Self-Adaptive and Lightweight Real-Time Sleep Recognition With Smartphone

Ennio Gambi, *Member, IEEE,* Simone Barbetta, Adelmo De Santis, and Manola Ricciuti

Original scientific paper

Abstract—**It is widely recognized that sleep is a basic physiological process having fundamental effects on human health, performance and well-being. Such evidence stimulates the research of solutions to foster self-awareness of personal sleeping habits, and correct living environment management policies to encourage sleep. In this context, the use of mobile technologies powered with automatic sleep recognition capabilities can be helpful, and ubiquitous computing devices like smartphones can be leveraged as proxies to unobtrusively analyse the human behaviour. To this aim, we propose the implementation of a real-time sleep recognition methodology relied on a smartphone equipped with a mobile app that exploits contextual and usage information to infer sleep habits. As an improvement of already presented solutions, in this proposed application an initial training stage is required, during which the selected features are processed by k-Nearest Neighbors, Decision Tree, Random Forest, and Support Vector Machine classifiers, in order to select the best model for each user. Moreover, a 1st-order Markov Chain is applied to improve the recognition performance. Experimental results demonstrate the effectiveness of the proposed approach, achieving acceptable results in term of Precision, Recall, and F1-score.**

Index Terms—**Mobile application, smartphone sensing, sleep monitoring, activity recognition.**

I. Introduction

SLEEP is a fundamental aspect for humanity and influences performances, health status and psychological and physical well-being significantly. Most of the adult individuals need about 7-9 hours of continuous sleep [1]. However, this time interval greatly depending on the activities that daily involve people. For example, one study analyzed that student sleep duration is decreased over the years, from about 7.5 hours per night in 1969 to 6.5 hours per night in 1989 [2]. Roenneberg, in [3], affirmed that sleep disorders affect about 70 million people only in the United States. Moreover, the author stated that the knowledge of real world sleep patterns could be very useful in order to determine how much sleep is optimal, how its quality can be measure or predicted, and the role of genetic and environmental factors.

The automatic sleep recognition, carried out without direct intervention of the subject, can promote the self-awareness of sleeping patterns, enabling the subject to collect objective information that can be shown to a consultant [4]. Many studies showed that ambient factors like light, temperatures or external interruptions tend to bother sleep [5], [6].

Among the technological tools to improve people's awareness about their sleep habits, the wearable sleep trackers have the advantage of being portable and, staying in contact with the user's body, provide information on movement, heart rate and temperature. The disadvantages are that they may be considered as intrusive and susceptible to lose tracking continuity for a variety of reasons, including uncomfortable, health problems and battery life [7]. In fact, each time the device battery needs to be recharged, the device is removed by the subject and this can cause the missing detection of possible events of interest.

To overcome these limitations, smartphone-based solutions exploiting suitably designed mobile apps can be very helpful. Campbell and Choudhury [8] stated that smartphones are the first really ubiquitous computing devices that are becoming extraordinary proxies to analyze human behaviour [9] and health [10]. Shin and Cho presented an application [11] to detect snoring through the smartphone built-in microphone, which must therefore located in a position close to the subject. In the same year, Abdullah *et al.* presented in [12] a rules-based method which determines sleep onset, duration, and midpoint, from screen on-off patterns.

Several mobile applications are available in app stores for Android- and iOS-based devices. Among them, *Sleep as Android* [13] is a smart alarm clock with sleep cycle tracking, designed to wake up the user in the moment identified as the optimal one. The app still requires the user to notify the moment in which the sleep period starts, in order to activate the sleep quality monitoring functionality. Another app designed for Android devices, called *Sleep Detector* [14], detects the beginning of the sleep time based on the fact that usually people check their emails or social accounts just before going to sleep, and then leave their phone on charging. The app "remembers" the last user activity on the phone and sets this time as the start of sleeping time. As a consequence, the sleep time detection is basically performed by analysing the user interaction with the smartphone. *Samsung Health* [15], available for all Android mobile phones, is able to track the user activity, nutrition and sleep, and to provide health insights; this app may also allow a video chat with board-certified doctors to get immediate help.

In the domain of mobile apps for iOS-based devices, the most interesting ones are *Under Armour Record* [16], *Sleep Cycle* [17], that is available for Android devices too. The former is actually a connected health and fitness system,

Manuscript received July 17, 2018; revised August 30, 2018. Date of publication September 8, 2018. Prof. Mladen Russo has been coordinating the review of this manuscript and approved it for publication.

Authors are with the Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, Ancona, Italy, e-mails: e.gambi@univpm.it, S1074218@studenti.univpm.it, a.desantis@univpm.it, m.ricciuti@staff.univpm.it.

Digital Object Identifier (DOI): 10.24138/jcomss.v14i3.584

1845-6421/09/584 © 2018 CCIS

but it provides a built-in sleep monitoring function and the possibility to cross-compare data such as fitness vs. sleep. To operate properly, the app needs to be activated every day, and the sleep time should be longer than 4 hours to be correctly detected. The latter tracks the user's sleep patterns and wakes him/her up during light sleep. The app uses sound analysis based on machine learning to identify sleep states by tracking movements in bed, so, as the built-in microphone is used to analyze the movements, it's important to correctly place the phone during the night, over the matress and near to the pillow. In any case the user has to activate the app on purpose before going to sleep, and place the phone with its screen downward. Details on the machine learning algorithm are covered by patent. In all the cases herein presented, it is requested a strong user interaction with the phone, and in particular to notify the moment at which the sleeping time starts and ends.

Contrary to the previously described applications, the aim of the app presented in this paper is just related to the definition of the sleep time duration. Furthermore, this results is obtained without any interaction with the user, once ended the initial phase of classifiers training. In fact, it is not required to the user to modify its habits, or to remember to digit any information. Indeed, in order to recognize the sleep activity in real-time it is sufficient that contextual information acquired from smartphone's onboard sensors are combined with that resulting from mobile phone usage. Being based on the sleep recognition through a classification method, the app is able to recognize the sleep, or not, if the classifier was suitably trained. In other words, the classifier may recognize just the situations that have been presented during the training phase. This makes the app robust against particular, and different, habits of the users, as sleep during daily hours, particular charging condition and so on.

In a previous paper [18], the proposed methodology was first applied offline, in a Matlab environment, and in real time, on data collected through a quite simple and preliminary mobile application running on users' smartphones. Having verified the effectiveness of the approach, in this work we present a stable and fully functional mobile application for Android-based devices, implementing the aforementioned real time and self-adaptive methodology. In fact, contrary to the first version of the app, where a pre-stored dataset was used for classifier training, the application here proposed requires a personalized training period. Each user is in fact required to ”communicate” to the app the sleep start and end time. In this way the app may be adapted to the different users, being so able to recognize the sleep of users having really different habits. At the same time, the app finds the best classifier for each user, considering its habits and the different characteristics of the sensors implemented on the smartphones. In this way, the same app working on different users may choose different classifiers; with this approach, an increase of the classification performance may derive. Once the most performing classifier is selected, following a time-limited training period over which the user is requested to annotate the true sleeping intervals by means of a control available on the app, no additional direct action from the user is requested. The app has been provided to users for a second, real field trial. Results demonstrate the real time feasibility of the proposed method and the good performance in correctly recognizing sleep habits.

The paper is organized as follows. Section II describes the proposed methodology for sleep recognition, while Section III presents the preliminary tests performed offline, and the results obtained. Section IV describes the fully functional Android application designed to implement the proposed methodology, and Section V presents the outcomes of the field trial performed by users adopting the mobile application on their personal devices. Finally, Section VI draws the main conclusion of the work.

II. Methodology

The real-time sleep recognition methodology presented in this paper aims to exploit the physical sensors onboard the smartphone to detect the sleep activity in an unobtrusive way. Additionally, data related to the usage of the device can be also acquired to provide a more complete set of features to be used. The main idea is to sample data on the environment surrounding the user, and on the smartphone usage, process them, and distinguish between sleep and wakefulness. The method is schematized in the Figure 1.

The set of features chosen for sleep recognition, identified by Chen *et al.* in [19], are the following ones:

1) *Light* [L], is the light intensity measured by the phone sensor in lx.
2) *Motion* [M], is based on the onboard triaxial accelerometer.
3) *Sound* [S], a noise index achieved from the microphone, the only featured stored, important aspect to preserve user's privacy issues.
4) *Phone Block*[PB], is such that the state 1 means “phone locked”, while the state 0 means “phone unlocked”.
5) *Phone Charging* [PC], is such that the state 1 means “phone in charge”, while the state 0 means “phone not in charge”.

Every 5 minutes a feature vector X of values (X_1, X_2, .., X_5) is filled with the normalized values (L, M, S, PB, PC) obtained from the sensors, and then used for sleep recognition, as input for classification algorithms. The output of the first phase of the classification method provides the best classifier, chosen from four possible ones, i.e. k-Nearest Neighbors (KNN), Random Forest (RF), Decision Tree (DT), and Support Vector Machine (SVM), on the basis of performances obtained from the preliminary test campaign, described in Section III. As an example of classification procedure, KNN classifier is based on the measure of the distance between the ”test” data (to be classified) and each of the ”training” data, acquired during the learning phase of the algorithm, where each set of data is associated to the sleep or not [20]. The chosen distance function can affect the classification accuracy, and in this work the Minkowski metric was considered:

$$Distance = \sqrt[r]{\sum_{j=1}^{5} |X_j^{test} - X_j^{training}|^r} \qquad (1)$$

When in (1) the value r is equal to 1 the distance is sometimes known as the Manhattan distance, while if equal to 2 is the

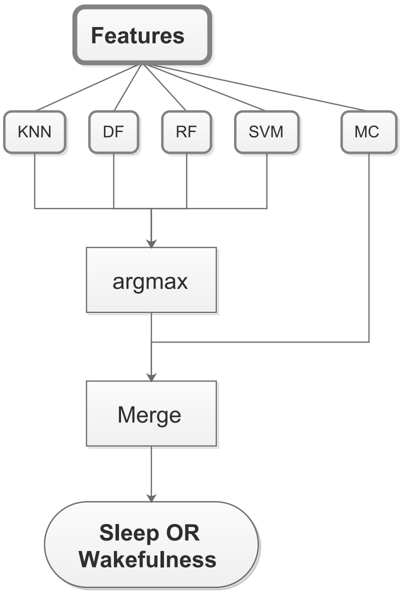


Fig. 1. General scheme of the proposed sleep recognition method.

Euclidean distance. The classification is then assigned to the largest class among its k nearest neighbours (where k is an odd integer to avoid ambiguity).

Integrating an N-order Markov Chain (MC), which takes into account everything that happened previously [21], the algorithm's robustness increased. Therefore, if in the previous sampling instants, the user was sleeping, it is assumed that probably, in the current instant he/she is still sleeping. The transition matrix is calculated taking into account feedbacks provided by the single user or multiple users so that, progressively, it will fit to the users' habits.

The value returned by the transition matrix provides information on the reliability of the classification process. If the classifier produces a highly improbable result, the algorithm will make decisions to the next sampling instant. Only when the probability value is sufficiently high, the algorithm decides on the action to carry out, such as silencing incoming calls, or turning off television when sleeping, if the application is associated with a specific phone or house management systems.

III. Preliminary Experimental Results

In order to check the effectiveness of the proposed methodology, a preliminary test campaign has been conducted, in which real data have been collected from different users running a simple version of the mobile app on their smartphones, but the test on the classification results have been executed offline, in a Matlab environment.

The dataset used for this work consisted of eight young users, five male and three female, aged between 20 and 35 years. Every 5 minutes light, motion, sound, phone block, and phone charging information have been extracted, through a smartphone application featuring a background service and based on the Android Operating System (O.S.).

In a previous study [18] the data have been collected and divided into training and test sets. The classification test was performed first on the basis of a single user dataset and then taking all the collected data as a whole. In the former approach, all the collected data are acquired over a single day, instead, the training set is made up by all the other data. In the latter approach, they are formed by all users' test and training sets, respectively.

The classification process is carried out with KNN (Minkowski metric, 5 neighbours), DT (Gini criterion), RF (Gini criterion, 10 estimators), and SVM (polynomial kernel, 3rd degree) classifiers.

The robustness of the algorithm increased by introducing a 1st-order MC to implement an error correction process. In this way, following the classification step, the system is able to differentiate between *likely*, when the transition probability is greater than 0.5, and *unlikely* outputs, on the basis of the relative value given by the transition matrix.

Table I shows the system performances Precision (P), Recall (R), and F1-score (F1-s) after the error correction process, for each subject (S). The performances reached a F1-score of 0.95 for all subjects.

Table II exhibits the different classifiers (C) selected and the performances Precision (P), Recall (R), and F1-score (F1-s) resulting over all the users. Also in this case, the best F1-score is achieved by the RF classifier, and it is equal to 0.93. It is evident from the Table II that the performance of the classifiers are not so many different, even if a slight difference is present, making so possible to identify the best classifier for the considered user.

So, although performed on a limited dataset, the preliminary experimental results discussed above demonstrate the capability of the proposed methodology to correctly distinguish between sleep and awake states. Moving from these premises, a fully-functional mobile application for devices running an Android O.S. has been developed, in order to test in a real field trial the possibility to run the proposed methodology in realtime, exploiting the computational capabilities of consumer devices.

IV. Mobile App Design and Implementation

The real-time sleep recognition methodology discussed in previous sections and tested offline has been implemented as a mobile application for Android O.S. devices, with compatibility from Android 5.0 (Lollipop), up to Android 7.1.1 (Nougat). As already mentioned, the app aims to recognize the sleep/awake condition of the subject using the smartphone by collecting five values every 5 minutes: three generated by on-board physical sensors (accelerometer, light sensor and microphone), and two from the device O.S., related to the phone block and charging status.

A. Application timeline

The proposed methodology requires to train the four available classifiers before being able to select the best one for the specific subject, and to use it to generate automatic classification. To this aim, the application features an execution timeline, shown in Figure 2, according to which, following the first start, it will spend a few days collecting data and asking

TABLE I
SYSTEM PERFORMANCES AFTER THE ERROR CORRECTION PROCESS FOR EACH SUBJECT (S).

S	Value	P	R	F1-s
1	0	0.93	0.80	0.86
	1	0.70	0.89	0.78
	avg/tot	0.85	0.83	0.83
2	0	0.99	1.00	1.00
	1	1.00	0.99	1.00
	avg/tot	1.00	1.00	1.00
3	0	0.96	0.98	0.97
	1	0.97	0.92	0.94
	avg/tot	0.96	0.96	0.96
4	0	1.00	0.90	0.95
	1	0.86	1.00	0.92
	avg/tot	0.95	0.94	0.94
5	0	1.00	0.99	0.99
	1	0.96	1.00	0.98
	avg/tot	0.99	0.99	0.99
6	0	1.00	1.00	1.00
	1	1.00	1.00	1.00
	avg/tot	1.00	1.00	1.00
7	0	0.99	0.90	0.94
	1	0.75	0.98	0.85
	avg/tot	0.94	0.92	0.92
8	0	1.00	1.00	1.00
	1	1.00	1.00	1.00
	avg/tot	1.00	1.00	1.00
TOT	0	0.98	0.94	0.96
	1	0.89	0.96	0.92
	avg/tot	**0.95**	**0.95**	**0.95**

TABLE II
CLASSIFICATION RESULTS FOR DIFFERENT CLASSIFIERS, CONSIDERING ALL SUBJECTS.

C	Value	P	R	F1-s	Support
KNN	0	0.96	0.91	0.93	1442
	1	0.84	0.92	0.88	753
	avg/tot	0.92	0.91	0.91	2177
DT	0	0.95	0.90	0.92	1442
	1	0.82	0.90	0.86	753
	avg/tot	0.91	0.90	0.90	2177
RF	0	0.97	0.92	0.94	1442
	1	0.86	0.94	0.90	753
	avg/tot	**0.93**	**0.93**	**0.93**	2177
SVM	0	0.93	0.94	0.93	1442
	1	0.87	0.85	0.86	753
	avg/tot	0.91	0.91	0.91	2177

the user to provide the ground truth (GT) information. Then, once a sufficient amount of data records have been collected to populate both the training and the testing sets, the app will check the classifiers and select the one that provides the best recognition performance. From that moment on, the automatic recognition mode of operation can be activated by the user, and the app will apply the classifier model that has been selected during the former phase. Finally, after a month from the first start, the timeline will be executed again from scratch, so that the application can self-adapt to possible changes in the sleep habits which have occurred meanwhile.

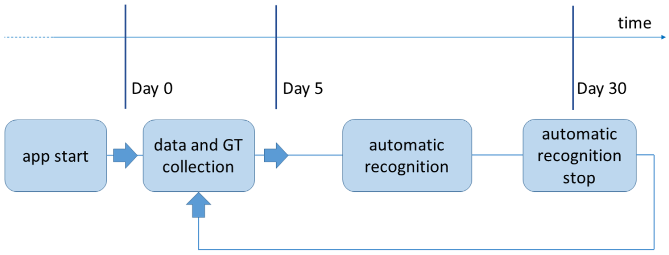


Fig. 2. Mobile application execution timeline.

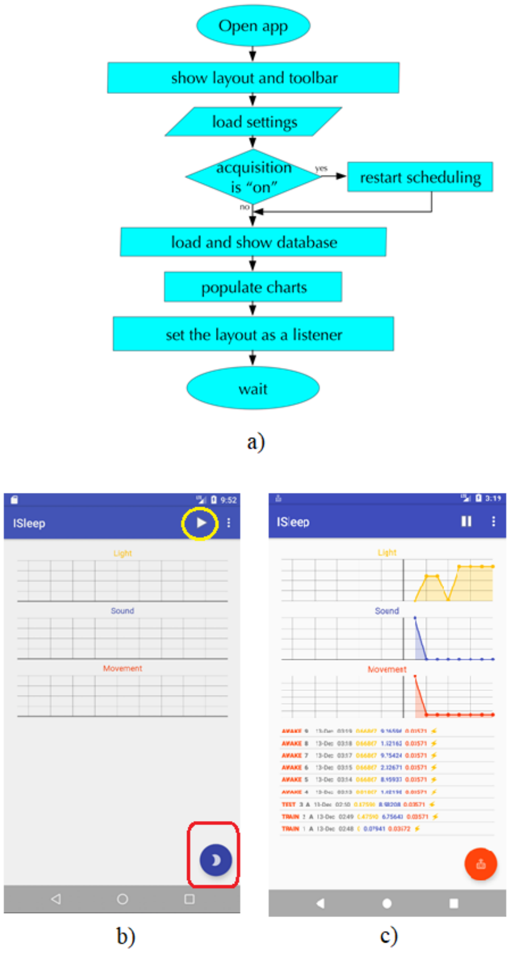


Fig. 3. Mobile application during the initial phase of the timeline following the first start: a) flow diagram of the operations executed by the app, b) user GUI, c) charts populated at the end of the acquisition stage.

B. Data collection stage

Following the first start, the application shall collect both the data from sensors and smartphone usage, and the GT information on sleep or awake state input by the user. In Figure 3 a), the flow diagram of the operations executed by the app in this first stage is illustrated. As soon as the app is opened by the user, the underlying code verifies if the acquisition process is turned off or on. In the former case, the outcome of previously performed acquisitions is shown to the user by a number of charts in the GUI, as in Figure 3 c); in the latter case, the alarm in charge of managing the acquisitions is rescheduled. This is motivated by the experimental observation that different Android O.S. perform a different management of scheduled alarms, i.e. temporized processes that have to be repeated in future time. In fact, scheduled alarms may be stopped for several reasons: typically, when the smartphone is running out of battery, or needs to use its computational resources to perform high-priority tasks, scheduled alarms are interrupted, especially the ones activated since the longest time. Then, from tests performed with different devices, we found that interrupted alarms usually are not resumed, and the corresponding processes don't get reactivated. As a consequence, in order to avoid that a missing re-activation of the acquisition process gets unnoticed, each time the app is opened, the corresponding alarm is scheduled again from scratch. In Figure 3 b), the Graphical User Interface (GUI) provided by the app during the data collection stage is shown. It is possible to notice, in the upper right corner, the touch command to start acquisition (given by the right arrow), whereas in the bottom right corner of the app, an icon is used to allow the user input the GT status. When the user is going to sleep, the moon icon will be activated; the sun icon appears to input the awake state.

Data acquisition from the three physical sensors (microphone, accelerometer and light sensor) and the O.S., is performed once every 5 minutes, for 1000 ms. It is triggered by a *scheduled alarm* executed by the Android O.S..Specifically: for the sound, the average amplitude over the observation period of 1000 ms is computed from samples obtained by invoking the *getAmplitude()* function; for the light, the maximum light intensity value over the observation time is collected through the *getMaximumRange()* function, and, for the movement, a single value relative to g, from the accelerations measured along the three axes, is computed and saved. The information about the smartphone state (blocked or recharged) is stored in two boolean variables, the value of which is obtained directly from the O.S.. All the above values are saved as a new record in a table of a local SqLite database implemented within the mobile app, joint with a timestamp, and the GT label about the sleep/awake state, provided by the user through the app GUI. Simple queries are used to input data, modify a value (according to the MC control), and get data from the table, if they are to be shared. Finally, the app provides a configuration menu in which it is possible to activate the Artificial Intelligence (AI) for automatic status recognition (once enough records have been collected to train and test the classifiers and select the best one), delete the collected data, or share them. The whole process of the data collection stage is represented by the flow diagram in Figure 4. Considering the frequency of data acquisition, after a period of 30 days there are 8640 data records in the local database.

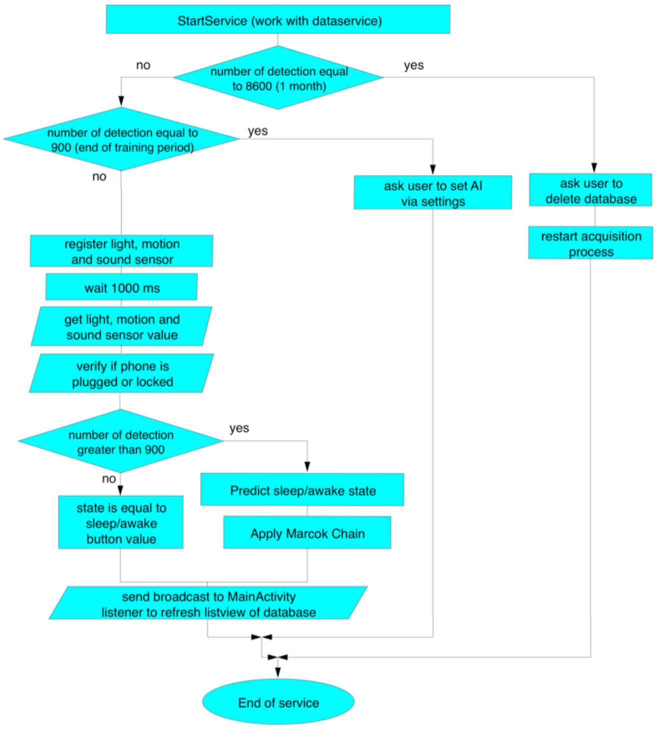

StartService (work with dataservice)
no
number of detection equal to 8600 (1 month)
yes
number of detection equal to 900 (end of training period)
yes
no
ask user to set AI via settings
ask user to delete database
restart acquisition process
register light, motion and sound sensor
wait 1000 ms
get light, motion and sound sensor value
verify if phone is plugged or locked
number of detection greater than 900
yes
no
Predict sleep/awake state
state is equal to sleep/awake button value
Apply Marcok Chain
send broadcast to MainActivity listener to refresh listview of database
End of service

Fig. 4. Flow diagram of the operations performed in the data collection stage.

C. Automatic recognition stage

The automatic recognition stage, during which the AI is active, starts as soon as enough labeled data records have been collected. In fact, on the basis of the results of experimental evaluations, almost 600 data records are necessary to train the classifiers, and 300 data records to test them, i.e. a total amount of at least 900 records are needed. This way, 2/3 and 1/3 of the available dataset are being used for training and testing, respectively. As described in the previous paragraphs, the recognition of the duration of sleep occurs by comparing the data acquired from the smartphone sensors with those acquired during the training phase. Therefore, during this phase, the precision with which the user inserts the start and end instants of sleep plays a crucial role, able to compromise the performance of the app.

Once 900 labeled records are stored in the local database, the app notifies the user to activate the AI through the configuration menu. The app takes less than 1 minute to obtain the classifier model: in fact, the four classifiers (KNN, RF, DT, and SVM) are trained and tested on the available dataset, and the one providing the best recognition score is selected as the active model. Then, the user is prompted to get back to the app home page; from that moment on, the app automatically restarts the acquisitions from the sensors and, by applying the selected classifier model, each collected data record is labeled as *sleep* or *awake*.

D. Design issues and countermeasures

The implementation of the automatic recognition methodology described in the previous sections through the Android mobile app presents a number of issues. During the data collection stage, the sleep/awake user's status input manually has to be added to the right data records, that are acquired from sensors by an *Intent*. To enable such an operation, the Android *SharedPreferences* are used, i.e. a memory partition in the smartphone that is dedicated to save primary data type.

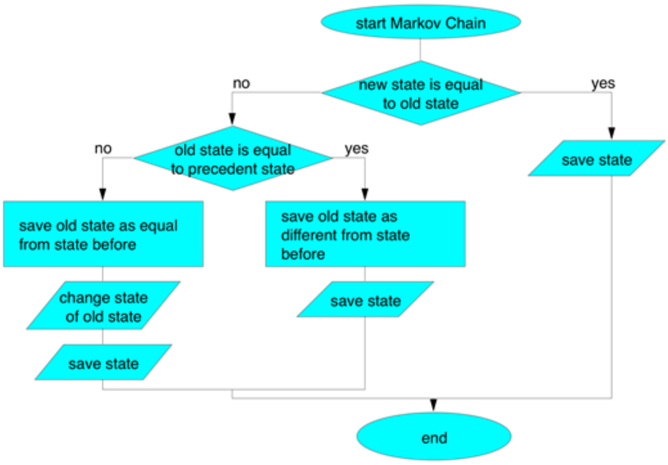

start Markov Chain
new state is equal to old state
no
yes
old state is equal to precedent state
no
yes
save state
save old state as equal from state before
save old state as different from state before
change state of old state
save state
save state
end

Fig. 5. Use of two *SharedPreferences* to implement the MC.

In the same stage, it is important to find a suitable time interval to allow the acquisition of an adequate number of data records without draining too much battery. Tests on different devices, including quite old ones, show that 5 minutes is a good tradeoff. During the processing of the data performed to train and test the available classifiers, the classifier model serialization requires the use of an *ObjectOutputStream*, to save the serialized classifier model in a folder accessible to the mobile app. This way, the model can be deserialized everytime it is needed. It is not possible to use again the *SharedPreferences*, as they do not allow to save non-primary types of data. Additionally, the app method in charge of creating the different classifiers' models is implemententd as an *AsyncTask*; this way, even when the model creation requires longer time, the process is not identified as blocked and the task is not aborted. A progress dialog is shown to the user on the GUI, to make it clear that the model creation task is active and working. The *SharedPreferences* are used also to implement the MC that improves the automatic status recognition performances, as shown in the flow driagram of Figure 5.

V. Mobile App Field Trial

The mobile app has been tested on different consumer and mass market Android smartphones; none of the users reported any malfunctions or excessive drain of battery or memory. Table III provides details on the participation in the mobile app field trial. The number of days refers to duration of the automatic recognition stage following the data collection, and classifier model creation.

In Table IV, the best classifier for each subject participating in the app field trial, together with the corresponding performances, is detailed.

The smallest F1-Score (0.572) is obtained with the KNN classifier for user no. 4, whereas the highest F1-score is obtained with the RF classifier for user no. 1 (0.98). Apart from a couple of cases for which the F1-score is under 80%, on average the methodology implemented with the app provides satisfactory automatic sleep recognition capabilities.

TABLE III
Details about the participation in the mobile app field trial.

Subject (gender)	N. of days	N. of records	N. of sleep records	N. of wake-fulness records
1 (F)	7	1569	434	1135
2 (M)	7	1520	512	1008
3 (M)	6	1133	319	814
4 (F)	8	1745	418	1327
5 (M)	7	1752	552	1200
6 (F)	8	1826	501	1325
7 (M)	8	2504	563	1941
Total	51	12049	3299	8750

TABLE IV
Best classification results for each subject in the app field trial.

Subject	Best Classifier		Precision	Recall	F1-score
1	RF	0	0,95	0.977	0.986
		1	0.941	0.988	0.964
		avg/tot	0.981	0.98	0.98
2	SVM	0	0.982	0.977	0.979
		1	0.939	0.951	0.945
		avg/tot	0.97	0.97	0.97
3	RF	0	0.869	0.969	0.916
		1	0.932	0.745	0.828
		avg/tot	0.892	0.887	0.884
4	KNN	0	0.665	0.934	0.777
		1	0.48	0.115	0.186
		avg/tot	0.601	0.65	0.572
5	SVM	0	1.00	0.719	0.837
		1	0.46	1.00	0.63
		avg/tot	0.896	0.773	0.797
6	SVM	0	0.828	0.959	0.889
		1	0.931	0.736	0.823
		avg/tot	0.873	0.863	0.86
7	RF	0	0.755	1	0.861
		1	1	0.28	0.437
		avg/tot	0.831	0.777	0.729

Thanks to the execution of the data and GT collection stage at the beginning of the app timeline, the automatic recognition of the sleep/awake status performed during the corresponding stage is tailored to the specific subject using the app. In fact, the winning classifier is selected after training and testing on a dataset that belongs to the same subject, and not on the basis of a generic model. Additionally, the forced regeneration of the classifier model after 30 days of automatic recognition allows to take into account changes in the subject's habits that may have happened meanwhile. This way the application is able to self-adapt to the user's profile and to provide more reliable results. The use of the MC allows to artificially remove evaluation errors due to inconsistent data generated by sensors. This is quite common especially with the accelerometer. Finally, according to the users' subjective evaluation, the application is nonintrusive and does not affect their habits. The increase in battery consumption is negligible.

VI. Conclusion

Among the wide range of smartphone-based solutions for sleep monitoring, we propose an application able to classify sleep and wakefulness in real time, without the need for any external sensor, and with minimal requested user interaction. In fact, the user's feedback on the ground truth status (sleep or awake) is requested only during the initial stage of data collection and classifier model creation. The proposed methodology exploits environmental and smartphone usage information to determine whether the user is sleeping. The selected set of features has been tested using KNN, DT, RF, and SVM classifiers. Best performances are achieved with the RF classifier, reaching an F1-score of 0.93 in offline testing, and up to 0.98 in a real field trial. The automatic sleep recognition results are further improved by introducing an error correction process based on a 1st-order Markov Chain, that is able to compensate for inconsistent sensor data acquisitions [21].

The implementation of the methodology presented in the paper as an Android mobile app has requested a number of programming solutions to adapt the desired functionalities to the available O.S. primitives. In any case, the resulting app is nonintrusive and does not affect the smartphone performances in terms of battery and memory consumption.

References

[1] M. Hirshkowitz, K. Whiton, S. M. Albert, C. Alessi, O. Bruni, L. DonCarlos, N. Hazen, J. Herman, E. S. Katz, L. Kheirandish-Gozal *et al.*, "National sleep foundation's sleep time duration recommendations: methodology and results summary," *Sleep Health: Journal of the National Sleep Foundation*, vol. 1, no. 1, pp. 40–43, 2015, doi:10.1016/j.sleh.2014.12.010.

[2] R. A. Hicks and R. J. Pellegrini, "The changing sleep habits of college students," *Perceptual and Motor Skills*, vol. 72, no. 3_suppl, pp. 1106–1106, 1991, doi:10.2466/pms.1991.72.3c.1106.

[3] T. Roenneberg, "Chronobiology: the human sleep project," *Nature*, vol. 498, no. 7455, pp. 427–428, 2013, doi: 10.1038/498427a.

[4] A. A. Ong and M. B. Gillespie, "Overview of smartphone applications for sleep analysis," *World Journal of Otorhinolaryngology-Head and Neck Surgery*, vol. 2, no. 1, pp. 45–49, 2016, doi: 10.1016/j.wjorl.2016.02.001.

[5] C. A. Cmiel, D. M. Karr, D. M. Gasser, L. M. Oliphant, and A. J. Neveau, "Noise control: A nursing teams approach to sleep promotion: Respecting the silence creates a healthier environment for your patients." *AJN The American Journal of Nursing*, vol. 104, no. 2, pp. 40–48, 2004, doi: 10.1097/00000446-200402000-00019.

[6] M. A. Carskadon, W. C. Dement *et al.*, "Normal human sleep: an overview," *Principles and practice of sleep medicine*, vol. 4, pp. 13–23, 2005, doi:10.1016/B0-72-160797-7/50009-4.

[7] W. Liu, B. Ploderer, and T. Hoang, "In bed with technology: challenges and opportunities for sleep tracking," in *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*. ACM, 2015, pp. 142–151, doi:10.1145/2838739.2838742.

[8] A. Campbell and T. Choudhury, "From smart to cognitive phones," *IEEE Pervasive Computing*, vol. 11, no. 3, pp. 7–11, 2012, doi:10.1109/MPRV.2012.41.

[9] A. Cuttone, P. Bækgaard, V. Sekara, H. Jonsson, J. E. Larsen, and S. Lehmann, "Sensiblesleep: a bayesian model for learning sleep patterns from smartphone events," *PloS one*, vol. 12, no. 1, p. e0169901, 2017, doi:10.1371/journal.pone.0169901.

[10] K. Wac, "Smartphone as a personal, pervasive health informatics services platform: literature review," *arXiv preprint arXiv:1310.7965*, 2013, doi:10.1055/s-0038-1639436.

[11] H. Shin and J. Cho, "Unconstrained snoring detection using a smartphone during ordinary sleep," *Biomedical engineering online*, vol. 13, no. 1, p. 116, 2014, doi:10.1186/1475-925X-13-116.

[12] S. Abdullah, M. Matthews, E. L. Murnane, G. Gay, and T. Choudhury, "Towards circadian computing: early to bed and early to rise makes some of us unhealthy and sleep deprived," in *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*. ACM, 2014, pp. 673–684, doi:10.1145/2632048.2632100.

[13] Urbandroid Team, "Sleep as android," https://play.google.com/store/apps/details?id=com.urbandroid.sleep, online; accessed March 31st, 2018.

[14] FleiCOM, "Sleep detector," https://play.google.com/store/apps/details?id=fleicom.app&hl=it, online; accessed March 31st, 2018.

[15] SAMSUNG, "Samsung health," https://www.samsung.com/us/samsung-health.

[16] Under Armour, Inc., "Under armour record," https://itunes.apple.com/us/app/record-by-under-armour/id895425891?mt=8, online; accessed March 31st, 2018.

[17] Northcube AB, "Sleep cycle," https://itunes.apple.com/it/app/sleep-cycle-alarm-clock/id320606217?mt=8, online; accessed March 31st, 2018.

[18] L. Montanini, N. Sabino, S. Spinsante, and E. Gambi, "Smartphone as unobtrusive sensor for real-time sleep recognition," in *Proceedings of the 2018 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2018, doi:10.1109/ICCE.2018.8326220.

[19] Z. Chen, M. Lin, F. Chen, N. D. Lane, G. Cardone, R. Wang, T. Li, Y. Chen, T. Choudhury, and A. T. Campbell, "Unobtrusive sleep monitoring using smartphones," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2013 7th International Conference on*. IEEE, 2013, pp. 145–152, doi:10.4108/icst.pervasivehealth.2013.252148.

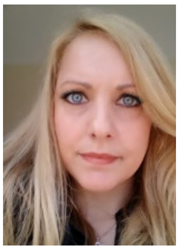
[20] K. S.-W. Hu Li-Yu, Huang Min-Wei and T. Chih-Fong, "The distance function effect on k-nearest neighbor classification for medical datasets," *SpringerPlus*, vol. 5, no. 1, p. 1304, 2016, doi:10.1186/s40064-016-2941-7.

[21] T. Liu, "Application of markov chains to analyze and predict the time series," *Modern Applied Science*, vol. 4, no. 5, p. 162, 2010, doi:10.5539/mas.v4n5p162.

Ennio Gambi received the Electronic Engineering MD at the Universita' Politecnica delle Marche in 1986 and the Master of Microwave Engineering in 1989. Since 1992 he joined the Universita' Politecnica delle Marche in Ancona, where he is, currently, an Associate Professor. His present research interests are on smart systems for Ambient Assisted Living and Enhanced Living Environments, and on spread spectrum solutions in wireless communications.

Simone Barbetta was born on 1996 in Pescara, where attended the Scientific and Technological Course of High School Marconi. He graduated at the Universita' Politecnica delle Marche on a first level degree in Computer Science and he is starting a Master Degree on Artificial Intelligence.

Adelmo De Santis has been working at Universita' Politecnica delle Marche since 2001 as a member of technical staff, system and network administrator. He received his MD in 2006 and PhD in 2015. He is Huawei Lecturer for HCNA Routing and Switching since 2017. His research and interests fields are millimeters wave, SDR, computer networks and RF measurement systems.



Manola Ricciuti received the Bachelor Degree in Electronic Engineering in July 2009, then she obtained the Master Degree in Electronic Engineering in July 2015 with Summa cum laude. Since November 2017 she is a Ph.D. student in Telecommunications at the Department of Information Engineering, Universita' Politecnica delle Marche. Her research interests are focused on signal processing and computer vision techniques for Ambient Assisted Living applications.