SleepLess: Personalized Sleep Monitoring Using Smartphones and Semi-supervised Learning

Priyanka Mary Mammen[1*], Camellia Zakaria[2] and Prashant Shenoy[1]

[1]University of Massachusetts Amherst, USA.
[2]University of Toronto, Canada.

*Corresponding author(s). E-mail(s): pmammen@cs.umass.edu;
Contributing authors: camellia.zakaria@utoronto.ca;
shenoy@cs.umass.edu;

Abstract

Purpose: Sleep affects our bodily functions and is critical in promoting every individual's well-being. To that end, sleep health monitoring research has gained interest recently, including coupling data-driven AI techniques with mHealth adaptations of wearable, smartphone, and contactless-sensing modalities. Regardless, prior works, by and large, require gathering sufficient ground truth data to develop personalized and highly accurate sleep prediction models. This requirement inherently presents a challenge of such models underperforming when inferring sleep on new users without labeled data.
Methods: In this paper, we propose *SleepLess*, which uses a semi-supervised learning pipeline over unlabeled data sensed from the user's smartphone network activity to develop personalized models and detect their sleep duration for the night. Specifically, it uses a pre-trained model on an existing set of users to produce pseudo labels for unlabeled data of a new user and achieves personalization by fine-tuning over selectively picking the pseudo labels.
Results: Our IRB-approved user study found *SleepLess* model yielding around 96% accuracy, between 12-27 minutes of sleep time error and 18-25 minutes of wake time error. Comparison against other approaches that sought to predict with fewer labeled data found *SleepLess*, similarly yielding best performance.
Conclusion: Our study demonstrates the feasibility of achieving personalized sleep prediction models by utilizing unlabeled data extracted from network activity of users' smartphones, using a semi-supervised approach.

Keywords: Semi-supervised Learning, Time-Series Analysis, Sleep, Passive Sensing

1 Introduction

Sleep is an essential daily human activity that significantly affects a person's health and well-being. Despite its importance, sleep disorder is common among adults, with prior studies reporting 20-40% adults suffering from a form of sleep disorder [1]. Sleep deprivation is a widespread problem, with a third of the population getting less sleep than the recommended 8 hours of regular sleep [2]. Since poor sleep hygiene can influence various health problems, sleep monitoring has become a critical technology enabler for researchers and clinicians to understand daily sleep habits better and identify poor sleep health.

Wearable sleep trackers such as FitBit [3] and Oura ring[4] have become popular for users to keep track of their daily sleep in recent years. Although they are simple to use, these contact-based methods may be less favorable among users who prefer not to wear a device during their sleep time [5]. To overcome this challenge, researchers have responded by developing several contactless solutions. For example, radar-based approaches [5], use radio frequency signals that bounce off users to monitor their breathing and sleep. This technology, in particular, is adopted by smart speakers such as Google Nest [6] and Amazon Alexa [7] for contactless sleep tracking using a built-in radar [8]. While wearable and smart speakers can monitor sleep duration and quality, smartphones are more ubiquitous. Researchers have attempted to leverage the ubiquity of smartphones as an inexpensive means of tracking users' sleep. The primary approach of such solutions is based on indirect sensing, where passive observations of smartphone activities are used to infer a user's sleep duration. An early work [9], which utilizes smartphone screen activity as a proxy of their awake states, correspondingly estimating sleep based on users' inactivity. More recent work has successfully generalized this notion to utilizing network activity generated by smartphones and smart devices where long periods of inactivity were used to detect sleep periods [10]. A general drawback of these solutions is the fundamental need to collect labeled ground truth data from users for training prediction models that will accurately infer their sleep. Due to a lot of user-related issues, such as inaccurate data logging, missing data, and eventually, user attrition, conducting long-term user studies to specifically collect large amounts of ground truth data is challenging [11]. Consequently, many research efforts to study sleep have been limited to a small sample population of tens of users.

Designing models that generalize to a larger population using a small sample size and a small amount of labeled ground truth data is challenging – especially since sleep patterns can vary from one user to another. In contrast to labeled data that is difficult to collect via user studies, unlabeled data of a phone's network or screen activity is significantly easier to collect via automated apps; neither of these data sources require user involvement during data collection. Similarly, WiFi networks routinely logs client activities which can be used to infer a phone's network activity over time [12]. *The convenient availability of unlabeled data together coupled with the challenges gathering labeled data motivates the need to develop semi-supervised learning (SSL) methods* that can monitor a user's sleep using passive observations of the phone's network activity. Further, such SSL methods should enable personalization of the learnt model in order to handle each user's sleep patterns.

In this paper, we present *SleepLess*, a system that uses a semi-supervised learning pipeline over unlabeled phone activity data to develop personalized models for detecting a user's daily sleep patterns. In designing and evaluating *SleepLess*, we make the following contributions:

1. We propose a semi-supervised training pipeline to enable personalized sleep duration estimation in users from the network activity of their mobile-phones. We use a teacher-student framework to utilize a pre-trained sleep prediction model and a few weeks of unlabeled data from the end-user.
2. We implement a complete prototype of a semi-supervised learning pipeline to demonstrate the efficacy of our approach. We conduct a user study on a campus consisting of 20 users. Further, we present a case study to demonstrate the generalizability of the approach in residential settings.
3. The model validations show that our approach achieves around 96% accuracy, between 12-27 minutes of sleep time error and 18-25 minutes of wake time error.

2 Background

This section discusses prior work on phone-based sleep sensing and motivates the need for semi-supervised learning approaches.

2.1 Detecting Sleep using Smartphones

The use of smartphones for detecting sleep periods of their users has seen over a decade of research. The ubiquity of smartphones along with their numerous built-in sensors, make it a choice to augment or serve as an alternative to wearables for monitoring sleep periods. Much of the work done over the past decade is based on a simple premise - a user activity when awake correlates with phone activities, and consequently, a lack of user activity when asleep manifests as a lack of phone activity. Accordingly, studies have shown that most users sleep with their smartphones, with most using smartphones as an alarm clock and checking their phones as soon as they wake up. This premise has been validated by monitoring phone activity through various means, such as screen activity on a phone, motion sensed by the phone's motion sensors, and change in environment conditions sensed by light and microphone sensors. An early effort from Cornell University [13] was one of the first to establish that passive monitoring of a phone's screen activity can be used to detect sleep periods. Subsequent studies established the use of motion sensors, silence, and ambient light conditions to detect sleep. This notion gained mainstream adoption recently when Google incorporated this idea into all Android phones using Andriod's Activity Recognition API and Sleep API [14]. This functionality enables lack of motion or other activities to be used to determine a daily sleep segment, which is made available through the sleep API. Third-party applications like AutoSleep [15] and Rise [16] use similar concepts on iOS to detect sleep solely using phone activities or in conjunction with wearables when available. Thus, all modern smartphones are now capable of using local activities to detect sleep periods.

2.2 Detecting Sleep using Network Activity

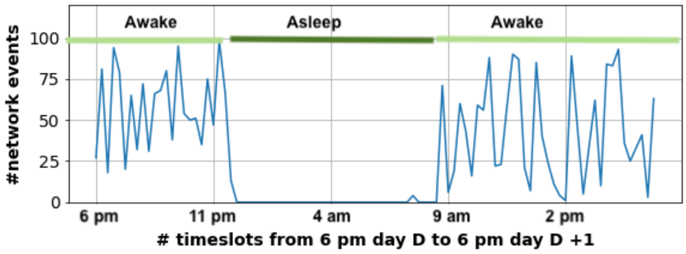

Fig. 1: Time-series data of network activity of mobile phones representing sleep and wake-up periods.

Our work draws inspiration from the above literature and uses phone network activity to detect sleep periods. The same premise holds in this case - when a user is awake, they use their phone from time to time, which generates WiFi network activity; when asleep, there is little WiFi activity, barring notifications or background tasks. Importantly, the level of a phone's network activity can be monitored on the phone itself (similar to prior work), or it can be monitored externally by the WiFi network since WiFi traffic is visible to the network. Thus, unlike prior work, the use of network activity as a sensing modality allows individual monitoring (on the device) or monitoring of a larger group or population (when done at the network level). At the same time, observing the network activity does not expose any sensitive private information, thus making it a safe choice for population scale monitoring. Therefore network activity based sleep monitoring can be used to complement clinical sleep monitoring by providing comprehensive insights into long-term sleep trends. In essence, our choice of sensing modality not only capitalizes on the ubiquity of smartphones but also provides an avenue for unobtrusive, safe, and scalable sleep trend analysis that can be of significance for both individual users and a group of users. Fig 1 shows a sample trace of the network activity of a smartphone, and the lack of night activity is clearly visible, which correlates to the user's sleep. Using two user studies, we validated our premise of using network activity to detect sleep periods. We obtained institute IRB approval and participants' consent before monitoring their devices' network activity. User study details are summarized in Table 1. The results are summarized in our prior efforts and are briefly summarized below. SleepMore[17] involved a user study of 46 users, conducted in collaboration with NUS School of Medicine, using smartphone WiFi activity and OuraRing as ground truth. This study showed that smartphones can provide sleep detection accuracy that approaches wearables (refer Fig 2 and Table 2). Separately, our WiSleep [18] work involved a user study of 23 users, with a mix of students from the UMass campus and non-students, and also validated the use of WiFi activity as a method for monitoring sleep patterns in large groups of users (i.e., population-scale monitoring).

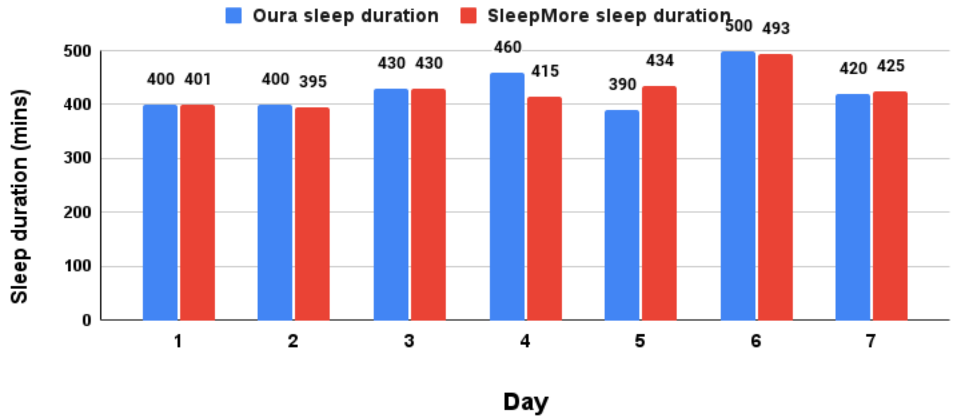

Fig. 2: SleepMore [10] achieves comparable sleep duration detection accuracy with Oura-ring for a user over a period of one week.

	UMass study	NUS study	home users
Users, N	20 (18 Male, 2 Female)	46 (23 Male, 23 Female)	3 (2 Male, 1 Female)
Age	18 - 21 years old mean: 20, stdev.: 0.75	20-28 years old mean: 21.95, stdev.: 1.43	36-46 years old mean: 42, stdev.: 4.71
Study duration	4 weeks	4 weeks	1-4 weeks
Sleep Tracker	Fitbit Inspire HR and manual logs	Oura-ring and manual logs	Fitbit Inspire HR and manual logs
Network activity logging (type)	Syslog	RTLS log	Syslog

Table 1: Our prior work [10, 18] conducted user studies on a range of users to demonstrate sleep detection using network activity.

Dataset	Sleep Prediction technique	Sleep,wake-up time error
UMass	unsupervised	51, 36 minutes
In-home	unsupervised	70, 42 minutes
NUS	supervised	42, 37 minutes

Table 2: Summary results from our prior work [10, 18] validating sleep detection using network activity using an unsupervised and a supervised approach without personalization.

Applications: We establish the premise of the paper on sleep inference using the network activity of smartphones due to several advantages we discuss in this section and section 4.1. Although our proposed sleep inference technique focuses on network activity, its applicability extends to various phone-related events. The integration of network activity-based sleep sensing holds promise for well-being monitoring applications, particularly within college campuses. A critical aspect here is that we need user involvement to enable individual-level monitoring that requires mapping their device

IDs - safeguarding user privacy. Thus, by leveraging cost-effective and easily accessible data sources, we can extend the benefits of network-based sleep monitoring to a broader scale.

Practical Considerations: Our prior works demonstrate the impact of network-related issues such as missing data/network absence, ping-pong events, and background noise. We address the issue of days with network absence/missing data and ping-ping events at the data pre-processing stage. Nevertheless, days with severe background noise could impact the prediction accuracy. Similarly, behavior-related issues such as delayed sleep onset of users after phone usage and delayed phone usage after wake-up could also lead to prediction inaccuracies. In our prior work using a supervised approach [17], we ensure the robustness of sleep predictions by using uncertainty quantification approaches, filtering out predictions with less confidence. Overall, the benefits of using network events as a low-cost sleep monitoring solution for population scale well-being analytics outweigh the shortcomings.

2.3 Need for Semi-Supervised Learning

To motivate the need for our approach, consider a supervised learning approach. We use a CNN-based classifier as it can effectively capture the temporal dependencies in the data and learn relevant features automatically. Given a WiFi network activity trace of a smartphone with the activity rates reported in 15 minutes intervals, we can train an supervised learning classifier to determine whether the user is asleep or awake in each interval based on the observed activity. With some smoothing function to remove noise predictions, the longest sequence of sleep states can be regarded as the nocturnal sleep period. Figure 3 shows the accuracy of a supervised learning approach when trained with 3 weeks of data from 10 users and then tested on one week of new data for the same users. The model can predict sleep duration with an average of 33 minutes of sleep and average of 27 minutes of wake time errors, yielding a 94% accurate model.

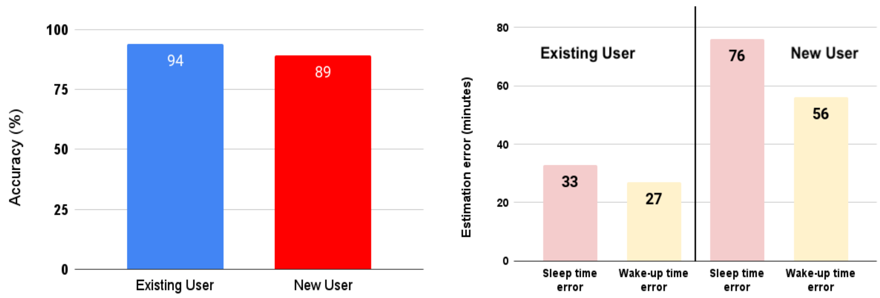

Fig. 3: Model performance using supervised learning comparing the sleep prediction on existing users and new users.

However, since the model is trained on a small user dataset, we expect that it will not generalize well to users who are not in the training set. To verify this hypothesis, we consider ten different users and examine the efficacy of the model for these new users. As expected, the model error increases from 33 to 76 minutes of sleep and 27 to 56 minutes of wake time errors. Note that these ten users are part of our user study, described in Section 5.

In scenarios where a model does not generalize to new users, transfer learning can be used to personalize the model with a small amount of training data. In this case, we freeze the earlier layers of our pre-trained model and add a small amount (e.g., 14 days) of labeled sleep data of the new user to develop a newly-personalized model. Figure 4 shows the mean accuracy of ten new users, where personalization using transfer learning helps boost model accuracy thus reducing sleep and wake errors by 45 minutes on average. This result is comparable to that of the initial model (see Figure 3 - existing user).

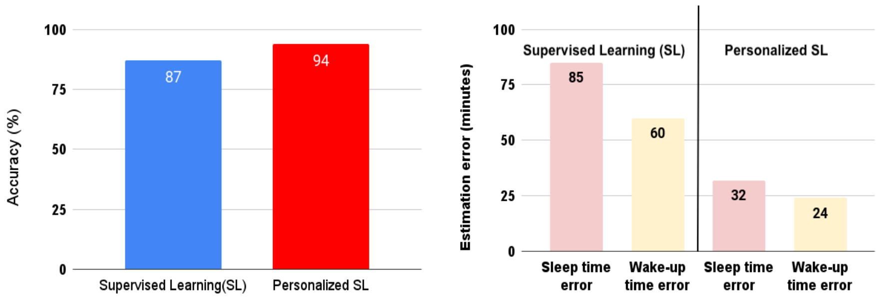

Fig. 4: Model performance comparing generalized supervised learning and personalization on new users.

While transfer learning provides a path to improve accuracy for each new user, this technique still requires labeled data for personalization of model. In our case of predicting sleep, acquiring unlabeled data from new users is much easier compared to obtaining ground truth of their daily sleep. Semi-supervised learning (SSL) can overcome this specific challenge of requiring labeled data. SSL methods are particularly attractive when combining unlabeled data with relatively small amounts of labeled data. We can utilize this unlabeled data along with labeled data of roughly similar characteristics because of smoothness assumption that data points closer in a sample space have similar samples. Since the model does not generalize well and personalization requires sufficient labeled data, the motivation of work builds on the need to use unlabeled data using SSL methods.

3 *SleepLess* Design

This section presents the design of our system, *SleepLess*, beginning with its problem formulation.

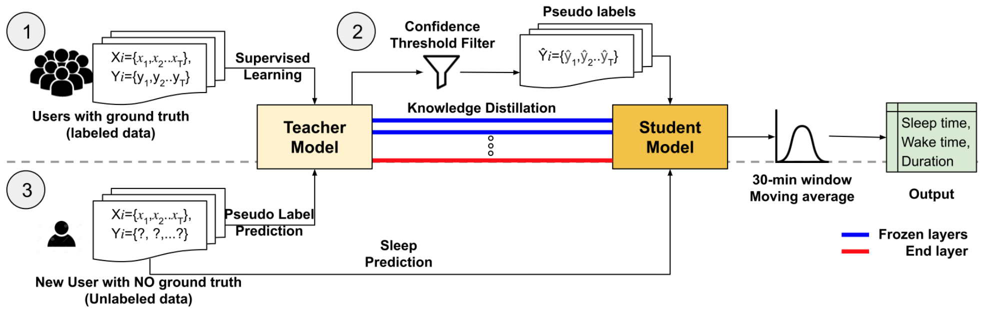

Fig. 5: *SleepLess*'s approach is a three-step process of training a Teacher model, generating high-quality pseudo labels from unlabeled data, and personalizing a Student model for a new user.

3.1 Problem Statement

The goal of our work is to develop personalized sleep detection models for each user based on unlabeled activity data from their smartphones. Our work assumes that a small amount of labeled data is available from a small group of users, which can be used for initial training. Additionally, we assume that only unlabeled data of new users is available, but their model needs to be personalized. We seek to design a semi-supervised learning approach to personalize an existing model, pre-trained on other sets of users, solely using unlabeled smartphone data for the new user. For the purpose of this paper, we consider utilizing coarse-grained WiFi activity data generated by users' smartphones as a measure of their phone activity[1]. Formally, we model this problem as a multivariate time series classification problem:

Users with Labeled Data: Consider $X^i = \{x_1, x_2 .... x_T\}$, which represents the multivariate time series of phone activity features for user, i, where $x_j = \{f_1, f_2 .... f_n\}$ is a vector of phone activity features at time, j. The features, $f_1, f_2 .... f_n$, in our case, are WiFi network activity features generated by user i's smartphone; for example, the number of observed WiFi events, number of WiFi access points connected by the phone. Collectively, these features represent the level of a phone's network activity. We assume time is discretized into fixed length intervals (i.e., 15-minutes) and these activity features are computed for each interval.

We assume a small group of users whose ground truth sleep information is available. This information includes the user's sleep duration, sleep time, T_{sleep}, and wake time T_{wake}. The ground truth yields a labeled time series for each user, i, where $Y^i = \{y_1, y_2 .... y_T\}$ and the label for each interval j is $y_j \in \{0, 1\}$. A label of 1 denotes the user as asleep, conversely, 0 denotes the user as being awake. All intervals between T_{sleep} and T_{wake} gets a label of 1.

[1]Our approach can be applied to other types of phone activity data such as screen activity. Here, phone activity data is represented by network activity rate.

New Users with Unlabeled Data: We assume a much larger group of users, whose phone activity data X^i is available but no labeled ground truth Y^i is known. In this case, the time series X^i simply represents unlabeled activity data for the user.

Henceforth, the problem is to train an (initially) supervised model on users with labeled data and personalize this model for each new user with only using their unlabeled data.

3.2 *SleepLess* Approach

Our approach to addressing the above problem involves three key steps, depicted in Figure 5.

3.2.1 Step 1: Train a Teacher Model

SleepLess first users the set of users with ground truth data to train an initial CNN-based deep learning model. We refer to this initial model as the teacher model, $Model_{Teacher}$. $Model_{Teacher}$, discussed further in Section 4.3, uses a cross-entropy loss function defined as:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i.log(p(y_i)) + (1-y_i).log(1-p(y_i)) \quad (1)$$

where y is the sleep or awake label.

In solving for a binary classification problem, prior work has reported sigmoid value as a confidence metric reliable despite overconfident predictions arising from unseen classes [19]. However, due to the imbalanced nature of the dataset where there are more awake labels than sleep, the model may be biased towards predicting more prevalent label (i.e., awake). To address this issue, we calibrate our models by leveraging a validation set to improve the accuracy of our prediction probabilities.

Note that $Model_{Teacher}$ is trained to perform a binary classification task by taking the phones activity features X_j and predict whether the user is awake or asleep. The longest sequence of sleep labels over the course of each 24-hour represent the sleep period for that day.

3.2.2 Step 2: Obtain Pseudo Labels from the Teacher Model

Given the teacher model, $Model_{Teacher}$, *SleepLess* then considers a new user, k, whose phone activity data is not accompanied with their sleep ground truth. It uses the time series of phone activity features, $X^k = \{x_1, x_2....x_T\}$, where x_T represents each time step, into $Model_{Teacher}$ to predict whether user k is asleep or awake. The output generated by $Model_{Teacher}$ constitute *pseudo labels* for the user.

It is possible that $Model_{Teacher}$ does not generalize well to the new user, likely due to low-quality pseudo labels. We use Dropout in the prediction network to improve the reliability of pseudo labels by reducing the effects of overfitting and improving the robustness of the model's predictions. Dropout works by randomly dropping out some neurons in the network during each training iteration. We also calibrate our models using a validation set. *SleepLess* performs label selection to only retain output predictions of high confidence, discarding all others. We use a confidence threshold, Δ, retaining predictions above this value as pseudo labels for the next phase. The Confidence score is chosen based on the average softmax scores of all predicted outcomes in a given 24-hour period as follows:

$$C_{avg}^{i} = \frac{\sum c_t^i}{T} \tag{2}$$

Thus, for each new user, we obtain pseudo labels $\hat{Y} = \{\hat{y}_1, \hat{y}_2....\hat{y}_T\}$. Not all time slots will have such labels but since it is easy to collect unlabeled data, this process can continue until adequate pseudo labels are generated. Our experiment in Section 5.2.1 will show the impact of the amount of pseudo labels *SleepLess* leverages to personalize the model a new user.

3.2.3 Step 3: Personalize a Student Model using Fine-tuning

The pseudo label data for user k can then be used to train a personalized model. Specifically, *SleepLess* performs transfer learning of the original teacher model, $Model_{Teacher}$, by freezing the initial layers of the CNN. The pseudo labels are then used to further train and fine-tune the subsequent layers, thus generating a student model, $Model_{Student}$, now personalized to the new user. This transfer learning approach helps the model re-use the learnt features in earlier layers and tailor the latter layers to specific sleep patterns of the new user. As noted in Section 2.3, such personalization can significantly improve the accuracy for a new user.

This algorithm is described in Algorithm 1.

Algorithm 1 *SleepLess* algorithm

Input (1) : Labeled data D from N users.
Input (2) : Unlabeled data D' from new user
Output (1): $Model_{Teacher}$
Output (2): $Model_{Student}$
Training Teacher model
Train $Model_{Teacher}$ using D with the following loss function
$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i.log(p(y_i)) + (1 - y_i).log(1 - p(y_i))$
Pseudo Label Selection:

for (each day i in D') **do**
  Predict labels $\hat{Y}_i = \{\hat{y}_1, \hat{y}_2....\hat{y}_T\}$ and confidence scores $C_i = \{c_1, c_2....c_T\}$
for X_i using $Model_{Teacher}$
  $C_{avg}^{i} = \frac{\sum c_t^i}{T}$
  if $C_{avg}^{i} > \Delta$ **then**
    $\mathcal{P}.add([X_i, \hat{Y}_i])$
  end if
end for

4 *SleepLess* Implementation

We implemented *SleepLess* prototype as a cloud-based service using Python [20] and Keras libraries [21]. In what follows, we elaborate on the steps we took to build the deep learning component of the system.

4.1 Data Pre-Processing

4.1.1 Network Activity Data

In essence, our technique utilizes users' phone activities generated from smartphone devices to predict their sleep. These activities can be represented by various measures including but not limited to the screen activity, application logs, accelerometer, and WiFi network activity logs [22].

Our work builds on utilizing WiFi network activity data for several operational reasons. As discussed in Section 5.1, our study sought to minimize user burdens and maximize privacy by avoiding dedicated app installations on their smartphone and, thus, directly sensing from their device. As such, we use a passive sensing technique where we acquired WiFi network activity data to collect *syslog* data directly from the WiFi APs, bypassing any connection to the user's device. In acquiring these logs, we filter out entries relevant only to our participants, specifically their primary smartphone device.

4.1.2 Data Cleaning

The coarse granularity of WiFi data presents inherent technical errors in the measurement instrument. To maintain the quality of our analysis, we cleaned these logs to reduce inaccurate and noisy data.

Noisy Data: Generally speaking, we can approximate user location based on the user's device connection to a WiFi AP. However, this approximation contributes to noisy data due to the ping-pong effect. Consider a user whose device remains stationary at a location and is within the range of neighboring APs with similar signal strengths. Their device connection may switch back and forth between different APs, causing a spectrum handoff known; this is known as the "ping-pong" effect. The noise from this effect can resemble network activity despite the absence of the user's direct interaction with their smartphone. To reduce noise, we group APs in an area, such as a dorm floor, and filter out patterns that resemble ping-pongs between nearby APs from the event logs.

Missing Data: The primary reliance on collecting logs directly from the WiFi infrastructure implies that users must be actively utilizing the WiFi for us to reasonably associate the records with their presence. Our user study procedure in the following section describes our steps to ensure their active participation. However, it is possible that participants were not on the premise during the entire duration of the study. Our data-cleaning process only retains daily records that consist of at least 75% of WiFi network activity logs over a 24-hour period. This cutoff threshold implies that users must be present either on campus or in their residence for at least 18 hours for the whole day.

4.2 Feature Extraction

SleepLess processes logs of WiFi network activity rates generated from a user's phone. These logs are in the following format:

```
<date> <hh:mm:ss> <controller> <event\_ID> <severity>

<AP, MAC and IP addresses> <message text with BSSID and SSID>
```

Timestamp is given by date and time, while WiFi access point(AP) and users' device MAC addresses. Note that our collection of WiFi logs ensures user privacy by hashing users' device MAC addresses. together allow us to approximate the user's location. Event ID particularly describes three events of interest. They are i) association, when a device connects to an access point , ii) dissociation, when a device disconnects from the access point, and iii) authentication of the device, thus allowing us to approximate user activity and movement from one place to another. The result of using a *timestamp*, *event_ID*, *WiFi AP address*, and user device (hashed) *MAC address* is four input features to predict the user's sleep.

Fig. 6: Feature correlation

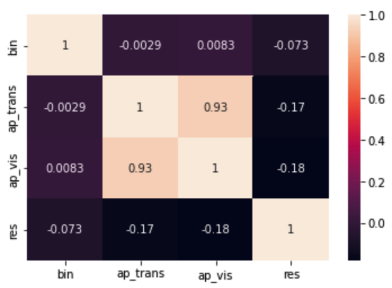

Table 3: Features used

Weight	Feature
0.647	Time
0.107	#WiFi AP Connections
0.132	#WiFi AP transitions
0.098	Dorm or not

- The *time of day* is marked in bins of every 15 minutes. Since we are interested in the nocturnal sleeping period, we consider a 24-hour time span that starts from 6 pm of Day_1 and ends at 5:45 pm the next day. 6 pm of Day_1 corresponds to bin 0, while 5:45 pm of Day_2 corresponds to bin 95.
- The *number of WiFi AP connections* denotes the total number of unique access points visited over every 15 minutes interval.
- The *number of WiFi AP transitions* denotes the total number of transitions approximated from WiFi AP switching over every 15 minutes interval.
- We categorize *Dorm or not* as a user in their residential or non-residential location. This assignment is based on mapping WiFi APs specific to our campus and campus housing.

Figure 6 shows the correlation map of our features. Then, we compute feature importance through the permutation importance method. That is, we recursively measure the model performance every time the values of a feature are randomly shuffled. Table 3 summarizes the values of the most important features in our model. In this case, our top two features represent the time at which the user's device generate high network activity rate.

4.3 Model Architecture

In developing the teacher model, $Model_{Teacher}$, we extract features from the time series data of a fixed set of users into bins of 15 minutes intervals and include label assignments of *sleep (1)* or *awake (0)* state corresponding to their supplied ground truth. Conversely, $Model_{Student}$ is accompanied by pseudo labels of sleep.

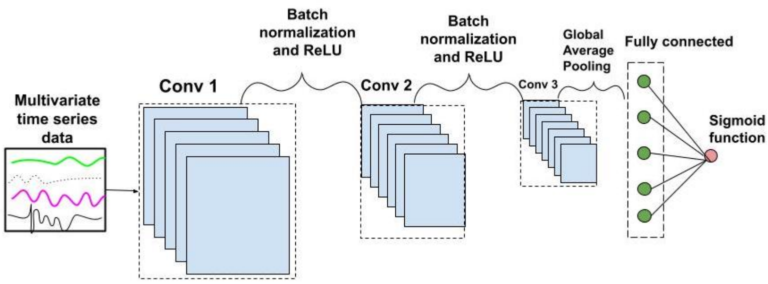

Fig. 7: CNN architecture for our Teacher-Student models.

We utilize a basic CNN architecture, depicted in Figure 7. Specifically, the model consists of three temporal convolution layers with a filter size of 32, 64, and 96. Each layer has a kernel size of 24, 16, and 8, respectively. We chose a uniform stride of 1 for all the layers. The ReLU function was chosen as the activation function. We also chose a dropout rate of 0.1 between layers, a global 1D maximum pooling layer at the last convolutional layer. Finally, we used Adam optimizer with a learning rate of 0.0003.

$Model_{Teacher}$ was trained for 20 epochs, while $Model_{Student}$ was trained for 10 epochs. In fine-tuning $Model_{Student}$, we froze the first two convolutional layers of the CNN to ensure that some knowledge from the earlier training is retained.

5 Evaluation of *SleepLess*

In this section, we evaluate the efficacy of *SleepLess*'s semi-supervised learning model with other prediction methods employed in similar prior work. Further, we evaluate its robustness on private home users with different phone activity profiles.

5.1 Experimental Setup

We begin by describing our user study details in acquiring two different datasets, as summarized in Table 4.

	Student dataset (main)	**Private home dataset** (supplementary)
Users, N	20 (18 Male, 2 Female)	3 (2 Male, 1 Female)
Age	18 - 21 years old mean: 20, stdev.: 0.75	36-46 years old mean: 42, stdev.: 4.71
Study duration	4 weeks	1 week - 4 weeks
Sleep summary	Bedtime: 06:00 pm - 11:00 am (mean: 01:20 am), Wake time: 03:00 am - 03:00 pm (mean: 10:10 am), Sleep duration: 60 - 660 mins (mean: 420 mins)	Bedtime: 11:20 pm - 12:45 am (mean: 11:27 pm), Wake time: 05:30 am - 08:00 am (mean: 06:16 am), Sleep duration: 300 - 511 mins (mean: 428 mins)
Sleep Tracker	Fitbit Inspire HR	Fitbit Inspire HR, Fitbit Versa 3 and manual logs
Device activity	anonymized logs of connected smartphones to campus WiFi.	unanonymized WiFi logs of connected smartphones and home devices to home WiFi

Table 4: Demographic information of two different datasets.

5.1.1 Study Protocol

We ran a month-long user study among college students living on campus upon receiving IRB approval from our institution. Our study protocol includes recruiting undergraduates and giving out Fitbit inspire HR [3] wearable to collect sleep logs automatically. In demonstrating how users' smartphone activities can be used to predict sleep, students simultaneously consented to us collecting WiFi network activity generated from their smartphones directly from the campus WiFi APs. The sleep logs generated from Fitbit are used as ground truth. In practice, WiFi logs generated from the campus APs only contain the timestamp and network activity rate of hashed MAC addresses per device. We specifically isolated our participants' device connection to a dedicated AP to identify our participant's smartphones, despite us dealing with only hashed records. As part of the study, participants must wear their Fitbit to bed every night and connect to the campus WiFi APs at all times.

Separately, we repeated the same protocol to a different set of private home users (non-student) over a one-week period. As each household has a dedicated WiFi set up, our home users directly supplied us with their WiFi network activity logs in the form of a .CSV file. Participants also supplied their Fitbit sleep logs, manual sleep entry logs or both.

5.1.2 Data Ethics and IRB Approval

Our user study is approved by the Institutional Review Board (IRB) and includes a Data Usage Agreement (DUA) with the campus network IT group. As mentioned above, the WiFi logs generated from students' devices on campus do not contain identifiable information. The identifier of their connected devices are anonymized using a strong encryption algorithm.

5.1.3 Evaluation Metric

The standard evaluation metrics used for classification, such as accuracy, F1 score, precision, recall alone, are less suitable in our case due to class imbalance. For example, on a typical day, we can expect approximately 33% of sleep and 77% of awake bins. Thus accuracy and F1 scores will be biased toward classifying awake bins. In evaluating *SleepLess*, we complement our results with sleep time estimation error and wake-up time estimation error, represented in minutes as per Equations 3 and 4.

$$T^{err}_{sleep} = |T^{true}_{sleep} - T^{est}_{sleep}| \tag{3}$$

$$T^{err}_{wake-up} = |T^{true}_{wake-up} - T^{est}_{wake-up}| \tag{4}$$

5.2 Efficacy of SSL-based Model

Our first experiment examines how *SleepLess* performs compared to a personalized supervised learning approach, which would require re-training a small amount of labeled data for a completely new user.

We utilize the student dataset in this experiment. Our results are achieved through conducting a train-test split. Specifically, models are trained on ten randomly picked student users and tested on the remaining ten users. Additionally, we set aside the first two weeks of labeled data of our test users to develop their personalized model and used the last two weeks to test the model.

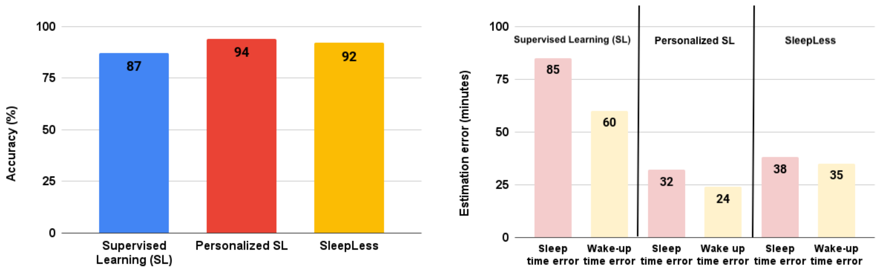

Fig. 8: Model performance comparing general and personalized supervised learning approaches with semi-supervised learning that *SleepLess* employs.

As shown in Figure 8, the model performance for *SleepLess* is comparable to that of a personalized model. Specifically, the sleep and wake times prediction errors by *SleepLess* are approximately 10 minutes more than a personalized model. As expected, personalization will yield better model performance for a new user (94% accuracy, 32 minutes sleep time prediction error and 24 minutes wake-up time prediction error). In contrast, *SleepLess* achieves 92% accuracy, 38 minutes sleep time prediction error and 35 minutes wake-up time prediction error.

Method	T_{sleep} mins	T_{wake} mins
Personalized SL	8± 5	23± 20
SleepLess	12± 4	25± 20

Table 5: Personalized supervised learning versus semi-supervised learning on home users.

We replicated this experiment on our small group of non-student users who reside in private homes, comparing a personalized supervised learning approach with *SleepLess*. As indicated in Table 4 of our user study, the device network activity of our home participants accounted for all the personal devices connected to their home WiFi AP. Our model yielded slightly increased errors in predicted sleep and wake times. Even though *SleepLess* recorded 2% less accuracy, our results remain favorable for two reasons. First, *SleepLess*'s model decrement is insignificant ($p < .01$). Second, personalized supervised learning model will only offer practical use to new users after providing two weeks of labeled data for model retraining. In a real-world application, *SleepLess* can begin prediction for a new user without their labeled data almost instantaneously. These considerations motivate us to explore and propose *SleepLess* as a more practical approach, appealing to new users.

5.2.1 Impact of model parameters

The development of *SleepLess*'s teacher model, illustrated in Figure 5, utilizes data of ten randomly picked student users over the entire 4-weeks of participation. Further, $Model_{Teacher}$ is implemented on the most optimal settings. These settings include the number of users we relied on for training labeled data and the cutoff prediction interval retained to minimize results uncertainty. Our empirical observations below systematically explains how we achieve high performance for $Model_{Student}$, comparable to a personalized one.

Prediction Interval Threshold: As illustrated in Figure 7, we applied a sigmoid activation function to produce the likelihood of a 'sleep' outcome. In this experiment, we varied the threshold from 0.5 to 0.85, where 0.85 denotes the upper limit of all predictions.

Figure 9 charts the cumulative errors for each threshold bin. We observe the lowest errors of 27 minutes sleep time error and 18 minutes wake time error when $Model_{Student}$ only includes predictions with at least 0.7 sigmoid score. By filtering out pseudo labels with lower thresholds, the model selectively builds on samples close to the data distribution of the users in the training data. This cutoff threshold should, however, consider the tradeoff between accuracy and model overfitting.

Impact of Pseudo Labels The reliance on pseudo labels generated by $Model_{Teacher}$ is critical in our approach. Our next experiment examines the amount of pseudo labels needed to fine-tune the $Model_{Student}$. Table 6 compares the impact of pseudo labels on the average sleep and wake time errors in minutes when *SleepLess* predicts the sleep of new users. Generally speaking, *SleepLess* yields decreasing sleep and wake time errors. Adding 21 days of pseudo labels for fine tuning result in errors decreasing to the lowest of 27 $\pm$ 6 minutes sleep time and 18 $\pm$ 5 minutes wake time.

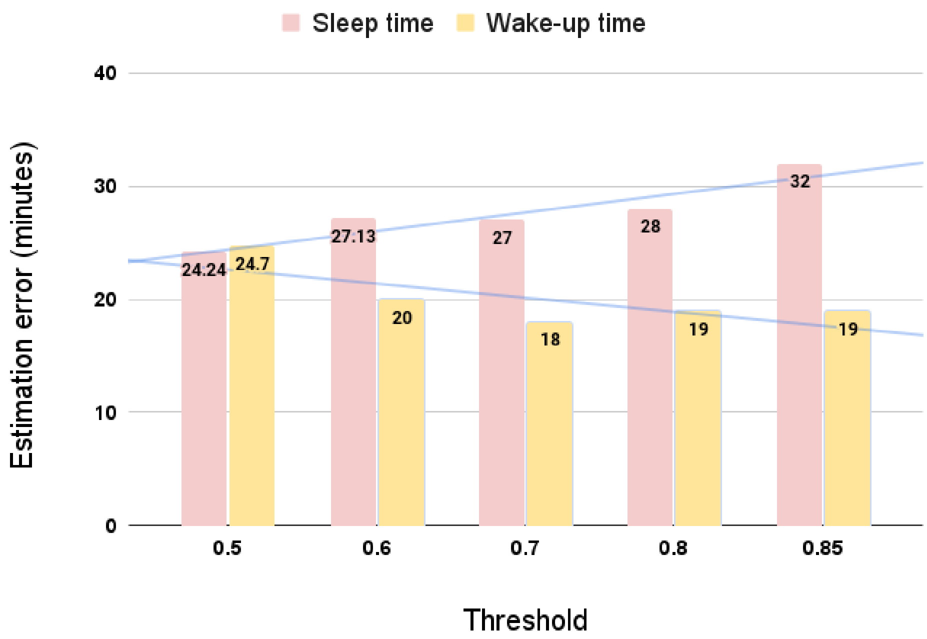

Fig. 9: Sleep and wake time errors after filtering pseudo labels with varying confidence threshold.

Days for Pseudo label	$T_{sleepErr}$ mins	$T_{wakeErr}$ mins
7	48± 5	35± 3
14	39± 6	29± 2
21	27± 6	18± 5

Table 6: Sleep and wake time estimation errors comes into acceptable values with 3 weeks of unlabelled data.

Window Threshold for Moving Average: Up to this point, the outcome from $Model_{Student}$ is in the form of binary labels, $sleep(1)$ and $awake(0)$ states at every 15 minutes interval. Recall in Figure 5, our system employs a moving average to estimate the longest sleep sequence and determine the start and end of a user's sleep and wake time.

Figure 10 charts the line graph of the average sleep and wake time errors on 10 new users when we applied moving average with varying threshold window. We empirically decided on a 30 minutes window size for this technique as our errors dipped to its lowest of 27 minutes sleep time and 18 minutes wake time.

Teacher vs. Student Model: As explained in our approach (Section 3.2), we applied transfer learning of $Model_{Teacher}$ to $Model_{Student}$ by freezing its initial layers and fine-tuning the subsequent layers. This step implies that the layer weights of $Model_{Teacher}$

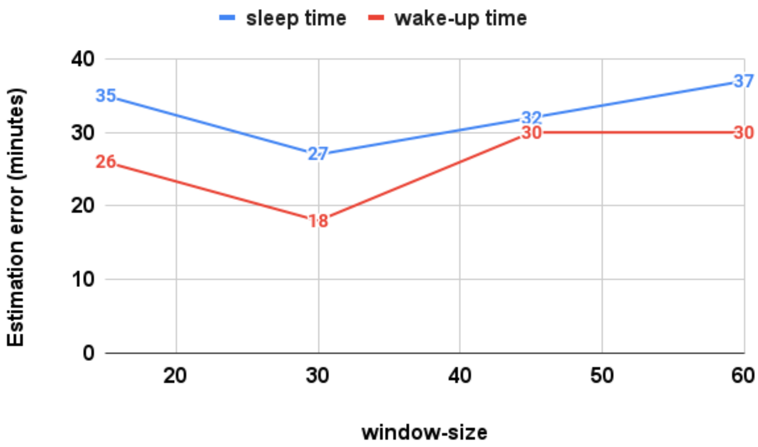

Fig. 10: Sleep and wake time errors with varying moving average window size.

remain unchanged in the subsequent tasks. We expect comparable, if not better, performance in $Model_{Student}$ by further fine-tuning. Figure 11 charts the confusion matrix between both models. In evaluating $Model_{Teacher}$, we conduct a leave-one-user-out validation of our ten randomly selected trained users. Indeed, our results confirm our hypothesis.

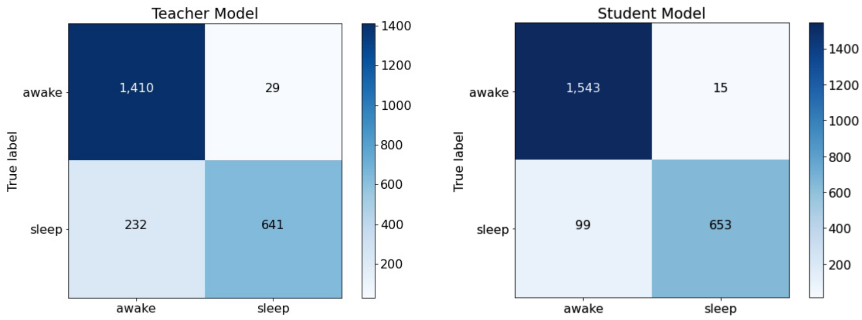

Fig. 11: Similarity between $Model_{Teacher}$ and $Model_{Student}$.

Figure 12 exemplifies the best and worst case prediction outcome for one new user, P1, without providing any labeled data. Note that our user study procedure did, in fact, collect their sleep ground truth for validation. However, their labeled data were not used as part of training $Model_{Student}$.

In the best case predicted outcome, *SleepLess* predicted within 20 minutes sleep time error and 20 minutes wake time error of their ground truth. In contrast, the worst-case predicted outcome happened on another day with less than 10 minutes of sleep time error and more than 120 minutes of wake time error. This error is attributed to the user not using the device immediately upon waking up. However, it remains an outlier for P1. Our analysis found such occurrences happening approximately 5% of the time in a month.

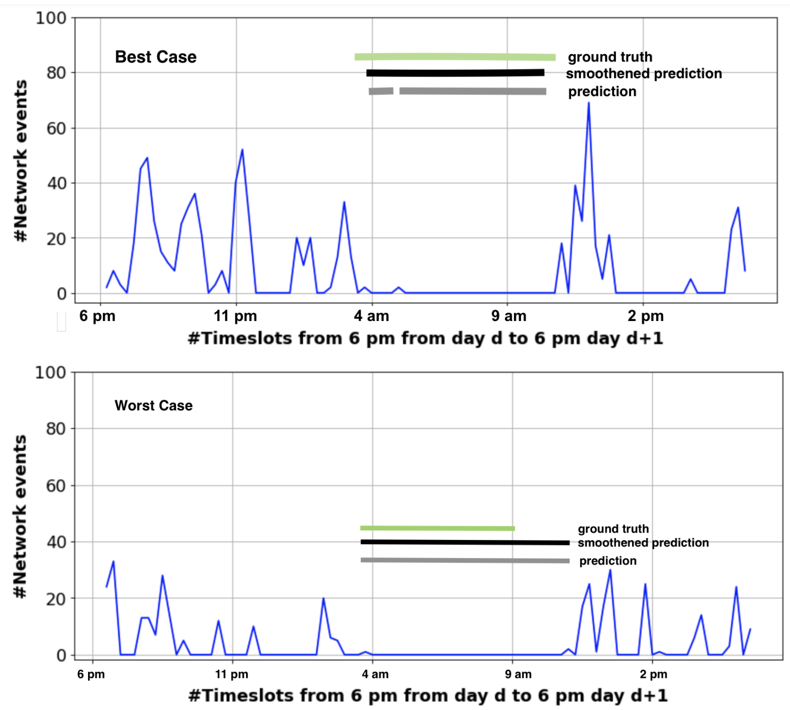

Fig. 12: Best case and worst case model performance on a new user, P1.

Key Takeaway Catering the model to new users without ground truth data is central to our work's motivation. Our approach employs a teacher-student model based on CNN architecture to predict the binary outcomes of sleep (and awake) states using features generated from the users' smartphone network activity rates. Then, it applies a moving average window of 30 minutes to determine the nocturnal bedtime of each user. Evaluating our model on students and home users yielded an average accuracy of 96%, 12-27 minutes of sleep time error, and 18-25 minutes of wake time error. More importantly, the performance of our technique improves with more pseudo labels being trained by the student model over time.

5.3 *SleepLess* vs. Baseline Algorithms

We follow up on the comparison of *SleepLess*'s performance with other baseline approaches proposed in prior work, including why these baselines would work less favorably for our case.

5.3.1 Baseline Algorithms

Given that the primary motivation of our work is to utilize fewer labeled data, we selectively picked on several methods that sought to learn with fewer labels.

The first is *semi-supervised learning using self-training*. Here, we train the teacher model using the labeled data from existing users and generate pseudo labels for the new users using their unlabeled data. The pseudo labels are picked based on the same criterion we used in *SleepLess*. The selected pseudo labels are combined with labeled data, D to train the student model. Compared to the typical self-training process, we stopped the pseudo label selection after one iteration to avoid error accumulation.

The second is *multi-head single-view co-training*. We adopt a similar structure suggested by Chen et al. [23], with several tweaks to the training pipeline. Rather than training multiple classifiers as in multi-view co-training, we will use predictions from multiple classification heads sharing a common module. First, we train the classifier using the labeled data, D, with only one classification head. Second, we generate pseudo labels for the new user using major voting by all the classification heads. Similarly, we filter the pseudo labels using the label selection criteria as per *SleepLess*. Finally, we combine the selected pseudo labels and labeled data to train the personalized model for the new user.

The third baseline employs an *encoder-decoder* approach, which exploits unlabeled data to learn the latent representation of the data [24]. The central idea of this approach is that the unlabeled data and labeled data can together help us select relevant features thus improving model robustness and generalizability. Here, we first combine unlabeled data from the new user to labeled data from existing users and train an encoder-decoder model to learn the latent representation of the new users' unlabeled data. After training the encoder-decoder model, we ingest only the labeled data, D, through the encoder decoder and obtain intermediate output from the encoder. We train the classifier using the encoded representation.

5.3.2 Results

Table 7 compares *SleepLess*'s model performance against the baseline models by testing on 4 weeks of data from 10 users. Specifically, *SleepLess* yields 96% accuracy, which is significantly higher than all other models ($p < .01$). Further, Figure 13 charts the sleep and wake time errors for all methods. We observed that multi-head single-view co-training yielded the least errors among all the baseline methods; 27 minutes of sleep time error and 18 minutes of wake time error. However, this difference remains significantly higher than *SleepLess*. Co-training, which relies on the mix of new and existing user data, can be more suitable in conditions where our goal is to improve a generalized model approach.

Method	Acc	Prec	Rec	F Scr.	p val.
SleepLess- SSL	0.96	0.98	0.87	0.93	-
Self-training	0.91	0.94	0.88	0.86	$p < .01$
Multi-head single-view Co-Training	0.92	0.87	0.88	0.88	$p < .01$
Auto encoder	0.87	0.80	0.76	0.78	$p < .05$

Table 7: *SleepLess* and baseline models performance.

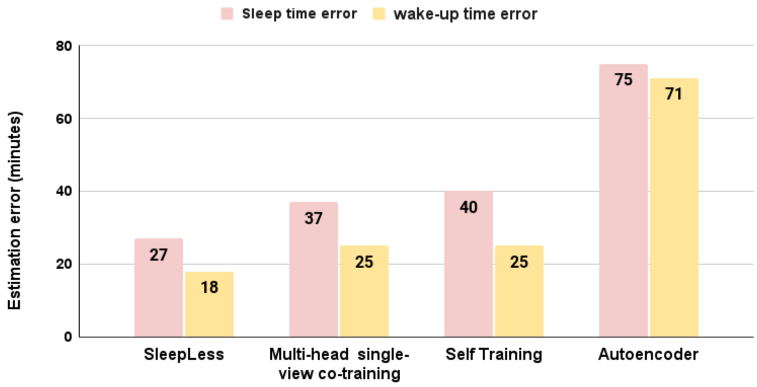

Fig. 13: Sleep and wake time errors by *SleepLess* and baselines.

Key Takeaway Given our proposal on a semi-supervised learning approach, we compared the performance of our technique with standard SSL-based techniques such as self-training and co-training. As briefly discussed, our decision to compare these standard techniques was informed by prior work's implementation for sleep prediction, however, using fine-grained data. In this case, the consideration for a teacher-student model is primarily to address the error accumulation problem, which we hypothesized will be more prominent from using coarse-grained data such as phone network activity rate. Our results yielded significantly better performance of 96% accuracy compared to these standard techniques.

5.4 Summary Findings

While much prior work related to sleep modeling and prediction has been established recently, many works would require collecting labeled data to perform model re-training to achieve user personalization. In contrast, our proposed technique demonstrated the promise of expanding sleep detection capabilities to new users without requiring labeled data. *SleepLess* employs a semi-supervised learning approach that performs transfer learning on a teacher-student model and relies solely on high-quality pseudo labels (above a 0.7 threshold) of new users' unlabeled data as a method of "personalization." This method could be a double-edged sword in that it inherently demands a highly accurate teacher model to produce pseudo labels that can further fine-tune to a new user. Even so, the model would require periodic maintenance of the

Work	Technique	Data Type	User-specific Labels	Validation Period
[25, 26]	Self-Training	EEG,PSG	✓	1 night
[27]	Adversarial Learning	EEG	✓	1 night
[28]	Gaussian Mixture Model	EEG	✓	2 nights
[9, 10, 13, 22, 29–33]	Supervised Learning	Phone activity	✓	1 week - 6 months
[32]	Unsupervised	Phone activity	Not required	1 month
SleepLess	SSL +Transfer Learning	Phone activity	Not required	1 month

Table 8: Summary of prior techniques for sleep monitoring. Note: phone activity comprises of activity inference using phone sensors and or WiFi activity.

student model as prediction errors accumulate over time. Nonetheless, our technique achieves the best performance of 27 minutes sleep time error and 18 minutes wake time error, compared to similar baselines that sought to learn with fewer user labels.

6 Discussions

Our study's objectives were to implement a semi-supervised learning approach to predict the sleep of new users without having them contribute labeled data for model training. Here we discuss the implications of our findings.

6.1 Privacy Implications

Safeguarding user privacy is a key concern in applications for health monitoring, including sleep. Even though our technique uses phone activity data, it ingests features based on network rates, specifically by counting the number of APs and transition rates. We extracted these features without knowing the users' actual locations visited, thus not exposing private information. Our proposed technique involves the design of this teacher-student training method to directly personalize the deep learning model on each user's phone and use it to detect sleep periods. The pseudo labels generated from one user are not used for model fine-tuning of a different user, thus avoiding situations of data poisoning.

6.2 Broader Sensing and Sleep Application

Our current prediction mechanism utilizes WiFi network rates as one of the available sources of smartphone sensing, representing users' wakeful interaction and thus helping predict sleep. Evidently, other types of phone sensor data have proven valuable data, such as screen activity, GPS, and application logs, to provide insights into students' health (e.g., the StudentLife project) [34]. Moreover, some works utilize audio signals from the smartphone to monitor breathing rates [31, 35] and detect sleep apnea [36, 37]. Our work continues to replicate this extensive user study with various sensor data collection and finer-grained sleep ground truth from our pool of participants. By sensing a phone's network activity rate to predict a sleep period and, correspondingly, using audio-sensing to monitor the user's breathing pattern, our semi-supervised learning approach can be employed for these existing applications to broaden the capabilities of sleep monitoring to new users, without their prior participation.

6.3 Limitation and Future Work

We expect tendencies where users will change their phone usage patterns and sleeping behavior, for example, during holiday periods, which are not yet trained in the teacher model. These occurrences can result to data drift [38] and affect the model prediction. Second, our approach is geared towards detecting the longest inactive period over a 24-hour window as the sleep period and needs to be extended to detect multiple inactive periods corresponding to multiple sleep periods, including daytime naps. Finally, our primary user study was among college students, considered digital natives. However, the three home users were full-time working professionals between the ages of 35-46. Findings from this investigation demonstrated that our technique applies to different user profiles. However, it includes the prerequisite use of active personal devices. Our work continues to conduct deeper evaluations to develop more robust models with finer-grained sleep measures.

7 Related Work

In this section, we discuss prior efforts on sleep monitoring using different sensing mechanisms and, more importantly, their model prediction techniques. Table 8 summarizes the key takeaway of each work and how *SleepLess* sets itself apart.

7.1 Sleep Monitoring Modalities

Polysomnography is the gold standard for sleep monitoring [39]. However, conducting sleep studies outside clinical settings would require less obtrusive monitoring techniques for long-term sleep monitoring. As an alternative, researchers explored using heart rate sensors and motion sensors, found in commercially available sleep trackers such as Fitbit and Apple watch [40], for detecting human activity [41]. Although such wearables are very convenient to use, there are cases where users may not always wear them to bed. Instead, contactless solutions such as doppler radar or RF signals are deployed as standalone devices to monitor breathing patterns [42, 43] and predict sleep [44, 45]. Other sensing techniques look into smartphone based sensing techniques for sleep tracking. These efforts utilize an array of phone sensors such as accelerometer [13, 22], light sensor [13, 22], microphone [13, 22, 30, 31], proximity sensor [13], and WiFi network activity rates [17]. These works above have proven rich (unlabeled) data we can use to predict sleep. However, collecting labeled ground truth of users continue to be a big challenge [46].

7.2 Sleep Prediction Techniques

Most of the techniques described above in the prior work are based on supervised learning approaches , whereby models require large amounts of training data to build accurate prediction models . Several works developed separate models for each user in the study which require at least 2 weeks of labelled data with sleep/wake-up estimation errors less than 40 minutes [17]. On the other hand, unsupervised methods have been explored specifically to do away with requiring training data. For example, Cuttone et. al develop a Bayesian approach to infer bed time and wake-up time from smart-phone screen events. Although convenient, these works have reported bed time and wake-time errors in the range of 1-2 hours and they don't offer any form of personalization. These results collectively informed our decision to explore semi-supervised learning (SSL) approaches, where we can leverage unlabeled data to improve the accuracy of the prediction models.

Our investigation on semi-supervised learning approaches began via Self Training, followed by other standard techniques such as Co-Training, Auto Encoder, Data Generative, and Adversarial Training. These approaches have been adopted by prior work to solve sleep stage classification [25, 26, 47, 48]. In understanding self-training, Zhang et al. reported error accumulation as a potential problem [49]. In contrast, other works have reported lesser error accumulation with co-training [50, 51], including successful detection of everyday human behavior such as walking, running, and climbing stairs [52]. In fact, much work in human activity recognition has utilized adversarial learning [53] and autoencoder [24, 54] to develop a generalizable and robust classification model for everyday human behavior.

Similar to these works, we aim to predict a person's sleep duration every night as an everyday human activity. However, these works rely on fine-grained time-series data sources such as EEG and actigraphy data, which are more likely to offer data completeness. In contrast, our work aims to develop a prediction model that can leverage unlabeled data, which is also suitable for coarse-grained data.

8 Conclusions

Fundamentally, the requirement of collecting a significant amount of ground truth holds for training any user behavioral models. Unlike many prior sleep detection techniques that rely on collecting and training large amounts of labeled data, our work sought to build a model that can cater to new users without collecting ground truth information from them. We proposed *SleepLess*, which uses semi-supervised learning over unlabeled data sensed from the user's smartphone network activity to develop personalized models and detect their sleep duration for the night. By using a generalized pre-trained model on an existing set of users to produce pseudo labels for unlabeled data of a new user, it achieves personalization by fine-tuning using selected pseudo-labels for the new user without requiring any labeled data. Our user study among 23 users found *SleepLess* model yielding around 96% accuracy, between 12-27 minutes of sleep time error and 18-25 minutes of wake time error. We also demonstrated applicability to private home users and compared our technique with similar learning techniques that rely on fewer labeled data. With our prediction technique yielding the best performance, our work shows promise for sleep monitoring to be more conveniently adapted to monitor new users' sleep immediately. Where the larger goal of our work aims to improve students' health, lack of sleep is linked to many major health challenges. Our work continues investigating the efficacy of this technique in complementary domains, including sleep quality.

Acknowledgments

This work is supported in parts by NSF grants 2211302, 2211888, 2213636, 2105494, and Army grant W911NF-17-2-0196.

Declarations

- Conflict of interest/Competing interests: On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

[1] Ohayon, M.M.: Epidemiological overview of sleep disorders in the general population. Sleep Medicine Research **2**(1), 1–9 (2011)

[2] Krueger, P.M., Friedman, E.M.: Sleep duration in the united states: a cross-sectional population-based study. American journal of epidemiology **169**(9), 1052–1063 (2009)

[3] Google, I. (2022 (accessed Oct 27, 2022)). https://www.fitbit.com/global/be/products/trackers/inspire

[4] inc. (2022 (accessed Oct 27, 2022)). https://ouraring.com/

[5] Rahman, T., Adams, A.T., Ravichandran, R.V., Zhang, M., Patel, S.N., Kientz, J.A., Choudhury, T.: Dopplesleep: A contactless unobtrusive sleep sensing system using short-range doppler radar. In: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 39–50 (2015)

[6] Google, I. (2022 (accessed Oct 27, 2022)). https://support.google.com/googlenest/answer/10388741?hl=en

[7] Amazon, I. (2022 (accessed Oct 27, 2022)). https://www.amazon.com/Paschar-LLC-Walabot-Sleep-Tracker/dp/B07C2HRYSX

[8] Dixon, M., Schneider, L., Yu, J., Hsu, J., Pathak, A., Shin, D., Lee, R.S., Malhotra, M.R., Mixter, K., McConnell, M., Taylor, J., Patel, S.: Sleep-wake detection with a contactless, bedside radar sleep sensing system. Technical report (2021)

[9] Abdullah, S., Matthews, M., Murnane, E.L., Gay, G., Choudhury, T.: 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, pp. 673–684 (2014)

[10] Zakaria, C., Yilmaz, G., Mammen, P.M., Chee, M., Shenoy, P., Balan, R.: Sleepmore: Inferring sleep duration at scale via multi-device wifi sensing. Proceedings

of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **6**(4), 1–32 (2023)

[11] Zhao, Y., Liu, H., Li, H., Barnaghi, P., Haddadi, H.: Semi-supervised federated learning for activity recognition. arXiv preprint arXiv:2011.00851 (2020)

[12] Trivedi, A., Zakaria, C., Balan, R., Becker, A., Corey, G., Shenoy, P.: Wifitrace: Network-based contact tracing for infectious diseases using passive wifi sensing. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **5**(1), 1–26 (2021)

[13] Min, J.-K., Doryab, A., Wiese, J., Amini, S., Zimmerman, J., Hong, J.I.: Toss'n'turn: smartphone as sleep and sleep quality detector. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 477–486 (2014)

[14] inc., A. (2023 (accessed Aug 17, 2023)). https://developers.google.com/location-context/sleep

[15] inc. (2023 (accessed Aug 17, 2023)). http://autosleep.tantsissa.com/home

[16] rise (2023 (accessed Aug 17, 2023)). https://www.risescience.com/

[17] Zakaria, C., Yilmaz, G., Mammen, P., Chee, M., Shenoy, P., Balan, R.: Sleepmore: Sleep prediction at scale via multi-device wifi sensing. arXiv preprint arXiv:2210.14152 (2022)

[18] Mammen, P.M., Zakaria, C., Molom-Ochir, T., Trivedi, A., Shenoy, P., Balan, R.: Wisleep: Inferring sleep duration at scale using passive wifi sensing. arXiv preprint arXiv:2102.03690 (2021)

[19] Dhamija, A.R., Günther, M., Boult, T.: Reducing network agnostophobia. Advances in Neural Information Processing Systems **31** (2018)

[20] Python, I. (2022 (accessed Oct 27, 2022)). https://www.python.org/

[21] keras, I. (2022 (accessed Oct 27, 2022)). https://keras.io/

[22] Saeb, S., Cybulski, T.R., Schueller, S.M., Kording, K.P., Mohr, D.C., *et al.*: Scalable passive sleep monitoring using mobile phones: opportunities and obstacles. J Med Internet Res **19**(4), 118 (2017)

[23] Chen, M., Du, Y., Zhang, Y., Qian, S., Wang, C.: Semi-supervised learning with multi-head co-training. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 6278–6286 (2022)

[24] Bhattacharya, S., Nurmi, P., Hammerla, N., Plötz, T.: Using unlabeled data in a sparse-coding framework for human activity recognition. Pervasive and Mobile

Computing **15**, 242–262 (2014)

[25] Haoran, B., Guanze, L.: Semi-supervised end-to-end automatic sleep stage classification based on pseudo-label. In: 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), pp. 83–87 (2021). IEEE

[26] Zhang, C., Yu, W., Li, Y., Sun, H., Zhang, Y., De Vos, M.: Cms2-net: Semi-supervised sleep staging for diverse obstructive sleep apnea severity. IEEE Journal of Biomedical and Health Informatics **26**(7), 3447–3457 (2022)

[27] Li, Y., Peng, C., Zhang, Y., Zhang, Y., Lo, B.: Adversarial learning for semi-supervised pediatric sleep staging with single-eeg channel. Methods **204**, 84–91 (2022)

[28] Munk, A.M., Olesen, K.V., Gangstad, S.W., Hansen, L.K.: Semi-supervised sleep-stage scoring based on single channel eeg. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2551–2555 (2018). IEEE

[29] Gu, W., Shangguan, L., Yang, Z., Liu, Y.: Sleep hunter: Towards fine grained sleep stage tracking with smartphones. IEEE Transactions on Mobile Computing **15**(6), 1514–1527 (2015)

[30] Hao, T., Xing, G., Zhou, G.: isleep: unobtrusive sleep quality monitoring using smartphones. In: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, pp. 1–14 (2013)

[31] Ren, Y., Wang, C., Yang, J., Chen, Y.: Fine-grained sleep monitoring: Hearing your breathing with smartphones. In: Computer Communications (INFOCOM), 2015 IEEE Conference On, pp. 1194–1202 (2015). IEEE

[32] Cuttone, A., Bakgaard, P., Sekara, V., Jonsson, H., Larsen, J., Lehmann, S.: Sensiblesleep: A bayesian model for learning sleep patterns from smartphone events. PLoS ONE **12**(1) (2017)

[33] Zhenyu, C., Lane, N., Cardone, G., Lin, M., Choudhury, T., Campbell, A.: Unobtrusive sleep monitoring using smartphones. In: Proceedings of Pervasive Health (2013)

[34] Wang, R., Chen, F., Chen, Z., Li, T., Harari, G., Tignor, S., Zhou, X., Ben-Zeev, D., Campbell, A.T.: Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 3–14 (2014)

[35] Wang, X., Huang, R., Yang, C., Mao, S.: Smartphone sonar-based contact-free respiration rate monitoring. ACM Transactions on Computing for Healthcare

2(2), 1–26 (2021)

[36] Tiron, R., Lyon, G., Kilroy, H., Osman, A., Kelly, N., O'Mahony, N., Lopes, C., Coffey, S., McMahon, S., Wren, M., *et al.*: Screening for obstructive sleep apnea with novel hybrid acoustic smartphone app technology. Journal of Thoracic Disease **12**(8), 4476 (2020)

[37] Kim, D.H., Kim, S.W., Hwang, S.H.: Diagnostic value of smartphone in obstructive sleep apnea syndrome: A systematic review and meta-analysis. Plos one **17**(5), 0268585 (2022)

[38] Hofman, J.M., Sharma, A., Watts, D.J.: Prediction and explanation in social systems. Science **355**(6324), 486–488 (2017)

[39] Rundo, J.V., Downey III, R.: Polysomnography. Handbook of clinical neurology **160**, 381–392 (2019)

[40] inc., A. (2022 (accessed Oct 27, 2022)). https://www.apple.com/watch/

[41] Witt, D.R., Kellogg, R.A., Snyder, M.P., Dunn, J.: Windows into human health through wearables data analytics. Current opinion in biomedical engineering **9**, 28–46 (2019)

[42] Lee, Y.S., Pathirana, P.N., Steinfort, C.L., Caelli, T.: Monitoring and analysis of respiratory patterns using microwave doppler radar. IEEE journal of translational engineering in health and medicine **2**, 1–12 (2014)

[43] Gu, C., Li, C.: Assessment of human respiration patterns via noncontact sensing using doppler multi-radar system. Sensors **15**(3), 6383–6398 (2015)

[44] Lin, F., Zhuang, Y., Song, C., Wang, A., Li, Y., Gu, C., Li, C., Xu, W.: Sleepsense: A noncontact and cost-effective sleep monitoring system. IEEE transactions on biomedical circuits and systems **11**(1), 189–202 (2016)

[45] Hong, H., Zhang, L., Zhao, H., Chu, H., Gu, C., Brown, M., Zhu, X., Li, C.: Microwave sensing and sleep: Noncontact sleep-monitoring technology with microwave biomedical radar. IEEE Microwave Magazine **20**(8), 18–29 (2019)

[46] De Zambotti, M., Cellini, N., Goldstone, A., Colrain, I.M., Baker, F.C.: Wearable sleep technology in clinical and research settings. Medicine and science in sports and exercise **51**(7), 1538 (2019)

[47] Yalniz, I.Z., Jégou, H., Chen, K., Paluri, M., Mahajan, D.: Billion-scale semi-supervised learning for image classification. arXiv preprint arXiv:1905.00546 (2019)

[48] Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European

Conference on Computer Vision (ECCV), pp. 289–305 (2018)

[49] Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Communications of the ACM **64**(3), 107–115 (2021)

[50] Zhou, Z.-H., Li, M.: Semi-supervised learning by disagreement. Knowledge and Information Systems **24**(3), 415–439 (2010)

[51] Wang, W., Zhou, Z.-H.: Analyzing co-training style algorithms. In: European Conference on Machine Learning, pp. 454–465 (2007). Springer

[52] Guan, D., Yuan, W., Lee, Y.-K., Gavrilov, A., Lee, S.: Activity recognition based on semi-supervised learning. In: 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2007), pp. 469–475 (2007). IEEE

[53] Faridee, A.Z.M., Chakma, A., Misra, A., Roy, N.: Strangan: Adversarially-learnt spatial transformer for scalable human activity recognition. Smart Health **23**, 100226 (2022)

[54] Gogna, A., Majumdar, A.: Semi supervised autoencoder. In: International Conference on Neural Information Processing, pp. 82–89 (2016). Springer