






Deep Learning-Based Event Counting for Apnea-Hypopnea Index Estimation Using Recursive Spiking Neural Networks

Lorin Werthen-Brabants , Yolanda Castillo-Escario , Willemijn Groenendaal, Raimon Jané , Senior Member, IEEE, Tom Dhaene , Senior Member, IEEE, and Dirk Deschrijver , Senior Member, IEEE

I. INTRODUCTION

Abstract—Objective: To develop a novel method for improved screening of sleep apnea in home environments, focusing on reliable estimation of the Apnea-Hypopnea Index (AHI) without the need for highly precise event localization. **Methods:** RSN-Count is introduced, a technique leveraging Spiking Neural Networks to directly count apneic events in recorded signals. This approach aims to reduce dependence on the exact time-based pinpointing of events, a potential source of variability in conventional analysis. **Results:** RSN-Count demonstrates a superior ability to quantify apneic events (AHI MAE 6.17 ± 2.21) compared to established methods (AHI MAE 8.52 ± 3.20) on a dataset of whole-night audio and SpO₂ recordings (N = 33). This is particularly valuable for accurate AHI estimation, even in the absence of highly precise event localization. **Conclusion:** RSN-Count offers a promising improvement in sleep apnea screening within home settings. Its focus on event quantification enhances AHI estimation accuracy. **Significance:** This method addresses limitations in current sleep apnea diagnostics, potentially increasing screening accuracy and accessibility while reducing dependence on costly and complex polysomnography.

Index Terms—Sleep apnea detection, AHI estimation, deep learning, spiking neural networks, wearables.

Received 19 February 2024; revised 3 June 2024 and 24 October 2024; accepted 10 November 2024. Date of publication 14 November 2024; date of current version 21 March 2025. This work was supported in part by the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” and the CERCA Program and 2021 SGR 01390 of Generalitat de Catalunya, and in part by the Spanish Ministry of Science and Innovation under Grant PID2021-126455OB-I00 MCIN/AEI/FEDER. (Corresponding author: Lorin Werthen-Brabants.)

Lorin Werthen-Brabants is with IDLab, Ghent University – imec, 9052 Gent, Belgium (e-mail: lorin.werthenbrabants@ugent.be).

Yolanda Castillo-Escario and Raimon Jané are with the Universitat Politècnica de Catalunya - BarcelonaTech, Spain, also with the Institute for Bioengineering of Catalonia, Spain, also with the Barcelona Institute of Science and Technology, Spain, and also with the Centro de Investigación Biomédica en Red de Bioingeniería, Biomateriales y Nanomedicina, Spain.

Willemijn Groenendaal is with OnePlanet Research Center, Stichting imec Nederland, The Netherlands.

Tom Dhaene and Dirk Deschrijver are with IDLab, Ghent University – imec, Belgium.

Digital Object Identifier 10.1109/TBME.2024.3498097

SLEEP apnea affects 25–50% of the adult population, particularly elderly and obese individuals, and stands as one of the most common sleep disorders [1]. However, about 80% of patients remain undiagnosed [2]. The gold standard for sleep apnea diagnosis is a polysomnography (PSG) conducted in a sleep laboratory. During the night, PSG measures multiple physiological signals pertaining to respiration, brain activity, sleep stages, heart rate, oxygen saturation and others. These signals are analysed and annotated by trained sleep specialists according to the American Academy of Sleep Medicine (AASM) guidelines [3]. This leads to the determination of the Apnea-Hypopnea Index (AHI), representing the number of apneas and hypopneas per hour of sleep. Depending on the outcome, patients are categorized as normal (AHI < 5), having mild sleep apnea ($5 \leq \text{AHI} < 15$), experiencing moderate sleep apnea ($15 \leq \text{AHI} < 30$), or facing severe sleep apnea ($\text{AHI} \geq 30$) [3]. Nevertheless, PSG has several limitations, including its high cost and complexity, patient discomfort, long waiting lists, interference with natural sleep patterns, and its reliance on a single night recording. Therefore, portable devices for home sleep apnea monitoring are being developed. Such devices measure a limited number of respiratory-related signals, such as respiratory flow, thoracic effort, oxygen saturation (SpO₂), bio-impedance or audio signals [4], [5].

Despite the availability of guidelines, the annotation of the signals involves some degree of subjectivity and can lead to inter-rater and intra-rater variability as shown by recent studies [6]. This can hamper the performance of existing techniques [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] that detect apneic events using deep learning models by sliding a window over the raw filtered signals and extracting relevant features. Especially in cases where the pinpointing of apneic events is approximate, there can be a mismatch between the considered windows and their correct annotation. Furthermore, annotations can be ill-defined when windows contain only a part of the apneic event, or multiple events.

This paper aims to address those problems by taking advantage of the fact that, in clinical practice, the diagnosis is mostly based on the AHI without requiring the precise location of events. To this end, a novel RSN-Count algorithm is proposed,

as a method that leverages Spiking Neural Networks to count apneic events in the recorded signals, treating the task as a true counting task where the events are treated as singular units in time. This is in contrast to previous methods, that either treat it as a regression problem, or otherwise make use of time thresholds, where there needs to be an uninterrupted positive prediction of several seconds. Instead, the proposed RSN-Count aims to estimate the AHI directly by counting the number of apneic events, while discarding information about the precise start and end times of those events. It is successfully applied to determine the AHI of patients, based on a home sleep apnea test that records acoustic signals with a smartphone, as well as oxygen saturation. It is noted that the algorithmic approach is generic and could be applied to other types of signals as well.

The structure of the paper is organized as follows. After this Section I, corresponding to the introduction, an overview is provided of related work on sleep apnea detection in Section II. Section III introduces some preliminary concepts and the methodology of the novel RSN-Count algorithm. Section IV describes the available data and experimental set-up. This is followed by numerical results and a performance validation in Section V. Finally, the paper provides a discussion in Section VI and concluding remarks in Section VII.

II. RELATED WORK

In literature, several advancements are reported in predicting the AHI from PSG recorded signals, hereby leveraging a variety of machine learning techniques. Classical approaches for sleep apnea detection are based on a sliding window approach where human-engineered features are extracted from physiological signals, followed by the application of classical algorithms such as k-nearest neighbor, Hidden Markov models, support vector machine, fuzzy logic and neural networks [19]. More recently, the use of deep learning algorithms, such as 1D or 2D Convolutional Neural Networks (CNN's), bi-directional Long Short-Term Memory (BiLSTM) networks, Gated Recurrent Units (GRU), self-attention mechanisms and transformer networks has emerged as a promising approach [20], [21], [22]. A systematic review of the latest developments is reported in [23]. With the uprise of wearable devices, various modalities can be recorded and analysed, such as nasal or oral airflow, electrocardiogram (ECG), ECG-derived respiration, bio-impedance, pulse oximetry, tracheal sound, accelerometer data, and various respiration signals. Such measurements can be obtained from a chest band, patch, pressure sensor, thermal sensor or other types of devices.

Nowadays, the analysis of acoustic breathing and snoring is gaining attention for sleep apnea monitoring as it only requires a low-cost sensor (microphone) and can be used to detect apneas and hypopneas as an absence or reduction in sound. In [4], a rule-based algorithm was presented, based on entropy analysis of acoustic signals recorded with a smartphone for home sleep apnea diagnosis. When traditional machine learning methods are applied to audio signals, a random forest approach yields a Mean Absolute Error (MAE) on the AHI of 9.64 using global audio features [9]. In [15], OSA harnesses deep learning methods,

such as a CNN, to achieve an improved MAE of 3 events per hour. In other papers, deep neural networks are combined with Mel-frequency cepstral coefficients (MFCC) to classify normal snoring, apneic snoring and not snoring by making use of a windowed approach [13]. Furthermore, it was shown in [24] that the integration of physiological signals, such as respiratory effort, can potentially enhance the AHI prediction. Additionally, ensemble methods, particularly gradient boosted models, have been highlighted for their promise in predicting OSA severity [25].

In this work, a novel Recursive Spiking Network is proposed that changes the overall objective to counting events, rather than detecting individual apneic events. As such, the algorithm does not require a precise localization of individual apneic events during annotation. This is an important advantage, as there can be subjective elements in the interpretation and scoring of sleep data, even though the AASM guidelines themselves are objectively defined [3]. By detecting entire events, rather than focusing solely on the fact an event occurred at or near a specific time step, the RSN-Count algorithm is able to estimate the AHI more accurately.

III. METHODS

This section gradually introduces RSN-Count, a technique that is not restricted to specific signal modalities. In this work, however, audio and SpO₂ are considered in particular. A transformation on the audio and SpO₂ signals is proposed in Section III-A using standard deep learning techniques, resulting in a fused time series. Then, an introduction to Spiking Neural Networks (SNN) is given in Section III-B, followed by the novel RSN-Count in Section III-C. The complete pipeline of an RSN-Count based model is shown in Fig. 1.

A. CNN-Based Feature Extraction

Prior to the proposed RSN-Count stage, a feature extraction step is applied on the input audio and SpO₂ data. First the raw audio data is transformed to a spectrogram making use of the Short-Time Fourier Transform (STFT). Then, a CNN is applied as it can extract meaningful features from audio spectrograms. It is composed of four sequential blocks, each comprising the following layers:

- 1) *Convolutional Layer*: Each block contains a convolutional layer with 100 filters. Each filter has a kernel size of 3, and Rectified Linear Unit (ReLU) activation is applied to the output of this layer.
- 2) *MaxPooling Layer*: Following the convolutional layer in each block is a max-pooling layer, responsible for downsampling the feature maps.
- 3) *Dropout Layer*: To prevent overfitting, a dropout layer with a probability of $p = 0.4$ is inserted after the max-pooling step to improve generalizability.

The output from the final block of the CNN is concatenated with the input SpO₂ maxdrop time series data, resulting in a new time series that contains a compressed, latent version of the audio. This combined data is then fed into a stacked BiLSTM, all with 50 hidden units, allowing the network to leverage both the

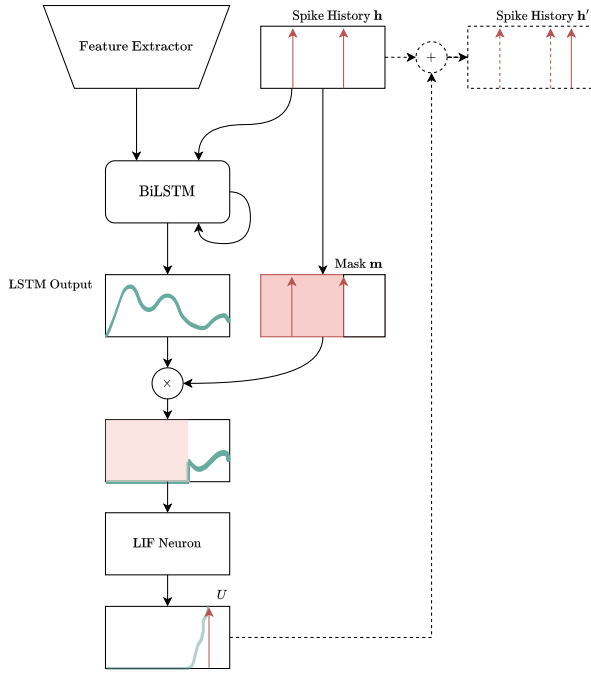


Fig. 1. Conceptual model of RSN-Count making use of a predetermined spike history h . Note that the BiLSTM block replaces the function g_θ from (3), which by itself is also recurrent. Hence, the output of this block is a time series.

learned features from the audio spectrograms and the temporal dynamics of the SpO₂ data. The output of this stacked BiLSTM is then used as the base latent time series used by RSN-Count.

B. Spiking Neural Networks

Before introducing RSN-Count, a basic foundation of Spiking Neural Networks (SNNs) [26], [27] is given. SNNs are inspired by the way biological neurons communicate and process information. In the human brain, neurons do not continuously send information. Instead, they transmit information sporadically through electrical impulses, known as spikes. When the electrical potential across a neuron's cell membrane reaches a specific threshold, the neuron fires this spike. This discrete, event-driven communication stands in contrast to traditional artificial neural networks, which rely on continuous values. SNNs attempt to mimic this biological spiking behavior, aiming to capture the efficiency and temporal dynamics seen in natural neural systems. These spiking neural networks are particularly well-suited for modeling temporal information and have been used in various applications, including neuromorphic computing and time-series analysis. The key advantage for sleep apnea detection is that the spikes can be interpreted as *discrete events*, whereas an LSTM or other techniques for modeling sequential data need additional post-processing and thresholding to discretize an output time series into events.

1) Leaky Integrate and Fire (LIF) Neuron: The *Leaky Integrate and Fire Neuron* is a core concept in SNNs. It is a simplified version of the biological process that occurs in real neurons. Unlike traditional artificial neurons that output a continuous

value, the Leaky Integrate and Fire Neuron models the *spiking* behavior of biological neurons. It integrates incoming signals until a threshold is reached, at which point it fires a spike and resets its internal state, also known as the *membrane potential*. The *leaky* aspect comes into play because the neuron also has a mechanism to gradually lose or *leak* some of its stored energy over time, mimicking the decay process in biological neurons. This allows SNNs to capture temporal dynamics and makes them particularly useful for time series data and tasks requiring temporal context.

2) Membrane Potential: The membrane potential u_t of a LIF neuron is defined as

$$u_t = \beta u_{t-1} + x_t, \quad (1)$$

and is determined by its previous value u_{t-1} and an input term x_t , where $u_0 = 0$. A decay parameter $\beta \in [0, 1]$ determines how fast the membrane potential decays over time, given an external input $x_t \in \mathbb{R}$ to the neuron. If the membrane potential u_t surpasses a defined threshold κ (often $\kappa = 1$), the neuron fires or “spikes” and the membrane potential is reset to zero. Depending on the application or task at hand, the spikes can be interpreted in different ways. In this paper, the timing of the spikes reflects critical events in the signal or temporal changes in the system being modeled. This threshold crossing is often described using the Heaviside step function Θ . The spike train, which represents the discrete spike emissions over time, can be defined as $\text{LIF} = \{s_t\}, \forall t$ with

$$s_t = \Theta(u_t - \kappa). \quad (2)$$

3) Solving the Dead Neuron Problem: The use of the heaviside function implies that the LIF neuron can not be used with back-propagation, which is the primary mechanism for training neural networks. Its derivative is the Dirac δ distribution, which is 0 everywhere except at 0, where it tends to infinity, causing the *dead neuron* problem. This occurs when certain neurons in a neural network become inactive, outputting constant values and ceasing to update during training, thereby reducing the model's learning capacity. Instead, the gradient of the arctangent function is used as a surrogate $s_t \approx \tilde{s}_t = \frac{1}{\pi} \arctan(\pi u_t)$ in the backward pass

$$\frac{\partial \tilde{s}_t}{\partial u_t} \leftarrow \frac{1}{\pi} \frac{1}{(1 + (\pi u_t)^2)},$$

where the left arrow denotes substitution. The forward pass, as described in (2), remains unchanged.

C. RSN-Count Algorithm

The novel Recursive Spiking Network for Counting (RSN-Count) technique leverages the recursive application of the Leaky Integrate and Fire (LIF) Neuron used in Spiking Neural Networks (SNNs) to assess apnea severity of a patient.

The input of RSN-count are recorded signals, e.g. based on a PSG or a wearable, whereas the output is the estimated AHI. The method is applied in a windowed manner, meaning segments x of a predefined size are extracted from the input signals. Based on a recursive approach, a binary spike history vector h is calculated

that contains a single spike for every detected apneic event in the window. By counting the occurrence of spikes over time for each window and subsequently dividing by the overlap in the sliding window, the patient's AHI is estimated.

Note that this network is not considered to be an SNN in the purest sense, but rather a hybrid approach. SNNs typically make use of spikes throughout the architecture, including the inputs, whereas RSN-Count only makes use of the LIF neuron to mark the location of a discrete apneic event, occurring approximately around the spike. This makes it possible to design a loss function that shifts the time of the spike closer to the time of an actual event. Hence, the methodology of RSN-Count revolves around the implementation of a Deep Neural Network with a LIF head, where its primary function is to identify the first event in a given window, given the previously detected events. This paradigm characterizes the approach, where \mathbf{h} is constructed through successive predictions obtained from previous recursive steps of RSN-Count.

Note that the event spikes emitted by RSN-Count are not exact, and indicate the presence of an event in its vicinity, rather than its exact location and duration. This makes it suitable in situations where the number of events in a given window is more important than their precise time localization.

1) Model Architecture: The architecture of the algorithm is visualized in Fig. 1. First, the input segments of the recorded signals are passed through the aforementioned feature extractor from Section III-A. The resulting time series data \mathbf{x} , together with a spike history \mathbf{h} that is initialized with zeros, are then fed into a Bidirectional Long Short-Term Memory layer, denoted as function $g_\theta(\mathbf{x}, \mathbf{h})$. Both \mathbf{x} and \mathbf{h} are of the same size. The output \mathbf{z} of this BiLSTM layer is then multiplied with a mask $\mathbf{m} = \mu(\mathbf{h})$. Initially, the mask consists of only ones, meaning no change is applied to output \mathbf{z} . However, in each recursive step, it is constructed by zeroing a ones vector until the time of the last event $\tau_\omega(\mathbf{h})$ in the spike history \mathbf{h} .

$$\mu(\mathbf{h}) = \mathbf{w}, \quad \text{where } w_i = \begin{cases} 0, & \text{if } i < \tau_\omega(\mathbf{h}) \\ 1, & \text{otherwise} \end{cases}.$$

$$\tau_\omega(\mathbf{h}) = \max(\{t : h_t = 1\} \cup \{0\}).$$

The masked signal $\mathbf{z} \odot \mathbf{m}$, calculated as a Hadamard product, is then fed into the LIF neuron (1) that potentially emits a spike. If it does, the spike is recorded in \mathbf{h} and the process is recursively repeated until either no spike is emitted, or $\tau_\omega(\mathbf{h}) = W$, where W represents the window size.

Hence, RSN-count starts from $\mathbf{h}^{(0)} = \mathbf{0}$ and recursively solves (3) in consecutive steps, n , until $\mathbf{h}^{(n+1)} = \mathbf{h}^{(n)}$.

$$\mathbf{m}^{(n)} = \mu(\mathbf{h}^{(n)}), \quad \mathbf{z}^{(n)} = g_\theta(\mathbf{x}, \mathbf{h}^{(n)})$$

$$\text{RSN}(\mathbf{x}, \mathbf{h}^{(n)}) = \mathbf{h}^{(n+1)} = \mathbf{h}^{(n)} + \iota_n \left(\text{LIF} \left(\mathbf{m}^{(n)} \odot \mathbf{z}^{(n)} \right) \right). \quad (3)$$

Algorithm 1: RSN-Count at Inference Time.

```

1: Given function RSN from (3)
2:  $\mathbf{h}^{(0)} \leftarrow \mathbf{0}_{1 \times l_w}$ 
3:  $\mathbf{h}^{(1)} \leftarrow \text{RSN}(\mathbf{x}; \mathbf{h}^{(0)})$ 
4:  $i \leftarrow 1$ 
5: while  $\mathbf{h}^{(i)} \neq \mathbf{h}^{(i-1)}$  do
6:    $\mathbf{h}^{(i+1)} \leftarrow \text{RSN}(\mathbf{x}; \mathbf{h}^{(i)})$ 
7:    $i \leftarrow i + 1$ 
8: end while
9: return  $\mathbf{h}^{(i)}$ 
    
```

In (3), an auxiliary function $\iota_n(\mathbf{x})$ is used that filters the LIF to only contain the first n occurrences of 2 as follows

$$\iota_n(\mathbf{v}) = \mathbf{w}, \quad \text{where } w_i = \begin{cases} v_i, & \text{if } \sum_{j=1}^i v_j \leq n \\ 0, & \text{otherwise} \end{cases}.$$

This final history vector $\mathbf{h}^{(n)}$ then becomes the resulting output of the network, indicating approximate locations of detected events within this window. Algorithm 1 provides a description on how apneic events are detected by RSN-Count through a recursive application of (3) and consecutive addition of predicted spikes to the spike history \mathbf{h} .

2) AHI Estimation: By sliding the window over an entire night recording and applying this procedure to the subsequent segments, the AHI can be determined by counting the number of detected apneic events over time and calculating the average number of events over each hour of sleep. The number of events is determined as follows: given a window stride length l and a window size W , it is anticipated that each event will be observed W/l times within any given time period. Consequently, the spikes are tallied to obtain the count estimate $\hat{c} = \frac{N_{tot}}{W/l}$, where N_{tot} denotes the total number of observed apneic events. It is worth noting that at the edges of the predictions, the accuracy of detected events tends to decrease. To enhance the counting accuracy when analyzing events over an extended time period using a sliding window approach, only the predictions within a calibrated sub-window $[t_{init}, t_{end}]$ are considered. Note that the optimal configuration of these hyperparameters is identified from an evaluation of the model on the validation set.

3) Model Training: The model training procedure is explained in Algorithm 2. For each input segment \mathbf{x} , a corresponding binary vector \mathbf{y} is defined, containing N target spikes that are located at the mid-point of each apneic event in the window, as determined from the ground-truth annotations. First, the predicted spikes $\hat{y}_i = \text{RSN}(\mathbf{x}; \iota_i(\mathbf{y}))$ are sequentially calculated by the model for every $i \leq N + 1$, by making use of the target spikes \mathbf{y} . The last prediction at iteration $N + 1$ is used as an “off” event, where no prediction signals the lack of any further events. Subsequently, the loss function $\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}, \mathbf{u}^{(N+1)})$, where $\mathbf{u}^{(N+1)}$ is the last membrane potential from RSN-Count is computed.

Algorithm 2: RSN-Count Training With Teacher Forcing.

```

1: Given function RSN from (3), input  $x$ , target spikes  $y$ 
2: for  $k \in [1 \dots N + 1]$  do
3:    $h' \leftarrow \iota_{k-1}(y)$ 
4:    $\hat{y}_k \leftarrow \text{RSN}(x; h')$ 
5: end for
6: Retrieve last membrane potential  $u^{(N+1)}$ 
7: Calculate  $\nabla_{\theta} \mathcal{L}(\hat{y}, y, u^{(N+1)})$ 

```

4) Loss Function: The loss function $\mathcal{L}(\hat{y}, y, u^{(N+1)})$ comprises of three terms, and is defined as follows.

$$\begin{aligned}
 \mathcal{L}(\hat{y}, y, u) = & \underbrace{\sum_{i=1}^N \left(\frac{\hat{t}_{y_i} - t_{y_i}}{\alpha} \right)^2}_{\text{target events}} \\
 & + \underbrace{\left(\frac{t_{\hat{y}_{N+1}} - \alpha}{\alpha} \right)^2}_{\text{no event}} + \underbrace{\frac{\lambda}{\alpha - \tau_{\omega}(u)} \sum_{i=\tau_{\omega}(u)}^{\alpha} |u_i|}_{\text{membrane penalty}}.
 \end{aligned}$$

The first term computes the Mean Squared Error (MSE) between the predicted spike times \hat{t}_{y_i} and the actual spike times t_{y_i} , normalized by the value $\alpha = t_{\text{end}} + 1$. The second term computes the MSE between the last predicted spike time and α . If no spike is predicted, this term vanishes because $t_{\hat{y}_{N+1}} = \alpha$. The last term is a penalty term that steers the membrane potential of the prediction after the last spike y_N towards 0, where $\lambda = 0.01$ is a regularization parameter. Without this penalty term, spikes may be incorrectly predicted at the end of the window, even though no event is present. For example, if $t_{\hat{y}_{N+1}} = \alpha - 1$, the “no event” term would become α^{-2} , resulting in a negligible contribution to the loss function.

Note that the derivative of each spike time with respect to the spike $\partial t_{\hat{y}} / \partial s$ is non-differentiable. Therefore, the gradient of each predicted spike time $t_{\hat{y}}$ is set to a sign estimator of -1. A positive gradient $\partial s / \partial u$ at the predicted spike when using gradient descent will increase the value of membrane potential u , therefore causing an earlier firing time.

By utilizing MSE in this event-based context, only an approximate event location becomes crucial. If a cross-entropy loss were employed instead, it would enforce a single precise location for each event detection, potentially leading to overfitting. This scenario is more likely when the timing of apneic event annotations is noisy or imprecise, and represent approximations of the actual underlying ground truth data.

IV. EXPERIMENTAL SETUP

A. Dataset Description

RSN-Count is applied to a sleep apnea dataset containing overnight recordings of 33 patients (18 men, 15 women) with a mean age of 55 ± 16 who were enrolled for an overnight sleep

test [4]. The data acquisition and analysis was performed by the BIOSPIN group of Institute for Bioengineering of Catalonia (IBEC) and received approval from the Ethics Committee of Hospital Clínic de Barcelona (protocol code HCB/2017/0106), and informed consent was obtained from all participants. Among the participants, 20 underwent in-lab polysomnography (PSG), while 13 underwent a home sleep apnea test with ResMed ApneaLink Air™ [28].

The PSG recordings consisted of various channels, including respiratory signals (nasal cannula, thermistor, and thoracic and abdominal effort) sampled at a rate of 32 Hz, single-lead electrocardiogram sampled at 256 Hz, and SpO₂ sampled at 1 Hz. On the other hand, the ApneaLink measurements included respiratory flow through a nasal cannula sampled at 100 Hz, thoracic movement sampled at 10 Hz, and SpO₂ sampled at 1 Hz. Simultaneously, overnight audio recordings were acquired at a rate of 48 kHz using the built-in microphone of a smartphone (Samsung Galaxy S5) placed over the subjects’ thorax using an elastic band. This configuration had been successfully tested in previous studies [4], [11]. The mean recording length for each subject was 6.4 ± 1.3 hours.

To ensure synchronization, timestamps were used to align the data from the smartphone and the reference system (either PSG or ApneaLink). Trained sleep specialists annotated the data from the reference system, following the guidelines of the American Academy of Sleep Medicine (AASM) [3]. Based on these annotations, 3 subjects were identified with a normal Apnea-Hypopnea Index (AHI), 4 with mild AHI, 17 with moderate AHI, and 9 with severe AHI.

B. Data Preprocessing

The dataset is divided into a train, validation, and test set, comprising 17, 8, and 8 patient recordings, respectively, and the model results are evaluated using 4-fold cross validation.

Audio recordings from the smartphone’s microphone and oxygen saturation SpO₂ are the two modalities that will be considered to determine the patient’s AHI using RSN-Count.

In a data-preprocessing step, a Short-Term Fourier Transform (STFT) is applied to the audio signals with an FFT window size of 512. Subsequently, each 60-second window is extracted from the STFT and resized to 256×960 using nearest neighbor sampling. Additionally, a “maxdrop” SpO₂ signal is defined at each second t by computing the maximum drop in SpO₂ within the time range $[t, t + 45]$. An example of a 60 s segment x , obtained from the STFT spectrogram and the SpO₂ maxdrop feature can be seen in Fig. 2.

The midpoints of the annotated apnea-hypopnea events, visible within each window, are used to define the corresponding target spikes y that can be used to train RSN-Count.

V. RESULTS

A. Baseline Models

In order to assess the model performance of RSN-count, the novel approach will be benchmarked against two state-of-the-art deep learning for sleep apnea detection using audio recordings

TABLE I
MODEL PERFORMANCE COMPARISON: AHI AND CORRELATION METRICS

Model Type	AHI MAE	AHI RMSE	Pearson r	ICC
CNN	11.49 \pm 1.76	15.42 \pm 3.19	0.68 \pm 0.28	0.66 \in [0.41, 0.81]
CNN-BiLSTM	8.52 \pm 3.20	12.79 \pm 2.97	0.82 \pm 0.08	0.81 \in [0.65, 0.90]
RSN-Count	6.17 \pm 2.21	9.58 \pm 3.44	0.86 \pm 0.12	0.87 \in [0.75, 0.93]

The best values per metric are indicated in bold.

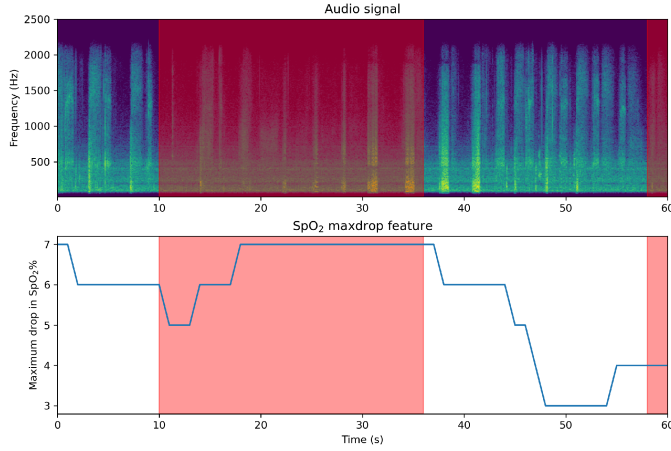


Fig. 2. Example of a 60 s window from the dataset. The SpO₂ maxdrop feature is shown in the lower half. Apnea/Hypopnea events are marked in red.

and SpO₂. The first technique is a CNN architecture that comprises 5 sequential blocks, as described in [29], whereas the second technique is a CNN-BiLSTM architecture, similar to other reported methods for sleep apnea detection [16], [17], [20], [21]. A key difference is that the reference models are trained by making use of the Cross-Entropy Loss, whereas RSN-Count makes use of the loss provided in Section III-C4, based on the distance of the predicted event and the center of the actual event.

For both the CNN and CNN-BiLSTM baseline, a similar strategy for event detection is applied, following the approach used by Kwon et al. [20]. This strategy involves counting one discrete event of Apnea-Hypopnea (AH) if six consecutive windows with a stride of 1 s in a night are classified as AH. In this work however, we adhere to the AASM guidelines and accept only events longer than 10 seconds as AH.

B. Performance Metrics

While RSN-Count is not geared towards predicting exact locations and durations of apneic events, it is possible to evaluate RSN-Count in a similar way by using commonly used metrics for sleep apnea detection. Typical metrics are the area under the ROC curve, average precision score, NPV, PPV, sensitivity, specificity and accuracy. The average precision is similar to the area under the precision-recall curve (AUCPR), but less optimistic. The formal definition is as follows, with recall (sensitivity) values r , and precision (PPV) values p :

$$AP = \sum_n (r_n - r_{n-1}) p_n.$$

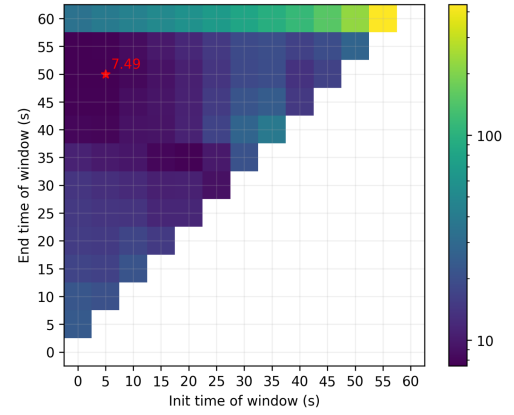


Fig. 3. Optimal hyperparameters t_{init} and t_{end} (red asterisk).

An often overlooked metric is the Intersection over Union (IoU), a.k.a. Jaccard index, defined for two sets A and B as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

It is frequently used in semantic segmentation for computer vision to score how well the predicted regions overlap with the labels, which is more important when detecting AH events.

A calculation of the regions is achieved by post-processing the spikes produced by RSN-Count. Each predicted spike \hat{y}_i is represented as an unnormalized Gaussian $f(t; t_{\hat{y}_i}, \sigma)$, and they are aggregated over subsequent windows by a summation

$$f(t; t_{\hat{y}_i}, \sigma) = \exp\left(-\frac{(t - t_{\hat{y}_i})^2}{2\sigma^2}\right). \quad (4)$$

The standard deviation σ is determined by optimizing the area under the receiver operating characteristic curve (AUC ROC) for the approximated scores on the validation set. Hyperparameter σ is optimized to 20.69. Fig. 7 shows the resulting, unnormalized scores for a sample segment as an example.

C. AHI Assessment

The main goal of RSN-Count is the assess the AHI of patients as accurately possible. While the complete window of predictions could be used of RSN-Count, through testing it can be observed that selecting a smaller window of the predictions (defined by a t_{init} and t_{end}) improves the counting performance. Through a grid search, it is determined that optimal value of hyperparameters is $t_{\text{init}} = 5$ and $t_{\text{end}} = 50$, as shown in Fig. 3

Table I shows a comparison between different models when evaluating the Apnea-Hypopnea Index (AHI) with a 4-fold

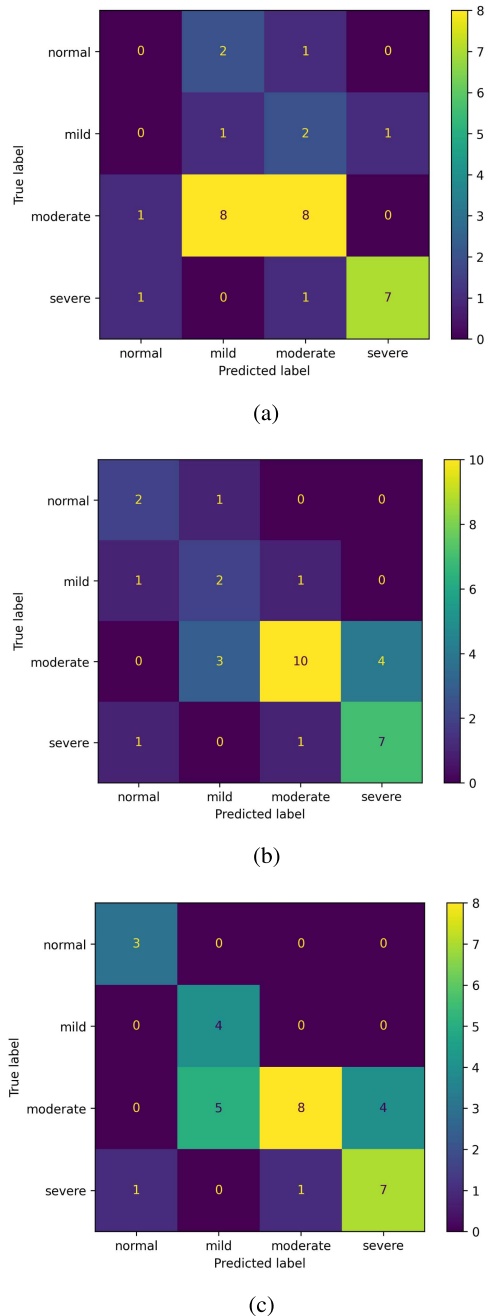


Fig. 4. (a) CNN: AHI severity confusion matrix on every test set. (b) CNN-BiLSTM: AHI severity confusion matrix on every test set. (c) RSN-Count: AHI severity confusion matrix on every test set.

cross-validation split on the patients over all folds. It can be seen that the MAE of RSN-Count is 6.17 ± 2.21 , while that of the base CNN-BiLSTM is 8.52 ± 3.20 , showing superior performance in a generalized setting as well. A similar observation can be made when comparing the AHI RMSE between both models. This demonstrates that RSN-Count significantly outperforms the other state-of-the-art methods.

The evaluation based on the AASM guidelines [3] involves the construction of a confusion matrix, shown in Fig. 4(c) for all folds. It can be seen that there is slight confusion among

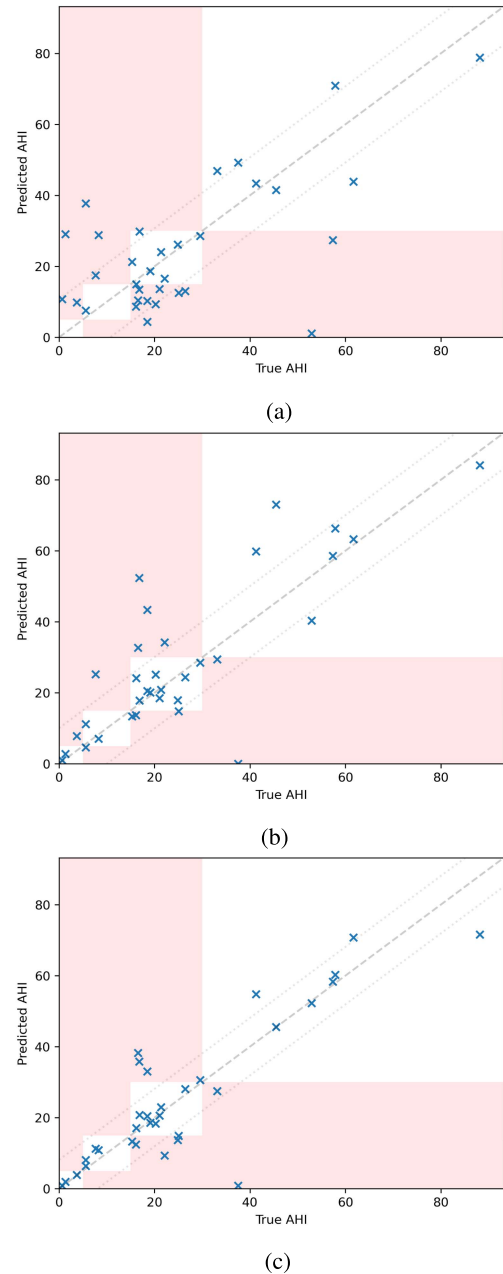


Fig. 5. Correlation between estimated AHI and true AHI for CNN, CNN-BiLSTM and RSN-Count. The red regions denote erroneous classifications w.r.t. the AASM annotations [3]. (a) CNN predicted AHI vs. true AHI. (b) CNN-BiLSTM predicted AHI vs. true AHI. (c) RSN-Count predicted AHI vs. true AHI.

the higher severities, with one severe AHI patient identified as normal. This outlier also persists in the CNN-BiLSTM model and suggests an outlier in the dataset. The Pearson correlation coefficient of 0.86 ± 0.12 and intraclass correlation (ICC) of 0.87 within the 95% confidence interval of 0.75 to 0.93 further highlights the positive correlation between the predicted AHI values and the actual AHI values obtained from the gold-standard measurement. The linear correlation is shown for each tested model in Fig. 5, with corresponding Bland-Altman plots in Fig. 6. The Bland-Altman plots show some degree of bias of all

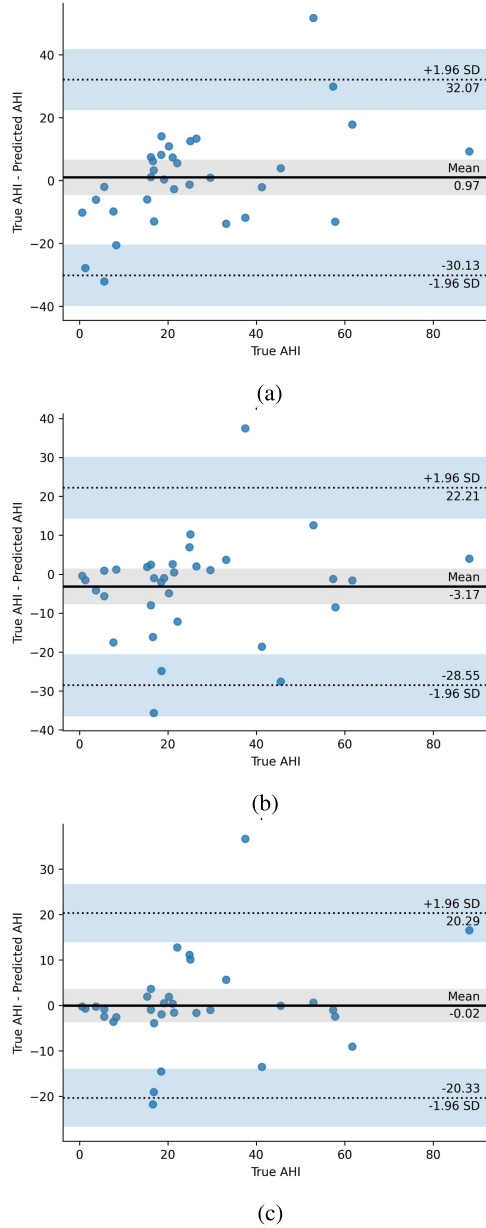


Fig. 6. Bland-altman plots for each model. (a) CNN predicted AHI vs. true AHI. (b) CNN-BiLSTM predicted AHI vs. true AHI. (c) RSN-Count predicted AHI vs. true AHI.

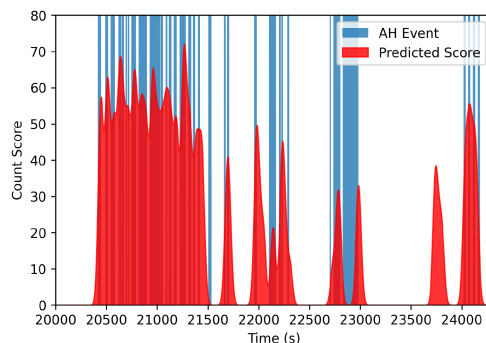


Fig. 7. Unnormalized RSN-Count scores making use of the method outlined in Section V-D.

three models, with the CNN-BiLSTM model having a negative bias (underprediction) and the CNN model showing a slight positive bias (overprediction), while RSN-Count shows virtually no bias. Additionally, RSN-Count has the narrowest limits of agreement, indicating more consistent predictions compared to the CNN and CNN-BiLSTM models.

D. Event Classification

By making use of (4), RSN-Count can provide predictions in line with those given by the state-of-the-art CNN and CNN-BiLSTM models. The classification results for RSN-Count and the reference methods are presented in Table II. Although event classification is not the aim of RSN-Count, it is observed that the method demonstrates competitive performance compared to state-of-the-art methods. In comparing the performance of convolutional neural networks (CNN), CNN-BiLSTM, and RSN-Count models, significant variations in classification metrics are evident. The CNN-BiLSTM model outperforms the others in terms of ROC AUC (0.87 ± 0.03) and Average Precision (0.71 ± 0.06), indicating its superior ability to distinguish between classes and its higher precision in classification. The RSN-Count model demonstrates the highest Intersection over Union (IoU) score (0.64 ± 0.07), suggesting better performance in overlapping class instances. In secondary metrics, the CNN-BiLSTM also shows the highest Negative Predictive Value (NPV) (0.93 ± 0.01), while the RSN-Count model leads in Positive Predictive Value (PPV) (0.73 ± 0.05), Sensitivity (0.83 ± 0.07), and Accuracy (0.83 ± 0.07). These results indicate that while CNN-BiLSTM has a balanced performance across various metrics, RSN-Count may be preferred for applications requiring high sensitivity and accuracy. The traditional CNN model, while outperformed by the others, still maintains a consistent baseline across all evaluated metrics.

VI. DISCUSSION

RSN-Count differentiates itself from existing deep learning-based sleep apnea detection methods by estimating the AHI directly via treating apnea events as single units in time, rather than a region with a precise begin and end time, as is the case with previous proposed DL based methods. The main methods of performing AHI estimation with deep learning are 1) direct regression on the AHI [30], 2) classification on windows of time, where the entire window is evaluated as containing an apnea event or not [29], or 3) treating every second (or other timestep) as a moment that needs to be classified as apnea or not [20], [31]. They all have their own drawbacks:

- 1) Regression methods need to infer the task, as no prior information about the events is given
- 2) Windowed classification methods suffer in cases when there is overlap, or multiple apnea events occurring in a single window
- 3) Per-second classification methods require an artificial threshold, both in probability and time.

RSN-Count treats its targets as single events that need to be predicted. Its novelty lies in the use of Spiking Neural Networks and a novel loss function that focuses the predictive power of

TABLE II
MODEL PERFORMANCE COMPARISON: CLASSIFICATION METRICS

Model Type	ROC AUC	IoU	AP	NPV	Specificity	PPV	Sensitivity	Accuracy
CNN	0.67 ± 0.05	0.26 ± 0.05	0.46 ± 0.04	0.85 ± 0.02	0.94 ± 0.02	0.57 ± 0.04	0.32 ± 0.09	0.82 ± 0.02
CNN-BiLSTM	0.87 ± 0.03	0.45 ± 0.05	0.71 ± 0.06	0.93 ± 0.01	0.83 ± 0.04	0.53 ± 0.04	0.75 ± 0.09	0.82 ± 0.02
RSN-Count	0.83 ± 0.03	0.64 ± 0.07	0.67 ± 0.06	0.90 ± 0.02	0.83 ± 0.04	0.73 ± 0.05	0.83 ± 0.07	0.83 ± 0.07

The best values per metric are indicated in bold.

deep learning models to output single spikes or events, close to the center of the actual event. This is done by using an MSE distance loss, rather than cross-entropy, which is the common method of training deep learning classifiers. This property make the approach more appealing in cases where annotation of apneic events is imprecise, or where the exact start and end times are not relevant. Although event annotations are based on strict AASM guidelines, there can be situations where the recorded signals are affected by noise or artifacts. Additionally, due to additional signals not being present in the training data used for this study (only audio and SpO₂ for convenience of the subject), there may be inherent uncertainty on the exact location of the apnea event given this limited set of data sources. Existing ML-based approaches often struggle to deal with this uncertainty [32], leading to a loss of performance, whereas the design of RSN-Count inherently addresses these challenges.

When comparing RSN-Count to other state-of-the-art Deep Learning-based methods for sleep apnea detection such as CNNs and CNN-BiLSTMs [16], [17], [20], [21], the novel approach shows superior performance for AHI estimation when making use of smartphone audio and SpO₂. While the accuracies can not directly be compared due to the differences in data, the findings in this work are similar and exceed those from previous studies making use of smartphone audio, such as [33], who report an average offset of 0.23 (95% CI [−28.73, 29.18]), similar to the performance of the CNN reported in this work, with 0.97 (95% CI [−30.13, 32.07]). Comparing these results to RSN-Count's −0.02 (95% CI [−20.33, 20.29]), it is clear that with a similar architecture and a novel loss function, superior results can be obtained. The combination of sleep sounds and SpO₂ has also already been used with deep learning, with similar results pertaining to classification metrics as in [34], where they report accuracy, sensitivity and specificity of 0.84, 0.84 and 0.84 respectively (rounded to two significant digits), whereas RSN-Count presents accuracy, sensitivity and specificity of 0.83, 0.83 and 0.83 respectively.

Unlike conventional methods that focus on pinpointing events, RSN-Count's counting-based learning scheme is more flexible and can more accurately tackle the problem of AHI estimation, while retaining similar "classical" classification performance. The increase in IoU also indicates that RSN-Count is more accurately aligned with the actual timestamps in which apneas occur compared to CNN and CNN-BiLSTM.

One limitation of the study is the size of the data set. Due to the unavailability of high quality audio recordings in public datasets such as the Sleep Heart Health Study (SHHS) [35], a small dataset is used (as in [4], [29]). However, there is a lot of information present in the audio recordings due to the heterogeneous data collection from sleep centers and at-home

recordings, resulting in approximately 211 hours of data. An interesting future study would be to validate the generalisability of the technique when tested on data sets from other sleep centres.

Another limitation of the study is that acoustic signals captured using a smartphone can be sensitive to background noise and environmental sounds that interfere with the detection of sleep apnea. While this data was used to validate and demonstrate the proposed methodology, its application is not necessarily exclusive to audio recordings and SpO₂ signals. As mentioned earlier, sleep apnea detection can also be based on other types of respiratory-related signals, such as respiratory flow, thoracic effort, bio-impedance, and others. Although the effectiveness of the novel method has not been exhaustively explored on all types of signals, it is hypothesized — as future work — that the conceptual model of RSN-Count (see Fig. 1) is sufficiently general to accommodate further expansion to other signal types as well.

VII. CONCLUSION

RSN-Count represents a paradigm shift in the development of ML-based solutions for the screening of patients or sleep apnea severity assessment by leveraging concepts from spiking neural networks. Apneic events can be counted from an overnight recording, rather than aiming to exactly pinpoint individual apneic events on a time scale. The novel algorithm is validated on recordings of acoustic signals and oxygen saturation that were recorded with a smartphone, an approach that is particularly suitable for use in a home environment. The results confirm that RSN-Count leads to a more accurate estimation of the AHI (MAE 6.17 ± 2.21) when compared to baseline models reproduced in this work (MAE 8.52 ± 3.20 and 11.49 ± 1.76).

ACKNOWLEDGMENT

The authors thank Dr. Josep Maria Montserrat and the technicians from the Hospital Clínic sleep laboratory for their help in patient recruitment and data acquisition and annotation.

REFERENCES

- [1] R. Heinzer et al., "Prevalence of sleep-disordered breathing in the general population: The HypnoLaus study," *Lancet Respir. Med.*, vol. 3, no. 4, pp. 310–318, 2015.
- [2] T. Young et al., "Estimation of the clinically diagnosed proportion of sleep apnea syndrome in middle-aged men and women," *Sleep*, vol. 20, no. 9, pp. 705–706, 1997.
- [3] R. B. Berry et al., "The AASM manual for the scoring of sleep and associated events," *Rules, Terminology Tech. Specifications, Darien, Illinois, Amer. Acad. Sleep Med.*, vol. 176, 2012, Art. no. 2012.

- [4] Y. Castillo-Escario et al., "Entropy analysis of acoustic signals recorded with a smartphone for detecting apneas and hypopneas: A comparison with a commercial system for home sleep apnea diagnosis," *IEEE Access*, vol. 7, pp. 128224–128241, 2019.
- [5] T. Van Steenkiste et al., "Portable detection of apnea and hypopnea events using bio-impedance of the chest and deep learning," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 9, pp. 2589–2598, Sep. 2020.
- [6] N. Collop, "Scoring variability between polysomnography technologists in different sleep laboratories," *Sleep Med.*, vol. 3, pp. 43–47, 2002.
- [7] M. Al-Mardini et al., "Classifying obstructive sleep apnea using smartphones," *J. Biomed. Informat.*, vol. 52, pp. 251–259, Dec. 2014.
- [8] J. Behar et al., "SleepAp: An automated obstructive sleep apnoea screening application for smartphones," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 1, pp. 325–331, Jan. 2015.
- [9] J.-W. Kim et al., "Prediction of apnea-hypopnea index using sound data collected by a noncontact device," *Otolaryngology–Head Neck Surg.*, vol. 162, pp. 392–399, Mar. 2020.
- [10] S. Nikkonen et al., "Automatic respiratory event scoring in obstructive sleep apnea using a long short-term memory neural network," *IEEE J. Biomed. Health Inform.*, vol. 25, no. 8, pp. 2917–2927, Aug. 2021.
- [11] H. Nakano et al., "Monitoring sound to quantify snoring and sleep apnea severity using a smartphone: Proof of concept," *J. Clin. Sleep Med.*, vol. 10, no. 1, pp. 73–78, 2014.
- [12] Á. Serrano Alarcón, N. Martínez Madrid, and R. Seepold, "A minimum set of physiological parameters to diagnose obstructive sleep apnea syndrome using non-invasive portable monitors. a systematic review," *Life*, vol. 11, 2021, Art. no. 1249.
- [13] A. Sillaparaya et al., "Obstructive sleep apnea classification using snore sounds based on deep learning," in *Proc. 2022 Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2022, pp. 1152–1155.
- [14] T. Van Steenkiste et al., "Automated sleep apnea detection in raw respiratory signals using long short-term memory neural networks," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 6, pp. 2354–2364, Nov. 2019.
- [15] B. Wang et al., "Obstructive sleep apnea detection based on sleep sounds via deep learning," *Nat. Sci. Sleep*, vol. 14, pp. 2033–2045, Nov. 2022.
- [16] J. Xie et al., "Audio-based snore detection using deep neural networks," *Comput. Methods Programs Biomed.*, vol. 200, 2021, Art. no. 105917.
- [17] J. Zhang et al., "Automatic detection of obstructive sleep apnea events using a deep CNN-LSTM model," *Comput. Intell. Neurosci.*, vol. 2021, 2021, Art. no. 5594733.
- [18] R. Haidar, I. Koprinska, and B. Jeffries, "Sleep apnea event detection from nasal airflow using convolutional neural networks," in *Proc. Neural Inf. Process., 24th Int. Conf.*, 2017, pp. 819–827.
- [19] J. Marcos et al., "Assessment of four statistical pattern recognition techniques to assist in obstructive sleep apnoea diagnosis from nocturnal oximetry," *Med. Eng. Phys.*, vol. 31, pp. 971–978, 2009.
- [20] H. B. Kwon et al., "Hybrid CNN-LSTM network for real-time apnea-hypopnea event detection based on IR-UWB radar," *IEEE Access*, vol. 10, pp. 17556–17564, 2022.
- [21] Y. Chen et al., "Prediction of sleep apnea events using a CNN transformer network and contactless breathing vibration signals," *Bioengineering*, vol. 10, no. 746, 2023, Art. no. 746.
- [22] E. Wang, I. Koprinska, and B. Jeffries, "Sleep apnea prediction using deep learning," *IEEE J. Biomed. Health Inform.*, vol. 27, no. 11, pp. 5644–5654, Nov. 2023.
- [23] D. Ferreira-Santos et al., "Enabling early obstructive sleep apnea diagnosis with machine learning: Systematic review," *J. Med. Internet Res.*, vol. 24, no. 9, 2022, Art. no. e39452.
- [24] J. Xie et al., "The use of respiratory effort improves an ecg-based deep learning algorithm to assess sleep-disordered breathing," *Diagnostics*, vol. 13, no. 13, 2023, Art. no. 2146.
- [25] H. Han and J. Oh, "Application of various machine learning techniques to predict obstructive sleep apnea syndrome severity," *Sci. Rep.*, vol. 14, no. 1, pp. 1–10, 2023.
- [26] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *Int. J. Neural Syst.*, vol. 19, no. 4, pp. 295–308, 2009.
- [27] A. Tavanaei et al., "Deep learning in spiking neural networks," *Neural Netw.*, vol. 111, pp. 47–63, 2019.
- [28] M. K. Erman et al., "Validation of the apnealink™ for the screening of sleep apnea: A novel and simple single-channel recording device," *J. Clin. Sleep Med.*, vol. 3, no. 4, pp. 387–392, 2007.
- [29] Y. Castillo-Escario et al., "Convolutional neural networks for apnea detection from smartphone audio signals: Effect of window size," in *Proc. 2022 44th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2022, pp. 666–669.
- [30] J. Levy et al., "Deep learning for obstructive sleep apnea diagnosis based on single channel oximetry," *Nat. Commun.*, vol. 14, no. 1, 2023, Art. no. 4881.
- [31] L. Cen et al., "Automatic system for obstructive sleep apnea events detection using convolutional neural network," in *Proc. 2018 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2018, pp. 3975–3978.
- [32] H. Song et al., "Learning from noisy labels with deep neural networks: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 8135–8153, Nov. 2023.
- [33] S.-W. Cho et al., "Evaluating prediction models of sleep apnea from smartphone-recorded sleep breathing sounds," *JAMA Otolaryngology–Head Neck Surg.*, vol. 148, no. 6, pp. 515–521, 2022.
- [34] C. Singtothong and T. Siriborvornratanakul, "Deep-learning based sleep apnea detection using sleep sound, SpO2, and pulse rate," *Int. J. Inf. Technol.*, vol. 16, pp. 4869–4874, 2024.
- [35] S. F. Quan et al., "The sleep heart health study: Design, rationale, and methods," *Sleep*, vol. 20, no. 12, pp. 1077–1085, 1997.