DeepApnea: Deep Learning Based Sleep Apnea Detection Using Smartwatches

Zida Liu, Xianda Chen, Fenglong Ma, Julio Fernandez-Mendoza, and Guohong Cao
The Pennsylvania State University
E-mail: {zjl5310, xuc23, fenglong, jfmendoza, gcao}@psu.edu

Abstract—**Sleep apnea is a serious sleep disorder where patients have multiple extended pauses in breath during sleep. Although some portable or contactless sleep apnea detection systems have been proposed, none of them can achieve fine-grained sleep apnea detection without strict requirements on the device or environmental settings. To address this problem, we present DeepApnea, a deep learning based sleep apnea detection system that leverages patients' wrist movement data collected by smartwatches to identify different types of sleep apnea events (i.e., central apneas, obstructive apneas, and hypopneas). Through a clinical study, we identify some special characteristics associated with different types of sleep apnea captured by smartwatch. However, there are many technical challenges such as how to extract informative apnea features from the noisy data and how to leverage features extracted from the multi-axis sensing data. To address these challenges, we first propose signal preprocessing methods to filter the raw accelerometer (ACC) data, smoothing away noise while preserving the respiratory signal and potential features for identifying sleep apnea. Then, we design a deep learning architecture to extract features from three ACC axes collaboratively, where self attention and cross-axis correlation techniques are leveraged to improve the classification accuracy. We have implemented DeepApnea on smartwatches and performed a clinical study. Evaluation results demonstrate that DeepApnea can significantly outperform existing work on identifying different types of sleep apnea.**

Index Terms—**Apnea Detection, Deep Learning, SmartWatch**

I. Introduction

Sleep apnea is a serious sleep disorder where patients have multiple extended pauses in breath during sleep. Sleep apnea is linked to many diseases, such as high blood pressure, chronic heart failure, depression, obesity, and daytime fatigue [1]. It is estimated that more than 22 million Americans suffer from sleep apnea [2]. Although the US government spends more than 150 billion on sleep apnea [3] every year, about 75% of people with moderate and severe apnea are still undiagnosed [4].

To diagnose sleep apnea, the commonly used method is the polysomnography (PSG) test, which requires the subjects to wear more than 20 wired sensors, including the pulse oximeter, pressure transducer, thermocouple, and electrodes placed at different parts of the body. It is uncomfortable for many patients and can even affect their sleep and the diagnosis results [5]. Moreover, such in-lab PSG test is expensive, cumbersome, and time-consuming, and thus many potential patients cannot be timely diagnosed, endangering their health.

In order to overcome these shortcomings of the PSG test, many contactless or wearable systems have been proposed for sleep monitoring and apnea detection. For example, with the wide deployment of wireless technology, many researchers [6] [7] [8] [9] leverage sound waves, WiFi or radio frequency signals to measure the chest movements during patients' sleep. The chest movement due to breathing can be identified by analysing the properties of the wireless signal, i.e., the channel state information or the shift in carrier frequency. Although wireless technologies can extract breathing information for monitoring sleep, they either require customized hardware, have strict environmental restrictions, and hence cannot be largely deployed or cannot detect abnormal breathing signals (i.e., sleep apnea).

Compared to these systems based on wireless technology, the wristband-based methods [10] [11] [12] and the geophone-based method [13] [14] can measure the respiration signal with widely adopted wearable devices such as smartwatches or through multiple geophone sensors. However, they can only provide some coarse-grained sleep data such as the respiratory rate, and they are not capable of detecting sleep apnea. *ApneaDet* [15] is the first smartwatch-based system which exploits the built-in sensors in smartwatch to detect sleep apnea. Specifically, it leverages the *accelerometer (ACC)* to monitor the wrist movements, then extracts respiratory information from the ACC data for apnea detection. However, it was designed for achieving binary classification, i.e., differentiating sleep apnea from normal sleep, which limits its general application.

There are three different kinds of respiratory events associated with sleep apnea (i.e., central apneas, obstructive apneas, hypopneas), and distinguishing these different kinds of respiratory events is very important. This is because different respiratory events have different etiology (e.g., central apneas are caused by the brain stopping the breathing process, while obstructive apneas are caused by local collapsibility of the upper airway) and they have links to different diseases. Thus, identifying all three types of sleep apnea can help clinicians provide better diagnosis and treatment [5].

There are many technical challenges for identifying three different types of sleep apnea. First, the wrist movement generated by breath or lung movement is very subtle. The raw ACC data recorded by smartwatch contains a large amount of noise, which makes it harder to extract the respiratory information. Second, existing machine learning features used for binary classification do not work well for multi-classification, i.e., identifying three types of sleep apnea. This is because it is

979-8-3503-2603-1/24/$31.00 ©2024 IEEE

relatively easier to differentiate normal sleep from abnormal sleep apnea, but it is much harder to identify different kinds of sleep apnea events. Third, based on the sleeping posture and the wrist position, the three ACC axes may carry different amount of respiratory information. How to leverage such information to identify sleep apnea remains as a challenge.

In this paper, to address these challenges, we propose a smartwatch-based system named *DeepApnea*, which can detect different types of sleep apnea. We first propose signal preprocessing methods to filter the raw ACC data, smoothing away noise while preserving the respiratory signal and potential features for identifying sleep apnea. Then, we design a deep learning architecture to extract features from three ACC axes collaboratively. Specifically, we apply self attention technique to accentuate more significant features and apply cross-axis correlation technique to exploit the correlations among different axes. The extracted deep features and the correlation information are merged through aggregated classification to further improve the classification accuracy.

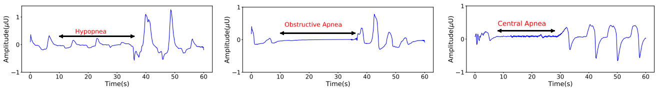
The main contributions of the paper are as follows.

- To the best of our knowledge, this is the first work to identify three types of sleep apnea (hypopneas, obstructive apneas, central apneas) only using wrist-worn ACC data.
- We propose signal preprocessing techniques to extract accurate representations of the respiratory signal from the raw noisy ACC data.
- We design a deep learning model to automatically extract informative apnea features from three ACC axes and wisely fuse these features to improve performance.
- We have implemented DeepApnea on smartwatches and performed clinical study. Evaluation results show that DeepApnea significantly outperforms existing work on identifying three types of sleep apnea.

II. Background and Motivation

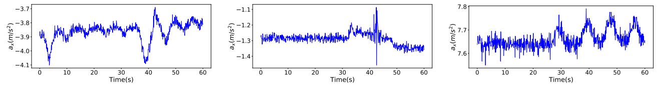
There are three types of respiratory events associated with sleep apnea [16]. A Central Apnea occurs when the subject holds his/her breath for a long period of time, typically 10 to 30 seconds. During central apnea, the human brain fails to provide the signal to inhale, resulting the absence of breathing effort. A hypopnea occurs when the subject's breathing becomes shallow. Specifically, patients will lose 30% to 90% of normal airflow. This procedure usually lasts more than ten seconds. An Obstructive Apnea occurs when there is a complete or partial blockage of the upper airway during sleep. The subject makes an effort to pull air into the lungs, however, the air does not reach the lungs because of blockage. Fig. 1 shows the airflow measured by the nasal pressure sensor for different sleep apneas types.

Based on our previous clinical study [15], we demonstrated the feasibility of apnea detection with a smartwatch. This clinical study was conducted with twenty subjects at Penn State Hershey Sleep Research & Treatment Center. Each subject wears a smartwatch to collect the ACC data of the wrist movement, during a regular PSG test.



(a) hypopnea (b) obstructive apnea (c) central apnea

Fig. 1. The airflow (nasal pressure)



(a) hypopnea (b) obstructive apnea (c) central apnea

Fig. 2. The raw ACC data (x axis), which corresponds to the subfigures in Fig. 1.

Fig. 2 shows the raw ACC data collected using smartwatches. In general, respiration leads to the periodic subtle movement of the chest, abdomen, arms and wrists and these movements can be recorded by the ACC in smartwatch. Fig. 2(a) shows the ACC data corresponding to Fig. 1(a). Between the 12th and 36th seconds, labeled by the technician, a hypopnea happens. During this time, the respiration becomes shallow, so the amplitude change of airflow will decrease and slimier change is reflected on the ACC data. Fig. 2(b) (corresponding to Fig. 1(b)) presents the ACC data during an obstructive apnea. In obstructive apnea, after a respiratory blockage for several seconds, the subject is likely to make one or several intense breaths before returning to normal breathing, leading to the signal spike around the 42th seconds. In Fig. 2(c), since the subject holds breath during central apnea, the ACC data are flat and there is no intense spike after this holding.

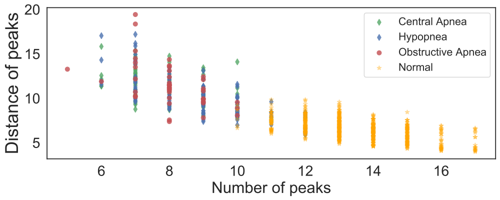
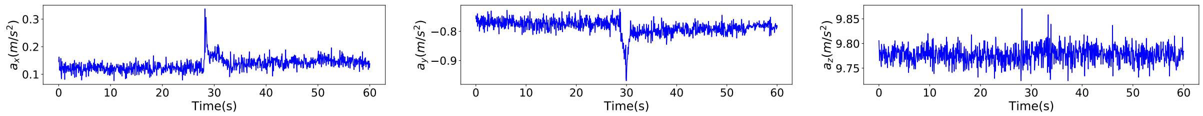


Fig. 3. The infeasibility of using hand-crafted features to differentiate three types of sleep apnea.

Based on this study, we can see that different sleep apnea can lead to different pattern of ACC data. With machine learning techniques, sleep apnea can be identified. Unfortunately, since the ACC data is very noisy and the wrist movement is very subtle, it is hard to use simple machine learning techniques to identify different types of sleep apnea (i.e., multiclass classification) although it is possible to differentiate sleep apnea events from normal sleep (i.e., binary classification) which was the design goal of [15]. For example, the number of respiration peaks and the maximum distance between two consecutive respiration peaks are commonly used as features [15] [17] for sleep apnea detection. Fig. 3 visualizes different types of apnea events using a two-dimensional scatter plot, where the horizontal dimension represents one feature (the number of peaks) and the vertical dimension represents another feature



(a) X axis

(b) Y axis

(c) Z axis

Fig. 4. The ACC data along three axes in an obstructive apnea.

(maximum distance of peaks). Although normal sleep events can be easily distinguished from apnea events, different types of apnea events are overlapped with each other and there is no obvious decision boundary to distinguish them.

To address this problem, we propose deep learning techniques to identify different types of sleep apnea. Although deep learning has been proved to help extract more representative features than traditional machine learning methods in many areas, simply applying widely used deep learning models such as CNN and LSTM to our problem may not work. This is because they only treat the triaxial data as one single input feature without considering the heterogeneity among different axes. In practice, based on the sleeping posture and the wrist position, the collected ACC data along each axis may be different. Instead of only using the data from one axis, this multi-dimensional data can help us obtain more information. However, if the ACC data is not processed properly, it may have adverse effects. For example, Fig. 4 shows the raw data recorded from one obstructive apnea event. X axis and Y axis have a similar data pattern (e.g., they both contain a signal spike near $29nd$ second) whereas Z axis does not have it. Simply fusing the data from three axes through basic deep learning operations (e.g., average pooling) may result in large errors.

To deal with this problem, we consider the data reliability from each axis. For example, we can assign a higher weight to more informative axes X, Y and lower weight to less informative axis Z in Fig. 4. However, in practice, it is hard to manually determine such weights due to the data heterogeneity. Since the ACC data from three axes jointly represents the wrist movement, there exist correlations between different axes' ACC data (e.g., Y is like an inverse of X in Fig. 4 whereas Z is less correlated with X and Y). Based on existing research [18], the learning performance can be improved by leveraging the correlations between different data sources or feature subsets. Since data from each axis can be treated as a single data source of the patient's wrist movement, the detection accuracy can be improved by exploiting the correlations among different axes. Thus, we propose the *Cross-axis correlation* technique (section V-C), to explore correlations between different axes and automatically assign different weights to different axes.

Additionally, within a single data segment collected from one axis, certain parts of the data may contain more valuable information than others. For example, a signal spike part might be more representative of recognizing obstructive apnea. Thus, it is crucial to focus more on the informative parts. To achieve this objective, we employ *self-attention* techniques (section V-B), which assign higher weights to the informative parts, thereby further enhancing the detection accuracy.

III. System Overview

The overall design of DeepApnea is shown in Fig. 5. During sleep, the smartwatch on the subject's wrist records data generated by the accelerator sensor. When the subject wakes up, the smartwatch stops recording and the collected data can be forwarded to the subject's smartphone through Bluetooth for further analysis. The raw acceleration data is preprocessed by the signal prepossessing module, and then forwarded to a deep learning module which extracts representative features and classifies into four sleep events - *normal sleep*, *hypopnea*, *obstructive apnea*, and *central apnea*.

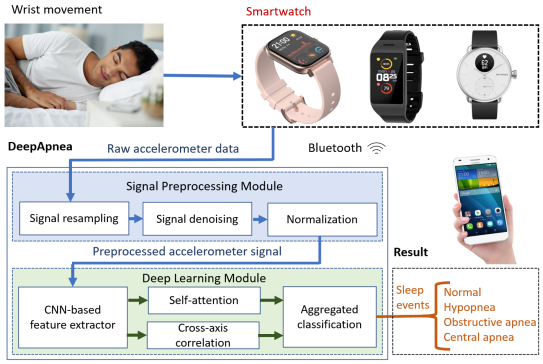


Fig. 5. System architecture of DeepApnea.

- **Signal preprocessing module**: The raw ACC data collected by the smartwatch contains a large amount of electronic and mechanical noises, which make it harder for the deep neural network to extract useful features from the time-series data. To deal with these problems, we have the following three steps for signal preprocessing: *1) signal resampling*: Resample the raw data at a fixed rate to mitigate fluctuations from the actual sampling rate caused by system operations. *2) signal denoising*: Utilize a signal filter to remove unnecessary noise while preserving the breathing signal. *3) signal normalization*: Normalize the data to mitigate high variations introduced by varying sleep poses and wrist positions. The details of these steps will be presented in Section IV.
- **Deep learning module**: In order to extract informative features from the prepossessed data and effectively leverage signals from three axes, the deep learning module has the following steps. First, the prepossessed data of each axis is fed to its specific *CNN-based feature extractor* to obtain the corresponding deep features. Second, *Self attention* and *Cross-axis correlation* techniques are applied to these deep features to obtain the weighted deep features of each axis and the correlation information between any two axes. Lastly, both the weighted deep features

and correlation information are merged with *Aggregated classification* to classify the sleep event, i.e., normal, hypopnea, obstructive apnea, or central apnea. The details of these steps will be presented in Section V.

IV. Signal Preprocessing

A. Signal Resampling

To collect data with smartwatch, we first setup a sampling rate higher than the respiration rate so that all useful signal can be preserved. However, in most commercial systems, the real sampling rate may fluctuate around the expected sampling rate due to many uncontrollable system operations. For example, we use a smartwatch (Huawei Watch 2) to collect the sleep data. Although the sampling rate was set to *SENSOR -DELAY-GAME* (i.e., 50Hz) through the Android API, the real sampling rate varies from from 40 Hz to 60 Hz. This will create problems for running the deep learning model which require the input data to have fixed size. Therefore, we need to re-sample the collected data with a certain sampling rate before sending it to the deep learning model.

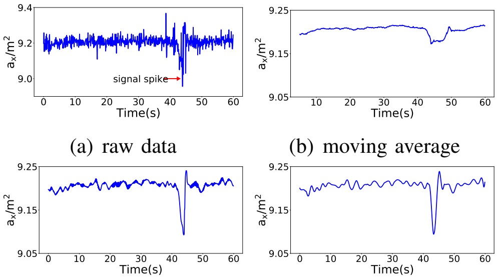
The Fourier method [19] is adopted to resample the raw data, because it can avoid information distortion (e.g., aliasing) during resampling and well preserve the information of the original signal. Fourier method first leverages Discrete-time Fourier Transform (DTFT) to transform the accelermoter signal into frequency domain. Then, during Inverse Discrete Fourier Transform (IDFT), we can eliminate aliasing by limiting the highest frequency to half of the sampling rate and obtain resampled data points with the same time interval. The procedure is expressed in the following equations: $\hat{X}[k] = \sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}nk} x[n]$ and $\dot{x}[m] = \frac{1}{M}\sum_{k=0}^{M/2-1} e^{j\frac{2\pi}{M}mk}\hat{X}[k]$.

The first equation represents DTFT, where j is the imaginary unit, N is the number of data points of the raw signal and $x[n]$ denotes the n^{th} raw data points. Through it, we can get different frequency components $\hat{X}[k]$, where $k = 0, 1, 2, ..., N-1$. The second equation represents IDFT, where M is the number of data points after resampling and $x[m]$ denotes the m^{th} data point of the resampled signal. When calculating $\dot{x}[m]$, we only consider 0^{th} to $(M/2-1)^{th}$ frequency components to avoid aliasing. Finally, we can obtain the rasampled signal $\dot{x}[m]$ from the raw signal $x[n]$.

B. Signal Denoising

The raw signal collected by the smartwatch contains a large amount of electronic and mechanical noise, which makes it harder for the deep neural network to extract useful features from the time-series data. Therefore, we need to design an effective filter to filter out the noise.

The moving average filter [20] is a widely-used filter for denoising. Since this method simply averages different subsequences of the signal, it also eliminates potential useful information for apnea classification. For example, Fig. 6(a) shows the raw ACC signal representing an obstructive apnea event. There is a signal spike when the subject tries to make an intense breath after an obstructive apnea event, and such a signal spike can serve as the feature for distinguishing obstructive apnea from other apnea classes. However, as shown in Fig. 6(b), the moving average filter smooths away the signal spike.



(a) raw data

(b) moving average

(c) TV filter

(d) ADA

Fig. 6. The raw and filtered ACC data with different denoising methods.

To preserve all useful signal information, e.g., the signal spike, most existing research [15] relies on the Total Variation filter (TV filter) [21] for signal denoising. Although, TV filter can preserve the respiratory spikes well, it cannot remove the low-amplitude noise as shown in Fig. 6(c). This is because TV filter only aims to minimize the sum of the variation between two adjacent signal values over the whole signal sequence without considering that the denoised signal should be locally smooth. As a result, TV filter eliminates the high-amplitude noise, but low-amplitude noise still remains (e.g., the $20th$ to $25th$, and the $48th$ to $50th$ seconds in Fig. 6(c)).

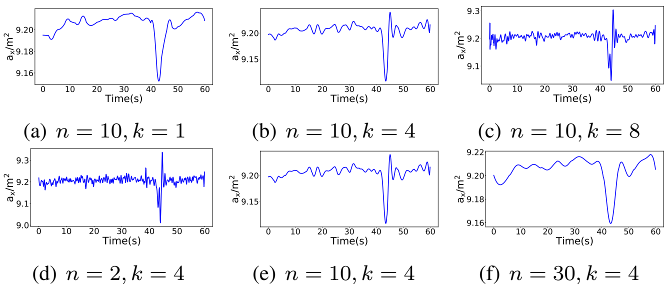
To keep the periodic respiratory information and remove all unnecessary noise, we design an adaptive denoising algorithm based on [22]. Our goal is to not only eliminate the high-amplitude noise but also achieve local smoothness. To achieve this goal, we first divide the raw signal segment into partially overlapped subsegments. Then, each subsegment is denoised separately based on their own signal trend and merged together by using a linear weighting mechanism.

The algorithm is shown in Algorithm 1. The input signal is divided into m subsegments. Each subsegment contains $2n+1$ data points, where adjacent subsegments overlap by $n+1$ points. For each subsegment, a polynomial function with order K is used to fit its data to extract the respiratory signal trend and eliminate noise. Then the denoised subsegments are concatenated together by leveraging the overlapping area, where a weighted sum is used to recalculate the data points in the overlapping area. The weighted sum ensures symmetry and effectively eliminates any jumps or discontinuities around the boundaries of neighboring subsegments, and the subsegment polynomial fitting ensures the local smoothness. Fig. 6(d) shows that designed adaptive denoising algorithm (ADA) can preserve the useful signal and filter out the noise.

The algorithm contains two important parameters: K and n. As shown in Fig. 7, choosing different K and n can lead to different denoising results. Since low polynomial order lacks the ability to represent complex signal trends, setting K to a small value may filter out the useful information. For example, as shown in Fig. 7 (a), the periodic respiratory movement signal is also filtered out. On the other hand, if K is too large

Algorithm 1: Adaptive Denoising Algorithm

```
Input: (1) raw ACC data $D$; (2) half length of
  subsegment $n$; (3) polynomial order $K$;
Output: denoised acclerometer data array $P_{denoised}$;
Function ADA($D$, $n$, $K$):
    Initialization: an empty array $P$;
    $D \rightarrow \{d^{(1)}, d^{(2)}, d^{(3)}, .., d^{(m)}\}$;
    for each data segment $d^{(i)}$ do
        $f \leftarrow \texttt{polyfit}(d^{(i)}(x), K)$;
        $p^{(i)}(x) \leftarrow f(x)$;
    end for
    $P$.append($p^{(1)}(x)$), where $x = 1, .., n$;
    for every overlap part of two adjacent denoised
     subsegment do
        $p^{(j,j+1)}_{overlap}(x) \leftarrow$
          $w_1 * p^{(j)}(x+n) + w_2 * p^{(j+1)}(x)$, where
          $x = 1, .., n+1$, $w_1 = (1-(x-1))/n$,
          $w_2 = (x-1)/n$;
        $P$.append($p^{(j,j+1)}_{overlap}$(x));
    end for
    $P$.append($p^{(n)}(x)$), where $x = n+1, .., end$;
    $P_{denoised} \leftarrow P$;
    Return: $P_{denoised}$;
```



(a) $n = 10, k = 1$ (b) $n = 10, k = 4$ (c) $n = 10, k = 8$

(d) $n = 2, k = 4$ (e) $n = 10, k = 4$ (f) $n = 30, k = 4$

Fig. 7. The filtered ACC data (along X axis) with different ADA parameter settings.

as shown in Fig. 7 (c), overfitting happens and there are still too much noises.

With a small n, as shown in Fig. 7 (d), the subsegment is too small and there is not enough information to fit the polynomial to filter out the noise. With a large n, as shown in Fig. 7 (f), the subsegment is too long and the fitted polynomial lacks the capability to represent all the data variations implying useful respiratory information. We experimentally determine the combinations of K and s and found that when $K = 4$ and $n = 10$, the noise is eliminated and the periodic respiratory movement can be preserved well, as shown in Fig. 7 (b) (e). Thus, we use such a setting in the rest of the paper.

V. Deep Learning Architecture

In order to extract informative features from the prepossessed data and effectively leverage signals from three axes, we propose a deep learning architecture as shown in Fig. 8. First, the prepossessed data of each axis is fed to its specific *CNN-based feature extractor* to obtain the corresponding deep features. Second, these deep features are sent to two modules in parallel - *Self attention* and *Cross-axis correlation* for obtaining the weighted deep features of each axis and the correlation information between any two axes. Lastly, both the weighted deep features and correlation information are merged in the *Aggregated classification* module to classify the sleep event. The rest of this section presents the details of these four modules.

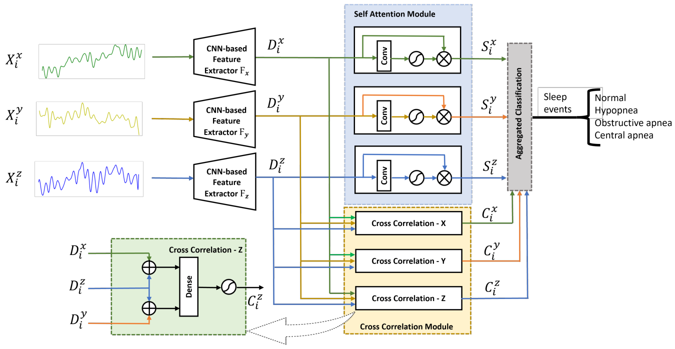


Fig. 8. The architecture of the proposed DeepApnea model.

A. CNN-based Feature Extractor

Traditional machine learning methods can only extract apnea features based on the designers' domain knowledge without exploiting the unknown apnea features, hindering their capability of distinguishing different types of apnea events. To solve this problem, we build a CNN to automatically extract apnea features from the collected data.

The CNN consists of four convolutional layers and two max pooling layers. Each layer is a non-linear operation. The multi-layer non-linear operations make the obtained features more sensitive to different apnea types, and less sensitive to irrelevant variations coming from other factors such as physical device, patient, environment, etc. The parameters of each layer are shown in Table I. In addition, we adopt a batch normalization layer after each convolutional layer and a dropout layer after each pooling layer, respectively, to prevent over-fitting.

Since different axes record wrist movements in different directions, we prepare a separate feature extractor for each axis. Let $\mathrm{X} = \{\mathrm{x}_i^x, \mathrm{x}_i^y, \mathrm{x}_i^z\}_{i=1}^n$ represents all the samples of input data, where x_i^x denotes the time-series signal along the X axis of the i^{th} sample. The feature extractor is denoted as $\mathcal{F}(\mathrm{x}, \theta_F)$, where θ_F represents the trainable parameters of the extractor. After feature extraction, we can obtain the deep features D_i^x, D_i^y, and D_i^z along the three axes:

$$D_i^x = \mathcal{F}(\mathrm{x}_i^x, \theta_{F_x}), D_i^y = \mathcal{F}(\mathrm{x}_i^y, \theta_{F_y}), D_i^z = \mathcal{F}(\mathrm{x}_i^z, \theta_{F_z}) \quad (1)$$

Specifically, D_i^x, D_i^y and $D_i^z \in \mathbb{R}^{\widetilde{T} \times F}$, where $\widetilde{T}$ represents the spatial dimension and F denotes the feature dimension. After obtaining the deep feature of each axis, they will be sent to the *Self attention* and *Cross-axis correlation* module for further processing.

TABLE I
THE ARCHITECTURE OF THE CNN-BASED FEATURE EXTRACTOR.

Layer	Size In	Size Out	Filter
conv1	584 × 1	98 × 64	24, 6
conv2	98 × 64	49 × 64	8, 2
pool1	49 × 64	24 × 64	2, 2
conv3	24 × 64	24 × 128	4, 1
conv4	24 × 64	24 × 128	4, 1
pool2	24 × 128	12 × 128	2, 2

B. Self Attention

The deep feature from different axes may carry different amount of respiratory information due to various reasons, e.g., the pose of patients' wrists and hence the smartwatch. It is natural to assign different weights to the data collected from different axis for measuring each axis's equality or reliability. Moreover, even within a single data segment collected from one axis, some parts of the data may contain more useful information than other parts. For example, the part of signal spike is more representative to recognize obstructive apnea. Therefore, in addition to setting different weights for different axes, we also set different weights for different parts of the data segment.

To achieve this goal, we leverage the self-attention mechanism. It is a weighted aggregation method to obtain better representations of the signal and it has been successfully applied to many deep learning applications such as sentence embedding [23], speech and activity recognition [24] and disease diagnosis [25]. Self-attention mimics cognitive attention. For a single data segment, it enhances some parts while diminishing other parts. Specifically, it forces deep neural network to devote more focus on that small but important part of the input data. After extracting the deep features from each axis, we apply self-attention mechanism on them separately to assign intra-axis weights. More specifically,

$$
\begin{aligned}
A_i^x &= \sigma(W^x D_i^x + b^x), \quad S_i^x = A_i^x \cdot D_i^x \\
A_i^y &= \sigma(W^y D_i^y + b^y), \quad S_i^y = A_i^y \cdot D_i^y \\
A_i^z &= \sigma(W^z D_i^z + b^z), \quad S_i^z = A_i^z \cdot D_i^z
\end{aligned}
\tag{2}
$$

where is the convolution operation and $\cdot$ is dot product. W and b are trainable parameters of one-layer convolution operation. $A_i \in \mathbb{R}^{\widetilde{T} \times F}$ is the self-attention weight which is controlled by the signal sequence itself. With such a mechanism, our model can learn to focus more on the informative locations of each axis. Finally, we can obtain the weighted deep features S_i^x, S_i^y and S_i^z.

C. Cross-axis Correlation

Although the self-attention module can assign weights automatically for each axis to obtain weighted features, it only considers the signal from each axis independently without leveraging the correlations among them. Since the ACC data from all three axes can record the wrists' movement information collaboratively during patients' sleep, as mentioned at the end of Section II, we should leverage the correlation between different axes to assign appropriate weight for each axis, which can further improve the performance.

We assess the correlation between different axes by analyzing the similarity of deep features extracted from each axis. High similarity suggests that the two axes convey similar apnea-related information. When such high similarity is identified, our model puts more weights on the information from these axes. In deep learning, the similarity of two feature vectors is often assessed using element-wise difference [26] [27]. Thus, we employ this method to quantify the correlation. Taking X-axis as an example, the cross-axis correlation vector is as follows:

$$
\begin{aligned}
&D_i^{x|y} = D_i^x - D_i^y, \quad D_i^{x|z} = D_i^x - D_i^z \\
&C_i^x = \sigma(W^{x|yz} \odot \{D_i^{x|y} \oplus D_i^{x|z}\} + b^{x|yz})
\end{aligned}
\tag{3}
$$

where $\odot$ represents the convolutional operation, and $\oplus$ represents the concatenate operation. $W^{x|yz}$ and $b^{x|yz}$ are trainable parameters. The superscript denotes the axis relationship, e.g., $D_i^{x|y}$ means the correlation between X-axis and Y-axis. After obtaining the correlation vector $D_i^{x|y}$ and $D_i^{x|z}$, we concatenate them together and use one fully connected layer to extract more information to represent X-axis correlation with the other two axes, which is C_i^x. The cross-axis correlations C_i^y for Y-axis and C_i^z Z-axis are calculated by the same way.

After capturing the correlation vectors C_i^x, C_i^y, C_i^z, they are sent the *Aggreded classification* model and served as the cross-axis weights. By incorporating all the mutual correlations among the three axes, our deep learning model can assign higher weight to more informative axis and lower weight to less informative axis.

D. Aggregated Classification

To capitalize on the valuable features from all axes and allocate suitable weights to each axis, we conduct a dot-multiplication between the output vectors of the self-attention and cross-axis correlation modules. This enables our framework to emphasize the informative parts within a single data segment of each axis and also prioritize the more informative axes. Specifically, we have:

$$
\begin{aligned}
&H_i^x = S_i^x \odot C_i^x, \quad H_i^y = S_i^y \odot C_i^y, \quad H_i^z = S_i^z \odot C_i^z \\
&H_i^{xyz} = H_i^x \oplus H_i^y \oplus H_i^z, \quad F_i^{xyz} = \text{Pooling}(H_i^{xyz})
\end{aligned}
\tag{4}
$$

where $H_i^x, H_i^y, H_i^z \in \mathbb{R}^{\widetilde{T} \times F}$ are the final features of each axis combining both the result of self-attention module and cross-axis correlation module. Then, fused feature H_i^{xyz} can be obtained by concatenating H_i^x, H_i^y, H_i^z along the feature dimension. After applying an average pooling layer on H_i^{xyz}, the final fused feature $F_i^{xyz} \in \mathbb{R}^{\widetilde{T} \times F}$ is calculated. Finally, the hybrid fused feature is fed into a 2-layer fully-connected network. The first layer is a fully-connected layer, which is activated by the ReLU function, while the the second layer is a softmax layer to calculate the probability of the four types of sleep events. The class with the maximum probability will be considered as the classification result.

VI. EVALUATIONS

A. Clinical Study

We conducted a clinical sleep study at at Penn State Milton S. Hershey Medical Center, with approval by our Institutional

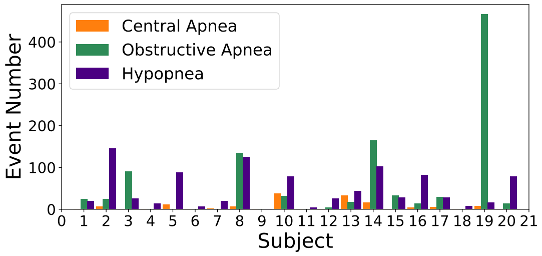


Fig. 9. The number of central apnea, Hypopnea, and obstructive apnea for each subject.

Review Board (IRB). The details of the clinical study were presented in [15]. The study includes twenty subjects (eight males and twelve females), and their ages vary from 36 to 72, with an average of 59.3. The subjects presented certain diversity in terms of the severity of sleep apnea and they were prescribed to undergo the regular polysomnography (PSG) study without receiving continuous positive airway pressure therapy. During the PSG study, all the patients were required to wear Huawei Watch 2 to collect the ACC data. The smartwatch is fully charged before recording the sensor data, to make sure that it is able to record the sensor data for whole night which is around eight hours. The smartwatch and the PSG equipment time are synchronized at the beginning of recording so that we can obtain the corresponding period of the sensor data when apnea events occur. The sleep apnea events are labeled by the sleep physician as the ground truth according to patients' PSG test. In total, we set the window size to be 60 seconds and recorded 2822 normal sleep events, 1018 obstructive apneas, 125 central apneas, and 818 hypopneas. Fig. 9 shows the number of sleep apnea events of each subject. Some subjects suffer from severe obstructive apnea but mild hypopnea such as subject 3 and subject 19, whereas others have more hypopnea but less obstructive apnea. In general, the number of central apnea is much smaller than that of obstructive apnea and hypopnea.

B. Comparing Methods

Since there is no existing work for identifying four types of sleep apnea events using purely wrist-worn ACC, we choose the following relevant methods as baselines.

1) Traditional machine learning methods: Many traditional machine learning approaches such as naive bayes **(NB)**, decision tree **(DT)** , random forest **(RF)**, adaboost **(ABT)** and support vector machine **(SVM)** have been widely used for recognizing time-series signal. For example, [28] successfully identifies fingerprints by recognizing fingerprint-induced sonic waves through LR, SVM and RF. *ApneaDet* [15] applies RF on different hand-crafted features of the ACC signal (e.g., peak distance, peak number, and peak amplitude) and realizes high recognition accuracy on binary sleep apnea classification task. In our evaluation, we compare with **NB, DT, SVM, RF,** and **ABT**. For fair comparison, we use the same signal processing pipeline introduced in *ApneaDet* to extract hand-crafted features, and use the same hand-crafted features such as peak distance, peak number, and peak amplitude introduced in *ApneaDet* as input features. Note that *ApneaDet* has the same performance as RF since it uses RF as the classifier.

2) Deep learning methods: Comparing to traditional machine learning methods, deep learning based methods have been proved to be more effective on analysing time-series signal in many applications. **AHF-CNN** [29] adopts a 6-layer convolutional neural network to automatically extract features based on the ACC data collected by IoT devices for human fall detection. However, this method only considers the triaxial ACC data as one single input feature with three dimensions without considering different modalities of the three inputs. For fair comparison, we also compare to the following two methods which consider multi-modal or multi-view data as their inputs. The first is **MM-CNN** [30], which designs a multi-channel CNN model for learning the features from different types of polysomnography signals (e.g., EEG, EMG, and EOG) to distinguish sleep stages. Although, for each type of PSG data, MM-CNN adopts a separate CNN channel to learn the signal's temporal context information, it treats all the channels equally and does not exploit the consensual and complementary information between them. The other one is **DeepSense** [31], which is a deep learning framework for analysing the signals from different mobile senors. It first converts the original signal of different mobile sensors into the frequency domain, and then leverages CNN and RNN to take advantage of the interactions among different input modalities. Although DeepSense has been demonstrated being effective in multiple challenging tasks through learning the correlation between different types of input signals, the quality of these inputs is not well considered. In DeepApnea, our self-attention module are designed to address this problem.

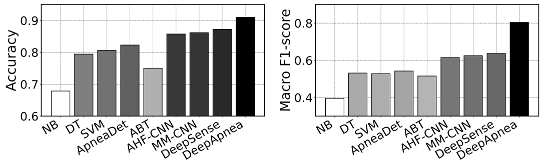
C. Experiment Setup

All the collected raw ACC data are preprocessed based on the techniques introduced in Section IV, i.e., resample at 8Hz, signal denoising, and normalization. Then, we train and examine the proposed DeepApnea model based on the processed data.

To train our DeepApnea model, we use categorical cross-entropy to calculate the training loss and adopt the Adam optimizer for updating the model's parameters. All the parameters are initialized using HeNormal initializer and we train the model for 500 epochs with initial learning rate of 0.005. Specially, we apply the 3-fold cross-validation for apnea classification. The final results are calculated based on the mean values of these cross-validation experiments. The performance is measured in terms of accuracy and F1-score.

D. Overall Performance Comparison

We compare the overall performance of DeepApnea with different machine learning methods introduced in Section VI-B, and the results are shown in Fig. 10. As can be seen from the figure, DeepApnea achieves the best performance in terms of accuracy and Macro-F1 scores. By comparing Fig. 10 (a) and (b), we can see that the performance improvement of



(a) Accuracy (b) Macro-F1 score

Fig. 10. Overall performance comparison.

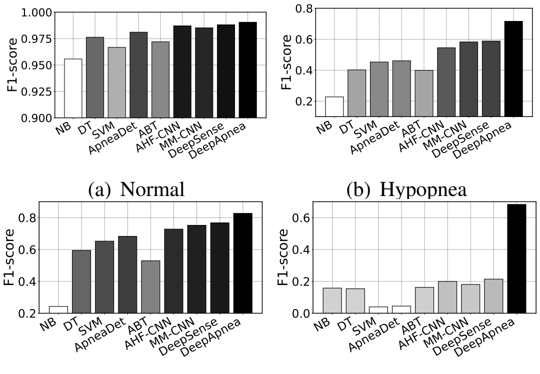
DeepApnea is much higher when Macro-F1 score instead of accuracy is used as the performance metric. As explained in the last section, accuracy is a good metric to show the overall classification result. Due to the imbalance of the data set, the result is dominated by the classification of normal events, and it cannot accurately reflect the classification results of different types of sleep apnea events. By treating different sleep events equally, macro-F1 score can better reflect the classification results of different types of sleep apnea events.

Traditional machine learning methods do not perform well on distinguishing different types of apneas. For instance, while ApneaDet [15] can achieve a high F1-score (Fig. 11 (a)) on binary classification task, its F1-score drops to around 55% when applied to this multi-classification task. Compared to traditional feature-based machine learning methods, deep learning methods perform better. As shown in Fig. 10 (a), AHF-CNN, MM-CNN, DeepSense, and DeepApnea have higher accuracy than NB, DT, SVM, ApneaDet, ABT. Similar advantage can also be seen in Fig. 10 (b). This is because traditional feature-based machine learning method only considers the properties of respiratory peaks as the hand-crafted features which ignores the signal trend and other potentially useful apnea characteristics. On the other hand, deep learning-based methods do not adopt hand-crafted features and automatically learn the appropriate features for sleep apnea classification.

Compared to other deep learning based methods, our DeepApnea model can achieve much better performance. Specifically, compared to AHF-CNN, our model improves the accuracy by 5.2% and improves the macro-F1 score by 18.9%. This is because AHF-CNN only considers the triaxial ACC data as one single feature without considering different modalities among the three inputs. As a result, the information from different axes cannot be effectively leveraged, and hence underperforms our model. Compared to MM-CNN, our model improves the accuracy by 4.9% and improves the macro-F1 score by 17.9%. Although MM-CNN considers the ACC signal from each axis, it treats data from each axis equally and does not take advantage of the correlation between different axes. Compared to DeepSense, our model improves the accuracy by 3.8% and improves the macro-F1 score by 16.7%. Although DeepSense extracts the correlation information between different axes, it does not consider the quality of the input signal, and hence underperforms our model.

E. Per-class Performance

In this subsection, we compare the performance of different methods on classifying sleep events into four types: normal, hypopnea, obstructive apnea, and central apnea. For normal sleep events, as shown in Fig. 11 (a), traditional machine learning methods can achieve almost perfect performance (i.e., about 97% using DT, RF, and ABT) based on hand-crafted features introduced in *ApneaDet* [15]. This is consistent with the results in [15], which only focuses on classifying sleep events into normal and sleep apnea.



(a) Normal (b) Hypopnea

(c) Obstructive apnea (d) Central apnea

Fig. 11. Per-class performance comparisons.

For hypopnea, obstructive and central apnea, as shown in Fig. 11 (b), (c) and (d), deep learning-based methods outperform traditional methods. In Fig. 11 (d), we can see that SVM and ApneaDet underperform other traditional methods. This is attributed to their necessity to map simple hand-crafted features into a highly dimensional space, leading to potential overfitting with limited central apnea data (Hughes Phenomenon). Different from APT-CNN, MM-CNN and DeepSense, DeepApnea leverages self-attention and cross-axis correlation modules to obtain better features and adopts several techniques for preventing overfitting such as the batch-normalization layers and dropout layers. Thus, DeepApnea can achieve the highest F1 score on central apnea.

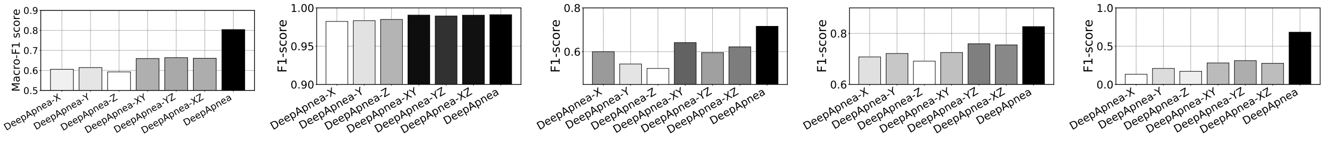
Overall, our DeepApnea model can achieve 99.3%, 71.6%, 82.8% and 68.2% F1 score for normal sleep, hypopnea, obstructive apnea, and central apnea.

F. Axis Data Fusion Study

In DeepApnea, data from three axes are leveraged. In this section, we demonstrate why such data fusion is necessary for improving performance. We compare DeepApnea with the following simplified models.

- DeepApnea-X (DeepApnea-Y/DeepApnea-Z): It only takes the ACC data from X-axis (Y-axis/Z-axis) as the input. Since there is only data from a single axis, there is no cross-axis correlation.
- DeepApnea-XY (DeepApnea-YZ/DeepApnea-XZ): It takes the ACC data from X-axis and Y-axis (Y-axis and Z-axis)/(X-axis and Z-axis) as the input. Since data from two axes are used, cross-axis correlation and self-attention techniques are also applied.

Fig. 12 (a) shows the overall performance of these simplified models. As can be seen, DeepApnea significantly



(a) overall (b) Normal (c) Hypopnea (d) Obstructive apnea (e) Central apnea

Fig. 12. performance comparisons for data fusion models leveraging inputs from various number of axes.

outperforms three other two-axis fusion models (DeepApnea-XY, DeepApnea-YZ, DeepApnea-XZ), which outperform the one-axis fusion models (DeepApnea-X, DeepApnea-Y, DeepApnea-Z). This demonstrates the benefits of leveraging data from three axes.

Fig. 12 (b)(c)(d)(e) show the per-class performance of these simplified models. Similar to the results in Section VI-E, for normal sleep events, all models can achieve almost perfect performance (i.e., above 98%). For hypopnea and obstructive apnea, as shown in Fig. 12 (c) and (d), DeepApnea significantly outperforms the three two-axis fusion models (DeepApnea-XY, DeepApnea-YZ, DeepApnea-XZ), which outperform the one-axis fusion models (DeepApnea-X, DeepApnea-Y, DeepApnea-Z).

For central apnea, as shown in Fig. 12 (e), DeepApnea significantly outperforms the three two-axis fusion models and the three one-axis fusion models. Because the number of central apnea events is very small compared to other sleep anpea events in our dataset, the single-axis and two-axis fusion models do not perform well. DeepApnea leverages information from all axes and considers their correlations, and hence obtains more informative and representative features, outperforming the simplified models.

G. Ablation Study

In this subsection, we validate the effectiveness of the proposed self-attention and cross-correlation modules by comparing with the following two models: **DeepApnea-SelAtt**, which only contains the CNN-based feature extractor and the self-attention module without the cross-correlation module, and **DeepApnea-CroCor** which only contains the CNN-based feature extractor and the cross-correlation module without the self-attention module.

TABLE II
ABLATION STUDY

Method	Normal	Hypopnea	Obstructive	Central
DeepApnea-SelAtt	98.75%	66.02%	76.95%	38.30%
DeepApnea-CroCor	98.63%	64.64%	75.72%	35.09%
DeepApnea	**99.53%**	**71.58%**	**82.86%**	**68.29%**

According to Table II, DeepApnea clearly outperforms the other two, which illustrates that both self-attention and cross-correlation modules help extract informative features from the raw ACC data. Specifically, for normal sleep event, compared to DeepApnea-SelAtt and DeepApnea-CroCor, there is little improvement when using DeepApnea model. This is because the breathing pattern of normal sleep is significantly different from that of sleep apnea so that even the incomplete models can achieve almost perfect F1 score. For hypopnea, compared to DeepApnea-SelAtt and DeepApnea-CroCor, the improvements are 5.56% and 6.93% respectively. For obstructive apnea, the improvements are 5.9% and 7.14%. Finally, for central apnea, the improvements are 29.9% and 33.2% respectively, which demonstrate that the self-attention and cross-correlation modules can significantly improve performance especially when the training dataset is small.

H. Comparison of the Denoising Methods

In this section, we compare how different denoising methods, i.e., ADA, moving average, and TV filter, affect the performance of the proposed DeepApnea model.

TABLE III
COMPARISON OF DIFFERENT DENOISING METHODS.

Denoising Method	Normal	Hypopnea	Obstructive	Central
Moving average	98.51%	42.55%	69.8%	24.02%
TV filter	99.05%	65.05%	78.57%	38.10%
ADA	**99.53%**	**71.58%**	**82.86%**	**68.29%**

As shown in Table III, DeepApnea can achieve the best classification result when ADA is used for denoising. The moving average filter has the worst performance because it may remove potential useful apnea information by simply averaging different sub-sequences of the raw signal. Specifically, the F1 score of using moving average filter is 42.55%, 69.8% and 24.02% for hypopnea, obstructive apnea, and central apnea, respectively. Although the TV filter can preserve the useful signal information, it cannot eliminate low-amplitude noise. As discussed in Section IV-B, our ADA method not only keeps the periodic respiratory information but also removes all unnecessary noise. Therefore, compared to TV filter, using ADA can further improve the F1 score by 6.7%, 4.3% and 30.19% on classifying hypopnea, obstructive apnea, and central apnea, respectively.

I. System Profiling

When running DeepApnea, the acclerometor data from the smartwatch can be moved to the smartphone through Bluetooth. Based on the clinical study introduced in Section VI-A, the total size of the raw ACC data from a single subject during 8 hours measurement is about 45 MB, which can be transformed from a smartwatch to a smartphone within three seconds through Bluetooth 5.0.

In order to measure the running time of proposed DeepApnea model on modern smartphones, we implemented DeepApnea on several smartphones based on TensorFlow Lite. As shown in Table IV, it only takes two or three seconds to generate the classification results from the data over a whole night measurement (e.g., eight hours). We also consider the energy consumption of DeepApnea on smartwatch. When running DeepApnea, the smartwatch has to constantly record the real-time data of ACC and write them into the watch's external storage (e.g., SD card).

TABLE IV
THE RUNNING TIME OF DEEPAPNEA ON SMARTPHONES.

Smart Phone	Processing Unit	Preprocessing (seconds)	Deep Learning Inference (seconds)
Google Pixel 3	Snapdragon 845	1.38	2.36
Huawei Mate30 Pro	Kirin 990	0.83	1.22
Google Pixel 6	Google Tensor	0.45	1.01

We measured two modern commercial smartwatches: (1) Huawei Watch 2; (2) Apple Series 6. During the measurement, we close all irrelevant applications and turn off the watches' screen to make sure the power usage comes from the operating system and DeepApnea. Table V shows their battery usages for running or not running DeepApnea for a whole night. As can be seen, running DeepApnea for eight hours on Huawei watch 2 drains 55% of battery. For more advanced smart watch Apple Series 6, the watch battery only drains 23%. It shows that modern smartwatches can support DeepApnea for running at least one night.

TABLE V
THE BATTERY USAGE (MAH)

Smart Watch	w/o DeepApnea	w DeepApnea
Huawei Watch 2 (420mAh)	49.2 mAh	229.8 mAh
Apple Serise 6 (303.8mAh)	24.6 mAh	69.6 mAh

VII. DISCUSSIONS

In this paper, we focus on identifying various types of sleep apnea using accelerometers in smartwatches. While evaluations demonstrate the superior performance of DeepApnea, it can be further improved by considering other sensors besides accelerometer. Sleep apnea not only disrupts typical respiratory patterns, affecting the accelerometer signal on smartwatches, but also induces cardiovascular variations, resulting in fluctuations in oxygen saturation (SpO2) and heart rate. In recent years, there has been an integration of new physiological sensors into smartwatches, such as the oximeter sensor and photoplethysmography (PPG) sensor. These sensors enable the measurement of oxygen saturation and heart rate. By incorporating data from these new sensors, we could achieve more accurate and robust apnea detection through smartwatches.

One limitation of our dataset is its small size, comprising data from only 20 subjects. Although the dataset has plenty of obstructive apneas and hypopneas, the number of central apneas is limited, with the majority occurring in subjects 10 and 13. Moreover, several subjects do not have central apneas such as subjects 4, 9, and 12. As a result, in our 3-fold cross-validation, the 125 central apnea events are split into training and testing sets, without checking whether they are from the same subject. From Fig. 9, we can see that the apnea distribution varies significantly across different subjects. Consequently, the limited dataset may impact the generalizability of DeepApnea when applied to new users, potentially leading to large variations in predictions. To provide a more comprehensive evaluation of our model, we plan to conduct a larger clinical study in the future.

VIII. CONCLUSION

In this paper, we presented DeepApnea, a deep learning based sleep apnea detection system that leverages patients' wrist movement data collected by smartwatches to identify different types of sleep apnea. We first proposed signal preprocessing methods to filter the raw ACC data, smoothing away noise while preserving the respiratory signal and potential features for identifying sleep apnea. Then, we designed a deep learning architecture to extract features from three ACC axes collaboratively. Specifically, we apply self attention technique to accentuate more significant features and apply cross-axis correlation technique to exploit the correlations among different axes. The extracted deep features and the correlation information are merged through aggregated classification to further improve the classification accuracy. Through a clinical study, we demonstrate that DeepApnea outperforms existing solutions on multiclass classification. More specifically, DeepApnea can detect different sleep apnea with high F1-score, i.e., normal sleep (99.5%), obstructive apnea (82.9%), hypopnea (71.6%), and central apnea (68.3%). Finally, by profiling DeepApnea on different commodity devices, we demonstrate that it is practical to apply our system on modern smartwatches and smartphones.

REFERENCES

[1] J. M. Parish, "Sleep-Related Problems in Common Medical Conditions," *Chest Journal*, 2009.

[2] A. S. Association, "Sleep and Sleep Disorder Statistics," April 2021, https://www.sleepassociation.org/about-sleep/sleep-statistics/.

[3] N. F. Watson, "Health care savings: the economic value of diagnostic and therapeutic care for obstructive sleep apnea," *Journal of Clinical Sleep Medicine*, 2016.

[4] S. L. Appleton, A. Vakulin, R. D. McEvoy, A. Vincent, S. A. Martin, J. F. Grant, A. W. Taylor, N. A. Antic, P. G. Catcheside, G. A. Wittert *et al.*, "Undiagnosed obstructive sleep apnea is independently associated with reductions in quality of life in middle-aged, but not elderly men of a population cohort," *Sleep and Breathing*, 2015.

[5] A. Sabil, C. Marien, M. LeVaillant, G. Baffet, N. Meslier, and F. Gagnadoux, "Diagnosis of sleep apnea without sensors on the patient's face," *Journal of Clinical Sleep Medicine*, 2020.

[6] 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 *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2015.

[7] L. Chen, J. Xiong, X. Chen, S. I. Lee, D. Zhang, T. Yan, and D. Fang, "Lungtrack: Towards contactless and zero dead-zone respiration monitoring with commodity rfids," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies(IMWUT)*, 2019.

[8] S. Yue, Y. Yang, H. Wang, H. Rahul, and D. Katabi, "Bodycompass: Monitoring sleep posture with wireless signals," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2020.

[9] Q. S. Xue, D. Shin, A. Pathak, J. Garrison, J. Hsu, M. Malhotra, and S. Patel, "Luckychirp: Opportunistic respiration sensing using cascaded sonar on commodity devices," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2022.

[10] D. Liaqat, M. Abdalla, P. Abed-Esfahani, M. Gabel, T. Son, R. Wu, A. Gershon, F. Rudzicz, and E. D. Lara, "WearBreathing: Real World Respiratory Rate Monitoring Using Smartwatches," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2019.

[11] X. Sun, L. Qiu, Y. Wu, Y. Tang, and G. Cao, "SleepMonitor: Monitoring Respiratory Rate and Body Position During Sleep Using Smartwatch," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2017.

[12] L. Zhao, F. Zhang, H. Zhang, Y. Liang, A. Zhou, and H. Ma, "Robust respiratory rate monitoring using smartwatch photoplethysmography," *IEEE Internet of Things Journal*, pp. 4830–4844, 2022.

[13] Z. Jia, A. Bonde, S. Li, C. Xu, J. Wang, Y. Zhang, R. E. Howard, and P. Zhang, "Monitoring a person's heart rate and respiratory rate on a shared bed using geophones," in *ACM SenSys*, 2017.

[14] J. Clemente, M. Valero, F. Li, C. Wang, and W. Song, "Helena: Real-time contact-free monitoring of sleep activities and events around the bed," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2020.

[15] X. Chen, Y. Xiao, Y. Tang, J. Fernandez-Mendoza, and G. Cao, "Apneadetector: Detecting sleep apnea with smartwatches," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2021.

[16] M. Shokoueinejad, C. Fernandez, E. Carroll, F. Wang, J. Levin, S. Rusk, N. Glattard, A. Mulchrone, X. Zhang, A. Xie *et al.*, "Sleep apnea: a review of diagnostic sensors, algorithms, and therapies," *Physiological measurement*, 2017.

[17] R. Nandakumar, S. Gollakota, and N. Watson, "Contactless Sleep Apnea Detection on Smartphones," in *ACM MobiSys*, 2015.

[18] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *arXiv preprint arXiv:1304.5634*, 2013.

[19] A. V. Oppenheim, A. S. Willsky, S. H. Nawab, G. M. Hernández *et al.*, *Signals & systems*. Pearson Educación, 1997.

[20] S. W. Smith, "The Moving Average Filter." April 2021, https://www.analog.com/media/en/technical-documentation/dsp-book/dsp_book_Ch15.pdf.

[21] I. W. Selesnick and I. Bayram, "Total Variation Filtering. White paper." 2010.

[22] J. Gao, H. Sultan, J. Hu, and W.-W. Tung, "Denoising nonlinear time series by adaptive filtering and wavelet shrinkage: a comparison," *IEEE signal processing letters*, 2009.

[23] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

[24] Y. Zhang, L. Wang, H. Chen, A. Tian, S. Zhou, and Y. Guo, "If-convtransformer: A framework for human activity recognition using imu fusion and convtransformer," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2022.

[25] H. Ren, J. Wang, W. X. Zhao, and N. Wu, "Rapt: Pre-training of time-aware transformer for learning robust healthcare representation," in *ACM SIGKDD*, 2021.

[26] H. Xue, W. Jiang, C. Miao, Y. Yuan, F. Ma, X. Ma, Y. Wang, S. Yao, W. Xu, A. Zhang *et al.*, "Deepfusion: A deep learning framework for the fusion of heterogeneous sensory data," in *ACM MobiHoc*, 2019.

[27] L. Mou, R. Men, G. Li, Y. Xu, L. Zhang, R. Yan, and Z. Jin, "Natural language inference by tree-based convolution and heuristic matching," *arXiv preprint arXiv:1512.08422*, 2015.

[28] A. S. Rathore, W. Zhu, A. Daiyan, C. Xu, K. Wang, F. Lin, K. Ren, and W. Xu, "Sonicprint: a generally adoptable and secure fingerprint biometrics in smart devices," in *ACM MobiSys*, 2020.

[29] G. L. Santos, P. T. Endo, K. H. d. C. Monteiro, E. d. S. Rocha, I. Silva, and T. Lynn, "Accelerometer-based human fall detection using convolutional neural networks," *Sensors*, 2019.

[30] S. Chambon, M. N. Galtier, P. J. Arnal, G. Wainrib, and A. Gramfort, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2018.

[31] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the 26th International Conference on World Wide Web*, 2017.