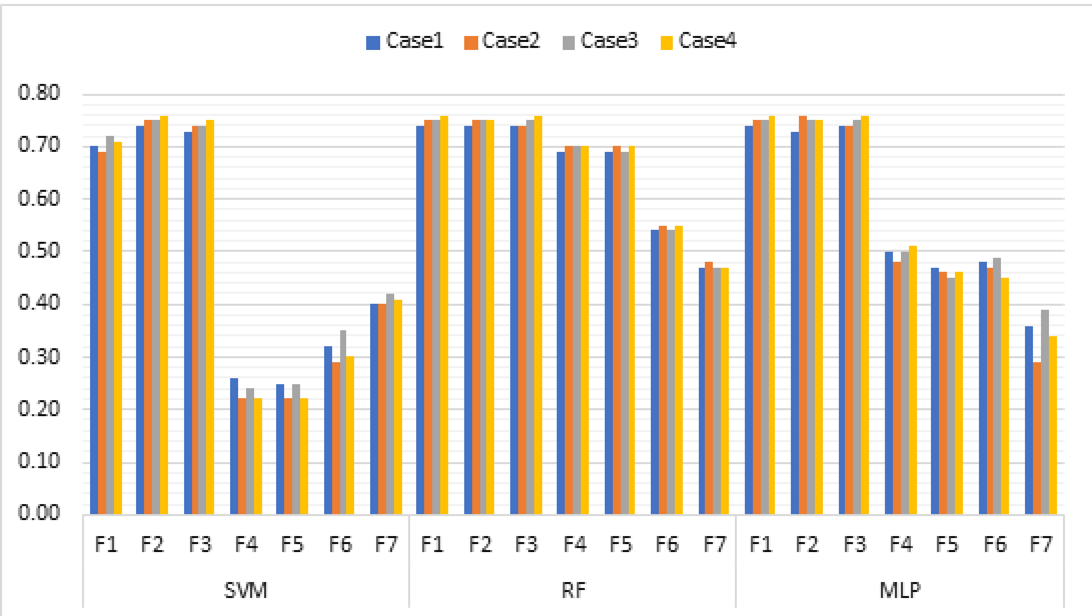


Figure 3.2: Impact of window size and overlap using accelerometer

In Figure 3.2, the results using SVM, RF and MLP are presented to compare different cases which were presented in Section 3.1. The y-axis shows the f1-score values obtained with different feature sets, whereas the x-axis shows the feature sets. When the results of different cases are compared, using Case 4 achieves the highest f1-score for all three classifiers, which is 76% for SVM using F2, for MLP using F1 and F3, for RF using F1 and F3. For these mentioned feature set and classifier combinations, there is only a small difference in other cases. The f1-scores obtained with Case 1, Case 2, and Case 3 are only 1-2% smaller than the results with Case 4. With other feature sets, namely F4 to F7, results obtained with all cases (1 to 4) are much lower, differing between 22% to 70%.

Furthermore, when we compare the performance of classifiers, RF is the best classifier, considering all feature sets. Particularly, the f1-scores of F4, F5 and F6 are 22%, 22% and 30% using SVM whereas it is 70%, 70% and 55% using RF, which results in 48%, 48% and 25% lower f1-score. Whereas, the results with other feature sets, particularly with F2 and F3, are similar with all the classifiers, ranging between 75 and 76% f1-score.

As mentioned, these results were obtained using only the accelerometer. The f1-scores for the other sensors are presented in Table A.1. Considering all cases, we observe that RF with Case 4 achieves the highest f1-score, followed by RF with Case 3. As mentioned, we tested smaller or larger window sizes as well, however recognition performance was lower in other cases. In the next sections, while evaluating the impact of feature set and sensors, we will continue to present and discuss the results obtained with Case 4 and the random forest classifier. However, all results are presented in the Appendix.

3.2.1.2 Impact of Feature Set

In this section, Case 4 is fixed using accelerometer with the random forest classifier. The f1-score results of the other classifiers with each feature set using accelerometer are presented in A.2.

In Figure 3.3, we present the f1-scores achieved per activity as well as the average f1-score considering all the activities. The highest performance for four activities (smokeST : 65% with F1 and F3 ; sit : 99% with F1 to F4 ; stand : 100% with F1 to F5 ; walk : 97% with F1 to F3) is obtained with more than one feature set. For the rest of activities, the best performance is achieved using F1 for drinkSD which is 62%, for smokeWalk (89%) and smokeGroup (55%), F2 for drinkST (61%) and F3 for smokeSD (64%) and for eat (88%). With other feature sets, relatively lower performances are

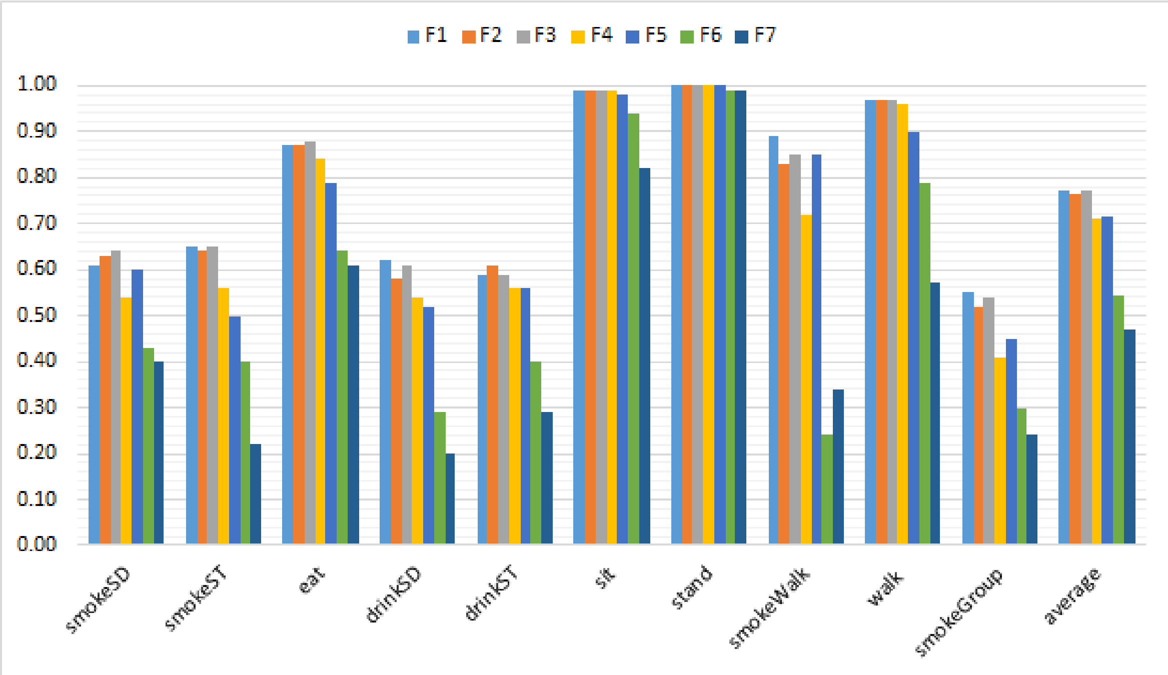


Figure 3.3: Impact of feature set using accelerometer with RF and Case 4

achieved. The worst feature set for all activities is F7 which contains only a string domain feature (levenshtein distance). The average differences between first three feature sets and the rest is relatively high (16%). Although F1 and F3 results are very close to each other, F3 is slightly higher than F1. When we also consider the performance with other classifiers in Table A.2, by ranking the feature sets in terms of f1-score, F3 again ranks as the best feature set, which is followed by F1 and F2. Since, F3 includes the features in F2, but also has median and range, this was an expected result. Moreover, F3 and F1 include min and max in common but F3 has more features. If computing more features is a concern, then F1 or F2 can also be used with a small tradeoff in f1-score.

For the smoking variations, the performance for the smokeWalk is the highest (89%) using F1 and the smokeGroup is the lowest 55% with F1. This is due to the fact that, smokeGroup is very similar to smokeSD. If we use both of these variations as one type of activity, the recognition performance improves as shown in one of our previous works where all these smoking variations were considered as one smoking activity (Shoaib, Scholten, Havinga and Incel, 2016). In smokeGroup sessions, all smokers did not talk or move their hand too much. This makes the smokeGroup activity very similar to smokeSD.

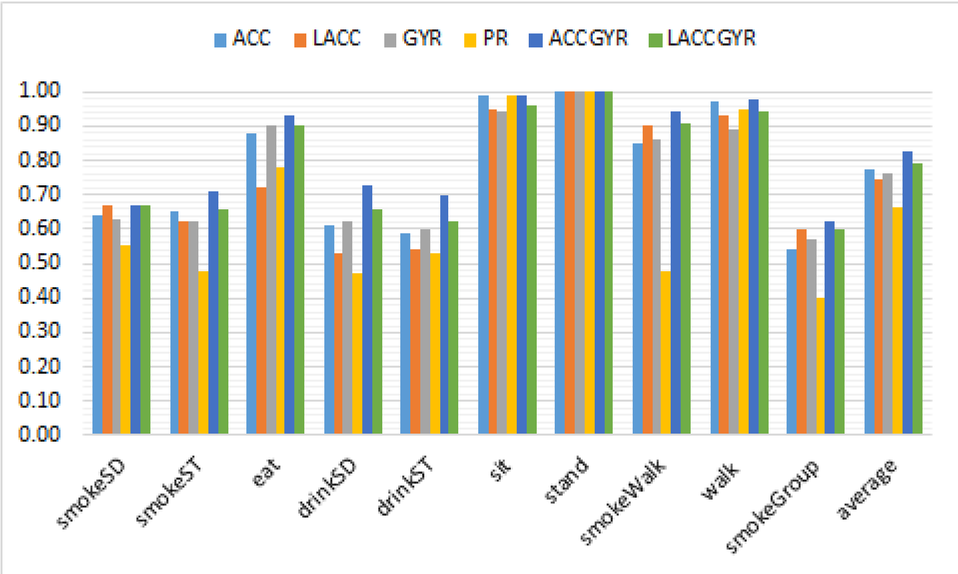


Figure 3.4: Impact of sensors using F3 with RF and Case 4

3.2.1.3 Impact of Sensors and Fusion of Sensors

In this section, impact of individual sensors (ACC, LACC, GYR, PR-pitch, roll), combination of accelerometer and gyroscope (ACCGYR) and combination of linear acceleration and gyroscope (LACCGYR) are analyzed. For this evaluation, RF, Case 4 and F3 are fixed. in Table A.3, the f1-score results for all feature sets with all classifiers are presented.

In Figure 3.4, we present the results per activity and average of all activities. The highest smoking performance is 94%, obtained for the smokeWalk using ACCGYR whereas it is 100% for stand and sit activities. In general, the best performances are obtained with the fusion of accelerometer and gyroscope for all activities. On average of all activities, using only accelerometer exhibits a performance of (77%) and only gyroscope (76%). Combining accelerometer and gyroscope improves the average performance of activity recognition (83%). Similarly, using fusion of linear acceleration and gyroscope increases performance compared to only using linear acceleration and only gyroscope. The worst performance is obtained with pitch and roll features which is 66% on average. As shown in Table A.3, using accelerometer gyroscope combination achieves the highest score considering the F1, F2 and F3 cases and the RF and MLP classifiers, which is followed by the combination of linear acceleration and gyroscope combination and this is followed by using only the accelerometer.

Example confusion matrices for only accelerometer sensor is given in Table 3.3 and for

Table 3.3: Confusion matrix using ACC with F3, RF and Case 4

		predicted class									
		smokeSD	smokeST	eat	drinkSD	drinkST	sit	stand	smokeWalk	walk	smokeGroup
actual class	smokeSD	**836**	43	0	36	4	0	1	18	0	318
	smokeST	79	**792**	15	59	239	3	0	1	0	66
	eat	0	11	**1132**	80	28	0	0	0	2	2
	drinkSD	50	88	120	**744**	213	0	4	3	3	29
	drinkST	9	206	48	233	**732**	19	0	1	0	7
	sit	0	1	0	1	0	**1253**	0	0	0	0
	stand	0	0	0	0	0	0	**1253**	1	0	1
	smokeWalk	29	1	0	8	0	0	0	**458**	7	51
	walk	0	0	0	10	0	0	0	8	**534**	1
	smokeGroup	339	54	5	12	5	0	0	35	1	**550**

Table 3.4: Confusion matrix using ACCGYR with F3, RF and Case 4

		predicted class									
		smokeSD	smokeST	eat	drinkSD	drinkST	sit	stand	smokeWalk	walk	smokeGroup
actual class	smokeSD	**860**	54	0	42	4	0	1	0	0	295
	smokeST	102	**856**	9	32	206	4	0	0	0	45
	eat	0	5	**1164**	79	5	0	0	0	0	2
	drinkSD	57	45	45	**917**	150	0	3	1	1	35
	drinkST	2	175	18	163	**871**	24	0	0	0	2
	sit	0	0	0	0	1	**1254**	0	0	0	0
	stand	1	0	0	0	0	0	**1253**	0	0	1
	smokeWalk	4	0	0	5	1	0	0	**520**	9	15
	walk	0	0	0	4	1	0	0	10	**537**	1
	smokeGroup	292	30	8	17	1	0	0	19	0	**634**

the fusion of accelerometer and gyroscope is given in Table 3.4. We use F3 and Case 4 in both these tables. We observe that smoking and drinking activities are confused with each other. However, two activities that are mostly confused with each other are smokeSD and smokeGroup. This may be due to the fact that people do not talk much in the group while collecting smokeGroup data and do not actively use hand movements. This may cause the activity to be confused with smoking while standing. We have previously observed higher recognition performance for smoking when its different variations were considered as one smoking activity (Shoaib, Scholten, Havinga and Incel, 2016). By comparing Table 3.3 and Table 3.4, as mentioned, we observe that the use of fusion of accelerometer and gyroscope reports an increase, compared to use of only accelerometer except stand and sit activities which are already well recognized.

As a conclusion, fusion of sensors improves the recognition performance, however, resource consumption due to additional sensors increases (Shoaib et al., 2017). Hence, the trade-off between high recognition rate and high resource consumption should be further investigated since battery lifetime is a limitation with smartwatches.

3.2.2 Scenario 2 : Less Activities

We define a second scenario for our evaluation. As mentioned, not all activities were performed by all eleven participants. Activity smokeWalk, walk and smokeGroup were performed in total only by 3, 3 and 8 participants, respectively. In order to look at the common smoking variations of smoking while sitting and standing, we removed the smoking in a group and smoking while walking. This also allowed us to create a balanced dataset where all these activities were performed by all participants. Thus, in Scenario 2, we consider all participants, but we do not include smoking while walking, smoking while in group conversation, and only walking. We evaluate this scenario by choosing the best case (Case 4 and F3) that is explored in Section 3.2.1.

3.2.2.1 Impact of Sensor Fusion and Classifiers

In this section, our aim is to analyze the effect of sensors and classifiers using Case 4 and F3 to better understand if focusing on less activities improves the performance of recognition. We present f1-score results of F3 for all classifiers with all sensors in Table A.4.

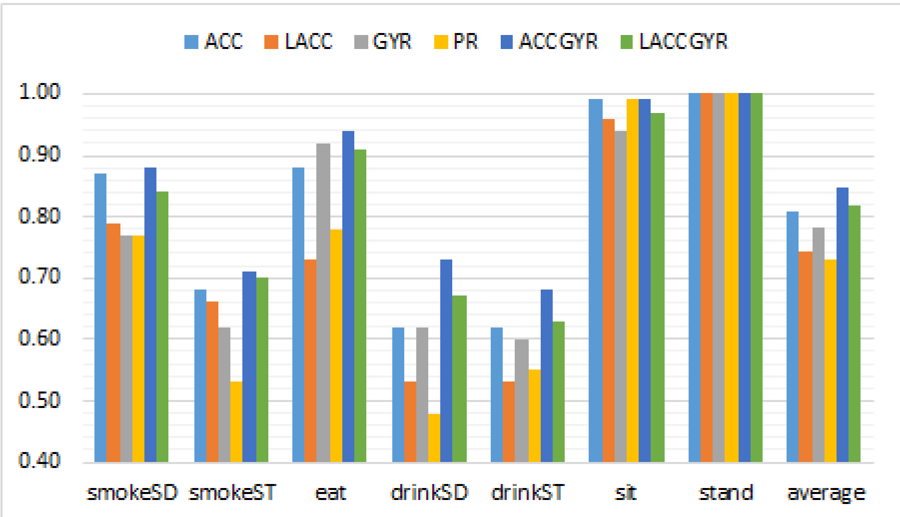


Figure 3.5: Impact of sensors using F3 with RF and Case 4

We present the results in Figure 3.5 for all sensors. This scenario brings significant improvement for the performance for smokeSD which is an increase of 23%, 22% and 21% using ACC, PR and ACCGYR and an increase of 12%, 13% and 17% using LACC, GYR and LACCGYR, compared to the previous scenario whose results are shown in Figure 3.4. Removing some variations of smoking activities, such as smokeGroup improves the performance for smokeSD. For the smokeST, eat and drinkSD, it performs equally or slightly better in a range of 5% and 1%.

On average, the best performances are obtained with the fusion of accelerometer and gyroscope for all activities. Similar to Scenario 1, the worst performance on average of all activities is obtained again using pitch and roll sensors (73%). Using only accelerometer gives a performance of 81% and only gyroscope 78%, which were 77% and %76 in Scenario 1. Combining accelerometer and gyroscope improves the average performance of activity recognition (85%). Similarly, using fusion of linear acceleration and gyroscope (82%) increases performance compared to only linear acceleration (74%) and only gyroscope (78%).

Table 3.5: Confusion matrix using ACC with F3, RF and Case 4

		predicted class						
		smokeSD	smokeST	eat	drinkSD	drinkST	sit	stand
actual class	smokeSD	**1131**	80	0	44	0	0	1
	smokeST	123	**843**	14	65	207	2	0
	eat	0	16	**1116**	87	35	1	0
	drinkSD	87	86	109	**767**	200	1	4
	drinkST	6	197	39	239	**759**	15	0
	sit	0	0	0	0	3	**1252**	0
	stand	0	0	0	0	0	0	**1254**

In Table 3.5 and Table 3.6, we show the confusion matrices of accelerometer and fusion of accelerometer and gyroscope respectively. Once again, it can be seen that mainly smoking and drinking are confused with each other. We observe less confusion for eat, sit and stand activities. Compared to Scenario 1, we see a significant decrease in the number of confused smokeSD activities. As mentioned in Scenario 1, smokeSD was

Table 3.6: Confusion matrix using ACCGYR with F3, RF and Case 4

		predicted class						
		smokeSD	smokeST	eat	drinkSD	drinkST	sit	stand
actual class	smokeSD	**1137**	80	1	33	4	0	1
	smokeST	120	**873**	4	37	215	5	0
	eat	3	4	**1167**	75	6	0	0
	drinkSD	68	63	53	**903**	163	0	4
	drinkST	3	184	14	185	**849**	20	0
	sit	1	0	0	0	2	**1252**	0
	stand	0	0	0	0	0	0	**1254**

mainly confused with smokeGroup activity. In this scenario, we removed smokeGroup, so this change positively affects the recognition of smokeSD. This shows that if smoking while standing and smoking while in group conversation are considered as one variation, it can easily be recognized. A similar result is observed previously in (Shoaib, Scholten, Havinga and Incel, 2016).

Table 3.7: Increase in f1-scores compared to Scenario 1

	SVM	**Increase**	**RF**	**Increase**	**MLP**	**Increase**
ACC	0.76	0.03	0.79	0.03	0.79	0.02
LACC	0.66	0.01	0.70	0.01	0.69	0.00
GYR	0.72	0.02	0.76	0.03	0.72	0.01
PR	0.45	0.09	0.71	0.06	0.65	0.06
ACCGYR	0.75	0.03	0.83	0.03	0.82	0.02
LACCGYR	0.70	0.02	0.78	0.05	0.77	0.00
average	0.67	0,03	0.76	0.03	0.74	0.02

To understand the best classifier among SVM, RF and MLP, in Table 3.7, we show the results of three classifiers as well as their improvements compared to Scenario 1. We observe that RF is the best classifier with an average performance of 76%. For the best sensor which is ACCGYR, the performance of corresponding classifier is 83%. There is not so much difference observed between RF and MLP. Performance of SVM is slightly lower compared to the other two. Our approach improves the recognition performance and the highest improvement is observed for PR sensor which is around 7%.

3.2.2.2 Impact of Height

After collecting the dataset, some extra questions were asked to the participants. These were about how often they smoke, their height and their age, as presented in Table 3.1. In this section, we explore whether the height of participants impacts the performance of activity recognition or not. For this purpose, among the eleven participants, we have created two separate groups consisting of participants with closer heights. Besides this, to balance the amount of data in groups, we determined that each group should contain an equal number of participants. The heights of participants in the first group (G1) are 180 cm, 181 cm and 181 cm for participant 1 (P1), P7 and P10, respectively. In the second group (G2), the heights are 164 cm, 167 cm and 170 cm for P6, P9 and P11, respectively. As mentioned in Section 3.2.2, all activities are not performed by every participant, we analyze the activities that were common to these six. Based on the results shown in Section 3.2.1 and 3.2.2, we performed our analysis with Case 4, feature set 3, accelerometer and gyroscope combination (ACCGYR) and RF classifier.

Table 3.8: Impact of height in-group using ACCGYR, F3, RF and Case 4

	smokeSD	**smokeST**	**eat**	**drinkSD**	**drinkST**	**sit**	**stand**
P1	0.97	0.93	0.96	0.81	0.84	0.96	1.00
P6	0.94	0.9	0.96	0.95	0.94	0.95	0.99
P7	0.92	0.86	0.96	0.84	0.82	0.94	0.99
P9	0.99	0.93	0.97	0.94	0.92	0.97	1.00
P10	0.95	0.91	0.96	0.88	0.84	0.93	1.00
P11	0.93	0.9	0.89	0.8	0.81	0.91	0.99
avg	0.95	0.91	0.95	0.87	0.86	0.94	1.00

Table 3.9: Impact of height out-group using ACCGYR, F3, RF and Case 4

	smokeSD	**smokeST**	**eat**	**drinkSD**	**drinkST**	**sit**	**stand**
P1	0.96	0.9	0.96	0.83	0.85	0.96	1.00
P6	0.96	0.95	0.92	0.93	0.92	0.95	0.99
P7	0.95	0.91	0.95	0.82	0.81	0.94	1.00
P9	0.99	0.92	0.97	0.93	0.9	0.96	0.99
P10	0.95	0.94	0.96	0.89	0.88	0.95	0.99
P11	0.93	0.9	0.9	0.82	0.85	0.94	1.00
avg	0.96	0.92	0.94	0.87	0.87	0.95	1.00

In the first phase, we perform in-group analysis which means the aim is to find the performance of each participant in his own group. More clearly, we train the data of G1 and G2 separately to create two models, then we test each participant of G1 with G1's model and same for G2.

In Table 3.8, only the results with the fusion of accelerometer and gyroscope are presented for ease of presentation and to focus on the impact of the height. If we compare the groups, all participants have over 90% performance for all activities except drinkSD and drinkST. The performance of G2 (P6, 9 and 11) is better than G1 (P1, 7 and 10), particularly for participant 6 and 9. Generally, in in-group analysis, there is no participant with a performance less than 80% for any activity. We achieve that the activity recognition performance of every participants in their height groups is improved compared to the previous sections, especially for smoke and drink activities. f1-score results of all sensors for six participants are presented in Table A.5.

Table 3.10: Impact of height for P4 using G1 and G2 separately as training data with ACCGYR, F3, RF and Case 4

	smokeSD	**smokeST**	**eat**	**drinkSD**	**drinkST**	**sit**	**stand**
Gr1	0.86	0.83	0.95	0.85	0.85	0.97	1.00
Gr2	0.83	0.78	0.96	0.86	0.86	0.98	1.00

In the second phase, we perform the out-group analysis. We test each participant with the training data from the other group rather than their own group. For example, P1 belongs to group 1, we test his performance by using group 2 (G2) as the training set. In Table 3.9, the results of ACCGYR (fusion accelerometer and gyroscope) are presented. We observe that the results are slightly different or not at all for some participants. Again, the participant 11 has the worst results for all activities except for drinkST. As observed in the first phase, the performance of G2 is better than, especially for participant 6 and 9 which have an average performance of 95% for both. All the sensor results are presented in Table A.6.

The average performance difference between in and out group analysis does not exceed 2% for all participants. Even if we look at the activities of individual participants, the performance difference never exceeds 5%, but usually is either 0% or 1%. Based on this, we find that using different height groups does not significantly affect the performance results.

Additionally, we decided to analyze an extreme case particularly using G1 and G2 as the training set, and we chose participant 4 as the outlier, with height of 156 cm, to test. Recognition performances with P4 are presented in Table 3.10. Comparing training set G1 and G2, we did not observe a difference over than 1% for all activities except for the smoking activities. On average, considering in-group and out-group analysis, the smoking performances were between 91% and 96%. However, in this analysis, for smokeSD and smokeST, we observe a performance of 86% and 83% using G1, and 83% and 78% using G2. As the height differences are too much between P4 and the two groups, it might affect the performance for smokeSD. Because in these two cases, usually, participants take their hand really down and then it comes to the mouth where the angular distance might make a difference. However, smokeST is a way complex activity. Because the hand to mouth angular distance can take many shapes depending how someone smokes. We can report that height does affect the smokeSD if it is significantly different as the case of P4. However, smokeST can be affected by the different variations of activity.

In the third phase, we realize a per participant analysis. The aim is to find whether the height of participants has an impact on the recognition performance using statistical analysis. For this purpose, we train the data of all eleven participants together for creating a classification model. Then, we test each participant with a particular height, separately, using this model. Firstly, f1-score results of all sensors were obtained for all participants (see Table A.7). Then, we explore whether the participants' activity performances change significantly based on the height parameter. To examine this, we used regression analysis with the default confidence level which is 95%. After creating the regression model for each activity, particularly, we focused on the significance-F value. In Table 3.11, we presented the significance-F results of the regression model for all activities based on the height parameter. It shows that, all generated regression results are not significant for all the considered activities.

Table 3.11: Significance-F results

Activity	smokeSD	smokeST	eat	drinkSD	drinkST	sit	stand
Significance	0.13	0.37	0.77	0.43	0.71	0.30	0.10

In Table 3.12, we also present the detailed analysis of the performance differences between per participant analysis (see Table A.7) and cross validation analysis (Section 3.2.2) considering Scenario 2. In Scenario 2, cross validation results do not contain performance per participant, we use average f1-scores of this analysis to make comparisons. It is clear that, per participant analysis performs better for all activities except sit activity with all sensors and stand with the gyroscope. Particularly for smokeST, drinkSD and drinkST activities, this approach provides the high improvement on the performance which is in a range of 11% and 22%. As the performance for stand was very high which is almost 100%, it is not possible to achieve an improvement for this activity. However, we observed a decrease of 2% for gyroscope sensor. In Table A.7, for the extreme case P4, we see that his performance is remarkably low for smokeSD and smokeST. However, in other activities, it is similar. In this case, we observe similar trends between in group and per participant analysis. So, these two activities may be affected by height. Although, out-group and in-group results were similar and regression analysis considering all participants and all activities do not show a significant difference in recognizing these activities, if we consider a participant with a significantly different height than all participants, we observe a decrease in the recognition

Table 3.12: Performance difference between per participant analysis and cross validation results

	smokeSD	smokeST	eat	drinkSD	drinkST	sit	stand
ACC	0.07	0.20	0.05	0.21	0.20	-0.05	0.00
LACC	0.07	0.13	0.12	0.22	0.20	-0.11	0.00
GYR	0.08	0.14	0.02	0.15	0.11	-0.15	-0.02
ACCGYR	0.07	0.20	0.02	0.13	0.19	-0.04	0.00
LACCGYR	0.05	0.13	0.03	0.13	0.14	-0.11	0.00

of smokeSD and smokeST activities. Further research is needed to investigate this in detail.

3.3 Discussion

In this section, we summarize our findings and identify the open issues for further investigation.

— Simple features perform sufficiently well : When we analyze the performance of recognition with different feature sets, we observe that F1 (min, max, skewness, kurtosis) and F3 (median, std, min, max, range, mean) perform the best. These features are all time domain features and can be computed easily. If smoking recognition is performed online on watches or on other wearables (Rezaie and Ghassemian, 2017), this will be an advantage.

— Larger windows perform better : Compared to shorter window sizes, 1 to 5 seconds, used in the recognition of simple activities, such as walking, sitting, larger window sizes perform better for smoking recognition. In our analysis, Case 4 (30 seconds window size with 50% overlap) performed slightly better than the other cases.

— When efficient features are used, performance of classifiers is similar : In our analysis, RF classifier is the best performing classifier in most cases, which is followed by MLP and then SVM. However, when efficient feature sets are used, such as F1 or F3, their performances are very similar. Again, if online recognition is to be performed, in another study (Shoaib et al., 2017), we show that these classifiers perform well in terms of resource consumption, such as battery, memory and CPU cycles.

— Combination of gyroscope with accelerometer improves the results : The best performances are obtained with the fusion of accelerometer and gyroscope for all activities. Considering all activities, using only accelerometer exhibits a performance of (77%) on average and only gyroscope (76%). Combining accelerometer and gyroscope improves the average performance of activity recognition to 83% on average, considering all the activities. Fusion of linear acceleration and gyroscope increases performance compared to only using linear acceleration and only gyroscope as well, however the results are not as high as the combination of accelerometer and gyroscope. We should also note that linear acceleration sensor consumes more battery than accelerometer.

— Impact of height needs to be further investigated : While there is no significant difference between in group and out group analysis, the performance for smoking while standing is affected when tested on a participant with a different height than those participants in the training data. On the other hand, impact of height is not very clear in other activities, such as smoking while sitting since it is a more complex activity and the hand to mouth angular distance can take many shapes depending how someone smokes.

— Smoking in group is more difficult to recognize : Compared to the other variants, smoking in a group is more difficult to recognize due to different patterns exhibited by the participants and it is mostly confused with smoking while standing. This can be further investigated with the use of both phone and watch data to obtain better performance.

4 DYNAMIC SMOKING RECOGNITION WITH SMARTWATCH SENSORS

4.1 Methodology For Dynamic Smoking Recognition

In this section, we explain the methodology followed for the recognition of smoking activities. First, we will present the datasets and feature sets used. Then, we will give the details of our approach to create the dynamic parameter selection framework for resource efficient activity recognition.

4.1.1 Background : Dataset, Feature Sets

We use the same dataset used in the previous chapter. During this study, we are particularly interested in four feature sets which are found as the most successful feature sets in the previous chapter. They contain only time domain features. We only calculate these features for the accelerometer and gyroscope sensors based on our findings presented in Chapter 3. In addition to x, y and z axis, we calculate a fourth dimension called the magnitude (m axis) to better understand movement changes. It is calculated as the square root of the sum of the x, y and z axis. All features are calculated for four dimensions except the correlation feature in F4 which is the relationship between each axis and it gives one feature. Details about feature lists and number of features are presented in Table 4.1.

The best two feature sets are identified as F1 and F3. Firstly, in the dynamic parameter selection system, we choose a feature set based on the activity. The principle is simple, when the end of an activity session is detected, in the first session, the algorithm computes both F1 and F3 features, chooses the best in terms of accuracy and saves the activity name and its best feature set. In the next sessions of this activity, gets the name of the activity's best feature set and compute its features to make the classification. Additionally, it checks regularly the best feature set of activity and updates its best feature set at regular session intervals of activity to prevent a permanent error. More details will be explained in Section 4.1.5.3

Secondly, we try to create feature sets using feature selection algorithms available in Weka tool (Witten et al., 2016) to select the best features in order to reduce the

Table 4.1: Details of feature sets

Feature Set	Features	# of features
F1	min, max, skewness, kurtosis	16
F2	min, max, mean, standard deviation	16
F3	min, max, mean, standard deviation, median, range	24
F4	mean, standard deviation, integration, correlation, root mean square, absolute difference	21

computation cost and hence reduce the resource consumption. In Weka tool, feature selection is divided into two parts which are attribute evaluator and search method. Attribute evaluator is the technique which evaluates each attribute in the dataset. Search method investigates different feature combinations to observe a list of chosen features. Rather than using a specific method, we consider different methods with different criteria and select the best feature set according to their rankings. Feature selection techniques used in this study are described below. The first seven belongs to the attribute evaluator and the rest to the search method. These brief definitions are taken from the online documentation of Weka.

— CfsSubsetEval : Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them.

— CorrelationAttributeEval : Evaluates the worth of an attribute by measuring the correlation (Pearson's) between it and the class.

— GainRatioAttributeEval : Evaluates the worth of an attribute by measuring the gain ratio with respect to the class.

— InfoGainAttributeEval : Evaluates the worth of an attribute by measuring the information gain with respect to the class.

— OneRAttributeEval : Evaluates the worth of an attribute by using the OneR classifier.

— ReliefFAttributeEval : Evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class.

— SymmetricalUncertAttributeEval : Evaluates the worth of an attribute by measuring the symmetrical uncertainty with respect to the class.

— BestFirst : Searches the space of attribute subsets by greedy hillclimbing augmented with a backtracking facility.

— GreedyStepwise : Performs a greedy forward or backward search through the space of attribute subsets.

— Ranker : Ranks attributes by their individual evaluations.

Feature extraction was made separately, one for the complex activities and one for the simple activities. Features are extracted from four dimensions of sensors : x, y, z and magnitude (the sum square root of x, y and z axes). As the focus of using a gyroscope with accelerometer is recognizing complex activities, we extract both accelerometer and gyroscope features for the complex activities while extracting only accelerometer features for the simple ones. We obtain two feature files, one for simple activities and one for complex activities. We give them as inputs to the feature selection algorithms. We observe results of each feature selection algorithm for each of simple and complex activities groups. Finally, we choose the best feature set of each group based on results. More details about how we choose the best feature sets using different feature selection algorithms' results can be found in Section 4.2.4.

4.1.2 Scenario Creation

We created a scenario for each participant with the aim of making the order of activity sessions more realistic. More clearly, for example, in our scenario dataset for P1, we consider that the participant first smokes while standing later sits for a while, then he starts to walk and again smokes in a group conversation. After that, he sits and drinks a coffee, then he stands again, etc. Our aim is to create a scenario using each participant's data to be more realistic and close to a daily life pattern. As all activities were not performed by all participants, the scenarios have some differences which are noted in Table 4.2. After creating eleven scenario files, we merged files respectively to make our whole scenario dataset. In participants' raw data, activity sessions are not specified. We took every 5 minutes as an activity session, based on the experience we had when collecting data. The details in the scenario can be found in Table 4.3.

Table 4.2: Activity combination for each scenario type

Scenario Type	**Activity combination**
T1	smokeSD→walk→smokeGroup→sit→drinkST→stand→smokeWalk→stand→eat→sit→smokeST→stand→drinkSD→walk
T2	smokeSD→sit→smokeGroup→sit→drinkST→stand→eat→sit-→smokeST→stand→drinkSD→stand
T3	smokeSD→sit→drinkST→stand→sit→smokeST→stand→drinkSD→stand

Table 4.3: Information about participants' sessions

Participant	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
Scenario Type	T1	T1	T1	T2	T2	T2	T2	T2	T3	T3	T3
Number of cycle	9	10	10	8	4	4	4	5	5	4	4
Total number of session	126	140	140	96	48	48	48	60	50	40	40

Table 4.4: Activities and their states

State	**Activities**
Complex	smokeSD, smokeST, eat, drinkSD, drinkST, smokeWalk, smokeGroup
Simple	sit, stand, walk

4.1.3 Rule Based State Detection Algorithm

In our dataset, there are two different types of activities which are simple and complex (see Table 4.4). The parameters required for recognizing simple and complex activities using smartwatches are different. The state of activity (simple or complex) can be used as a trigger to activate or deactivate some parameters, such as gyroscope. In our case, we identified sitting, standing and walking as the simple activities whereas the other remaining activities are complex. In (Shoaib, 2017), Shoaib proposes to use the standard deviation value of each window as a trigger for activating gyroscope to recognize the smoking activity. He states that the average value of standard deviation is almost zero when device is in an inactive state. But in our case, there is not an inactive state, all states are composed of active activities and continuous calculation of standard deviation may have a negative effect on the resources of the device.

Firstly, using the accelerometer sensor, we tried to use the standard deviation and variance of magnitude as a trigger. The variance was a more successful indicator, an accuracy of 65% is achieved to differentiate simple and complex activities. Since this value is not good enough for starting, we investigate range, min, max and the results shows us there is not a clear threshold value to differentiate simple activities from complex ones (see Table 4.5).

We thought that it would be better to go over the gyroscope data for complex activities. It also failed with less than 50% accuracy. We decided to examine the ACC axis individually, instead of the magnitude, in order to see if using only ACC axis can be more successful. For variance, we tried ACC axis x, axis y and axis z separately. For each axis, the variance values of activities were very close to each other. This prevented

Table 4.5: Average std, variance, range, min and max values of activities using accelerometer

	std	variance	range	min	max
smokeSD	0.91	0.96	9.81	5.6	15.4
smokeST	0.68	0.57	8.46	6.6	15.1
eat	0.45	0.24	4.55	7.6	12.15
drinkSD	0.42	0.25	3.36	7.8	12.09
drinkST	0.33	0.19	3.37	8	12.37
smokeWalk	1.73	3.24	12.61	4.77	17.38
smokeGroup	1.14	1.49	12.13	4.65	16.78
sit	0.17	0.04	1.64	9.2	10.87
stand	0.07	0.006	0.71	9.5	10.22
walk	2.21	5	12.34	5.28	17.62
threshold	if 0.2$<$ value $<$ 2.0 complex ; else simple	if 0.1$<$ value $<$ 4.0 complex ; else simple	-	-	-

to determine a threshold to distinguish the state of activities. It is difficult to detect the state of activity using simply low and high threshold values, more clearly giving an interval for complex and simple states.

Secondly, we analyzed the data distribution of accelerometer's x, y, z, and magnitude axis individually, on the Weka tool. Particularly for the simple states, the graphic does not correspond to a standard normal distribution (see Figure 4.1). Thus, we discontinue to work on the standard deviation and its derivatives. It seems more efficient to work on the mean value. We investigate x, y, z axes and magnitude individually and combination of three axes using a rule-based method of Weka which is called JRIP. Using axis x, y and z in combination, we extracted 14 rules with a success of 99% for complex and of 98% for simple states. Details about other results are presented in Table 4.7, confusion matrix of mean x, y, and z axis is in Table 4.6. We imported rules in Python and tested these generic rules for each participant and in overall, we detect the state of activity between 96.3%-93.6% accuracy. Finally, we examine false detections state by state, there were similar types of incorrect state predictions. Most of the false detections were caused by only one time window. For example, among 10 windows, all are predicted as a complex state except the fifth which is predicted as simple. This is impossible to happen since the activities are continuous, so this can be an incorrect prediction. As proposed in (Shoaib, Scholten, Havinga and Incel, 2016), we followed a similar state correction method explained in Section 4.1.5.3, that utilizes neighboring states to

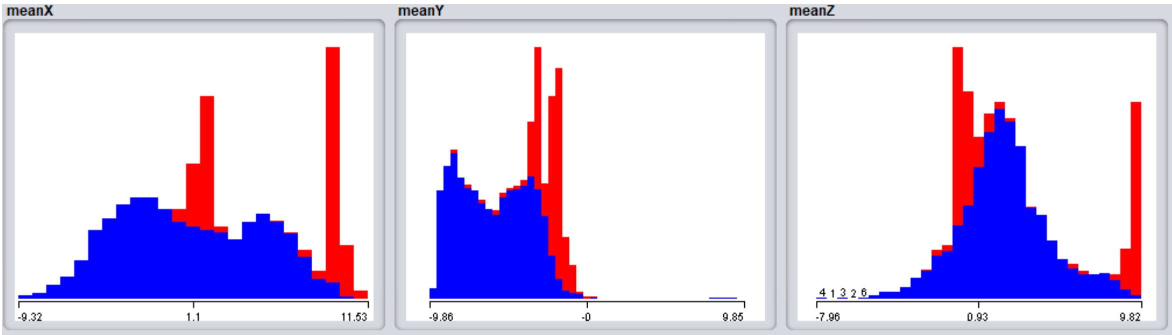


Figure 4.1: Visualization of mean data on Weka (blue is distribution of complex and red simple states

improve the detection. Correcting these false predictions brings an increase of 3% on the performance of state detection.

Table 4.6: Confusion matrix using accelerometer's mean x, y, and z features in JRIP

		predicted class	
		complex	simple
actual class	complex	5831	60
	simple	39	2266

Table 4.7: Results of JRIP using mean of different axis

	# of rules generated	Time taken to build model (sec)	f1-score of complex activities	f1-score of simple activities	Weighted average f1-score of all activities
Mean X	21	1.37	0.958	0.892	0.939
Mean Y	20	0.41	0.957	0.889	0.938
Mean Z	10	0.19	0.956	0.871	0.932
Mean M	23	1.60	0.934	0.856	0.912
Mean X, Y, Z	14	0.82	0.990	0.983	0.988

4.1.4 Parameter Decision

In our dataset, there are two different types of activities which are simple and complex. We need to detect this state as mentioned in Section 4.1.3 and create a dynamic parameter selection algorithm depending on activity. The parameters are sampling rates, window sizes, sensor(s) and feature sets. This algorithm will be able to make an energy-efficient activity recognition process while retaining a good enough f1-score.

Dataset is collected from sensors at 50 Hz. Choice of sampling rate depends on activity. Using a fixed high sampling rate will be sufficient to recognize smoking related activities but it will consume more battery. Simple activities can be recognized using a low sampling rate and complex vice versa. The sampling rate has an important impact on the resource consumption. Therefore, we investigate 1, 2, 5, 10, 25 and 50 Hz based on related studies. Similarly, for window size, we investigate 5, 10, 20 and 30 seconds windows without overlap. These are the most common window sizes used in activity recognition studies. Additionally, we need to generate data at lower sampling rates from our original 50 Hz data. Thus, we look at the ratio between 50 Hz and the new sampling rates. Then we add each repetition of this ratio to the new sampling rate's file and this creates a file with the new sampling rate. The pseudo code of this conversion is presented in Figure 4.2.

```
Original sampling rate ← 50
New sampling rate list ← {25, 10, 5, 2, 1}
Input file ← Original sampling rate's data file
for each New sampling rate in New sampling rate list do
        Output file ← An empty New sampling rate's data file
        i ← 0
        for each Row in Input file do
                if i % (Original sampling rate/ New sampling rate) is equal to 0 do
                        add Row to the Output file
                i ← i + 1
```

Figure 4.2: Decrease sampling rate algorithm

Simple activities can be recognized using only the accelerometer sensor but for the complex activities, it is better to use the accelerometer sensor with the gyroscope. For example, walking activities can be better recognized using only an accelerometer with a low sampling rate, whereas smoking while sitting can be better recognized using fusion of accelerometer and gyroscope with a higher sampling rate. Thus, in this work, we focus on using only accelerometer and accelerometer with the gyroscope. As observed in Chapter 3, the performance of feature sets depends on the activity. The best two feature sets are identified as F1 and F3. As mentioned in Section 4.1.1, firstly we work with these two feature sets than we apply feature selection methods.

4.1.5 Model Creation

In this section, we create our classification models using smoking data. We use random forest classifier (RF) which is one of the extensively used classifiers in the human activity recognition due to its high performance in the field (Shoaib, Scholten, Havinga and Incel, 2016; Sen et al., 2016) and it was found to be the best classification algorithm in the previous chapter. We create training models offline then use them in our algorithm. Models are created using Scikit-learn which is a Python based library and it is extensively used in machine learning studies. In the activity recognition process, if there is a large number of activities to classify, the model will be more complex, this will consume more CPU and power (Seneviratne et al., 2017). Based on this, instead of using a classification model that includes 10 activities, we can say that it would be better to use separate models for three simple activities and separate models for seven complex activities. Besides, if we use a model that contains all of the activities, it will have to be generated with both the accelerometer and the gyroscope data's features. Because of using two sensors model for both simple and complex activities, the complexity will increase, and this will consume more resource. For the complex activities, to observe a high recognition performance, we need to use the fusion of accelerometer and gyroscope, but for the simple activities only accelerometer data will suffice. For each combination of the sampling rate, window size and feature set, we created three different models : complex activities' model (ComplexAct), simple activities' model (SimpleAct) and all activities' model (AllAct). In the dynamic system, we will use only ComplexAct and SimpleAct but only for the first phase as we have not already differentiate the state of activity, we use AllAct models.

4.1.5.1 Static Parameters

The first phase of the study contains a static activity recognition system. The aim is to examine both the static recognition system and to provide a benchmark for the dynamic system. In this context, we used all the combinations of parameters, shown in Figure 4.3, to experiment with the scenario data. As we mentioned in Section 2.2, we use five different sampling rates, four window sizes, one individual sensor and one fusion of two sensors and two feature sets, thus, there are 96 combinations for the static analysis. We test scenario data with the created training models. This benchmark will be effective for resource consumption analysis as well as for evaluating performance.

4.1.5.2 Semi-Dynamic Recognition

In the second phase, we first experiment with dynamic sensor and feature selection. Based on Chapter 3, considering the trade-off between resource consumption and per-

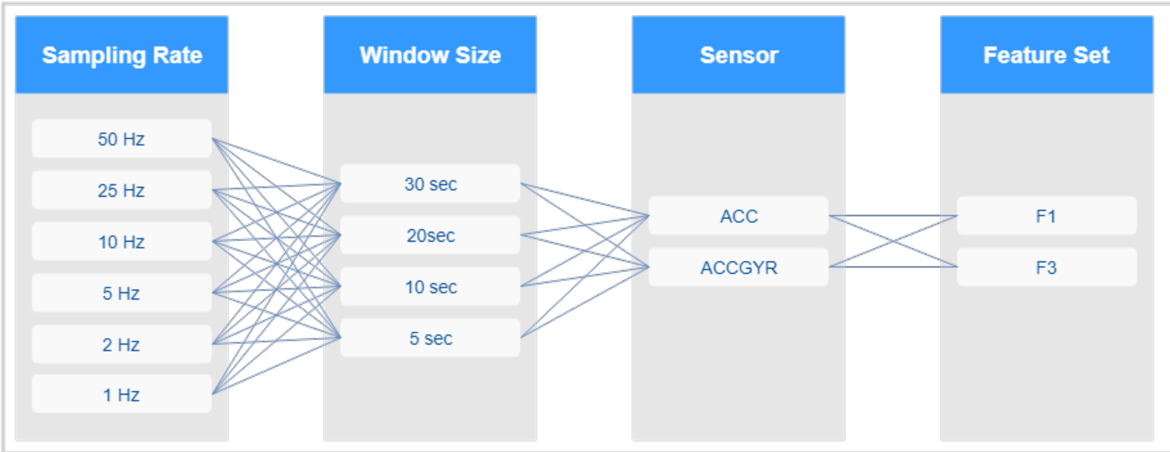


Figure 4.3: Parameters used in static parameters analysis

formance balance, it is clear that for simple activities ACC and for complex ACCGYR yield a better performance. As mentioned in Section 4.1.1 the performance of feature sets depends on the activity so in our semi-dynamic recognition system, choice of the feature set is given in the beginning. When an end of an activity session is detected, then both F1 and F3 are extracted for this session and after classification, the best performing set is chosen and this information is saved for the upcoming decisions. In subsequent repetitions of the activity, only the classification for the selected feature set is made. The best feature set is updated at certain repetition intervals of activity, and this is done for each activity. We realize semi-dynamic recognition for each combination of sampling rate and window size. Hence, this gives us 24 different results for each combination.

4.1.5.3 Dynamic Activity Recognition

In the third phase, we also dynamically select the sampling rate and window size based on the state of activity. In Figure 4.4, we present our dynamic parameter selection algorithm for activity recognition. This algorithm contains three main modules which are state detection module, session module and classification module.

Firstly, in the state detection module, we load training models for simple and complex activities. We initialize the window size of simple activities to 30 seconds and complex activities to 20 seconds, then sampling rate of simple activities (sRSimple) to 1 Hz and complex activities (sRComplex) to 5 Hz. Besides, in order not to miss a complex window, we set default value of lastState to complex. The current samplingRate, sizeOfWindow, sensor(s) are determined by lastState information and then, reading process starts. When a window is read, thanks to the state detection method explained in Section 4.1.3, we detect whether this window is a piece of simple or complex activity, called currentState. If currentState is different from lastState which signifies a change of state, we check this change. As we mentioned in Section 4.1.3, the majority of false state detections are caused by only one window. Therefore, we correct such errors, if needed, at verify state change. If there is a state change, either this may be the end of the current session, or in other words the activity is ended, or there can be a false detection. In order to differentiate these two cases, we continue with the next window, and then if it is again the same state then this means the state has changed, or a new activity has started. If it is not the same state then there was a false decision. Simply, if the sequence is as CCCSC (C :complex, S :simple) this is a false detection but if it is as CCCSS then it means the complex state has ended. The rest of the false detections are caused by the continuous 2 or 3 windows errors. So, there were a maximum of 3 continuous false detection in a session and in this analysis a window size of 20 seconds was used. That's why we set a minimum duration of activity to accept an end of activity session. This duration is 60 seconds for analysis, so we did not classify sessions less than that. For example, if it is identified as complex, then we update window size and sampling rate settings, also activate gyroscope in addition to the accelerometer. Then, we continue to read data with complex activity settings until the end of activity session is detected. To determine the end of a session, in addition to 60 second limit, after observing a number of the same type of activity (complex or simple), the observation of two cumulative different window states is needed as mentioned above. This signifies the transition to a different session.

Secondly, session module is the module where the parameters and metrics related to the session are set which are sessionNumber, avgF1Score and FeatureSet. Each activity, such as smokeWalk or smokeSit has a specific sessionNumber which keeps track of how many times this activity has been performed. Thus, we will be able to calculate the average f1-score (avgF1Score) of each activity. If it is the first time that algorithm encounters this type of activity, sessionNumber is set to zero, if not the sessionNumber of activity will be increased by one. For the first session, it calculates both F1 and F3 features, chooses the best in terms of accuracy and saves activity name and its best feature set. If this is not the first session of this activity, gets activity's best feature set and makes classification using only with this feature set. Additionally, we check regularly the best feature set of activity and update its best feature set. The reason is to keep the system up-to-date and to ensure that the wrong decisions are not made constantly. During this experiment, we update the best feature set of each activity every 5 repetitions, this number can be different depending on the dataset.

Thirdly, we continue with the classification module. Based on state of activity session read, we extract features determined above and for related sensors. We make a classification using RF training models already loaded. We get a new f1-score and calculate new avgF1Score value of activity. If the sessionNumber of activity is 1 or divisible by 5, by comparing classification results (f1-scores) of F1 and F3, we choose the best feature set of activity. Finally, for activity read, we update its best feature set, avgF1Score and sessionNumber and set durationOfSession to zero.After classification, we go back to the first module and wait for the incoming window. Differences between static, semi-dynamic and dynamic analysis can be found in Table 4.8.

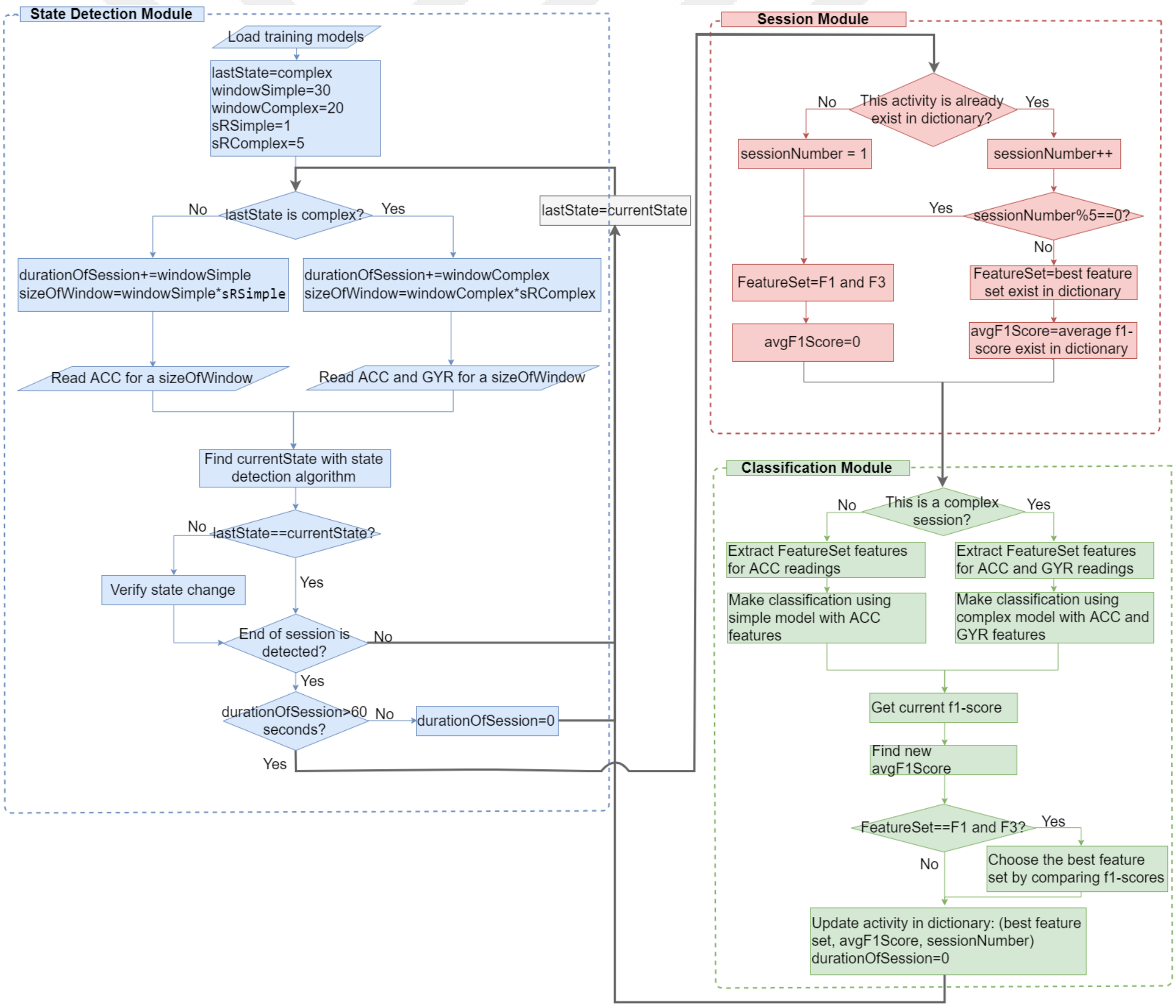


Figure 4.4: Dynamic activity recognition algorithm

Table 4.8: Differences between the three analyzes

	Static Analysis	**Semi-Dynamic Analysis**	**Dynamic Analysis**
Sampling rate	Static (50, 25, 10 ,5, 2 and 1 Hz)	Static (50, 25, 10 ,5, 2 and 1 Hz)	Dynamic (5 or 1 Hz)
Window size	Static (30, 20, 10 and 5 sec)	Static (30, 20, 10 and 5 sec)	Dynamic (20, or 30 sec)
Sensor	Static (ACC and ACCGYR)	Dynamic (ACC and ACCGYR)	Dynamic (ACC and ACCGYR)
Feature set	Static (F1 and F3)	Dynamic (F1 or F3)	Dynamic (F1 or F3)
Number of classification model extracted	1 (from all data)	2 (from complex data and simple data)	2 (from complex data and simple data)
Tested dataset	Scenario data	Scenario data	Scenario data
Min. duration of activity expected	0 second	60 second	60 second

4.2 Performance Analysis

In this section, the results obtained by the methodology explained in Section 4.1 are presented. In Section 4.2.1, we analyze all the combinations of sampling rates, window sizes, sensors and feature sets shown in Figure 4.3. This gives us a benchmark for the next steps of both resource consumption and recognition performance analysis. In Section 4.2.2, we perform a semi-dynamic analysis. Here, the feature set and the sensor are selected dynamically. In Section 4.2.3, all four parameters are selected in a dynamic way. To create two effective feature sets depending on the state of activity, such as complex and simple, in Section 4.2.4, we apply the feature selection methods, then repeat static, semi-dynamic and dynamic analysis with the new feature sets. In the last remaining section, we make a resource consumption analysis in terms of CPU, memory and energy usage. The scenario data explained in Section 4.1.2 was used in all of these analyses. In the performance evaluation, we present f1-score as a performance metric which ranges between zero and one.

4.2.1 Static Parameters

In this section, we evaluate separately all combinations of parameters mentioned in Section 4.1.5.1 on the recognition performance of simple and complex activities. Compared to complex activities, recognizing simple activities, such as sitting is always more successful. The main challenge is to achieve success in the complex activities. Therefore, in Figure 4.5, we present f1-score results of complex activities using the fusion of accelerometer and gyroscope sensors with F1 feature set for each pair of window size and sampling rate. We observe that eating activity achieves the highest scores comparing all activities (around 94%) and smokeGroup achieves lowest scores (around 72%) for all cases. When we look at the effect of sampling rate, using a high sampling rate slightly increases the performance for all activities. Activities have almost the same performance for 50, 25 and 10 Hz, but have a noticeable decrease for 5, 2 and 1 Hz. For example, the performance for smokeST is 83%, 83%, 82%, 81%, 79% and 77% using 50, 25, 10, 5, 2 and 1 Hz respectively, with 10 seconds. The difference is 1% among the three highest sampling rates and 4% for the lowest three. Additionally, we observe that changing window size has a similar effect compared to sampling rate. For example, using 10 Hz, smokeSD is recognized with a performance of 80%, 77%, 76% and 74% using 30, 20, 10 and 5 seconds, respectively. Increasing the window size increases the performance with some exceptions for 30 seconds. For example, drinkST using 20 seconds exhibits better performance compared to 30 seconds for all sampling rates. Similarly, recognition performance of smokeWalk using 20 seconds is higher than using 30 seconds for most of the sampling rate values.

Figure 4.5: Performance comparison of complex activities using ACCGYR with F1 at different sampling rates and window sizes

In Figure 4.6, we present a smoking related activity which is smokeWalk using different parameter combinations. In addition to what we observed for sampling rate and window size in Figure 4.5, here, we consider the effect of using different sensors and feature sets. To better analyze, firstly, we compare ACC F1 with ACCGYR F1 and ACC F3 with ACCGYRF F3. We see clearly that there is a difference between using only ACC and using ACCGYR. Using ACC has a lower performance than using ACCGYR and the difference reaches 6% for F1 using 1 Hz - 5 sec and 10 seconds, and 9% for F3 using 1 Hz - 10 seconds. Secondly, we compare feature sets, so ACC F1 with ACC F3 and ACCGYR F1 with ACC F3. Performance of a feature set is not significantly better than the other, it varies depending on the sampling rate and window size. The variations are in a range between 0% and 4% which are not very high.

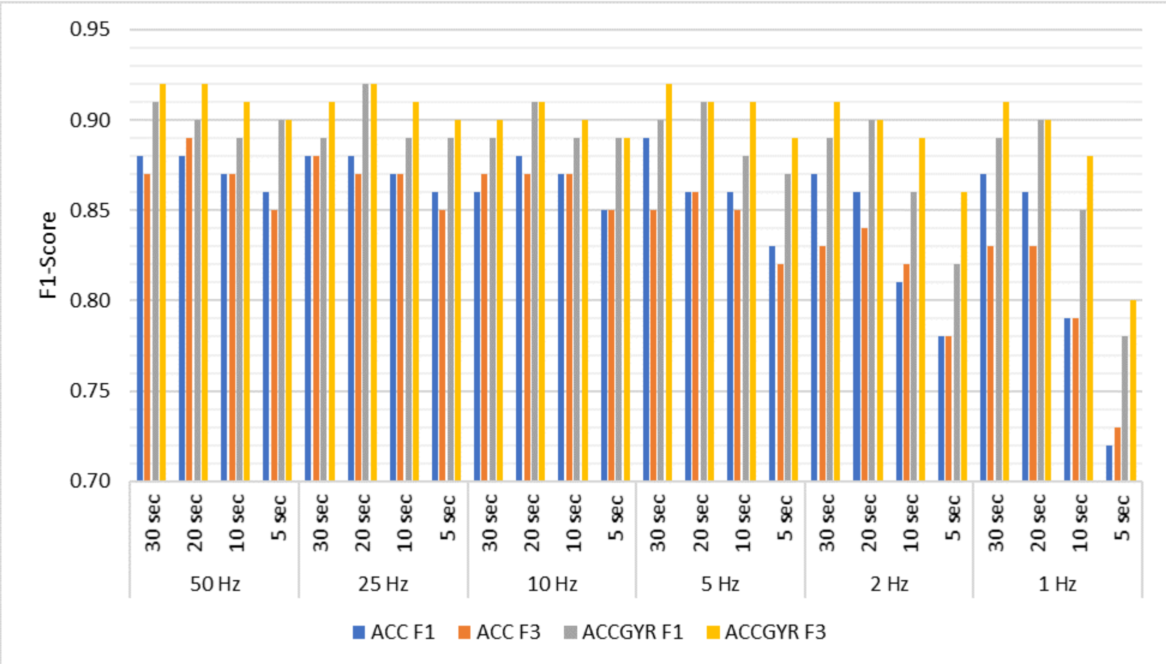


Figure 4.6: Performances of smokeWalk activity with different feature sets, sensors, sampling rates and window sizes

4.2.2 Semi-Dynamic Parameters

After static parameter experiments, for both complex and simple activities, the goal in this section is to select a sampling rate as low as possible and a window size as high as possible which guarantee a certain f1-score for all activities. We chose the limit of f1-score as 85%, but depending on the importance of activity recognized, it can be chosen very high or vice versa. The performance results of 10, 25 and 50 Hz are already higher than 85%, so we present only 5, 2, 1 Hz results in this section.

In Table 4.9, the semi-dynamic analysis results are presented. Using dynamic sensor and feature set selection, smokeSD, eat and smokeWalk activities achieve a higher recognition rate than 85% with all sampling rates and window size pair. smokeGroup has the lowest performance compared with the rest. Since we set a limit of 85%, it is better not to select 1 and 2 Hz as a sampling rate because it is always under 85% for

Table 4.9: Performance of complex activities using dynamic feature set and sensor selection at different sampling rates and window sizes

		smokeSD	smokeST	eat	drinkSD	drinkST	smokeWalk	smokeGroup
5 Hz	5 sec	0.90	0.89	0.97	0.88	0.88	0.95	0.87
	10 sec	0.93	0.89	0.97	0.87	0.85	0.95	0.85
	20 sec	0.92	0.87	0.94	0.90	0.87	0.96	0.87
	30 sec	0.92	0.87	0.91	0.94	0.85	0.96	0.87
2 Hz	5 sec	0.88	0.86	0.96	0.86	0.86	0.96	0.84
	10 sec	0.92	0.88	0.96	0.86	0.84	0.93	0.82
	20 sec	0.91	0.85	0.93	0.88	0.86	0.96	0.83
	30 sec	0.91	0.87	0.91	0.93	0.86	0.97	0.82
1 Hz	5 sec	0.88	0.85	0.96	0.84	0.85	0.93	0.84
	10 sec	0.91	0.87	0.96	0.81	0.81	0.95	0.80
	20 sec	0.90	0.85	0.92	0.87	0.85	0.96	0.83
	30 sec	0.90	0.85	0.90	0.91	0.85	0.95	0.81

smokeGroup and for some cases it is under for drinkSD and drinkST. So, we chose 5 Hz as a sampling rate for complex activities. Among window size values, it is more advantageous to select 30, 20, 10 and 5 seconds respectively for lower resource consumption. 5 Hz with 30 seconds is at the limit value for drinkST so we chose 20 seconds as a window size for complex activities. Using 5 Hz with 20 seconds gives a performance in a range between 87% and 96%. Since the performance of 20 seconds is already high enough, it is not needed to look at the 10 and 5 seconds which will consume more resource and do not have a high performance difference compared to 20 seconds.

Table 4.10: Performance of simple activities using dynamic feature set and sensor selection at different sampling rates and window sizes

		sit	stand	walk
5 Hz	5 sec	0.99	0.99	0.96
	10 sec	0.97	0.99	0.96
	20 sec	0.98	0.99	0.97
	30 sec	0.99	0.99	0.97
2 Hz	5 sec	0.99	0.99	0.95
	10 sec	0.97	0.99	0.96
	20 sec	0.98	0.99	1.00
	30 sec	0.99	0.99	0.97
1 Hz	5 sec	0.99	0.99	0.95
	10 sec	0.97	0.99	0.99
	20 sec	1.00	0.99	1.00
	30 sec	0.99	0.99	0.94

Recognizing simple activities is easier than the complex ones. In Table 4.10, the performance for simple activities is presented. We observe that the semi-dynamic algorithm recognizes very good even with the lowest sampling rate and highest window size. Therefore, we chose 1 Hz with 30 seconds for simple activities.

4.2.3 Dynamic Parameters

In addition to dynamic selection of feature set and sensors made in the semi-dynamic parameter selection, in this section, the sampling rate and window size are also dynamically selected. Performance results of dynamic parameter selection are presented in Figure 4.7. For simple activities we achieve very high recognition performance as expected. For the rest of activities, the best performance is achieved for smokeWalk which is 95% and the worst for smokeGroup which is 84%. Standing related complex activities, such as smokeSD and drinkSD are better recognized compared to their sitting related forms, such as smokeST and drinkST. For eating we achieve the second best performance (94%) after smokeWalk among complex activities.

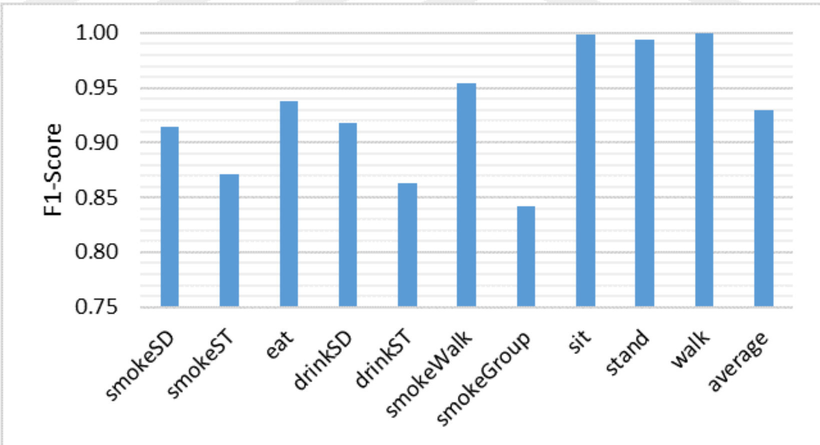


Figure 4.7: Performance of all activities using dynamic parameters

4.2.4 Feature Selection

In this section, the aim is to find two feature sets, one for simple and one for complex activities, using feature selection algorithms. In the first phase, we merge the list of best four feature sets to create an overall feature set. So, it contains mean, std, median, skewness, kurtosis, min, max, range, integration, correlation, rms, and absolute difference features for all x, y, z axes and magnitude value. From the raw accelerometer data of simple activities and from the raw accelerometer and gyroscope data of complex activities, we extract all features for a window size of 20 seconds with 50 Hz.

Table 4.11: According to complex and simple activities, the number of high, medium and low features selected by each feature selection algorithm

		Simple Activities				Complex Activities			
Attribute Evaluator	Search Method	# of selected features	# of high impact features	# of medium impact features	# of low impact features	# of selected features	# of high impact features	# of medium impact features	# of low impact features
CfsSubsetEval	BestFirst	6	4	2	0	26	9	17	0
CfsSubsetEval	GreedyStepwise	6	4	2	0	26	9	17	0
CorrelationAttributeEval	Ranker	45	4	15	26	90	9	34	47
GainRatioAttributeEval	Ranker	45	4	15	26	90	9	34	47
InfoGainAttributeEval	Ranker	45	4	15	26	90	9	34	47
OneRAttributedEval	Ranker	45	4	15	26	90	9	34	47
ReliefFAttributeEval	Ranker	45	4	15	26	90	9	34	47
SymmetricalUncertAttributeEval	Ranker	45	4	15	26	90	9	34	47

This window size and sampling rate were the best in our static parameters analysis investigated in Section 4.2.1.

In the second phase, we feed the data for simple and complex activities to the feature selection algorithm using the Weka tool (Witten et al., 2016). Most of the algorithms' search method was based on the ranking of all features between a certain range. For example, OneRAttributedEval ranks features between approximately 17 and 45 and GainRatioAttributeEval ranks between 0.02 and 0.28. However, certain feature selection algorithms use other search methods, such as BestFirst which makes a binary decision. More clearly, the algorithm gives 0 to the not selected and 1 to the selected feature. These different score ranges of algorithms complicate understanding which features are more successful. To avoid this difficulty, we normalized data to 0-1 range, using min-max normalization. For each feature, such as accelerometer mean x or gyroscope rms m, we sum up the normalized scores given by each algorithm, which gives us a total score for each feature. For the simple activities the total scores of features were between a range of 0.19 and 7.96 and for the complex, scores were between 0.11 and 7.56. We created three separate classes by dividing these intervals into three parts and named these classes as high impact, medium impact, and low impact features. The exact names of the algorithms in Weka, the number of features selected by the algorithms, and details of how many of these selected features belong to which group are presented in Table 4.11.

In Tables 4.12 and 4.13, the names of the features in each class, and the score ranges of the classes are presented. For simple activities, we obtain 26, 15 and 4 features extracted from the accelerometer in low, medium and high class, respectively. For the complex activities, we obtain 23, 17 and 5 features extracted from the accelerometer and 24, 17 and 4 in low, medium and the high class, respectively. Compared with low and medium class, high class has fewer features. In the rest of the analysis, instead of using F1 or F3 based on the type of activity, we use high impact features based on the type of activity. As we mentioned in Table 1, we calculate 16 features in F1 for simple activities, 32 for complex and in the same way, 24 features in F3 for simple and 48 for the complex activities. Thanks to this feature selection, if the state of activity is simple, we calculate only 4 features (MIN-X, MAX-X, RMS-X and RMS-Z of accelerometer), and if it is complex the number of features is 9 (STD-X, MEDIAN-X, MAX-X, RANGE-X, RMS-X of accelerometer and STD-Z, MEDIAN-M, MIN-Z, MAX-Z of gyroscope). Updated dynamic activity recognition algorithm is presented in Figure 4.8. In the text, we use the notation FSel for the features used in this section.

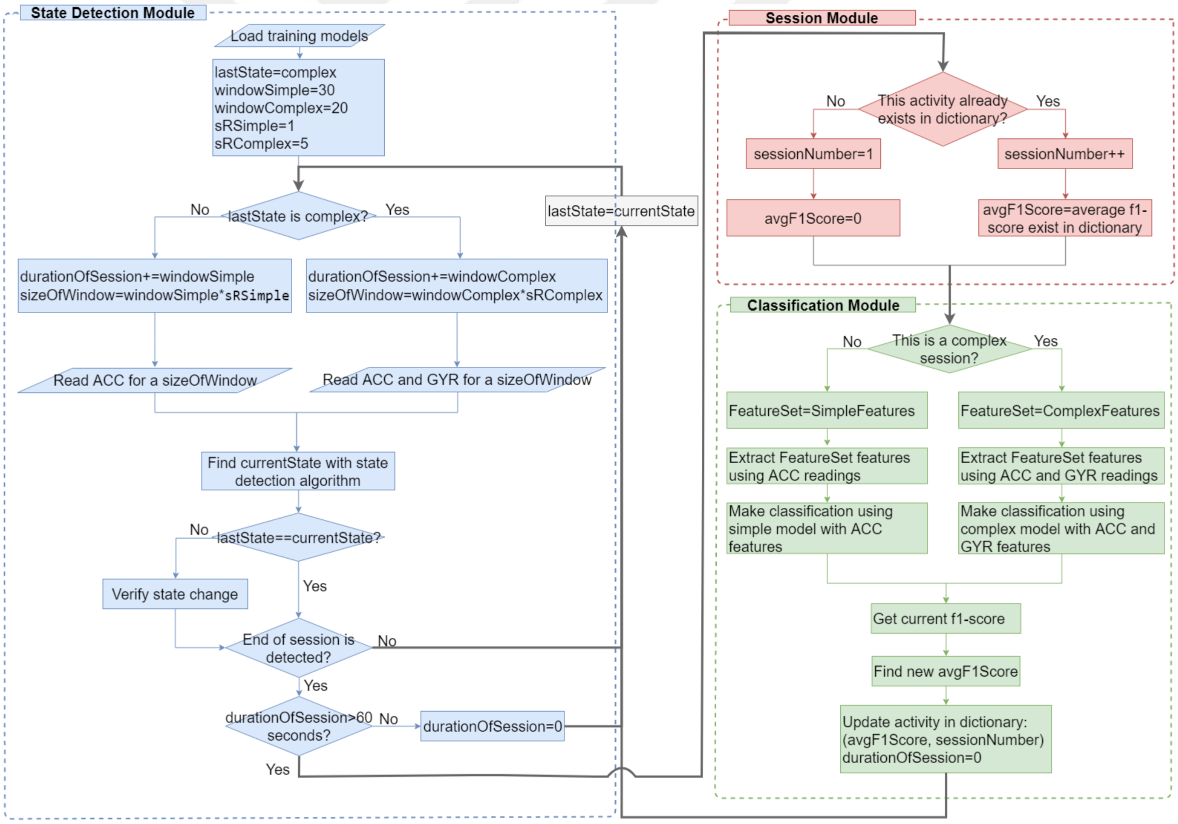


Figure 4.8: New dynamic activity recognition algorithm

We perform static, semi-dynamic and dynamic analysis using these new features. First of all, we create training models using new features as we mentioned in Section 4.1.5. After that, in the static parameter analysis as we did not already detect the state of activity we merge simple and complex high impact features and use them together for all activities. After this analysis, we use only selected high impact simple features for

Table 4.12: Selected low, medium and high features for simple activities

Simple Activities		
LOW	**MEDIUM**	**HIGH**
ACCMEAN-Y, M	ACCMEAN-X, Z	ACCMIN-X
ACCSTD-X, Y, Z	ACCSTD-M	ACCMAX-X
ACCMEDIAN-Y, M	ACCMEDIAN-X, Z	ACCRMS-X, Z
ACCSKEWNESS-X, Y, Z, M	ACCMIN-Z, M	
ACCKURTOSIS-X, Y, Z, M	ACCMAX-Z, M	
ACCMIN-Y	ACCRANGE-X, M	
ACCMAX-Y	ACCINTEGRATION-X, Z	
ACCRANGE-Y, Z	ACCABSDIFF-X, Z	
ACCINTEGRATION-Y, M		
ACCCORRELATION-XYZ		
ACCRMS-Y, M		
ACCABSDIFF-Y, M		

simple activities and selected complex features for complex activities.

Average recognition performances for complex and simple activities for three phases of analysis are presented in Table 4.14. In static analysis, we present ACCGYR results for complex and ACC results for simple activities. We observe that the performance increases depending on the increase in sampling rate and also window size with some exceptions. Best performance is observed using 25 Hz with 30 seconds and the worst 1 Hz with 5 seconds which are 83% and 70% respectively. For the simple activities, the average performance is always over the 94%. As we mentioned previously, in semi-dynamic and dynamic analysis, we use only selected 4 features for simple activities and 9 for complex activities. In the semi-dynamic analysis, we observe an increase in performance from 5% to 12%. This is due to the positive effects of both using the simple and complex features for the needed state and making the sensor selection accordingly. In the dynamic analysis, we use the already selected sampling rates and window sizes explained in Section 4.2.2. On average for complex activities, we observe a performance of 88%, which corresponds to an increase of 1% compared to semi-dynamic and 6% compared to static analysis results for 5 Hz with 20 seconds.

In Table 4.15, we present the performance results of three analysis for each complex and simple activity. As we use 5 Hz – 20 seconds for complex activities and 1 Hz – 30 seconds for simple, in the table we present only related sampling rate and window sizes. We present ACCGYR results for complex and ACC results for simple activities, if we take

Table 4.13: Selected low, medium and high features for complex activities

Complex Activities		
LOW	**MEDIUM**	**HIGH**
ACCMEAN-Z, M	ACCMEAN-X, Y	ACCSTD-X
ACCSTD-Y, Z	ACCSTD-M	ACCMEDIAN-X
ACCMEDIAN-M	ACCMEDIAN-Y, Z	ACCMAX-X
ACCSKEWNESS-X, Z, M	ACCSKEWNESS-Y	ACCRANGE-X
ACCKURTOSIS-X, Y, Z	ACCKURTOSIS-M	ACCRMS-X
ACCMIN-Y	ACCMIN-X, Z, M	GYRSTD-Z
ACCMAX-Y, Z, M	ACCRANGE-M	GYRMEDIAN-M
ACCRANGE-Y, Z	ACCINTEGRATION-X, Y	GYRMIN-Z
ACCINTEGRATION-Z, M	ACCCORRELATION-XYZ	GYRMAX-Z
ACCRMS-Z	ACCRMS-Y, M	
ACCABSDIFF-X, Z, M	ACCABSDIFF-Y	
GYRMEAN-X, Y, Z	GYRMEAN-M	
GYRSTD-X, Y	GYRSTD-M	
GYRSKEWNESS-X, Y, Z, M	GYRMEDIAN-X, Y, Z	
GYRMIN-X, Y	GYRMIN-M	
GYRKURTOSIS-M	GYRKURTOSIS-X, Y, Z	
GYRMAX-Y	GYRMAX-X, M	
GYRRANGE-X, Y	GYRRANGE-Z, M	
GYRINTEGRATION-X, Y, Z	GYRINTEGRATION-M	
GYRRMS-X, Y	GYRRMS-Z, M	
GYRABSDIFF-X, Y, Z	GYRABSDIFF-M	
GYRCORRELATION-XYZ		

the average performance values of the ACC and ACCGYR, especially the performance for complex activities are much lower. However, we observe that we have an important increase with the dynamization of sensor and feature set compared to static parameters. We observe an increase of 14%, 9%, 6%, 5%, 4% and 2% for smokeSD, smokeGroup, drinkSD, smokeWalk, drinkST and smokeST, respectively. But we observe a decrease of 1% for eat. Comparing semi-dynamic and dynamic analysis, there is not a significant difference. For certain activities, such as drinkSD and smokeGroup the performance is slightly increased and for certain activities vice versa. For simple activities, we observe very high performance in all three analyses.

Table 4.14: Performance comparison using high impact features on average of complex and simple activities at different sampling rates and window sizes

		Complex Activities			Simple Activities		
		Static Analysis	**Semi-Dynamic Analysis**	**Dynamic Anaysis**	**Static Analysis**	**Semi-Dynamic Analysis**	**Dynamic Anaysis**
50 Hz	30 sec	0.82	0.88		0.96	0.69	
	20 sec	0.82	0.87		0.97	0.83	
	10 sec	0.79	0.86		0.97	0.88	
	5 sec	0.76	0.86		0.97	0.99	
25 Hz	30 sec	0.83	0.87		0.96	0.69	
	20 sec	0.81	0.87		0.97	0.83	
	10 sec	0.80	0.86		0.97	0.88	
	5 sec	0.76	0.86		0.97	0.90	
10 Hz	30 sec	0.82	0.87		0.96	0.69	
	20 sec	0.81	0.88		0.97	0.96	
	10 sec	0.78	0.86		0.97	0.97	
	5 sec	0.76	0.86		0.97	0.98	
5 Hz	30 sec	0.81	0.87		0.96	0.85	
	20 sec	0.82	0.87	0.88	0.97	0.86	
	10 sec	0.78	0.86		0.97	0.95	
	5 sec	0.75	0.85		0.96	0.97	
2 Hz	30 sec	0.81	0.86		0.96	0.97	
	20 sec	0.81	0.86		0.97	0.99	
	10 sec	0.77	0.84		0.96	0.97	
	5 sec	0.72	0.83		0.95	0.98	
1 Hz	30 sec	0.79	0.85		0.96	0.97	0.99
	20 sec	0.79	0.85		0.96	1.00	
	10 sec	0.75	0.83		0.95	0.98	
	5 sec	0.70	0.82		0.94	0.98	

4.2.5 Resource Consumption Analysis

The aim of this section is to make a resource consumption analysis for better understanding the effect of dynamic parameter selection algorithm in comparison to using static parameters and semi-dynamic parameters. For this analysis, we use three common metrics used in resource consumption which are CPU time (in seconds), memory (in MB) and energy (in mWh) usages with static, semi-dynamic and dynamic parameters. In order to investigate these analyses, we use the tools provided by python to help us compute the performance metrics. List of tools are as follows :

— cProfile : It is a profiler that analyzes the performance of the code. The tool can be used from the command line. Thus, it will ensure execution of the original Python code directly. In the output, the tool gives five information which is called as *ncalls*, *tottime*, *cumtime*, *percall* and $filename$: *lineno* (Lanaro, 2017). In our analysis, we are particularly interested in *tottime*. It signifies the total time spend on CPU to execute the entire code and it is given in seconds.

— psutil : It is a Python library that takes the information of a running process and system utilization. This library's has a method called $memory\_info()$ which

Table 4.15: Performance comparison using high impact features for complex and simple activities

		Static Analysis	Semi-Dynamic Analysis	Dynamic Anaysis
5 Hz - 20 sec	smokeSD	0.75	0.89	0.88
	smokeST	0.82	0.84	0.84
	eat	0.93	0.92	0.91
	drinkSD	0.76	0.82	0.85
	drinkST	0.80	0.84	0.84
	smokeWalk	0.91	0.96	0.96
	smokeGroup	0.75	0.84	0.85
1 Hz - 30 sec	sit	0.97	0.99	1.00
	stand	0.95	0.99	0.98
	walk	0.96	0.94	1.00

gives us the memory information about the running process in bytes (). Then, we convert bytes to megabytes for the ease of reading.

— wmi : It is a module that provides much information about computer system, such as the capabilities and management of the battery.

In our experiments, we need to learn battery used during the execution of the code. For this, just before and after the execution, we get the remaining capacity. Then, we take the difference between these two values as the battery consumed. We report power usage in miliwatt hour (mWh) which is a measure of how much energy used to execute the code.

During this analysis, we did not run any other program and kept the computer idle until the end of execution to better observe the resource consumed only for the code. Similarly, we test with the scenario data. Additionally, to make the results more accurate and comparable, we did not take measurements while the battery is charging, or the battery level is low. We repeat each three anaysis both using F1-F3 features and using FSel features.

4.2.5.1 Static Parameters

In this section, resource consumption results of the static parameter analysis explained in Section 4.2.1 are presented. In Figure 4.9, the CPU consumption of sensors, feature sets, sampling rates and window sizes are shown. Using ACC and ACCGYR sensors, we

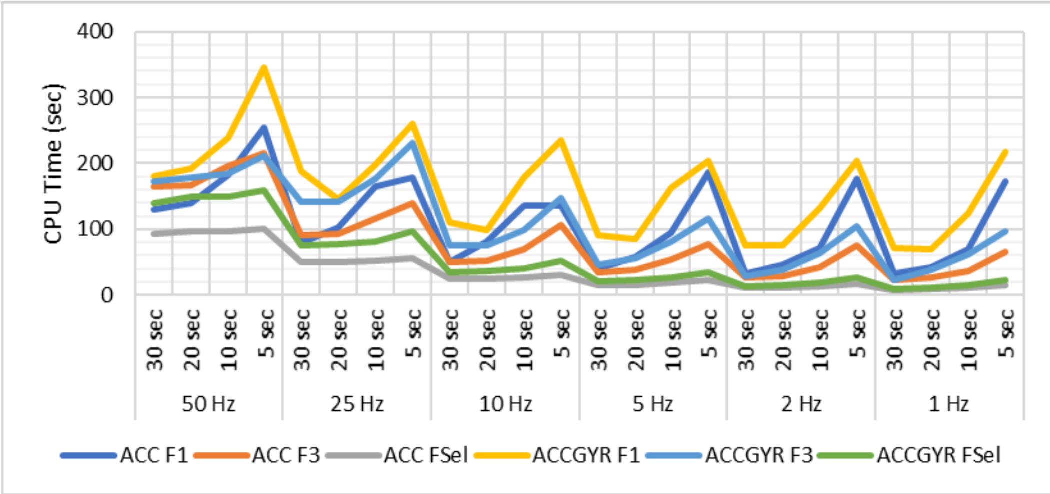


Figure 4.9: Comparison of all sensors and feature sets in terms of CPU usage at different sampling rates and window sizes

observe that when sampling rate decreases, CPU time is increasing and when window size decreases, it increases. Compared to others, using ACCGYR with F1 consumes the maximum energy for each sampling rate and window size pair. Using F1 consumes more energy compared to F3. As expected, using ACCGYR consumes more energy compared to using only ACC. For example, using 25 Hz with 10 seconds, when ACC data is read and F1 features are extracted, the code takes 164 seconds and when F3 features are extracted, it takes 115 seconds. And if ACCGYR is used, the time spends are 197 and 176 seconds with F1 and F3, respectively. We observe an important improvement using FSel features. Using ACC sensor with FSel features, the time spent is very low compared to the others. Even with ACCGYR, results are very good compared to using only ACC with F1 or F3. For example, using 10 Hz with 10 seconds, times consumed on CPU is 179, 136, 99, 69, 39 and 26 seconds for ACCGYR F1, ACC F1, ACCGYR F3, ACC F3, ACCGYR FSel and ACC FSel lines, respectively. We see clearly that in our case, the impact of the feature set has a higher effect on the CPU time than using ACCGYR instead of only ACC. Additionally, if we use FSel instead of F1, we observe an increase of 110 seconds both for ACC and ACCGYR.

In Figure 4.10, memory usages in term of megabytes (MB) are presented for the different combination of parameters. We see clearly that only the choice of sampling rate and sensor have an important effect on the memory consumption as expected, because they are directly related to the amount of data sampled. Choice of feature set and window size are more preliminary as there is a computational operation. Additionally, we observe that using two sensors instead of one doubles the memory consumption and with the decrease in sampling rate, the memory consumption decreases almost linearly.

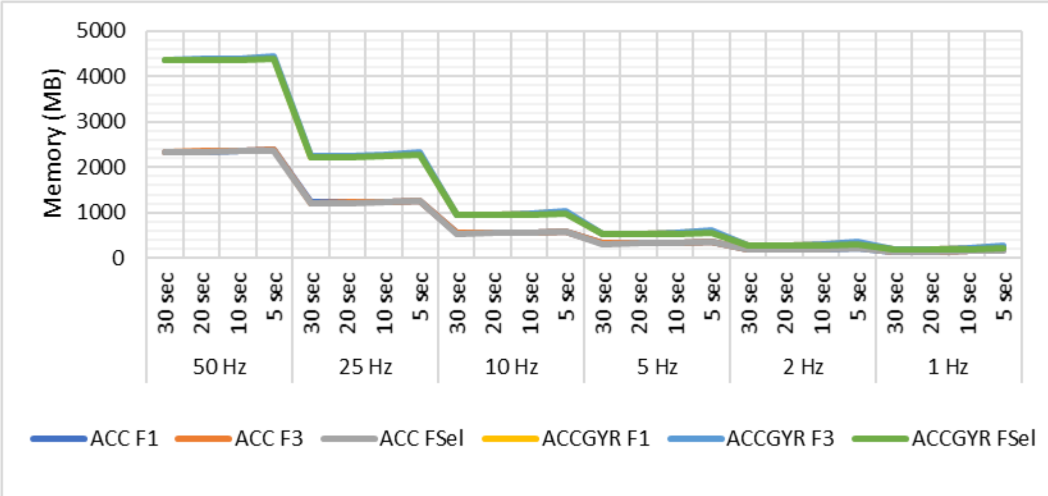


Figure 4.10: Comparison of all sensors and feature sets in terms of memory usage at different sampling rates and window sizes

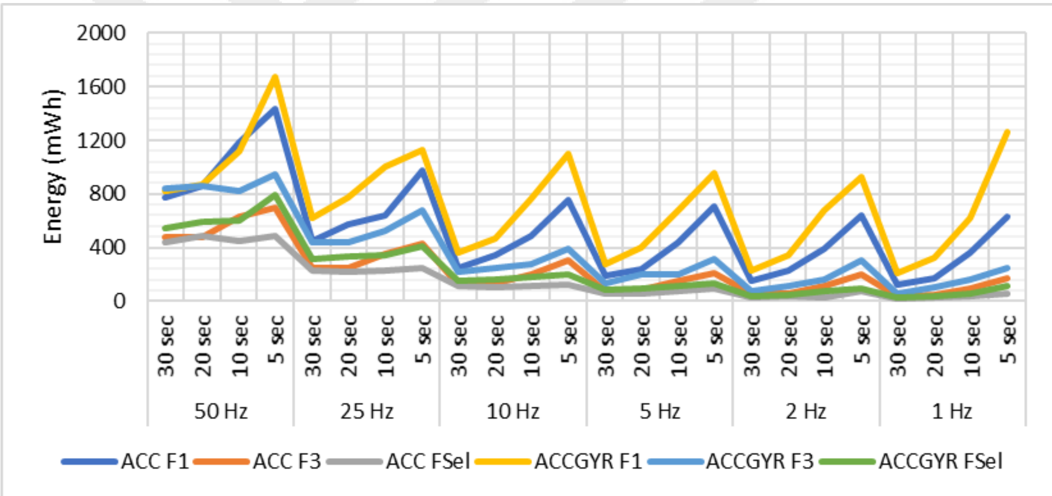


Figure 4.11: Comparison of all sensors and feature sets in terms of energy usage at different sampling rates and window sizes

In Figure 4.11, the energy consumption results of the static analysis are presented. This metric has a particular importance on battery usage. We observe similar patterns as we observed in the CPU usage. Decreasing the sampling rate decreases the energy consumption, while increasing the window size at a specific sampling rate increases it. Using the highest sampling rate with the lowest window size (50 Hz – 5 seconds) consumes maximum energy which is 1670 mWh and the lowest is observed using 1 Hz – 30 seconds which is 20 mWh. To guarantee a certain limit of performance, it is possible to reduce energy consumption excessively. If we look at the effect of sensors and feature sets, we observe that feature sets bring more change than the using ACC or ACCGYR in terms of energy.

4.2.5.2 Semi-Dynamic Parameters

In this section, we present resource consumption results when the feature sets and sensors are chosen dynamically. For each metric, firstly, we present the results obtained using the dynamic choice of F1-F3 depending on activity and then, results obtained with the features chosen in Section 4.2.4 depending on the type of activity.

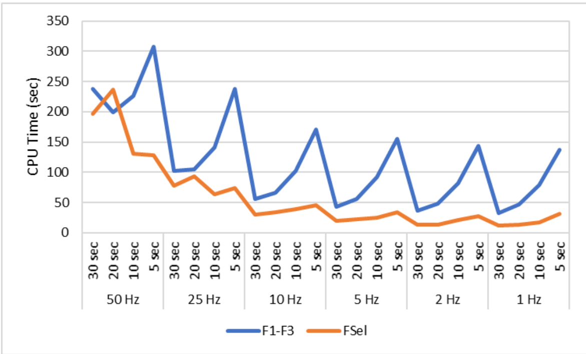
In Figure 4.12(a), (b) and (c), results of CPU, memory and energy consumption both using dynamic choice of F1-F3 and FSel features are presented. In Figure 4.12(a), we observe that for three metrics, the consumption decreases rapidly between the sampling rate of 50 and 10 Hz, then it decreases slowly using dynamic selection between F1 and F3 features. FSel line also follows a similar pattern. As expected, CPU consumption with using F1-F3 features are higher than FSel features due to the effect of the reduction in number of features on the computational cost.

In Figure 4.12(b), in semi-dynamic analysis, also the changes in memory consumption depends on changes in the sampling rate, but not exactly on the window size and using different features have not an impact on the memory consumed, similar to the results of static analysis. Therefore, F1-F3 and FSel lines overlap.

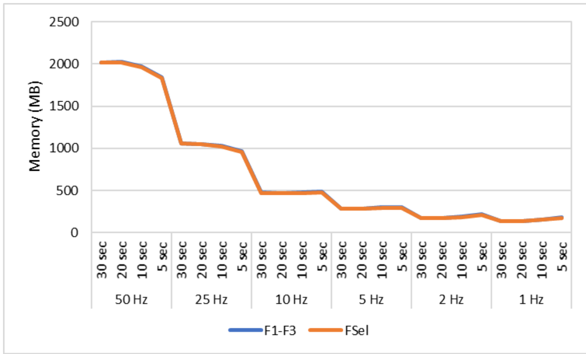
In Figure 4.12(c), similar patterns are observed also in the CPU time. For each sampling rate, using a window size of 5 seconds requires more computation and more time, so the CPU usage and energy consumption are higher than the other window sizes. The maximum CPU and energy are consumed using 50 Hz – 5 seconds, which are 308 seconds and 1400 mWh and the minimum values are using 1 Hz 30 seconds, which are 32 seconds and 120 mWh, respectively. We notice that changing window size and sampling rate can have an order of magnitude effect on the energy consumption.

4.2.5.3 Dynamic Parameters

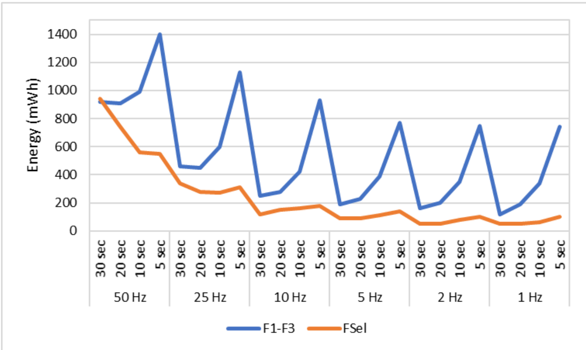
For the dynamic parameter analysis, resource consumption results both using FSel features and F1-F3 features are presented in Figure 4.13. We observe that the dynamic parameter selection consumes only 57 and 20 seconds of CPU time, using F1-F3 and FSel features, respectively. Memory consumption is around 230 MB for both. Energy consumption is 200 mWh using F1-F3 features and 70 mWh using FSel features. Comparing to the static parameter selection results, using only accelerometer consumes less



(a)



(b)



(c)

Figure 4.12: Resource consumption using dynamic feature sets and sensors : (a) CPU ; (b) memory ; (c) energy consumption

CPU time than ACCGYR or dynamic sensor selection, however as we discussed in Section 4.2.1, the recognition rate of complex activities is lower with only using the accelerometer. When the sensor is dynamically selected, the memory usage is below the memory usage of ACCGYR since the gyroscope is turned on/off dynamically. Both for F1-F3 and FSel features, the memory requirement of the dynamic parameter selection algorithm is 20 times less than the memory requirements of ACCGYR when 50 Hz sampling rate is used.

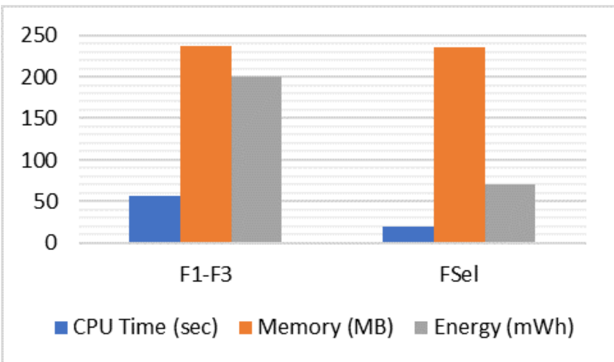
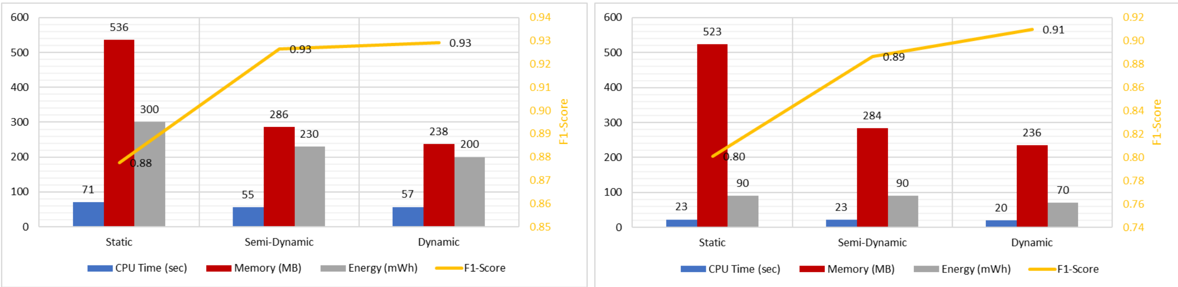


Figure 4.13: Resource consumption results of dynamic parameter selection using F1-F3 and FSel feature sets.

In Figure 4.14, we present a summary of the three methods in terms of both resource



(a) (b)

Figure 4.14: Comparison of static, semi-dynamic and dynamic analysis in term of cpu, memory, energy and f1-score : (a) using F1-F3 features ; (b) using FSel features

consumption and recogition accuracy, where the static case uses both accelerometer and gyroscope, 5 Hz, 20 seconds windows, and semi-dynamic setting uses 5 Hz and 20 seconds where sensor and features are selected dynamically. These results show that, before the feature selection (Figure 4.14(a)), the dynamic parameter selection algorithm achieves 33% less energy consumption, 55% less memory size and 20% less CPU time compared with using static parameters. Using FSel features(Figure 4.14(b)), for static and dynamic parameter selection, the consumption of CPU and energy is decreasing so much without compromising the recognition performance. If we use the dynamic parameter selection with FSel features, instead of using static with F1-F3 features, we reduce the energy consumption from 71 to 20 mWh, memory consumption from 536 to 236 MB and CPU consumption from 300 to 70 sec.

5 CONCLUSION

In this thesis, we studied the recognition of smoking activity with the motion sensors available on smartwatches using a challenging dataset which includes different variations of smoking as well as activities including similar hand gestures to smoking.

Firstly, in Chapter 3, we followed a detailed analysis : with a focus on using different classifiers, different and comprehensive set of features, different window sizes and different window overlap ratios. Besides, we investigated the impact of height in the training phase on smoking recognition performance. The results show that smoking activities can be recognized with simple features, such as median, std, min, max, range, mean. Compared to smaller window sizes used in the recognition of simpler activities, larger window sizes, such as 30 seconds, perform better. When we compare the performance of different classifiers, when efficient features are used, their performances are similar. Smoking in a group is more difficult to recognize compared to other variants of smoking due to different patterns exhibited by different participants. When we analyze the impact of height on smoking recognition, it does not have a significant effect when all activities are considered. However, it does have an effect on smoking while standing activity, particularly considering participants with a different height than others.

Secondly, in Chapter 4, we analyzed a dynamic smoking recognition system where we use the findings from Chapter 3. We investigated the resource consumption of different parameter sets on the wearable devices besides the recognition performance. For this purpose, we investigated context-aware activity recognition and propose a dynamic parameter selection algorithm which activates different sensors, sampling rates and features on demand according to the type of the activity. We defined two types of activities, namely simple and complex and presented a state detection algorithm to identify states together with a state correction method for the use of the dynamic parameter selection algorithm. We utilized a dataset that contains 10 different activities from 11 participants for training the activity recognition model. To choose features, firstly, we used two best feature set of Chapter 4 then we applied several feature selection algorithms to determine features for simple and complex activities. By doing this, we reduced the number of features and find the most efficient ones. Using a test scenario, we evaluated the performance of the algorithm both in terms of recognition rate and resource consumption and compare with using static and semi-dynamic parameters. Results show that, compared to using static parameters, the dynamic parameter selection algorithm achieves 2 to 13% better recognition rate depending on the type of activity. It consumes 23% less energy and 20% less CPU time with F1-F3 features. Using FSel features in the dynamic parameter selection algorithm, we achieve a decrease of 65% on the CPU and energy compared to the dynamic parameter selection algorithm using F1-F3 features and a decrease of 72% on the CPU and 77% on the energy compared to static parameter selection using F1-F3. It may be argued that we tested the performance with a specific set of activities using a specific dataset. However, the algorithm is not designed to be specific to these activities and new parameters can be added on demand.

We are currently implementing the dynamic activity recognition algorithm on smartwatches and will explore the resource consumption on these devices, considering different types of activities. In this thesis, we focus on smoking recognition, however, this algorithm can be extended to recognize other activities.

REFERENCES

Ali, A. A., Hossain, S. M., Hovsepian, K., Rahman, M. M., Plarre, K. and Kumar, S. (2012). mpuff : automated detection of cigarette smoking puffs from respiration measurements, *Proceedings of the 11th international conference on Information Processing in Sensor Networks*, ACM, pp. 269–280.

Bao, L. and Intille, S. S. (2004). Activity recognition from user-annotated acceleration data, *International Conference on Pervasive Computing*, Springer, pp. 1–17.

Cole, C. A., Anshari, D., Lambert, V., Thrasher, J. F. and Valafar, H. (2017). Detecting smoking events using accelerometer data collected via smartwatch technology : Validation study, *JMIR mHealth and uHealth* **5**(12).

Cornacchia, M., Ozcan, K., Zheng, Y. and Velipasalar, S. (2017). A survey on activity detection and classification using wearable sensors, *IEEE Sensors Journal* **17**(2) : 386–403.

Coskun, D., Incel, O. D. and Ozgovde, A. (2015). Phone position/placement detection using accelerometer : Impact on activity recognition, *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pp. 1–6.

Figo, D., Diniz, P. C., Ferreira, D. R. and Cardoso, J. M. (2010). Preprocessing techniques for context recognition from accelerometer data, *Personal and Ubiquitous Computing* **14**(7) : 645–662.

Fiscus, J. G., Ajot, J., Radde, N. and Laprun, C. (2006). Multiple dimension levenshtein edit distance calculations for evaluating automatic speech recognition systems during simultaneous speech, *The International Conference on language Resources and Evaluation (LERC).*

Incel, Ö. D. (2015). Analysis of movement, orientation and rotation-based sensing for phone placement recognition, *Sensors* **15**(10) : 25474–25506.
URL: *https ://doi.org/10.3390/s151025474*

Khalifa, S., Hassan, M. and Seneviratne, A. (2014). Feature selection for floor-changing activity recognition in multi-floor pedestrian navigation, *Mobile Computing and Ubiquitous Networking (ICMU), 2014 Seventh International Conference on*, IEEE, pp. 1–6.

Konak, S., Turan, F., Shoaib, M. and Incel, Ö. D. (2016). Feature engineering for activity recognition from wrist-worn motion sensors., *PECCS*, pp. 76–84.

Lanaro, G. (2017). *Python High Performance*, Packt Publishing Ltd.

Lara, O. D. and Labrador, M. A. (2013). A survey on human activity recognition using wearable sensors, *Communications Surveys & Tutorials, IEEE* **15**(3) : 1192–1209.

Lockhart, J. W., Pulickal, T. and Weiss, G. M. (2012). Applications of mobile activity recognition, *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, ACM, pp. 1054–1058.

Lu, H., Yang, J., Liu, Z., Lane, N. D., Choudhury, T. and Campbell, A. T. (2010). The jigsaw continuous sensing engine for mobile phone applications, *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, ACM, New York, NY, USA, pp. 71–84.
URL: *http ://doi.acm.org/10.1145/1869983.1869992*

Mukhopadhyay, S. C. (2015). Wearable sensors for human activity monitoring : A review, *IEEE Sensors Journal* **15**(3) : 1321–1330.

Nambu, M. (2007). Body surface mounted biomedical monitoring system using bluetooth, *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1824–1825.

Ortiz, J. L. R. (2015). Human activity dataset generation, *Smartphone-Based Human Activity Recognition*, Springer, pp. 59–78.

Parate, A., Chiu, M.-C., Chadowitz, C., Ganesan, D. and Kalogerakis, E. (2014). Risq : Recognizing smoking gestures with inertial sensors on a wristband, *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, ACM, pp. 149–161.

Preece, S., Goulermas, J., Kenney, L. P. J. and Howard, D. (2009). A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data, *IEEE Transactions on Biomedical Engineering* **56**(3) : 871–879.

Rachuri, K. K., Mascolo, C., Musolesi, M. and Rentfrow, P. J. (2011). Sociablesense : exploring the trade-offs of adaptive sampling and computation offloading for social sensing, *Proceedings of the 17th annual international conference on Mobile computing and networking*, ACM, pp. 73–84.

Rawassizadeh, R., Tomitsch, M., Nourizadeh, M., Momeni, E., Peery, A., Ulanova, L. and Pazzani, M. (2015). Energy-efficient integration of continuous context sensing and prediction into smartwatches, *Sensors* **15**(9) : 22616–22645.

Rezaie, H. and Ghassemian, M. (2017). An adaptive algorithm to improve energy efficiency in wearable activity recognition systems, *IEEE Sensors Journal* **17**(16) : 5315–5323.

Rodgers, J. and Nicewander, W. (1988). Thirteen ways to look at the correlation coefficient, *The American Statistician* **42**(1) : 59–66.

Ryder, J., Longstaff, B., Reddy, S. and Estrin, D. (2009). Ambulation : A tool for monitoring mobility patterns over time using mobile phones, *Computational Science and Engineering, 2009. CSE'09. International Conference on*, Vol. 4, IEEE, pp. 927–931.

Saeedi, R., Schimert, B. and Ghasemzadeh, H. (2014). Cost-sensitive feature selection for on-body sensor localization, *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing : Adjunct Publication*, ACM, pp. 833–842.

Saleheen, N., Ali, A. A., Hossain, S. M., Sarker, H., Chatterjee, S., Marlin, B., Ertin, E., al'Absi, M. and Kumar, S. (2015). puffmarker : a multi-sensor approach for pinpointing the timing of first lapse in smoking cessation, *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ACM, pp. 999–1010.

Scholl, P. M. and Van Laerhoven, K. (2012). A feasibility study of wrist-worn accelerometer based detection of smoking habits, *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, IEEE, pp. 886–891.

Sen, S., Rachuri, K. K., Mukherji, A. and Misra, A. (2016). Did you take a break today ? detecting playing foosball using your smartwatch, *2016 IEEE Internatio-*

nal Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), IEEE, pp. 1–6.

Seneviratne, S., Hu, Y., Nguyen, T., Lan, G., Khalifa, S., Thilakarathna, K., Hassan, M. and Seneviratne, A. (2017). A survey of wearable devices and challenges, *IEEE Communications Surveys & Tutorials* **19**(4) : 2573–2620.

Shoaib, M. (2017). *Sitting is the new smoking : online complex human activity recognition with smartphones and wearables*, PhD thesis, University of Twente, Netherlands. CTIT Ph.D. thesis series no. 17-436.

Shoaib, M., Bosch, S., Incel, O. D., Scholten, H. and Havinga, P. J. (2014). Fusion of smartphone motion sensors for physical activity recognition, *Sensors* **14**(6) : 10146–10176.

Shoaib, M., Bosch, S., Incel, O. D., Scholten, H. and Havinga, P. J. (2015). A survey of online activity recognition using mobile phones, *Sensors* **15**(1) : 2059–2085.

Shoaib, M., Bosch, S., Incel, O. D., Scholten, H. and Havinga, P. J. (2016). Complex human activity recognition using smartphone and wrist-worn motion sensors, *Sensors* **16**(4) : 426.

Shoaib, M., Bosch, S., Scholten, H., Havinga, P. J. and Incel, O. D. (2015). Towards detection of bad habits by fusing smartphone and smartwatch sensors, *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*, IEEE, pp. 591–596.

Shoaib, M., Incel, O. D., Scholten, H. and Havinga, P. (2017). Resource consumption analysis of online activity recognition on mobile phones and smartwatches, *36th IEEE – International Performance Computing and Communications Conference*.

Shoaib, M., Incel, O. D., Scholten, H. and Havinga, P. (2018). Smokesense : Online activity recognition framework on smartwatches, *International Conference on Mobile Computing, Applications, and Services*, Springer, pp. 106–124.

Shoaib, M., Scholten, H., Havinga, P. J. and Incel, O. D. (2016). A hierarchical lazy smoking detection algorithm using smartwatch sensors, *e-Health Networking, Applications and Services (Healthcom), 2016 IEEE 18th International Conference on*, IEEE, pp. 1–6.

Skinner, A., Stone, C., Doughty, H. and Munafo, M. (2018). Stopwatch : The preliminary evaluation of a smartwatch-based system for passive detection of cigarette smoking, *Nicotine and Tobacco Research* .

Tang, Q., Vidrine, D. J., Crowder, E. and Intille, S. S. (2014). Automated detection of puffing and smoking with wrist accelerometers, *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 80–87.

Wang, S., Yang, J., Chen, N., Chen, X. and Zhang, Q. (2005). Human activity recognition with user-free accelerometers in the sensor networks, *2005 International Conference on Neural Networks and Brain*, Vol. 2, pp. 1212–1217.

Wang, Y., Lin, J., Annavaram, M., Jacobson, Q. A., Hong, J., Krishnamachari, B. and Sadeh, N. (2009). A framework of energy efficient mobile sensing for automatic user state recognition, *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ACM, pp. 179–192.

Witten, I. H., Frank, E., Hall, M. A. and Pal, C. J. (2016). *Data Mining : Practical machine learning tools and techniques*, Morgan Kaufmann.

Yan, Z., Subbaraju, V., Chakraborty, D., Misra, A. and Aberer, K. (2012). Energy-efficient continuous activity recognition on mobile phones : An activity-adaptive approach, *Wearable Computers (ISWC), 2012 16th International Symposium on*, Ieee, pp. 17–24.

Yang, J.-Y., Wang, J.-S. and Chen, Y.-P. (2008). Using acceleration measurements for activity recognition : An effective learning algorithm for constructing neural classifiers, *Pattern recognition letters* **29**(16) : 2213–2220.

Zhang, M. and Sawchuk, A. A. (2011). A feature selection-based framework for human activity recognition using wearable multimodal sensors, *Proceedings of the 6th International Conference on Body Area Networks*, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 92–98.

Zheng, L., Wu, D., Ruan, X., Weng, S., Peng, A., Tang, B., Lu, H., Shi, H. and Zheng, H. (2017). A novel energy-efficient approach for human activity recognition, *Sensors* **17**(9) : 2064.

A F1-SCORE RESULTS OF SPECIFIC PARAMETERS

In Tables A.1, A.2, A.3, A.4, A.5, A.6, A.7, each cell shows the f1-score performance of a specific feature-set with a specific classifier in a specific case, using a specific sensor.

Table A.1: Scenario 1: F1-scores considering all cases

		Case 1			Case 2			Case 3			Case 4		
		SVM	RF	MLP	SVM	RF	MLP	SVM	RF	MLP	SVM	RF	MLP
ACC	F1	0.7	0.74	0.74	0.69	0.75	0.75	0.72	0.75	0.75	0.71	0.76	0.76
	F2	0.74	0.74	0.73	0.75	0.75	0.76	0.75	0.75	0.75	0.76	0.75	0.75
	F3	0.73	0.74	0.74	0.74	0.74	0.74	0.74	0.75	0.75	0.75	0.76	0.76
	F4	0.26	0.69	0.5	0.22	0.7	0.48	0.24	0.7	0.5	0.22	0.7	0.51
	F5	0.25	0.69	0.47	0.22	0.7	0.46	0.25	0.69	0.45	0.22	0.7	0.46
	F6	0.32	0.54	0.48	0.29	0.55	0.47	0.35	0.54	0.49	0.3	0.55	0.45
	F7	0.4	0.47	0.36	0.4	0.48	0.29	0.42	0.47	0.39	0.41	0.47	0.34
LACC	F1	0.6	0.65	0.65	0.6	0.66	0.65	0.63	0.67	0.67	0.62	0.68	0.67
	F2	0.57	0.7	0.67	0.59	0.71	0.69	0.6	0.71	0.7	0.6	0.72	0.7
	F3	0.59	0.71	0.67	0.58	0.71	0.67	0.61	0.71	0.69	0.6	0.73	0.68
	F4	0.3	0.69	0.5	0.27	0.7	0.48	0.32	0.7	0.54	0.28	0.71	0.51
	F5	0.27	0.64	0.47	0.24	0.65	0.44	0.27	0.65	0.47	0.24	0.66	0.45
	F6	0.46	0.52	0.51	0.45	0.52	0.52	0.49	0.52	0.5	0.46	0.53	0.52
	F7	0.38	0.43	0.22	0.4	0.45	0.17	0.38	0.44	0.31	0.41	0.45	0.22
GYR	F1	0.66	0.71	0.69	0.66	0.72	0.7	0.69	0.72	0.7	0.68	0.74	0.71
	F2	0.58	0.72	0.68	0.61	0.73	0.69	0.6	0.73	0.7	0.62	0.74	0.71
	F3	0.58	0.72	0.7	0.61	0.73	0.71	0.6	0.73	0.72	0.62	0.75	0.71
	F4	0.34	0.68	0.54	0.32	0.7	0.43	0.35	0.69	0.57	0.33	0.71	0.49
	F5	0.29	0.68	0.4	0.25	0.69	0.42	0.29	0.7	0.44	0.26	0.7	0.4
	F6	0.51	0.56	0.52	0.5	0.57	0.55	0.52	0.56	0.53	0.53	0.57	0.54
	F7	0.35	0.42	0.25	0.37	0.41	0.24	0.35	0.42	0.27	0.37	0.42	0.25
PR	F1	0.39	0.63	0.6	0.36	0.64	0.6	0.39	0.64	0.6	0.37	0.64	0.6
	F2	0.28	0.64	0.61	0.27	0.65	0.6	0.3	0.65	0.63	0.27	0.65	0.62
	F3	0.27	0.65	0.61	0.26	0.66	0.6	0.29	0.66	0.62	0.27	0.66	0.61
	F4	0.26	0.6	0.24	0.22	0.6	0.22	0.27	0.6	0.27	0.22	0.61	0.23
	F5	0.24	0.6	0.22	0.21	0.61	0.24	0.25	0.6	0.26	0.21	0.61	0.24
	F6	0.36	0.37	0.11	0.34	0.35	0.08	0.37	0.36	0.1	0.36	0.35	0.1
	F7	0.17	0.29	0.12	0.17	0.29	0.09	0.18	0.29	0.14	0.17	0.28	0.11

Table A.2: Scenario 1: F1-scores of each feature set using ACC and Case 4

		F1	F2	F3	F4	F5	F6	F7
SVM	smokeSD	0.60	0.63	0.62	0.24	0.24	0.15	0.42
	smokeST	0.55	0.65	0.63	0.00	0.00	0.26	0.22
	eat	0.84	0.90	0.89	0.00	0.12	0.10	0.63
	drinkSD	0.60	0.66	0.65	0.00	0.00	0.04	0.09
	drinkST	0.48	0.62	0.61	0.00	0.00	0.07	0.21
	sit	0.93	0.99	0.99	0.64	0.56	0.90	0.66
	stand	1.00	1.00	1.00	0.99	0.99	1.00	0.97
	smokeWalk	0.86	0.75	0.74	0.00	0.00	0.00	0.08
	walk	0.88	0.93	0.91	0.00	0.00	0.08	0.60
	smokeGroup	0.46	0.49	0.48	0.00	0.00	0.04	0.12
RF	smokeSD	0.61	0.63	0.64	0.54	0.60	0.43	0.40
	smokeST	0.65	0.64	0.65	0.56	0.50	0.40	0.22
	eat	0.87	0.87	0.88	0.84	0.79	0.64	0.61
	drinkSD	0.62	0.58	0.61	0.54	0.52	0.29	0.20
	drinkST	0.59	0.61	0.59	0.56	0.56	0.40	0.29
	sit	0.99	0.99	0.99	0.99	0.98	0.94	0.82
	stand	1.00	1.00	1.00	1.00	1.00	0.99	0.99
	smokeWalk	0.89	0.83	0.85	0.72	0.85	0.24	0.34
	walk	0.97	0.97	0.97	0.96	0.90	0.79	0.57
	smokeGroup	0.55	0.52	0.54	0.41	0.45	0.30	0.24
MLP	smokeSD	0.65	0.63	0.65	0.37	0.29	0.39	0.34
	smokeST	0.63	0.62	0.66	0.41	0.28	0.34	0.17
	eat	0.88	0.91	0.89	0.57	0.49	0.37	0.58
	drinkSD	0.60	0.61	0.61	0.25	0.21	0.14	0.10
	drinkST	0.57	0.57	0.59	0.40	0.42	0.34	0.16
	sit	0.96	0.97	0.97	0.91	0.88	0.87	0.53
	stand	0.99	0.99	0.99	0.92	0.90	0.94	0.82
	smokeWalk	0.91	0.83	0.85	0.16	0.22	0.15	0.08
	walk	0.97	0.96	0.96	0.72	0.58	0.63	0.43
	smokeGroup	0.60	0.49	0.55	0.26	0.22	0.26	0.01

Table A.3: Scenario 1: F1-score of each sensor and fusion of sensors using Case 4

		ACC	LACC	GYR	PR	ACCGYR	LACCGYR
SVM	F1	0.71	0.62	0.68	0.37	0.69	0.65
	F2	0.76	0.6	0.62	0.27	0.78	0.68
	F3	0.75	0.6	0.62	0.27	0.78	0.68
	F4	0.22	0.28	0.33	0.22	0.22	0.26
	F5	0.22	0.24	0.26	0.21	0.22	0.24
	F6	0.3	0.46	0.53	0.36	0.27	0.47
	F7	0.41	0.41	0.37	0.17	0.54	0.51
RF	F1	0.76	0.68	0.74	0.64	0.8	0.77
	F2	0.75	0.72	0.74	0.65	0.8	0.77
	F3	0.76	0.73	0.75	0.66	0.81	0.78
	F4	0.7	0.71	0.71	0.61	0.78	0.77
	F5	0.7	0.66	0.7	0.61	0.78	0.75
	F6	0.55	0.53	0.57	0.35	0.71	0.68
	F7	0.47	0.45	0.42	0.28	0.6	0.55
MLP	F1	0.76	0.67	0.71	0.6	0.8	0.75
	F2	0.75	0.7	0.71	0.62	0.8	0.77
	F3	0.76	0.68	0.71	0.61	0.81	0.76
	F4	0.51	0.51	0.49	0.63	0.6	0.56
	F5	0.46	0.45	0.4	0.24	0.51	0.52
	F6	0.45	0.52	0.54	0.1	0.29	0.66
	F7	0.34	0.22	0.25	0.11	0.44	0.33

Table A.4: Scenario 2: F1-scores of each sensor and fusion of sensors using F3 and Case 4.

	ACC	LACC	GYR	PR	ACCGYR	LACCGYR
SVM	0.80	0.61	0.64	0.34	0.83	0.70
RF	0.81	0.74	0.78	0.73	0.85	0.82
MLP	0.80	0.71	0.75	0.67	0.84	0.80

Table A.5: F1-scores of in-group analysis using F3, RF and Case 4

		smokeSD	smokeST	eat	drinkSD	drinkST	sit	stand
P1	ACC	0.95	0.89	0.90	0.73	0.85	0.96	1.00
	LACC	0.88	0.82	0.81	0.70	0.65	0.89	1.00
	GYR	0.84	0.78	0.90	0.67	0.61	0.82	1.00
	ACCGYR	0.97	0.93	0.96	0.81	0.84	0.96	1.00
	LACCGYR	0.87	0.80	0.92	0.74	0.66	0.87	0.99
P6	ACC	0.98	0.96	0.92	0.90	0.87	0.95	0.99
	LACC	0.92	0.86	0.86	0.94	0.92	0.87	0.99
	GYR	0.82	0.69	0.88	0.84	0.84	0.87	0.99
	ACCGYR	0.94	0.90	0.96	0.95	0.94	0.95	0.99
	LACCGYR	0.86	0.83	0.89	0.88	0.89	0.93	1.00
P7	ACC	0.94	0.91	0.88	0.82	0.81	0.93	1.00
	LACC	0.89	0.73	0.71	0.76	0.74	0.89	0.98
	GYR	0.83	0.79	0.92	0.72	0.70	0.78	0.95
	ACCGYR	0.92	0.86	0.96	0.84	0.82	0.94	0.99
	LACCGYR	0.84	0.83	0.96	0.87	0.82	0.92	0.99
P9	ACC	0.99	0.87	0.98	0.96	0.89	0.96	1.00
	LACC	0.92	0.85	0.85	0.65	0.71	0.80	1.00
	GYR	0.88	0.86	0.97	0.79	0.71	0.71	0.98
	ACCGYR	0.99	0.93	0.97	0.94	0.92	0.97	1.00
	LACCGYR	0.90	0.85	0.95	0.79	0.71	0.74	1.00
P10	ACC	0.94	0.85	0.96	0.84	0.80	0.94	1.00
	LACC	0.90	0.83	0.91	0.70	0.77	0.87	1.00
	GYR	0.93	0.87	0.97	0.87	0.78	0.83	0.97
	ACCGYR	0.95	0.91	0.96	0.88	0.84	0.93	1.00
	LACCGYR	0.93	0.90	0.97	0.87	0.79	0.86	0.99
P11	ACC	0.93	0.95	0.92	0.84	0.90	0.94	1.00
	LACC	0.80	0.62	0.77	0.62	0.69	0.70	0.99
	GYR	0.84	0.72	0.76	0.55	0.63	0.69	0.97
	ACCGYR	0.93	0.90	0.89	0.80	0.81	0.91	0.99
	LACCGYR	0.82	0.72	0.85	0.68	0.68	0.76	1.00

Table A.6: F1-scores of out-group analysis using F3, RF and Case 4

		smokeSD	smokeST	eat	drinkSD	drinkST	sit	stand
P1	ACC	0.97	0.90	0.88	0.72	0.83	0.94	1.00
	LACC	0.91	0.84	0.79	0.70	0.63	0.87	1.00
	GYR	0.84	0.77	0.93	0.68	0.62	0.85	0.99
	ACCGYR	0.96	0.90	0.96	0.83	0.85	0.96	1.00
	LACCGYR	0.88	0.82	0.93	0.77	0.70	0.86	1.00
P6	ACC	0.97	0.93	0.94	0.94	0.93	0.95	0.99
	LACC	0.88	0.80	0.89	0.90	0.92	0.88	0.99
	GYR	0.84	0.75	0.88	0.86	0.88	0.83	0.97
	ACCGYR	0.96	0.95	0.92	0.93	0.92	0.95	0.99
	LACCGYR	0.87	0.84	0.93	0.92	0.91	0.95	1.00
P7	ACC	0.95	0.87	0.88	0.83	0.82	0.95	1.00
	LACC	0.89	0.91	0.75	0.77	0.74	0.92	0.99
	GYR	0.79	0.75	0.92	0.78	0.71	0.77	0.95
	ACCGYR	0.95	0.91	0.95	0.82	0.81	0.94	1.00
	LACCGYR	0.90	0.88	0.94	0.88	0.83	0.91	0.99
P9	ACC	0.99	0.88	0.96	0.95	0.89	0.96	1.00
	LACC	0.92	0.84	0.91	0.74	0.73	0.72	1.00
	GYR	0.92	0.86	0.96	0.80	0.75	0.72	0.97
	ACCGYR	0.99	0.92	0.97	0.93	0.90	0.96	0.99
	LACCGYR	0.89	0.82	0.96	0.79	0.75	0.78	1.00
P10	ACC	0.96	0.87	0.95	0.83	0.82	0.93	0.99
	LACC	0.90	0.85	0.91	0.71	0.78	0.87	0.99
	GYR	0.94	0.89	0.96	0.85	0.76	0.84	0.99
	ACCGYR	0.95	0.94	0.96	0.89	0.88	0.95	0.99
	LACCGYR	0.90	0.90	0.97	0.87	0.83	0.88	1.00
P11	ACC	0.94	0.93	0.89	0.90	0.86	0.94	0.99
	LACC	0.81	0.65	0.83	0.70	0.69	0.75	1.00
	GYR	0.78	0.72	0.75	0.58	0.72	0.68	0.97
	ACCGYR	0.93	0.90	0.90	0.82	0.85	0.94	1.00
	LACCGYR	0.77	0.67	0.83	0.66	0.66	0.69	1.00

Table A.7: F1-scores of per participant analysis using F3, RF and Case 4

		smokeSD	smokeST	eat	drinkSD	drinkST	sit	stand
P1	ACC	0.97	0.90	0.93	0.75	0.84	0.96	0.99
	LACC	0.91	0.84	0.80	0.71	0.66	0.87	1.00
	GYR	0.83	0.75	0.94	0.75	0.64	0.80	0.99
	ACCGYR	0.96	0.93	0.96	0.83	0.87	0.96	1.00
	LACCGYR	0.88	0.81	0.92	0.75	0.68	0.89	1.00
P2	ACC	0.97	0.92	0.94	0.75	0.78	0.97	1.00
	LACC	0.94	0.88	0.96	0.86	0.84	0.83	0.99
	GYR	0.94	0.84	0.97	0.84	0.69	0.70	1.00
	ACCGYR	0.97	0.92	0.95	0.86	0.90	0.97	1.00
	LACCGYR	0.97	0.92	0.97	0.87	0.82	0.83	1.00
P3	ACC	0.92	0.79	0.92	0.86	0.66	0.93	1.00
	LACC	0.83	0.80	0.81	0.65	0.54	0.93	0.99
	GYR	0.84	0.76	0.96	0.71	0.63	0.87	0.99
	ACCGYR	0.94	0.86	0.98	0.88	0.79	0.94	1.00
	LACCGYR	0.86	0.81	0.94	0.75	0.70	0.92	0.99
P4	ACC	0.82	0.74	0.96	0.81	0.87	0.98	1.00
	LACC	0.84	0.76	0.89	0.67	0.59	0.80	1.00
	GYR	0.77	0.73	0.94	0.78	0.60	0.71	1.00
	ACCGYR	0.88	0.85	0.96	0.86	0.86	0.97	1.00
	LACCGYR	0.84	0.81	0.95	0.81	0.65	0.79	1.00
P5	ACC	0.91	0.86	0.92	0.73	0.70	0.91	1.00
	LACC	0.78	0.77	0.79	0.69	0.80	0.95	1.00
	GYR	0.83	0.76	0.95	0.72	0.71	0.89	0.97
	ACCGYR	0.92	0.86	0.94	0.71	0.72	0.91	1.00
	LACCGYR	0.89	0.81	0.92	0.73	0.78	0.94	1.00
P6	ACC	0.97	0.92	0.92	0.92	0.91	0.94	0.99
	LACC	0.92	0.81	0.89	0.92	0.92	0.86	1.00
	GYR	0.81	0.69	0.90	0.88	0.87	0.87	0.99
	ACCGYR	0.97	0.93	0.94	0.94	0.93	0.97	0.99
	LACCGYR	0.90	0.83	0.93	0.94	0.94	0.92	1.00
P7	ACC	0.95	0.90	0.87	0.82	0.79	0.94	1.00
	LACC	0.90	0.70	0.73	0.79	0.77	0.86	0.99
	GYR	0.76	0.69	0.95	0.75	0.68	0.75	0.95
	ACCGYR	0.95	0.91	0.98	0.84	0.85	0.97	1.00
	LACCGYR	0.90	0.85	0.95	0.89	0.78	0.87	0.99
P8	ACC	0.99	0.96	0.98	0.95	0.90	0.91	0.99
	LACC	0.83	0.80	0.96	0.85	0.80	0.89	1.00
	GYR	0.91	0.68	0.92	0.73	0.80	0.81	0.97
	ACCGYR	0.99	0.94	0.99	0.97	0.93	0.93	1.00
	LACCGYR	0.88	0.86	0.97	0.84	0.90	0.90	0.99
P9	ACC	0.99	0.83	0.95	0.95	0.85	0.96	1.00
	LACC	0.91	0.87	0.89	0.68	0.69	0.80	0.99
	GYR	0.90	0.88	0.97	0.83	0.70	0.70	0.99
	ACCGYR	0.99	0.94	0.98	0.94	0.92	0.96	0.99
	LACCGYR	0.89	0.87	0.96	0.77	0.71	0.79	1.00
P10	ACC	0.94	0.89	0.97	0.83	0.84	0.93	0.99
	LACC	0.89	0.85	0.91	0.72	0.75	0.87	1.00
	GYR	0.92	0.86	0.98	0.86	0.80	0.84	0.97
	ACCGYR	0.96	0.92	0.98	0.89	0.87	0.93	0.99
	LACCGYR	0.92	0.80	0.99	0.88	0.81	0.86	0.99
P11	ACC	0.92	0.93	0.92	0.80	0.87	0.93	0.99
	LACC	0.76	0.58	0.77	0.67	0.64	0.66	1.00
	GYR	0.81	0.70	0.82	0.61	0.68	0.72	0.96
	ACCGYR	0.93	0.91	0.88	0.75	0.89	0.93	0.99
	LACCGYR	0.82	0.72	0.83	0.62	0.65	0.73	0.99

BIOGRAPHICAL SKETCH

Sümeyye Ağaç was born in Diyarbakır in 1994. She received her BSc degree in Computer Engineering from Galatasaray University in 2016. She is continuing her graduate study with thesis master program in Computer Engineering at Galatasaray University. Her research interest includes human activity recognition with smartphones and wearables using machine learning.

PUBLICATIONS

- Konak, S., Turan, F., Shoaib, M. and Incel, Ö. D. (2016). Feature engineering for activity recognition from wrist-worn motion sensors., *PECCS*, pp. 76–84.
- Agac, S., Shoaib, M. and Incel, Ö. D. (2018). Fusion of smartphone and smartwatch sensors for smoking recognition., *ICENTE'18*, pp. 158-162.