LuckyChirp: Opportunistic Respiration Sensing Using Cascaded Sonar on Commodity Devices

Qiuyue (Shirley) Xue*‡, D Shin†, Anupam Pathak†, Jake Garrison†, Jonathan Hsu†, Mark Malhotra†, Shwetak Patel*†
*Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA 98195, USA
†Google LLC., Mountain View, CA 94043, USA
*{qxue2,shwetak}@cs.washington.edu, †{deshin, anupampathak, jakegarrison, jonathanhsu, markmalhotra}@google.com
‡This work was completed during the author's internship at Google.

***Abstract*—We present LuckyChirp, a contactless, passive, opportunistic respiratory tracking solution for commodity device using cascaded sonar modeling. Compared to conventional sonar methods that only solve the respiratory estimation problem ("what is the respiratory rate"), LuckyChirp also solves the additional respiratory detection problem ("is the human present and static enough for respiration sensing"). LuckyChirp uses a custom neural network on pulsed sonar's wavelet transformed features to detect respiration. The classifier is then cascaded with a respiratory rate estimator. Such holistic design eliminates user friction of manually activating the system and enables passive respiration monitoring for all-day natural use. With Google Nest Hub and Pixel 4 as experimental devices, LuckyChirp achieves a mean absolute error of 0.48 ± 0.98 and 1.07 ± 1.67 breaths/min, respectively, for 20 users participating in a whole-night study. Compared to direct respiratory estimation without respiration classification, this is a $\times$6 (Nest Hub) and $\times$4 (Pixel) reduction in error.**

***Index Terms*—Contactless respiration monitoring; Sonar Sensing; Mobile Health; Pervasive Computing**

I. Introduction

Estimating the Respiratory Rate (RR) of a user at rest can offer insights into respiratory health and potentially predict condition onsets or exacerbations [1]. Providing effortless, non-contact daily respiratory measures can thus benefit many users. There are several contactless sensing modalities for respiratory sensing in literature, such as radar [2], [3], camera [4], and sonar [5], [6]. Compared to radar, sonar does not need additional hardware like RF transceivers, and it is widely accessible on commodity devices with speakers and microphones. Sonar is also much less privacy-intrusive than cameras. Thus, sonar is a promising modality for ubiquitous health monitoring on consumer devices. One known drawback of existing sonar solutions is that they compute RR only in active mode, which requires the user to start the system manually for measurements [7]. As a result, an active sensing app for daily health monitoring usually has a high drop-out rate [8]. Additionally, when users are aware that the system is measuring their respiratory rate, they may adjust their breathing, which may result in an altered, non-rest measurement [9]. In contrast, passive mode sensing can operate in the background (with user's consent) and does not require any user attention once set up.

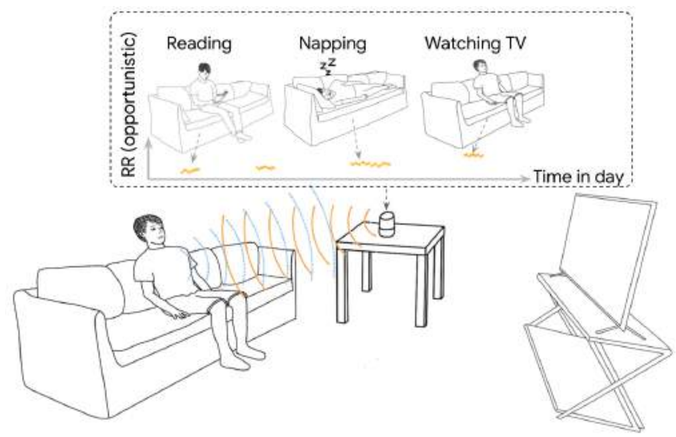

Fig. 1. LuckyChirp provides opportunistic respiratory measurements on a user without active device engagement.

We propose LuckyChirp, an opportunistic sonar system that aims to provide an accurate spot check of respiratory rates without active user engagement. Fig. 1 shows an example application: a smart home device (or smartphone) detecting a proximal user's RR. To extract an accurate RR for potentially moving user, LuckyChirp answers the question "is the condition good for respiration sensing?" before activating the step of estimating respiratory rate.

There are several challenges in building such a fully passive opportunistic sonar sensing system:

1) Power consumption for sonar can be high. Sonar sensing uses on-device speakers for ultrasonic transmission. Continuous-mode sonar, such as Frequency Modulated Continuous Wave (FMCW), operates the speakers in always-on mode and thus have a quicker battery drain for mobile devices.

2) Sensing time window needs to be be short. Users need to stay relatively static during the measurement to get a valid RR result. However, in practice, the times that a proximal user stays static are sporadic. It is challenging to confirm a static state in a short time because of limited signal context. Most existing systems assume a longitudinal recording (several minutes) to evaluate results.

3) The respiration sensing range needs to be long. The system should operate with a reasonable field-of-view to handle real-world scenarios (e.g. bedside monitoring). However, longer sensing distancs or oblique angles means lower Signal-to-Noise-Ratio (SNR), making respiration detection and estimation more difficult.

In order to address the above challenges, LuckyChirp employs three novel design approaches:

1) We use a correlation-based pulse mode respiratory sensing, which duty-cycles the speaker-on times and thus consumes significantly less power than FMCW without compromising sensing accuracy.

2) We design a novel Continuous Wavelet Transform (CWT)-based feature extraction method. The CWT scalogram offers high feature resolution for a short decision time window and thus a high classification accuracy.

3) We introduce a novel RR classification pipeline for sonar using a Range-Selecting Convolutional Neural Network (RS-CNN), which accurately detects respiration and finds user distance from a low SNR recording.

We evaluated LuckyChirp on commercial smartphones (Pixel 4) and smart home devices (Google Nest Hub Max). We conducted overnight experiments on 20 users sleeping up to 3 meters distance from device. The recorded data consists of 162.3 hours for smart home and 163.9 hours for smartphones sonar. We segmented sonar data into 30 seconds windows for classification and RR estimation. And we experimented with 30 seconds, 60 seconds, and 90 seconds windows for user distance detection. Polysomnography (PSG) and under-mattress Ballistocardiography (BCG) data were recorded as ground truth. There was no calibration or assumption of user distance and orientation for processing.The results show that:

- For smart home devices, LuckyChirp achieves a Mean Absolute Error (MAE) of 0.48 ± 0.98 Breaths Per Minute (BPM) for data that LuckyChirp detects respiration.
- For smartphones, LuckyChirp achieves an MAE of 1.07 ± 1.67 BPM for data that LuckyChirp detects respiration.
- The experiment of different window lengths for distance detection shows that a 60-second distance searching window is optimal for opportunistic sensing because it is the shortest window size that maintains a low MAE.

In summary, LuckyChirp aims to provide passive opportunistic respiration sensing using sonar. It passively screens for periods of clean, periodic breathing patterns, and provides a point estimate of respiratory rate for that period of time. We highlight our critical contributions here:

- **Systems:** We introduced **correlation based pulsed-mode sonar for respiration sensing**, which leverages chirp compression techniques, for improved power efficiency and peak-SNR compared to traditional FMCW decoding.
- **Modeling:** We proposed a **cascaded sonar design** that solves both the respiratory detection and estimation tasks. We detect ”is the condition good for respiration sensing?” using supervised deep learning and estimate ”what is the user’s RR?” using spectral methods.
- **Evaluation:** We evaluated LuckyChirp on over 327 hours of sonar data from off-the-shelf smart home devices and smartphones. We demonstrated that LuckyChirp has great potential to attain passive contactless respiration monitoring in practical, everyday scenarios. Also, for the first time, we compared the end-to-end performance between smart home devices and smartphones and showed that smart home devices have $\times 2$ accuracy in MAE.

	Implementation	Audible	Form factor	Max Distance	Opportunistic
ApneaApp	FMCW + freq tracking	No	Smartphones	<1m,on bed	No, by trigger
SonarBeat	FMCW + Phase	No	Smartphones	<0.5m	No, by trigger
C-FMCW	FMCW + correlation	No	Customized	<1.1m	No, by trigger
BreathJunior	white noise + freq tracking	Yes	Customized	<0.7m	No, by trigger
Opioid Overdose	FMCW + freq tracking	No	Smartphones	<1m	No, by trigger
BreathListener	Energy spectrum + GAN	No	Smartphones	<2m	No, by trigger
RespTracker	Zadoff-Chu modulation	No	Customized	<3m	No, by trigger
LuckyChirp	Chirp Pulse + correlation	No	Smartphones, home devices	<3m	Yes

Fig. 2. Comparison of related work: sonar for respiration sensing

II. Related Work

A. Contactless Sonar Sensing

Researchers have demonstrated using sonar to track locations, user motions and gestures [10]–[14], as well as vital signals [5], [6], [15]–[17]. LuckyChirp is most related to works that use sonar to monitor breathing. Fig. 2 shows the related work for respiration monitoring using sonar. ApneaApp [5] used FMCW sonar to track user motion, respiration and apnea events by putting the smartphone on the bed. SonarBeat [6] used FMCW phase information to monitor the user’s RR. C-FMCW [15] proposed correlation-based FMCW medthod for respiration monitoring with customized hardware. BreathJunior [16] demonstrated infant respiration monitoring using white noise signals with customized hardware. Opioid overdose detection [18] used FMCW to monitor users’ RR during the opioid injection. BreathListener [19] extracted the user’s RR in driving scenarios using energy spectrum density by training GAN model. RespTracker [20] leveraged Zadoff-Chu sequence to measure the amplitude and phase of the signal and extract the RR. In addition, sonar has also been demonstrated to monitor user’s other health conditions such as lung function [17] and heart rate [21]. However, most of the work used continuous-wave sonar to sense respiration, which can be limiting for all-day, opportunistic sensing applications.

B. Contactless vitals monitoring using radar or camera

Researchers have explored other sensing modalities for contactless vitals monitoring despite sonar, including Radio Frequency (RF) and camera. For RF, researchers have demonstrated using Doppler effect [22], FMCW [23], [24], millimeter waves [2], [25], [26], ultra-wide radar [3], and WiFi [27]. The theory of operation for RF sensing is pretty similar to sonar. Though some RF methods could get better accuracy than sonar because of shorter wavelengths and higher power, RF methods usually require additional bulky hardware that is not available on commodity devices.

Besides radar, cameras have also been demonstrated to monitor the user’s RR [4] and heart rate [28], [29]. However, one biggest concern for camera sensing is privacy, making it difficult for opportunistic passive sensing in daily scenarios.

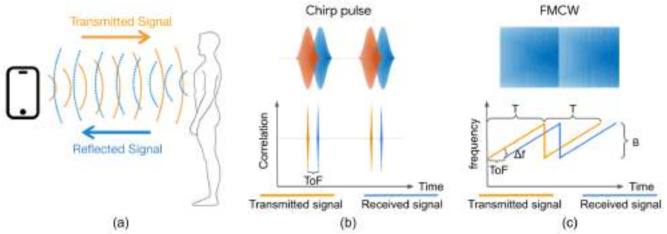

Fig. 3. (a) Respiratory sensing using sonar. (b) Chirp pulse signal and tracking its ToF using correlation. (c) FMCW signal and tracking its ToF using frequency shift.

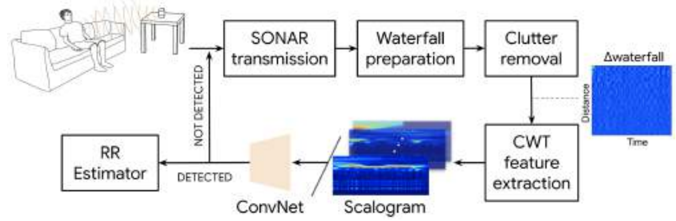

Fig. 4. LuckyChirp design overview

C. Opportunistic Health Monitoring

Most contactless systems described above require users to start the measuring process manually. Opportunistic sensing can automatically detect users' vital signs and is more practical for daily health monitoring. However, most of the current opportunistic sensing methods are contact-based wearables, such as Fitbit and Apple Watch. Both Fitbit and Apple watch can opportunistically monitor the user's heart rate, SpO2, and estimate the user's RR. Besides wearables, Sinabr [30] provides opportunistic ECG monitoring when the user touches the phone screen. Li et al. [31] proposed Doppler radar sensing for opportunistic respiration monitoring. To the best of our knowledge, LuckyChirp is the first sonar model that enables install-and-forget, contactless respiratory monitoring.

III. LuckyChirp Design

A. Sonar Sensing Background

Fig. 3 shows the idea of sonar sensing. The phone or smart home device transmits the pulsed sonar signal using the speaker and simultaneously records the reflected signal with the microphone. The transmitted pulses are reflected by objects in the environment, including the user's body. Fig. 3(b) shows the waveform and correlation of the transmitted and reflected signal. The time delay between transmitted and received chirp pulses corresponds to the Time-of-Flight (ToF) between the sonar device and the object that reflects the signal. And the distance between the object and the sonar device can be computed by $d = ToF \times V_s$, where V_s is the speed of sound. We can then extract the user's respiratory rate by computing the distance change Δd over time.

B. System Overview

LuckyChirp proposes a cascade two-stage algorithm: 1) Classify if the user is in the scene and is static enough to provide a clear respiratory pattern at any distance. 2) Compute the RR if the model detects a valid respiration signal.

Fig. 4 shows the algorithm pipeline. The received sound signal contains the reflection of the transmitted chirp pulses from the environment. We apply cross-correlation on the received signal to get the distance information of the environment. The correlation envelope was stacked to construct a 2D sonar "waterfall", whose x-axis is time, and the y-axis is distance. We further remove static reflections from the environment (walls, tables, etc.) by computing the difference of waterfall over time and get the Δwaterfall. Then we slice the 2D "Δwaterfall" signal at each distance and compute its scalogram feature using CWT. The CNN model classifies the scalograms and outputs a probability score of the scalogram having a valid respiratory signal. We estimate the user distance by picking the highest probability positive scalogram. We then compute the user's RR at this estimated distance.

C. Chirp Compression

Instead of FMCW, LuckyChirp chooses a frequency-modulated ultrasonic pulse signal for sonar sensing. Fig. 3(b) and (c) show a comparison of chirp pulse mode and FMCW mode. We show that correlation-based chirp pulse consumes less power than FMCW, and achieves a much higher peak-SNR than traditional FMCW decoding.

1) Chirp Pulse: LuckyChirp uses a frequency-modulated chirp pulse, and its frequency $f(t)$ varies linearly with time: $f = f_0 + \frac{B}{\tau}t$, where f_0 is the lowest frequency, B is the bandwidth, τ is the pulse time length, and T is the period. The time-domain function for the chirp is: $s(t) = sin(2\pi(f_0 + \frac{B}{\tau}t)t + \phi_0) \times sin^2(\frac{\pi t}{\tau})$, if $0 < t < \tau$; and $s(t) = 0$, if $\tau < t < T$. The $sin^2(\frac{\pi t}{\tau})$ at the end is a Hann window to taper off the signal, which eliminates audible spectrum leakage caused by the discontinuities at the pulse edges. The signal after the Hann window is confirmed inaudible.

The physical range resolution for chirp pulse is $\Delta R = \frac{V_s}{2B}$, which is same as FMCW. As for the power consumption, chirp pulse sonar only transmits for time τ during period T while FMCW transmits ceaselessly, so theoretically, pulse-mode consumes only $\frac{\tau}{T}$ percent of FMCW power.

2) Cross-Correlation: Traditional FMCW decoding compares the received signal frequency with the transmitted signal to get the frequency shift Δf, then computes the $ToF = \Delta f \times \frac{T}{B}$ as shown in fig. 3(c).

Instead of tracking frequency shift, LuckyChirp leverages cross-correlation in decoding to gain higher peak-SNR. By correlating the received signal with the transmitted template, we essentially implemented matched filtering that compresses the chirp pulse to a pulse with a much smaller width length $\tau' = \frac{1}{B}$ [32]. The pulse compression ratio is $\frac{\tau}{\tau'} = \tau \times B$. The transmitted signal energy E does not change with pulse compression, so the total energy is compressed into a narrower pulse with a smaller pulse length. Thus SNR is amplified as well. Fig. 5(a) shows the simulated peak-SNR results of using FMCW frequency shift method vs. cross-correlation. Correlation-based chirp obtains higher PSNR (around 4 times) than FMCW front-end globally over the parameter space of noise level and chirp length.

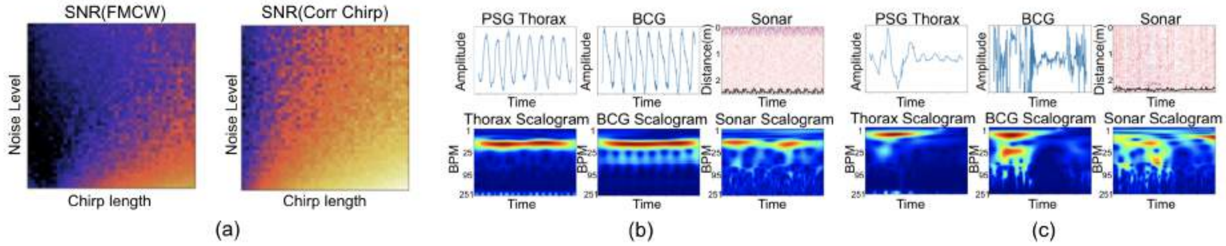

Fig. 5. (a) Simulated peak-SNR of FMCW vs. cross correlation on chirp pulses. (b) A 30-seconds window example that contains respiration signals (labeled as 1), (c) A 30-seconds window example that does not contain clear respiration signals (labeled as 0).

D. Sonar Signal Preprocessing

The sonar's preprocessing refers to steps from the recorded sound to the extracted sonar waterfall. The steps include:

1) Cross-correlate the transmitted chirp template with the received signal. The correlation result consists of spatial information of the environment. Each peak in the correlation envelope corresponds to a roundtrip path's ToF between the sonar device and an object (including the user's body). 2) Compute the envelope of the correlation signal with average pooling. 3) Stack the correlation envelope of each chirp vertically to construct a 2D waterfall. Each chirp works as a scan of the environment objects' distance information. By stacking them vertically, we get a 2D waterfall that contains distance information over time. 4) Remove the static clutters in the 2D waterfall. There are static objects (e.g., walls, tables) that are not related to respiration. We remove the static reflections by computing the difference of waterfall over time.

E. Scalogram Feature

We applied CWT to the sonar waterfall at each distance to compute the signal's scalogram. CWT is the convolution of the input signal with a set of functions generated by the mother wavelet at different scales. We use the Morlet wavelet for the CWT scalogram computation because it is closely related to human perception of vision and hearing. CWT mainly has two advantages over Short-Time Fourier Transform (STFT):

1) CWT offers good frequency resolution at low frequencies, preferred for detecting respiratory signals (respiratory frequency < 0.5Hz) and computing respiratory rate. In addition, CWT provides better time resolution at high frequencies, helping detect motions (with many high-frequency components) with precise time.

2) CWT offers good frequency flexibility. The scales of CWT's daughter wavelets control the scalogram's frequencies, which can be self-defined. We set the CWT frequencies fine-grained resolution in the breathing frequency range and coarse resolution in other frequency ranges. In this way, we magnify the frequency details in the breathing signals.

With CWT scalogram, the respiration detection problem is re-defined as an image classification problem, which is much more robust than just looking for a periodic pattern in a time-domain signal.

F. Classification

We feed the CWT scalogram at each distance index into a CNN model to classify whether the captured sonar signal has a respiration pattern. The CNN consists of two convolutional layers, each followed by a max-pooling layer, then a flatten layer, and two fully connected layers. The last layer uses the sigmoid activation function that outputs a probability between 0 and 1. The probability is used as a confidence score for distance searching later.

We randomly selected 460 30-seconds windows of smartphone sonar from 12 different users and manually labeled them for training. During labeling, we plot the corresponding ground truth (PSG thorax and BCG) raw signal and their scalograms to verify. Fig. 5(b) and (c) shows an example of a positive and negative 30-second window. We labeled the data following clinical advice: 1) Thorax, BCG, and sonar signals should all have clear periodic patterns in the respiratory frequency range (5 –30 BPM), showing a horizontal line in the scalograms. 2) The periodic patterns should have significantly higher energy than other noises. 3) PSG thorax, BCG, and sonar signals should have the same dominant frequency. It is fine if the periodic signal's amplitude or frequency gradually changes over time, as long as PSG thorax, BCG, and sonar show the same pattern. The typical scenarios that cause a window classified as 0 include: user motions, user not in the scene, bad angle or distance that makes the SNR low, disruptive breathing patterns such as sleep apnea, etc.

We augmented the 460 samples by flipping the scalogram horizontally and got 920 labeled samples. The 920 samples were randomly split into 80% training and 20% testing data. The model achieved 94.8% and 94.6% accuracy on the training and testing datasets, respectively. We also evaluated the model with smart home device data, though the training data were from smartphones only. The results showed that LuckyChirp is generalizable across devices with different hardware and ultrasound settings.

G. Distance Searching

To find the distance between the user and the sonar device, we iterate through all the distances and select the one whose signal is closest to a respiration signal.

1) Naive Algorithm: An naive algorithm to find the user distance is to search for the most periodic signal in the breathing frequency range. We developed a rule-based algorithm to find the user distance. For each distance's scalogram, we compute the percentage of time with the same dominant frequency (in the breathing frequency range). The dominant frequency is also gated by an empirical SNR threshold. The

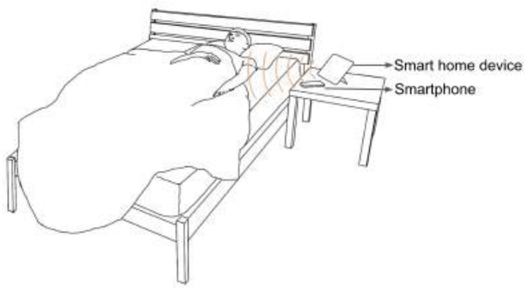

Fig. 6. The set up for night time experiment.

distance with the highest percentage is selected as the user distance. In practice, we found that this algorithm is not robust for different scenarios, and we use this algorithm as a baseline in evaluation later.

2) Classifier-based Algorithm: Since we have already trained a model to detect the respiration signals, we can use the probability output from the last layer as an indicator to find the user distance. So we developed a classifier-based algorithm to find the user distance. We computed the CWT scalogram of the sonar waterfall at each distance and fed each scalogram to the CNN model separately. The last layer of the model outputs a probability number between 0 and 1, indicating the probability that the scalogram contains a respiration signal. We pick the distance with the highest probability and consider it as the user's distance to the device.

H. Respiratory Rate Estimation

For each 30-second window sonar data, LuckyChirp computes the RR using FFT on the selected distance's sonar waterfall signal. The signal is zero-padded on both sides, making it a total of 120 seconds, thus each FFT bin is $\frac{sampling\_rate}{total\_samples\_of\_signal} = 0.5BPM$. We extracted the frequency bin with the highest energy in FFT results ranging from 5 BPM to 30 BPM, covering more than adults' normal RR range (12-20BPM) [33]. No fundamental barriers stop LuckyChirp from estimating RR in a larger range.

IV. Night-time Evaluation

A. Implementation

We implemented LuckyChirp on Google Pixel 4 and Google Nest Hub Max. We developed an app to transmit and record the ultrasound signal simultaneously. The signals were processed offline using Python. On Pixel 4, the chirp frequency linearly increases from 19kHz to 21.5kHz, with a 20ms chirp duration, 50ms period, and a 48000Hz sampling rate. On Nest Hub Max, the chirp frequency linearly increases from 26kHz to 31kHz, with an 8ms chirp duration, 50ms period, and a 96000Hz sampling rate. We trained the LuckyChirp CNN model using TensorFlow 2.0. The training data was randomly selected from the smartphone sonar (discussed in sectionIII-F).

B. Night-time experiment setup

This sleep-breathing analysis was conducted on volunteers recruited from SleepMed (a national healthcare research organization) and the surrounding community of South Carolina.

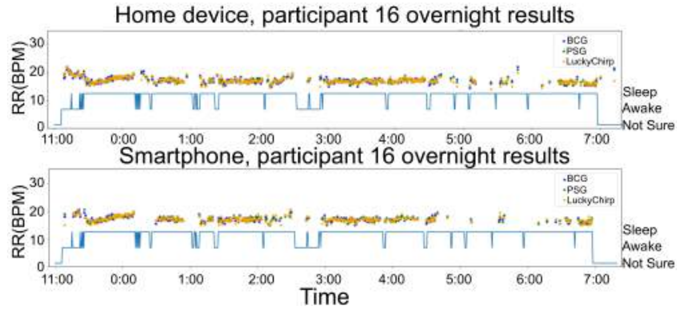

Fig. 7. The whole night respiration estimation results of participant 16 from (a) smart home device and (b) smartphone.

The study was approved by the Advarra institutional review board, and all participants provided written informed consent before participation. The characteristics of the group are: 20 adults (10 male, 10 female); ages ranging from 25 to 70-years-old (49.5 ± 14.6); BMI $26.5 \pm 4.4kg/m^2$; a range of sleep apnea severity (6 no apnea, 8 mild apnea, 3 moderate apnea, and 3 severe apnea); collected from 4 different rooms of the sleep laboratory. Fig. 6 shows an example of the experiment setup. The smartphone and smart home device were put on the table, and the table may be on the left or right side of the bed in actual settings. The bed width was around 1.5 meter, and users wore blankets during the overnight data collection.

The PSG thorax and the under-mattress BCG signal were used as ground truth. PSG monitors many body functions, including EEG, ECG, and respiratory activity (via respiratory inductive plethysmography [RIP]). The BCG uses accelerometer and piezoelectric sensors to capture the user's respiration. The timestamps of the smart home devices sonar, smartphone sonar, PSG thorax, and BCG signals were recorded every 10 minutes and were used to sync these signals later.

For the 20 participants, we recorded 162.3 hours of sonar data from smart home devices and 163.9 hours from smartphones (the smart home device and smartphone may start and end at a slightly different time). The 20 participants' data were segmented into 30-second windows without overlap, which resulted in 19475 windows for smart home, and 19677 windows for smartphones. LuckyChirp searched up to 3 meters to find the user distance and compute the RR.

C. Methodology

1) We evaluated LuckyChirp's RR estimation error using the Mean Absolute Error (MAE) of sonar RR to the ground truth RR. The ground truth was computed from the average of the PSG thorax RR and BCG RR. If the PSG RR does not agree with the BCG RR for more than 1 BPM difference, we disregard this window because its ground truth is invalid. The LuckyChirp RR results were computed using a 30-second window for classification and RR estimation and a 60-second window to search the distance.

2) We evaluated the classification efficiency by comparing LuckyChirp to a baseline method. The baseline algorithm blindly computes all the sonar data without classification and uses the naive algorithm to find user distance. In contrast, LuckyChirp only computes RR for windows classified as

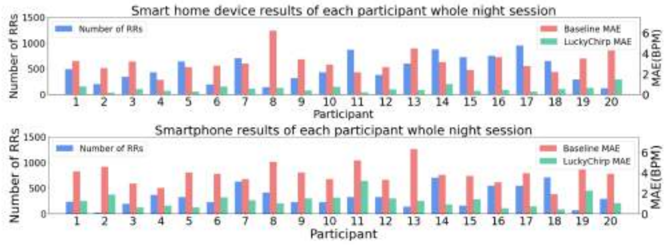

Fig. 8. Smart home and smartphone results of 20 participants' whole night data.

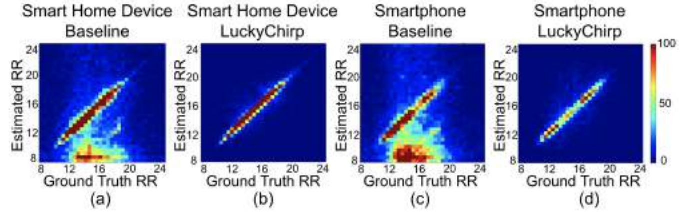

Fig. 9. Estimated RR vs. ground truth RR from home devices and smartphones.

positive. In addition, LuckyChirp uses the classifier-based algorithm instead of the naive algorithm (section III-G).

3) We compared LuckyChirp's performance between smart home devices and smartphones, and how different time window affects the results.

D. Smart Home Device Results

Fig. 7(a) shows an example of overnight results from smart home devices. The participant was sleeping around 1 meter away from the device. The blue line is sleep status labeled by a board-certified technologist, which is only plotted for reference here. The estimated RRs shown in fig.7 are only windows classified as positive by LuckyChirp. A few results samples deviated from the central results in the plot, which may be because of LuckyChirp's classification failures, estimation errors, or the user's sudden breathing change (apnea events), .

The whole data length for fig. 7 night is 8.3 hours, consisting of 997 30-second windows, and there are 751 windows classified as a positive (respiration detected) by LuckyChirp. The MAE for fig. 7(a) night is 0.39±0.50 BPM for the 751 positive windows. In contrast, the baseline algorithm's MAE for the same data is 3.62 ± 4.10 BPM, which is 10 times larger than LuckyChirp's MAE. This comparison shows that the classification gating is necessary to output a trusty RR.

Fig.8 shows smart home devices' 20 nights results. For each participant, the blue bar is the number of windows classified as positive by LuckyChirp. The red bar is the baseline algorithm's MAE, the green bar is LuckyChirp's MAE, which was computed on positive windows only. LuckyChirp's average MAE for 20 participants is 0.52 ± 0.29BPM. And LuckyChirp's aggregated results of the 20 nights' data are: Out of 19479 total windows, 10074 windows were classified as positive by LuckyChirp, and they have an MAE of 0.48 ± 0.98BPM. In contrast, the baseline algorithm's MAE is 3.1 ± 3.02BPM, which is x6 times larger than LuckyChirp. The standard deviation of LuckyChirp's MAE distribution is rather spread out. The reason is that a few classification failures can have random RR, causing the MAE to be much larger.

E. Smartphone Results

Fig.7(b) is an example of overnight results from the same participant as fig. 7(a) but recorded by a smartphone. The data length for fig. 7(b) is 8.3 hours, consisting of 997 30-seconds window. 541 windows were classified as positive by LuckyChirp. The MAE for fig. 7(b) is 0.50 ± 0.66 BPM for the 541 positive windows. In contrast, the baseline algorithm's MAE for the same data is 3.08 ± 3.64 BPM.

Fig. 8(b) shows smartphones' 20 nights results. The average MAE among 20 participants is 1.25 ± 0.64BPM. The smartphone aggregated results of the 20 nights' data are: Out of total 19677 windows, 6648 windows were classified as positive, with an MAE of 1.07 ± 1.67BPM. In contrast, the baseline algorithm's MAE is 3.86 ± 3.30BPM without LuckyChirp's classification, which is about x4 times larger than LuckyChirp.

F. Smart Home Device vs. Smartphone

Fig. 9 shows the aggregated results heatmap of estimated sonar RR vs. ground truth. Fig. 9(a) and (b) compare smart home RR results using baseline algorithm and LuckyChirp. And fig. 9(c) and (d) compare smartphone RR results using baseline algorithm and LuckyChirp. Both comparisons show that LuckyChirp can efficiently distinguish respiration signals from motion signals and reject wrong RR measurements.

Fig. 9 also shows that the smart home device captures more RRs than smartphones. For 20 nights aggregated results: the smart home device captured 10074 RRs, while the smartphone had 6568 RRs. Fig. 10(a) and (b) show the CDF of RR detecting ratio (how many RRs are detected during the unit measuring time) for smart home devices and smartphones, respectively. Fig. 10(a) and (b) also demonstrate that smart home devices have a higher probability of detecting an RR than smartphones. Smart home devices also have better MAE than smartphones. The MAE for aggregated smart home device results is 0.48 ± 0.98BPM, which is half of smartphones' MAE 1.07 ± 1.67BPM.

There are several reasons for smart home and smartphone's performance difference: 1) Smart home devices have higher transmitting power, which results in higher SNR than smartphones. 2) Smart home devices have twice the smartphone's sampling rate, allowing them to capture more details and improve the processing results. 3) Smart home devices transmit chirps with a larger bandwidth (5kHz) than smartphones (2.5kHz). So smart home devices sonar's range resolution is half of smartphone sonar's. Thus, smart home devices have a better ability to capture tiny movements.

G. Different Time Window

We evaluated LuckyChirp with three distance searching windows (30, 60, 90 seconds). Fig. 10(c) shows the MAE vs. window sizes. Smart home devices have an MAE of 1.00 BPM, 0.48 BPM, and 0.60 BPM for 30, 60, and 90 seconds windows, respectively. Smartphones have an MAE of 1.59

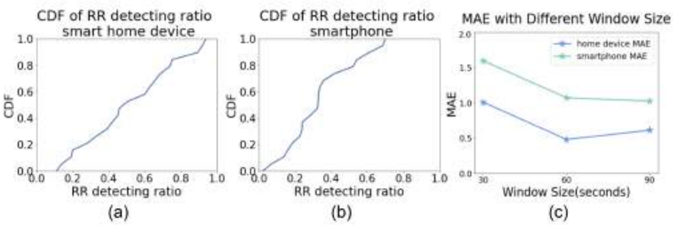

Fig. 10. The CDF of RR detecting ratio (a) from smart home devices, (b) from smartphones. (c) MAE with different window sizes.

BPM, 1.07 BPM, and 1.02 BPM for 30, 60, and 90 seconds windows, respectively. We noticed that increasing the distance searching window size from 30 to 60 seconds decreases MAE significantly because larger windows give more context information. However, increasing the window size from 60 to 90 seconds does not reduce MAE much. We think a 60 seconds window contains enough context information, so further increasing to 90-second does not help much.

H. LuckyChirp Classification Accuracy

We evaluated LuckyChirp's classification performance by precision. We assume that users' RR should not change dramatically in a short time, and any RR that is away from the central trend is considered as LuckyChirp classification failure. We used a rolling window to detect outliers that are more than 2 BPM away from the rolling window's mean. And we defined $precision = 1 - \frac{number\_of\_outliers}{number\_of\_positive\_samples}$. LuckyChirp has 95.97% precision for smart home data and 90.5% for smartphone data.

V. Discussion

A. Potential Daytime Sensing Scenarios

We explored several daytime scenarios that LuckyChirp can potentially sense the user's respiratory signal. Fig. 11 shows the scenario, its corresponding scalogram, and correlated signal at a specific distance. We demonstrated it is possible to opportunistically sense respiratory signals when the user is quietly reading on a laptop or a smartphone, even with subtle interactions on the trackpad or the phone touchscreen. Fig. 11 signal is classified as a valid respiration signal by the LuckyChirp classifier, and the estimated RR is 14 BPM, which aligns with the ground truth.

B. Multiple users

LuckyChirp is currently designed and evaluated on single-user scenarios. We tested that it is possible to monitor multiple users' RRs simultaneously, as long as they are reasonably apart(>range resolution). However, it is hard to identify each RR is from whom. This identification problem may be solvable by classifying users using breathing patterns or body reflection features. However, this is out of the scope of this paper.

C. Consent and Privacy

LuckyChirp only starts with the user's consent, and the user can stop or pause the passive sonar sensing at any time. LuckyChirp does not introduce audio privacy concerns, as it operates in the ultrasonic band. The correlation step removes all the audible sounds. The correlation step was implemented on smart home devices, so no audible signals were recorded. When implementing LuckyChirp on smartphones, correlation can also be on-device to avoid privacy concerns.

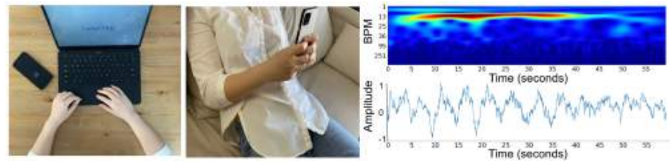

Fig. 11. Opportunistic sensing while the user is casually reading on a laptop or a phone (with touching and swiping gestures).

D. Overall Power Saving

We measured LuckyChirp's actual power consumption compared to FMCW on a Pixel 4 phone. We set the phone to transmit chirp pulses or FMCW continuously for 12 hours (with screen off, in airplane mode). After transmitting chirp pulses, the phone battery changed from 100% to 78%; After transmitting FMCW, the battery changed from 100% to 67%. The results show that, in practice, chirp pulses consume around 67% power of FMCW. Besides sonar, we also measured the power consumption of LuckyChirp's CNN. We deployed the CNN model on Pixel 4 using TensorFlow Lite. We repeated running the model 288000 times, which equals processing 12 hours of sonar data (1440 30-second windows$\times 200$ scalograms per window). The phone battery stayed at 100% after running the CNN model 288000 times (finished running in 3 minutes), which shows that the CNN consumes minimal power compared to sonar.

E. Pulsed Sonar

The time resolution for chirp pulse is $\frac{1}{T}$. We set $T = 50ms$ for LuckyChirp, which provides time resolution=20Hz. All breathing movements, including abnormal breathing (apnea, etc.), are much lower frequency (0.1-0.2Hz), so the pulsed sonar's time resolution is high enough to capture breathing details, including short-lived abnormalities. Since the breathing period is between 2 and 12 seconds, a duty-cycled sonar that turns off every few seconds will result in a high probability of missing a few respiration periods and other respiration events during the off time. Compared to duty-cycled sonar, LuckyChirp's pulsed sonar can monitor respiration continuously and will not miss any RR.

F. Limitations and Future Work

LuckyChirp's current CNN only detects uninterrupted periodic respiration signals. It may treat abnormal breathing signals (apnea, cessations, etc.) as user behavior artifacts and disregard them. This limitation can be addressed by adding clinician-labeled abnormal breathing data and re-training the model to detect breathing abnormalities. LuckyChirp sets the RR range from 5 BPM to 30 BPM, but in extreme cases, users' RR can be below 5 BPM or above 30 BPM. LuckyChirp did not train and evaluate such extreme data. However, a larger range of RR is possible with additional data. Due to LuckyChirp's opportunistic nature and the two limitations above, users should only use LuckyChirp as a "spot check" for their RRs instead of continuous abnormal breathing monitoring. For future work, it is also worth exploring using sonar to monitor heart rate, blood pressure, etc.

VI. Conclusion

This paper presents LuckyChirp, an opportunistic respiration-sensing model using sonar that provides a spot check for respiratory rates. We designed a cascaded sonar that solves both respiratory detection and estimation tasks. We introduced correlation-based pulse sonar that improves power efficiency and peak-SNR compared to FMCW. We evaluated LuckyChirp on over 327 hours of sonar data from off-the-shelf smartphones and smart home devices, and for the first time, compared their performance with the same environment setup. The evaluation results show that LuckyChirp can achieve sub-BPM accuracy on respiratory signals acquired from short time windows.

References

[1] J. F. Fieselmann, M. S. Hendryx, C. M. Helms, and D. S. Wakefield, "Respiratory rate predicts cardiopulmonary arrest for internal medicine inpatients," *Journal of general internal medicine*, vol. 8, no. 7, pp. 354–360, 1993.

[2] Z. Yang, P. H. Pathak, Y. Zeng, X. Liran, and P. Mohapatra, "Monitoring vital signs using millimeter wave," in *Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing*, 2016, pp. 211–220.

[3] D. Zito, D. Pepe, M. Mincica, F. Zito, A. Tognetti, A. Lanatà, and D. De Rossi, "Soc cmos uwb pulse radar sensor for contactless respiratory rate monitoring," *IEEE transactions on biomedical circuits and systems*, vol. 5, no. 6, pp. 503–510, 2011.

[4] M. Bartula, T. Tigges, and J. Muehlsteff, "Camera-based system for contactless monitoring of respiration," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2013, pp. 2672–2675.

[5] R. Nandakumar, S. Gollakota, and N. Watson, "Contactless sleep apnea detection on smartphones," in *Proceedings of the 13th annual international conference on mobile systems, applications, and services*, 2015, pp. 45–57.

[6] X. Wang, R. Huang, and S. Mao, "Sonarbeat: Sonar phase for breathing beat monitoring with smartphones," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2017, pp. 1–8.

[7] A. Trifan, M. Oliveira, and J. L. Oliveira, "Passive sensing of health outcomes through smartphones: systematic review of current solutions and possible limitations," *JMIR mHealth and uHealth*, vol. 7, no. 8, p. e12649, 2019.

[8] J. Torous, J. Lipschitz, M. Ng, and J. Firth, "Dropout rates in clinical trials of smartphone apps for depressive symptoms: a systematic review and meta-analysis," *Journal of affective disorders*, vol. 263, pp. 413–419, 2020.

[9] I. Wheatley, "Respiratory rate 3: How to take an accurate measurement," *Nursing Times*, vol. 114, no. 7, pp. 21–22, 2018.

[10] M. T. I. Aumi, S. Gupta, M. Goel, E. Larson, and S. Patel, "Doplink: Using the doppler effect for multi-device interaction," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, 2013, pp. 583–586.

[11] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 1515–1525.

[12] W. Mao, J. He, and L. Qiu, "Cat: high-precision acoustic motion tracking," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, 2016, pp. 69–81.

[13] W. Mao, M. Wang, W. Sun, L. Qiu, S. Pradhan, and Y.-C. Chen, "Rnn-based room scale hand motion tracking," in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–16.

[14] C. Zhang, Q. Xue, A. Waghmare, S. Jain, Y. Pu, S. Hersek, K. Lyons, K. A. Cunefare, O. T. Inan, and G. D. Abowd, "Soundtrak: Continuous 3d tracking of a finger using active acoustics," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–25, 2017.

[15] T. Wang, D. Zhang, Y. Zheng, T. Gu, X. Zhou, and B. Dorizzi, "C-fmcw based contactless respiration detection using acoustic signal," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–20, 2018.

[16] A. Wang, J. E. Sunshine, and S. Gollakota, "Contactless infant monitoring using white noise," in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–16.

[17] X. Song, B. Yang, G. Yang, R. Chen, E. Forno, W. Chen, and W. Gao, "Spirosonic: monitoring human lung function via acoustic sensing on commodity smartphones," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–14.

[18] R. Nandakumar, S. Gollakota, and J. E. Sunshine, "Opioid overdose detection using smartphones," *Science translational medicine*, vol. 11, no. 474, 2019.

[19] X. Xu, J. Yu, Y. Chen, Y. Zhu, L. Kong, and M. Li, "Breathlistener: Fine-grained breathing monitoring in driving environments utilizing acoustic signals," in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, 2019, pp. 54–66.

[20] H. Wan, S. Shi, W. Cao, W. Wang, and G. Chen, "Resptracker: Multi-user room-scale respiration tracking with commercial acoustic devices," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[21] A. Wang, D. Nguyen, A. R. Sridhar, and S. Gollakota, "Using smart speakers to contactlessly monitor heart rhythms," *Communications biology*, vol. 4, no. 1, pp. 1–12, 2021.

[22] T. Rahman, A. T. Adams, R. V. Ravichandran, M. Zhang, S. N. Patel, J. A. Kientz, and T. Choudhury, "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*, 2015, pp. 39–50.

[23] F. Adib, H. Mao, Z. Kabelac, D. Katabi, and R. C. Miller, "Smart homes that monitor breathing and heart rate," in *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, 2015, pp. 837–846.

[24] C.-Y. Hsu, A. Ahuja, S. Yue, R. Hristov, Z. Kabelac, and D. Katabi, "Zero-effort in-home sleep and insomnia monitoring using radio signals," *Proceedings of the ACM on Interactive, mobile, wearable and ubiquitous technologies*, vol. 1, no. 3, pp. 1–18, 2017.

[25] U. Ha, S. Assana, and F. Adib, "Contactless seismocardiography via deep learning radars," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–14.

[26] M. Dixon, L. Schneider, J. Yu, J. Hsu, A. Pathak, D. Shin, R. S. Lee, M. R. Malhotra, K. Mixter, M. McConnell *et al.*, "Sleep-wake detection with a contactless, bedside radar sleep sensing system," 2021.

[27] H. Wang, D. Zhang, J. Ma, Y. Wang, Y. Wang, D. Wu, T. Gu, and B. Xie, "Human respiration detection with commodity wifi devices: do user location and body orientation matter?" in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 25–36.

[28] P. V. Rouast, M. T. Adam, R. Chiong, D. Cornforth, and E. Lux, "Remote heart rate measurement using low-cost rgb face video: a technical literature review," *Frontiers of Computer Science*, vol. 12, no. 5, pp. 858–872, 2018.

[29] W. Chen and D. McDuff, "Deepphys: Video-based physiological measurement using convolutional attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 349–365.

[30] S. Kang, S. Kwon, C. Yoo, S. Seo, K. Park, J. Song, and Y. Lee, "Sinabro: Opportunistic and unobtrusive mobile electrocardiogram monitoring system," in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, 2014, pp. 1–6.

[31] W. Li, B. Tan, and R. Piechocki, "Passive radar for opportunistic monitoring in e-health applications," *IEEE journal of translational engineering in health and medicine*, vol. 6, pp. 1–10, 2018.

[32] A. Hein, *Processing of SAR data*. Springer, 2003.

[33] M. A. Cretikos, R. Bellomo, K. Hillman, J. Chen, S. Finfer, and A. Flabouris, "Respiratory rate: the neglected vital sign," *Medical Journal of Australia*, vol. 188, no. 11, pp. 657–659, 2008.