SmokeSense: Online Activity Recognition Framework on Smartwatches

Muhammad Shoaib[1], Ozlem Durmaz Incel[2(✉)], Hans Scholten[1], and Paul Havinga[1]

[1] Pervasive Systems Research Group, University of Twente, Enschede, The Netherlands
{m.shoaib,hans.scholten,p.j.m.havinga}@utwente.nl
[2] Department of Computer Engineering, Galatasaray University, Istanbul, Turkey
odincel@gsu.edu.tr

Abstract. In most cases, human activity recognition (AR) with smartphones and smartwatches has been done offline due to the limited resources of these devices. Initially, these devices were used for logging sensor data which was later on processed in machine learning tools on a desktop or laptop. However, current versions of these devices are more capable of running an activity recognition system. Therefore, in this paper, we present SmokeSense, an online activity recognition (AR) framework developed for both smartphones and smartwatches on Android platform. This framework can log data from various sensors and can run an AR process in real-time locally on these devices. Any classifier or feature can easily be added on demand. As a case study, we evaluate the recognition performance of smoking with four classifiers, four features, and two sensors on a smartwatch. The activity set includes variants of smoking such as smoking while sitting, standing, walking, biking, as well as other similar activities. Our analysis shows that, similar recognition performance can be achieved in an online recognition as in an offline analysis, even if no training data is available for some smoking postures. We also propose a smoking session detection algorithm to count the number of cigarettes smoked and evaluate its performance.

1 Introduction

Human activity recognition using smartphones and smartwatches has enabled many novel, context-aware applications in different domains, especially healthcare [1]. Such devices were initially considered as resource-limited [2] such as the battery capacity, for running an activity recognition system over an extended period. It is also a challenging task to implement and evaluate different recognition systems on these devices. Due to these reasons, most of the research on human activity recognition using these devices is done offline (not on the device) in machine learning tools, such as WEKA or scikit-learn [3–8]. In recent years, smartphones and some smartwatch models have become capable of running such recognition systems. They have become more powerful in terms of

K. Murao et al. (Eds.): MobiCASE 2018, LNICST 240, pp. 106–124, 2018.
https://doi.org/10.1007/978-3-319-90740-6_7

available resources, such as CPU, memory, and battery, so there has been a shift towards online activity recognition. In online recognition, the human activity recognition process is run on the device (smartphone or a smartwatch) in real-time. Offline analysis can be acceptable for applications where online recognition is not required [9]. For example, if the aim is to follow the sleeping patterns of a user, sensor data can be uploaded to a server and processed offline where real-time tracking is not necessary. However, if we aim to recognize smoking sessions of a user, online processing of the data on the watch or phone may be required. Online activity recognition on a local device (smartphone or smartwatch) does not depend on the internet connection all the time and also avoids the privacy concerns if users do not want their data to be uploaded to a server or cloud. It is important to note that "online activity recognition on smartphones" should not be confused with "online machine learning models". "Online machine learning models" are able to adapt themselves according to new data points, unlike offline or batch learning models [10]. We use the "online" term in a different way, for the practical implementation of activity recognition systems on mobile phones. These implemented systems can use either an online or a batch learning model.

There are a number of studies where activity recognition has been implemented on smartphones for real-time processing [11]. However, there are very few recent studies where such activity systems have been implemented and evaluated for their recognition performance as well as resource consumption on a smartwatch [12–14]. In most of these studies, especially on the smartwatch, it is very difficult to compare various aspects of an activity recognition system due to their different experimental setups. For example, they have used different classifiers, datasets, data features, platforms, performance metrics, validation methods, number of users, and implementations. Additionally, performance results of an online analysis and offline analysis can be different since conditions are usually idealized in an offline setting: there is no missing data, there is large size of training data, etc.

In this paper, we present SmokeSense, an online activity recognition framework for both smartphones and smartwatches. Our aim is to address the mentioned issues, validate the offline analysis results from our previous studies [15,16] and to compare various aspects of an activity recognition system in a similar environment and similar experimental setup. Based on this framework, we implemented a modular Android application for these devices where various classifiers, data features, sampling rates, and sensors can be evaluated for their recognition performance in an online maner. In this specific study, we evaluated the recognition performance of four commonly used classifiers on both a smartphone for recognizing seven physical activities[1] and on a smarwatch for recognizing the smoking activity. Smoking is one of the reasons for premature death and its reliable detection can enable tracking of smoking behavior [18,19]. Additionally, it can be used as an automated self-reporting tool in smoking cessation programs [18,19]. While performing these evaluations for smoking detection, we

[1] However, due to page size limitation we only present the results of smoking recognition. Interested readers can refer to [17].

consider two sensors, an accelerometer, a gyroscope and their combinations, considering both subject-specific training models and generic training models. Our results show that, similar recognition performance is achieved as in our offline analysis [16]. The use of the gyroscope besides an accelerometer has improved the recognition performance. Although subject-specific training models are observed to exhibit better performance, generic models also perform well at an acceptable level. Moreover, we learned that smoking while sitting is difficult to recognize compared to other postures due to its similarity to drinking while sitting. We also showed that smoking can be well recognized in different postures, such as while biking, from which no training data was available. Moreover, using a hierarchical classification approach for smoothing the results of windowing segments increased the recognition rates. We also evaluated the impact of various aspects (different classifiers, features, sampling rates, window sizes, activities, devices, sensors) of an activity recognition system on its resource consumption (CPU, memory, and battery) in [20]. We summarize our contributions as follows:

– From the system point of view, we developed a framework for online human activity recognition using smartphones and smartwatches. Compared to online recognition systems existing in the literature [], this framework can be used to detect any activity using smart watches and smart phones and it is an adaptive framework: any classifier, sensor, feature set can be added on demand.
– From the health care point of view, we proposed a rule-based smoking session detection algorithm where the aim is to detect the number of cigarettes smoked. This algorithm can be used as an automated self-reporting tool in smoking cessation programs.
– From the methodological point of view, we evaluated the recognition performance of the smoking activity on a smartwatch in real-time, considering different postures, such as while sitting, standing and in a group conversation. Compared to other studies in the literature where the analysis is done offline, all the analysis was performed for real-time recognition and even in a posture, smoking while biking, from which no training data was available.

The rest of the paper is organized as follows: In Sect. 2, we present the related studies and compare those with our study. We describe our framework for online activity recognition in Sect. 3. In Sect. 4, we present our performance evaluations. Finally, we present the conclusions, and future work in Sect. 5.

2 Related Work

Human activity recognition using smartphone sensors has been studied extensively for the last few years [1,11,21]. As mentioned, most of the work in this area is performed offline such that collected data is analyzed in machine learning tools, such as WEKA, Scikit-learn, R, and MATLAB. Activity recognition using smartwatch sensors is still relatively new, compared to smartphones. Most of the work using smartwatch sensors is also being done offline [22–26].

Recently, researchers have been moving towards online activity recognition in order to verify the offline results and to analyze the resource consumption of machine learning algorithms on mobile phones and other wearable devices such as smartwatches [27]. In a recent survey paper [11], we reviewed the studies that implement activity recognition systems on mobile phones. However, in these studies, only a few classifiers are tested, different platforms, datasets and experimental setups are used. Online activity recognition and in-device learning [28] on wearable device sensors is still relatively a new topic [12–14]. In these studies, battery consumption of sensor logging and online activity recognition process on smarwatches has been investigated. However, similar to the studies on mobile phones, different setups and use of different methods make it difficult to compare the results.

In this paper, our aim is to propose a conceptual framework and build an adaptive online activity recognition system that can run on smartphones and smarwatches where different classifiers, training methods, features, sensors can be added on demand. We aim to provide a testing platform also for other researchers to verify the results obtained in offline analysis.

Smoking recognition is one of the case studies that we performed for testing the framework, besides physical activity recognition. Most of the work on smoking recognition is done offline such that collected data is analyzed using machine learning tool on a desktop machine with no implementation on a smartwatch or smartphone. For example, studies in [18,29–31] follow such an offline approach for smoking recognition whereas in [32], the authors implemented the smoking recognition pipeline on the smartphone. However, to the best of our knowledge, none of these studies have evaluated online smoking recognition on a smartwatch. We previously published our results of offline analysis for smoking recognition using smartwatches and collected one of the largest smoking activity dataset [16]. In Table 1 in [16], we compare these smoking recognition studies in detail with our offline work and we also discuss the gaps in these existing studies. Unlike our work, most of these studies focused on an offline analysis, person-dependent evaluation, limited smoking postures, and combining other sensors with a smartwatch. Different than these studies, we perform an online smoking recognition analysis based on the proposed SmokeSense framework.

3 SmokeSense: A Framework for Online Activity Recognition

The activity recognition process can be divided into various components such as sensing, feature extraction, training, and classification. The generic activity recognition process is described in Fig. 1. It starts with sensing (step 1), if needed the collected sensor data can be preprocessed (step 2), features are extracted (step 3), a training model is created (step 4a) and in the final stage, the trained classifiers are used to classify new data instances into different activities (step 4b). This process can be divided into offline and online categories. In online activity recognition, the classification is done on the device (phone or wearable

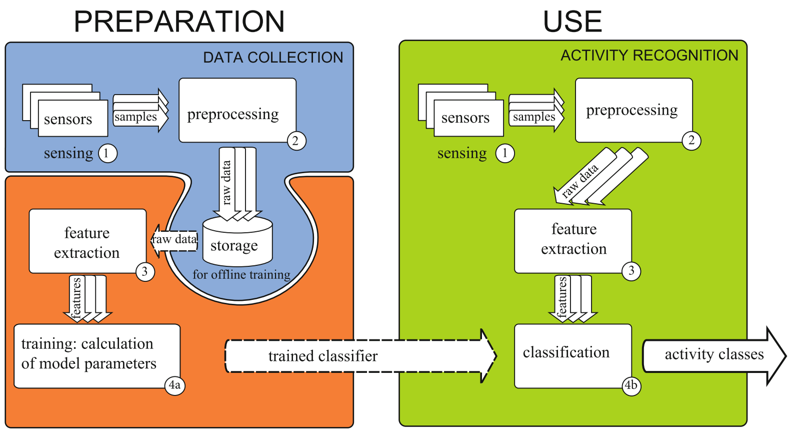


Fig. 1. Activity recognition process (The numbers (1, 2, 3, 4a, 4b shows the order of the steps involved in this process)

device). However, the training can still be done in two ways: online (on the device) and offline. The training can be very time and resource consuming, that is why it is usually done offline. We have opted for offline training in this work. We use offline training and then port these trained models to the mobile phone and the smartwatch.

3.1 Framework

We propose a conceptual framework for online activity recognition which integrates both smart phones and smart watches. These two devices are commonly used in combination and a lot of people already use/wear them. They are already connected with each other through Bluetooth. Therefore, we want to utilize sensing information from both these devices at different levels for richer context or activity recognition. For example, the smartwatch can send raw sensor information or extracted features to the smartphone where it can be combined with smartphone sensors for richer contextual information. However, we can also run the complete activity recognition process on the smartwatch and send the information about the recognized activities to the smartphone. In this case, models on smartwatch and smartphone can be trained to detect different activities, such as activities involving hand gestures can be detected using a smartwatch whereas the others can be identified using the smartphone. For example, the smartwatch can detect that a user is smoking or eating whereas the phone can detect the user's posture at that specific time such that if the user is doing this specific activity while sitting or standing or walking. We should note that, other wearable devices, such as smart glasses, that can run AR process can be integrated to the framework.

The framework consists of three main components: Activity recognition (AR) process on a smartphone, AR process on a smartwatch, and a machine learning tool (WEKA) for training models as shown in Fig. 2. In this framework, first a machine learning model is trained offline in WEKA. Then it is ported to the smartwatch and a smartphone in a serialized form. Afterwards, the AR process reads sensor data in real-time, preprocesses it, extract features over a segmented window (30 s) and then uses the trained model to predict the activity class of this segmented window. The smartwatch can also run the complete activity recognition process as well as sending raw sensor information or extracted features to the smartphone at the same time. For this purpose, we define the following modes of operation for this framework on how information can be processed and exchanged between the smartphone and the smartwatch.

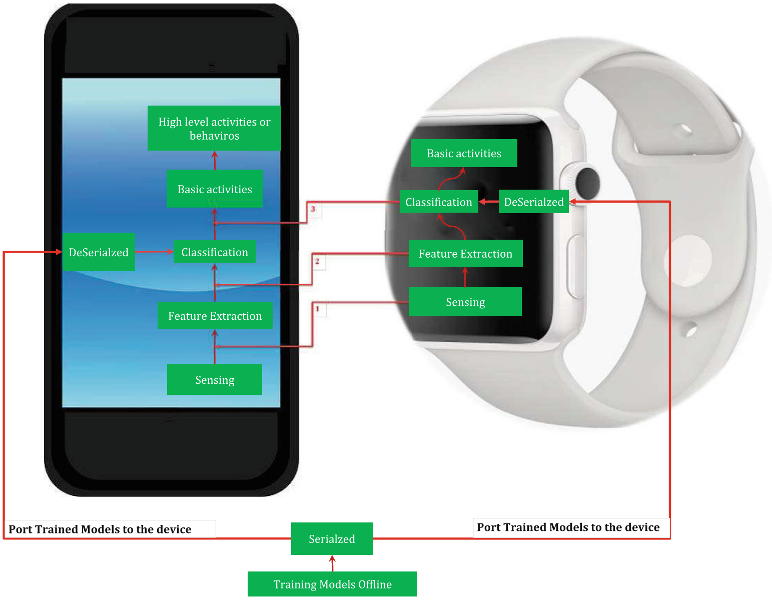


Fig. 2. Online activity recognition framework

- On phone only: In this mode, the whole AR process (sensing, feature extraction, and classification) runs on the phone and utilizes the phone sensors only. We do not utilize smartwatch in this mode. For example, if a user does not have a smartwatch, only this mode of operation can be used.
- On Watch only: In this mode, the complete AR process runs on the watch and utilizes the watch sensors only. It can send the predicted labels for the activities to the smartphone via Bluetooth for storing and displaying purpose. However, it can also store and display these on the watch.
- On both devices: In this mode, both smartphone and smartwatch are being used in the AR process. The smartphone always runs the complete AR process. However, we divide the AR process on the smartwatch into further modes of operation and these are shown with numbers (1, 2, and 3) in Fig. 2.

– Mode 1: In this case, only sensing is performed on the smartwatch such that smartwatch sends raw sensor information to the smartphone in real-time where they are combined with smartphone sensors before processing them for feature extraction and classification. Using this mode means more resource consumption because we will be using Bluetooth very frequently and will be sending a lot of data with each transfer. For example, we use 50 samples per second for reading sensor data.
– Mode 2: In this case, features are extracted from smartwatch sensors and these features are sent to the smartphone where they are combined with the features extracted from smartphone sensors before they enter into the classification phase. This mode should use relatively low resources because we will be sending only features after the window size is reached. However, if the number of features is very high then it may consume more resources as well.
– Mode 3: In this case, complete AR process is carried out on the smartwatch where only smartwatch sensors are used whereas in parallel smartphone runs its own AR process using its own sensors. We only send the recognized activities' labels to the smartphone. However, these labels can be stored and displayed on the watch too.
– Mode Hybrid: In this case, mode 3 can be combined with mode 1 or mode 2 such that smartwatch runs its own AR process and it also sends sensor information to the smartphone where it can be combined with the smartphone sensors for better activity recognition.

The decision to choose a specific mode of operation depends on many factors such as activities that need to be recognized, resources availability, and application requirements. For the training component, WEKA tool can be used where machine learning models are trained offline and then ported to these devices. After training these models, they are serialized in WEKA and stored in the relevant Android Apps where they are de-serialized at the time of their use. *Serialization* is the process of saving an object in a persistent form such as on a hard-disk as a byte stream. *Deserialization* is the reverse process where such serialized objects are converted back to its original form. This process is described in WEKA documentation [33].

We have implemented our Android app in a modular way based on the conceptual framework shown in Fig. 2, where the training is done offline in WEKA. For sensing, we have implemented the use of an accelerometer, a linear acceleration sensor, and a gyroscope. However, other sensors can easily be added to the implementation as per demand. For feature extraction, we have implemented min, max, mean and standard deviation. Other features can be added if needed. For classification part, the trained models from WEKA are used to predict the current window of sensor data and maps it to the relevant activity. These trained models can be placed in the asset or other folders in our app and they are ready to use. These three modules or parts are implemented as an Android service which runs in the background and does not need any user interaction. The app can be used in three modes: On phone only, on watch only, on both devices (only with option 3). Though we have implemented mode/option 3, the other options can be added a later stage. The current implementations are enough for the evaluation of the resources consumption and recognition performance of our use cases.

In our specific use case, we run smoking recognition process on the smartwatch. For training purpose, we used a dataset described in detail in [16]. We added an additional data of around 5 h to the data set in order to improve the null or other class, so smoking should not be confused with other activities. This additional data comes from a participant who took part in the evaluation of this study. He performed various activities such as drinking, eating, walking, biking, washing dishes, cooking, taking part in conversations, inactive (sitting, standing, laying in bed etc.) and others. We used WEKA tool for training the models because it is a java based toolkit which provides an easy to use serialization of these trained models. These serialized trained models can easily be ported to Android where we de-serialize them at run time to use them for real-time predictions.

We trained four classifiers in WEKA 3.7: decision tree (DT), support vector machine (SMO), random forest (RF), multilayer perceptron (MLP). We use these classifiers in their default settings except few changes. These changes were made for random forest. For random forest, we used two variants: one with 9 number of trees and other with 99. The default setting can easily be found in WEKA documentation [34]. For this specific study, we did not use any parameter optimization algorithms. These four classifiers were chosen because they have been previously shown to have reasonable recognition performance for recognizing various human activities [11]. Moreover, they were also chosen in a way that they represent various types of algorithms.

3.2 Smoking Session Detection Algorithm

It is important to detect the number of cigarettes smoked or smoking sessions. There are two ways to do so. One way is to sum the total number of smoking segments, convert that into total time spent while smoking and then divide it by the average smoking time per cigarette. This method works well if the underlying classification provides reasonable recognition performance for smoking. For example, it gives a higher number of smoking sessions if the false positive rate is high. Hence, it is important to correct as many as possible misclassified segments before we apply this method. The drawback with this method is that we cannot know the timing of each smoking session. If the underlying classification is poor, then it will lead to too many false positives. In the second method, smoking sessions can be calculated using a simple rule-based algorithm where it takes into account the neighboring segments for a specific amount of time and uses a threshold to decide if it is a smoking session or not. For this purpose, we developed a simple rule-based algorithm to detect these sessions. This algorithm is described in Fig. 3. We continuously monitor the prediction results of the classification function. We trigger a smoking session calculator as soon as we see a smoking segment. We also start counting the number of detected smoking segments after this trigger. We stop the session calculator in two cases:

– When an already defined session window size is reached. In our case, we use a window size of ten minutes as shown by *SWCthreshold* in Fig. 3.
– When there are no smoking segments for at least a specified amount of time. In our case, it was set to be at least 2 min as shown by the *OCthreshold* in Fig. 3. It helps in removing the random hand gestures classified as smoking.

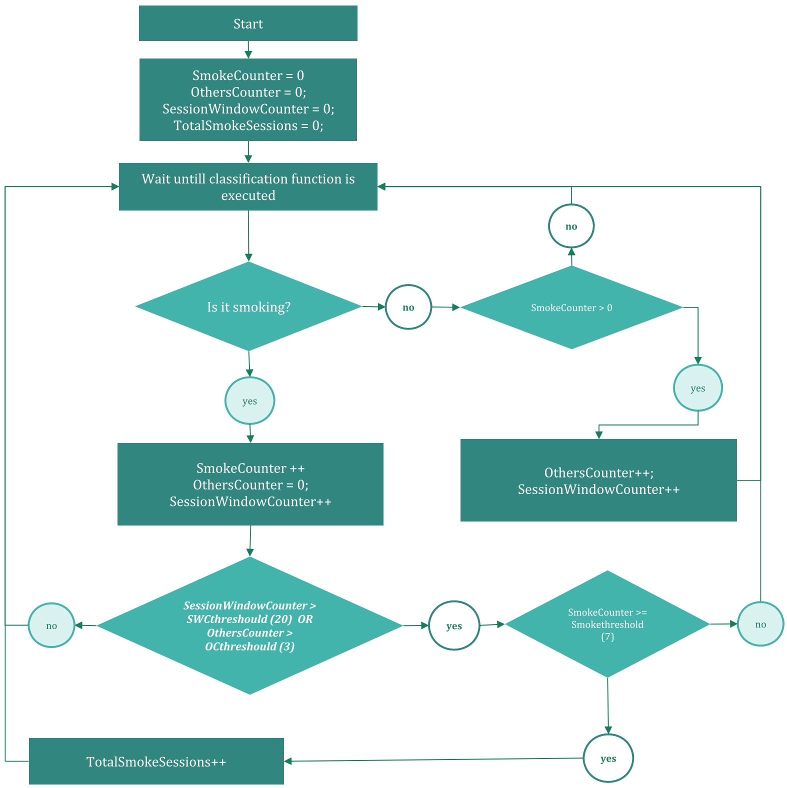


Fig. 3. Smoking session detection algorithm

After one these two conditions have been reached, then we see if there were at least seven segments of smoking in this session, shown in Fig. 3 by *Smokethreshold*. The average duration of a smoking session for our user in the testing phase was six minutes (12 segments). Therefore, we choose seven (7 segments: three and half minutes) as a threshold. If there are smoking predictions for more than half the average smoking session duration, then it should be classified as a smoking session. This value can be person-dependent. However, smokers can be asked at the start of using our app for their average smoking session duration. If we know that time, then this threshold can be automatically calculated for smokers according to their average smoking session duration. In Sect. 4.3, we evaluate the performance of this algorithm.

4 Performance Evaluation

For performance analysis, we trained four classifiers in WEKA: DT, SMO, RF, MLP. In this section, we first present the results of smoking recognition, using different sensors and different training models, namely subject-specific and generic training. Next, we evaluate the performance of smoking in a posture from which no training data was available. We also carried a resource consumption analysis while running online recognition algorithms. Our analysis show that the smartwatch's battery (LG R: 410 mAH) lasts for around eight hours while running a smoking recognition app using the accelerometer sensor only. The use of gyroscope in addition to an accelerometer decreases the battery life by almost an hour. The impact of classifiers' prediction task is very low on the battery except for KNN classifier because it runs through the whole dataset. In terms of memory usage, DT, MLP, SMO and RF (with 9 trees) classifiers occupy 12 to 13 MBs, while RF with 99 trees occupy 28 MBs. Model sizes are 19 KB for SMO, 137 KB for DT, 947 KB for MLP and RF (9 trees), while it is 10397 KBs for RF with 99 trees. The resource consumption analysis is discussed in detail in our other published work [20].

Finally, we analyze the performance of the proposed smoking session detection algorithm. Our evaluations involved a single participant and the results can be considered as indicative. However, these online results are found to be similar to our offline results from our previous study [16], and we expect it to be not so different if tested with a higher number of users. Additionally, in this paper our focus is more on the presentation of the framework and the smoking session recognition algorithm and the system can be tested with more participants for a more detailed performance analysis.

4.1 Smoking Recognition

For smoking recognition, we tested the four mentioned classifiers with one participant who also participated in the initial data collection phase presented in [16]. This participant wore a smartwatch (LG Watch R) at his right wrist and carried a smartphone (Samsung Galaxy S2) in his right pant's pocket for a couple of hours every day. The testing was spread over three weeks. He smoked 45 cigarettes over the testing period in various postures: 15 while standing, 15 while sitting, and 15 while walking. He also performed his daily activities during this testing time, such as working on a computer, taking lunch, drinking coffee, cooking, washing dishes, and many other activities.

In order to compare different scenarios, we created multiple versions of our app where each version was configured to run a specific scenario. Each scenario was defined by three components:

- Training Models: we trained the machine learning algorithms in two ways. (1) Using subject specific data: In this case, we only used data from this specific participant for training purpose who was taking part in our testing phase. (2) Using data from eleven participants: In this case, we used data from other nine participants and our current participant for training purpose. In this way, we can test both generic and subject-specific training models. In this dataset, we collected a dataset of 45 h for smoking and other similar activities such as eating and drinking coffee or tea. Out of these 45 h, the smoking activity was performed for 16.86 h in various forms such as smoking while sitting, standing, walking and in a group conversation. Each activity was performed multiple times by each participant on various days over a period of three months. Usually, the participants smoked 1–4 cigarettes (1 cigarette per session) in a day. In the meanwhile, they were also performing eating and drinking activities on different days according to their availability. Each participant wore a smart-watch (LG Watch R, LG Watch Urbane, Sony Watch 3) on the right wrist and a smartphone in the right pocket as all participants were right-handed. We collected data from multiple sensors from both smartwatch and smartphone, however, we only use accelerometer and gyroscope in this study. The data was collected at 50 samples per second from these sensors. For data collection, we developed our own Android application which can collect data from multiple sensors, both from the phone and smartwatch in real-time at a user-provided sampling rate. The details about this data is described in detail in [16]. We used 10-fold stratified cross-validation for evaluating our training models in WEKA.
- Sensor combinations: we used the accelerometer alone and also its combination with the gyroscope to see if there were any improvements due to such addition.
- Classifiers: For real-time activity recognition, we used four classifiers: SMO, RF9, MLP, and DT. Initially, we also tested with naive Bayes and KNN, however, we did not include them in this study because the recognition performance of naive Bayes was very low whereas running KNN was computationally expensive.

Each version of our app was running a specific scenario. These scenarios are shown in Table 1. For example, one version was running SMO classifier with the accelerometer alone whereas, the other was using the accelerometer with a gyroscope. In both these cases, the training data was coming from this specific participant who was doing the testing. However, at the same time, we were also running two other versions with similar configuration but the training data was coming from all ten participants. Similarly, four such versions were running for RF9. For MLP and DT, we used only its versions with accelerometer because we did not want to overload the CPU. All these different versions of our app were running at the same time.

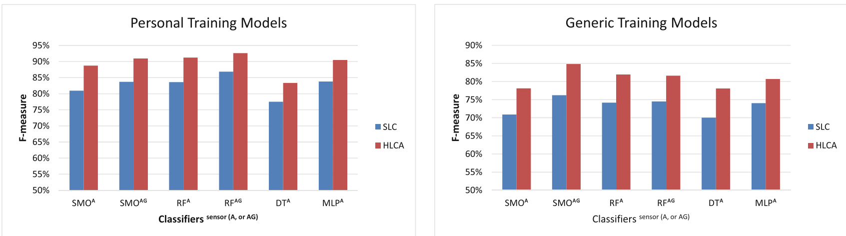
As the performance metric, we choose F-measure as our classes are imbalanced. For example, the time spent during smoking was around 5 h whereas the rest of the activities were performed for around 38 h. This can also be seen

Table 1. Real-time smoking recognition scenarios

Scenario	Classifier	Sensors	Training method
1	SMO^A	*Accelerometer*	*Subject-specific*
2	SMO^{AG}	*Accelerometer + Gyroscope*	*Subject-specific*
3	RF^A	*Accelerometer*	*Subject-specific*
4	RF^{AG}	*Accelerometer + Gyroscope*	*Subject-specific*
5	DT^A	*Accelerometer*	*Subject-specific*
6	MLP^A	*Accelerometer*	*Subject-specific*
7	SMO^A	*Accelerometer*	*Generic*
8	SMO^{AG}	*Accelerometer + Gyroscope*	*Generic*
9	RF^A	*Accelerometer*	*Generic*
10	RF^{AG}	*Accelerometer + Gyroscope*	Generic
11	DT^A	*Accelerometer*	Generic
12	MLP^A	*Accelerometer*	Generic

Table 2. Confusion matrices of various classifiers for smoking activity

			Predicted As											
			SMO^A		SMO^{AG}		$RF9^A$		$RF9^{AG}$		DT^A		MLP^A	
			Smoking	*Others*	*Smoking*	*Others*	*Smoking*	*Others*	*Smoking*	*Others*	*Smoking*	*Others*	*Smoking*	*Others*
Actual	***subject-specific Classifiers***	*Smoking*	467	115	495	87	495	87	522	60	443	139	509	73
		Others	105	4366	106	4365	107	4364	98	4373	118	4353	124	4347
	Generic Classifiers	*Smoking*	459	123	496	86	475	107	504	78	462	120	499	83
		Others	254	4217	223	4248	224	4247	267	4204	275	4196	267	4204



(a) Using subject-specific Training

(b) Using Generic Training

Fig. 4. Impact of various factors on recognition performance of smoking

from the confusion matrices in Table 2. However, we also use true positive rate or recall in some cases where we compare smoking in various postures because in such cases, true positive rate gives a better insight on their comparison. As shown in Table 2, when only accelerometer is used, MLP classifier performs the best in recognizing smoking, which is followed by RF both in subject-specific training and generic training. However, false positive rate of RF is lower (others recognized as smoking). When accelerometer is combined with gyroscope, missclassification rates decrease for all the classifiers.

In Fig. 4, we present the F-measure results obtained with different classifiers with either accelerometer or in combination with gyroscope. In Fig. 4a, results of using a subject-specific traning model are presented, while in Fig. 4b, results of using a generic training model are presented. Additionally, we ran single layer classification approach (mentioned as SLC in the figures) with our four classifiers in their default mode, as well as a hierarchical lazy classification algorithm (HLCA), proposed in [16]. Simply, HLCA is a rule based algorithm where classification results of activity recognition segments are smoothed/corrected by comparing with the results of neighboring data segments, considering the fact that human activities do not change instantly. In Fig. 4, results of using a single layer classification and HLCA are both presented. When we compare the results obtained with using a subject-specific model and a generic model, using a subject-specific model achieves better recognition rates, which is approximately 5% better. However, with generic models, still we achieve up to 85% F-measure with SMO and accelerometer and gyroscope combination. As mentioned, using gyroscope besides an accelerometer increases the recognition rates, however, this may increase the battery consumption on the devices, as we further investigated in [17].

When we compare the results of single layer classification with HLCA, as expected, we observed improvements in F-measure as observed also in our previous work [16] for offline recognition. Most of the misclassifications were corrected by taking into account the information among neighboring data segments. The observed improvements due to the addition gyroscope and due to the use HLCA can be seen in this figure.

4.2 Smoking Recognition with Different Postures

In Fig. 5, we present the recognition results (true positive rate) in different postures. We observe that it is relatively easy to recognize smoking while standing and while walking, but relatively difficult to detect smoking while sitting. The motion pattern of smoking while sitting can be very similar to drinking coffee or tea sometimes which makes it difficult to recognize. Due to this, it was mainly confused with drinking. Moreover, smoking can be done in many different ways while sitting compared to while standing and walking. Smoking while standing was recognized with the highest accuracy. We observed similar behavior when we calculated smoking sessions which we discuss in Sect. 4.3.

To see how well these trained models generalize, we also tested smoking while biking as we observed some smokers smoking while biking. It is important to note that these trained models had not included training data for smoking while biking. For this purpose, the participant smoked 5 cigarettes while biking from home to office on different days. We ran both SLC and HLCA for this detection and the results were reasonably well. HLCA outperformed the SLC approach. The F-measure and true positive for all twelve scenarios are given in Table 3.

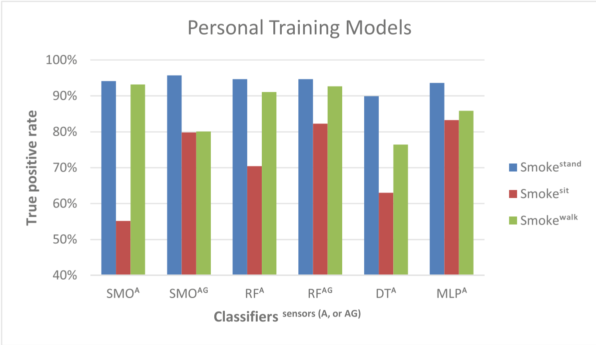


Fig. 5. Impact of smoking posture on its recognition performance

Table 3. Smoking$^{\text{biking}}$ recognition performance

		Generic Classifiers						subject-specific Classifiers					
		SMO^A	*SMO^{AG}*	*RF^A*	*RF^{AG}*	*DT^A*	*MLP^A*	*SMO^A*	*SMO^{AG}*	*RF^A*	*RF^{AG}*	*DT^A*	*MLP^A*
F-measure	*SLC*	*69%*	*74%*	*70%*	*71%*	*73%*	*65%*	*71%*	*83%*	*65%*	*69%*	*78%*	*73%*
	HLCA	*79%*	*81%*	*79%*	*82%*	*86%*	*79%*	*84%*	*93%*	*85%*	*85%*	*85%*	*84%*
True positive Rate	*SLC*	*86%*	*86%*	*84%*	*86%*	*81%*	*78%*	*69%*	*86%*	*57%*	*59%*	*81%*	*72%*
	HLCA	*100%*	*95%*	*98%*	*97%*	*95%*	*91%*	*84%*	*91%*	*79%*	*76%*	*84%*	*88%*

We observe that the recognition performance of smoking can be significantly improved by collecting more data for the null class (mentioned as others in our results). It is difficult to get a complete null class because we have to collect all possible hand gestures other than smoking for that. However, wearing the smartwatch for a few days should be sufficient for it. Initially, we tested our trained models with a limited null class which contained only drinking, sitting, standing, talking, and eating soup activities. The recognition results of smoking were relatively poor in terms of precision such that other random hand gestures were classified as smoking. After that, the participant who was involved in our testing process, collected more data for around 5 h while doing various activities such as washing dishes, laying in bed, working on computer, conversations, using stairs, walking, biking, drinking, watching TV etc. Using this additional data improved the overall recognition of the smoking activity. We believe collecting more data on daily activities can further improve the recognition of smoking.

4.3 Smoking Session Detection

We ran the smoking session detection algorithm, explained in Sect. 3.2 on the prediction results of all classifiers and compared its results with the ground truth. Based on this, we present the smoking sessions results in Table 4. It shows that this algorithm performs reasonably well for the subject-specific case. It can be seen from these results that the subject-specific classifiers provide better results than the generic ones as expected. Though the generic classifier performs

Table 4. Smoking sessions recognition

			Predicted as			
		Total predicted smoking sessions[a]	*Smoking*^{standing}	*Smoking*^{sitting}	*Smoking*^{walking}	*Others classified as smoking (false positives)*
Generic Classifiers	SMO^{A}	39	15	6	15	3
	SMO^{AG}	45	15	14	15	1
	RF^{A}	42	15	9	14	4
	RF^{AG}	46	15	14	13	4
	DT^{A}	40	15	8	15	2
	MLP^{A}	46	15	14	15	2
subject-specific Classifiers	SMO^{A}	40	15	10	15	0
	SMO^{AG}	45	15	15	14	1
	RF^{A}	42	15	11	15	1
	RF^{AG}	44	15	14	15	0
	DT^{A}	40	15	13	12	0
	MLP^{A}	45	15	14	15	1

[a] Actual smoking sessions = 45 where 15 while sitting, 15 while standing, and 15 while walking.

very well in some situations, it can lead to a higher number of false positives compared to a subject-specific classifier where a non-smoking activity is classified as smoking due to similar hand gestures. However, we expect this to improve when we add more data from these other participants in the context of the null class. Moreover, it is easy to identify smoking while standing whereas difficult while sitting as discussed earlier. In terms of sensors, the combination of an accelerometer and a gyroscope performs better than the accelerometer alone. In terms of classifiers, the support vector machine performs the best except for recognizing smoking while sitting using the accelerometer. We see higher recognition performance for support vector machine because it generalizes well and is resistant to over-fitting. However, it came at the cost of low performance for smoking while sitting because it is very similar to drinking tea or coffee.

We also ran this smoking session detection algorithm for the biking posture on top of our HLCA algorithm which corrects some of the misclassified smoking segments. We observed that the overall smoking session detection improved for all scenarios, however, in some cases, we had a higher number of false positives, especially for generic classifiers as shown in Table 5. It is an expected result as for HLCA to work better the underlying classification results should be reasonably high.

Finally, we ran of smoking session detection algorithm on top of SLC and HLCA and these results are shown in Table 6. It can be seen that we are able to recognize these smoking sessions with good accuracy even though there was no training data from such type of smoking. Our algorithm (HLCA) improves the smoking session recognition as well, however, occasionally it comes at the cost of false positives where a non-smoking session with random hand gestures or a drinking or eating session is classified as smoking. In the case of smoking session detection algorithm, running HLCA may not be very useful because both of them take into account the neighboring segments to improve the performance, making HLCA redundant.

Table 5. Smoking sessions recognition (HLCA)

			Predicted as			
		Total predicted smoking sessions	*Smoking^{stand}*	*Smoking^{sit}*	*Smoking^{walk}*	*Others classified as smoking (false positives)*
Generic Classifiers	***SMO^{A}***	39	15	6	15	3
	SMO^{AG}	45	15	14	15	1
	RF^{A}	42	15	9	14	4
	RF^{AG}	46	15	14	13	4
	DT^{A}	40	15	8	15	2
	MLP^{A}	46	15	14	15	2
subject-specific Classifiers	***SMO^{A}***	40	15	10	15	0
	SMO^{AG}	45	15	15	14	1
	RF^{A}	42	15	11	15	1
	RF^{AG}	44	15	14	15	0
	DT^{A}	40	15	13	12	0
	MLP^{A}	45	15	14	15	1

Table 6. Smoking^{biking} sessions recognition

			Total predicted smoking sessions		*Predicted as Smoke^{biking}*		*Others classified as smoking*	
		Actual Smoke^{biking} Sessions	SLC	HLCA	SLC	HLCA	SLC	HLCA
Generic Classifiers	***SMO^{A}***	5	5	6	4	5	1	1
	SMO^{AG}	5	5	5	5	5	0	0
	$RF9^{A}$	5	5	6	5	5	0	1
	$RF9^{AG}$	5	5	6	5	5	0	1
	DT^{A}	5	5	5	5	5	0	0
	MLP^{A}	5	5	5	5	5	0	0
subject-specific Classifiers	***SMO^{A}***	5	4	4	4	4	0	0
	SMO^{AG}	5	5	5	5	5	0	0
	$RF9^{A}$	5	3	4	3	4	0	0
	$RF9^{AG}$	5	3	4	3	4	0	0
	DT^{A}	5	5	5	5	5	0	0
	MLP^{A}	5	4	5	4	5	0	0

5 Conclusions and Future Work

In this paper, we presented a modular activity recognition system based on our conceptual framework, SmokeSense, for mobile phones and smartwatches where various classifiers, feature sets, and other parameters can be evaluated. As a case study, we analyzed the recognition performance of smoking on smartwatches and achieved an F1-measure of 92% for subject-specific classification and 85% for generic classification. In terms of recognition performance, we observed similar trends for online activity recognition as we observed previously in our offline analysis. For smoking recognition, the addition of the gyroscope to an accelerometer helped in improving recognition performance. Although, subject-specific training models are observed to exhibit better performance, generic models also perform well at an acceptable level. We learned that smoking while sitting is difficult to recognize compared to other postures due to its similarity to drinking while sitting. We also showed that smoking can be recognized in different postures, such as while biking, from which no training data was available. Finally, we proposed a smoking session detection algorithm and showed that it performs well in identifying the number of cigarettes smoked. Our evaluations involved a single participant, we are planning to test this system with more participants. We also plan to develop a context-aware activity recognition algorithm where sensors, sampling rates, window sizes are decided on demand.

Acknowledgements. This work is supported by Dutch National Program COMMIT in the context of SWELL project, by the Galatasaray University Research Fund under the grant number 17.401.004.

References

1. Lara, O.D., Labrador, M.A.: A survey on human activity recognition using wearable sensors. IEEE Commun. Surv. Tutor. **15**(3), 1192–1209 (2013)
2. Könönen, V., Mäntyjärvi, J., Similä, H., Pärkkä, J., Ermes, M.: Automatic feature selection for context recognition in mobile devices. Pervasive Mob. Comput. **6**(2), 181–197 (2010)
3. Fahim, M., Fatima, I., Lee, S., Park, Y.-T.: EFM: evolutionary fuzzy model for dynamic activities recognition using a smartphone accelerometer. Appl. Intell. **39**(3), 475–488 (2013)
4. Dernbach, S., Das, B., Krishnan, N.C., Thomas, B.L., Cook, D.J.: Simple and complex activity recognition through smart phones. In: 2012 8th International Conference on Intelligent Environments (IE), pp. 214–221. IEEE (2012)
5. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD Explor. Newsl. **11**(1), 10–18 (2009)
6. Shoaib, M., Scholten, H., Havinga, P.: Towards physical activity recognition using smartphone sensors. In: Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC), pp. 80–87, December 2013
7. Shoaib, M., Bosch, S., Incel, O.D., Scholten, H., Havinga, P.J.: Fusion of smartphone motion sensors for physical activity recognition. Sensors **14**(6), 10146–10176 (2014)
8. Wu, W., Dasgupta, S., Ramirez, E.E., Peterson, C., Norman, G.J.: Classification accuracies of physical activities using smartphone motion sensors. J. Med. Internet Res. **14**(5), e130 (2012)
9. Kose, M., Incel, O.D., Ersoy, C.: Online human activity recognition on smart phones. In: Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data, vol. 16, no. 2012, pp. 11–15 (2012)
10. Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of Machine Learning. MIT Press, Cambridge (2012)
11. Shoaib, M., Bosch, S., Incel, O.D., Scholten, H., Havinga, P.J.: A survey of online activity recognition using mobile phones. Sensors **15**(1), 2059–2085 (2015)
12. Rawassizadeh, R., Tomitsch, M., Nourizadeh, M., Momeni, E., Peery, A., Ulanova, L., Pazzani, M.: Energy-efficient integration of continuous context sensing and prediction into smartwatches. Sensors **15**(9), 22616–22645 (2015)
13. Poyraz, E., Memik, G.: Analyzing power consumption and characterizing user activities on smartwatches: summary. In: IEEE International Symposium on Workload Characterization (IISWC), pp. 1–2, September 2016
14. Liu, X., Chen, T., Qian, F., Guo, Z., Lin, F.X., Wang, X., Chen, K.: Characterizing smartwatch usage in the wild. In: Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys 2017, pp. 385–398. ACM, New York (2017)
15. Shoaib, M., Bosch, S., Incel, O.D., Scholten, H., Havinga, P.J.: Complex human activity recognition using smartphone and wrist-worn motion sensors. Sensors **16**(4), 426 (2016)

16. Shoaib, M., Scholten, H., Havinga, P.J., Incel, O.D.: A hierarchical lazy smoking detection algorithm using smartwatch sensors. In: IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), pp. 1–6. IEEE (2016)
17. Shoaib, M.: Sitting is the new smoking: online complex human activity recognition with smartphones and wearables. Ph.D. dissertation, cTIT Ph.D. thesis series no. 17–436, May 2017
18. Tang, Q., Vidrine, D.J., Crowder, E., Intille, S.S.: Automated detection of puffing and smoking with wrist accelerometers. In: Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare, pp. 80–87. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2014)
19. Chen, G., Ding, X., Huang, K., Ye, X., Zhang, C.: Changing health behaviors through social and physical context awareness. In: 2015 International Conference on Computing, Networking and Communications (ICNC), pp. 663–667. IEEE (2015)
20. Shoaib, M., Incel, O.D., Scholten, H., Havinga, P.J.: Resource consumption analysis of online activity recognition on mobile phones and smartwatches. In: Proceedings of the 36th IEEE International Performance Computing and Communications Conference. IEEE, 10–12 December 2017
21. Incel, O.D., Kose, M., Ersoy, C.: A review and taxonomy of activity recognition on mobile phones. BioNanoScience **3**(2), 145–171 (2013)
22. Gjoreski, M., Gjoreski, H., Luštrek, M., Gams, M.: How accurately can your wrist device recognize daily activities and detect falls? Sensors **16**(6), 800 (2016)
23. Gjoreski, M., Gjoreski, H., Luštrek, M., Gams, M.: Recognizing atomic activities with wrist-worn accelerometer using machine learning. In: Proceedings of the 18th International Multiconference Information Society (IS), Ljubljana, Slovenia, pp. 10–11 (2015)
24. Attal, F., Mohammed, S., Dedabrishvili, M., Chamroukhi, F., Oukhellou, L., Amirat, Y.: Physical human activity recognition using wearable sensors. Sensors **15**(12), 31314–31338 (2015)
25. Garcia-Ceja, E., Brena, R.F., Carrasco-Jimenez, J.C., Garrido, L.: Long-term activity recognition from wristwatch accelerometer data. Sensors **14**(12), 22500–22524 (2014)
26. Knighten, J., McMillan, S., Chambers, T., Payton, J.: Recognizing social gestures with a wrist-worn smartband. In: 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), pp. 544–549. IEEE (2015)
27. Documentation for Google activity recognition API. http://developer.android.com/training/location/activity-recognition.html. Accessed 21 July 2014
28. Seneviratne, S., Hu, Y., Nguyen, T., Lan, G., Khalifa, S., Thilakarathna, K., Hassan, M., Seneviratne, A.: A survey of wearable devices and challenges. IEEE Commun. Surv. Tutor. **19**(4), 2573–2620 (2017)
29. Scholl, P.M., Van Laerhoven, K.: A feasibility study of wrist-worn accelerometer based detection of smoking habits. In: 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp. 886–891. IEEE (2012)
30. Varkey, J.P., Pompili, D., Walls, T.A.: Human motion recognition using a wireless sensor-based wearable system. Pers. Ubiquit. Comput. **16**(7), 897–910 (2012)

31. Saleheen, N., Ali, A.A., Hossain, S.M., Sarker, H., Chatterjee, S., Marlin, B., Ertin, E., al'Absi, M., Kumar, S.: puffMarker: a multi-sensor approach for pinpointing the timing of first lapse in smoking cessation. In: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 999–1010. ACM (2015)
32. Parate, A., Chiu, M.-C., Chadowitz, C., Ganesan, D., Kalogerakis, E.: RisQ: recognizing smoking gestures with inertial sensors on a wristband. In: Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services, pp. 149–161. ACM (2014)
33. https://weka.wikispaces.com/Serialization. Accessed Dec 2017
34. https://weka.wikispaces.com/Primer. Accessed Dec 2017