Self-Supervised Representation Learning and Temporal-Spectral Feature Fusion for Bed Occupancy Detection

YINGJIAN SONG, The University of Georgia, USA
ZAID FAROOQ PITAFI, The University of Georgia, USA
FEI DOU, The University of Georgia, USA
JIN SUN, The University of Georgia, USA
XIANG ZHANG, University of North Carolina at Charlotte, USA
BRADLEY G. PHILLIPS, The University of Georgia, USA
WENZHAN SONG, The University of Georgia, USA

In automated sleep monitoring systems, bed occupancy detection is the foundation or the first step before other downstream tasks, such as inferring sleep activities and vital signs. The existing methods do not generalize well to real-world environments due to single environment settings and rely on threshold-based approaches. Manually selecting thresholds requires observing a large amount of data and may not yield optimal results. In contrast, acquiring extensive labeled sensory data poses significant challenges regarding cost and time. Hence, developing models capable of generalizing across diverse environments with limited data is imperative. This paper introduces SeismoDot, which consists of a self-supervised learning module and a spectral-temporal feature fusion module for bed occupancy detection. Unlike conventional methods that require separate pre-training and fine-tuning, our self-supervised learning module is co-optimized with the primary target task, which directs learned representations toward a task-relevant embedding space while expanding the feature space. The proposed feature fusion module enables the simultaneous exploitation of temporal and spectral features, enhancing the diversity of information from both domains. By combining these techniques, SeismoDot expands the diversity of embedding space for both the temporal and spectral domains to enhance its generalizability across different environments. SeismoDot not only achieves high accuracy (98.49%) and F1 scores (98.08%) across 13 diverse environments, but it also maintains high performance (97.01% accuracy and 96.54% F1 score) even when trained with just 20% (4 days) of the total data. This demonstrates its exceptional ability to generalize across various environmental settings, even with limited data availability.

CCS Concepts: • **Human-centered computing → Ubiquitous and pervasive computing**.

Additional Key Words and Phrases: Bed Occupancy, Self-Supervised Learning, Spectrum-temporal feature fusion

ACM Reference Format:
Yingjian Song, Zaid Farooq Pitafi, Fei Dou, Jin Sun, Xiang Zhang, Bradley G. Phillips, and WenZhan Song. 2024. Self-Supervised Representation Learning and Temporal-Spectral Feature Fusion for Bed Occupancy Detection. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 3, Article 124 (September 2024), 25 pages. https://doi.org/https://doi.org/10.1145/3678514

Authors' addresses: Yingjian Song, The University of Georgia, USA, Athens, Georgia, Yingjian.Song@uga.edu; Zaid Farooq Pitafi, The University of Georgia, USA, Athens, Georgia, zp18941@uga.edu; Fei Dou, The University of Georgia, USA, Athens, Georgia, fei.dou@uga.edu; Jin Sun, The University of Georgia, USA, Athens, Georgia, jinsun@uga.edu; Xiang Zhang, University of North Carolina at Charlotte, USA, Athens, Georgia, xiang.zhang@charlotte.edu; Bradley G. Phillips, The University of Georgia, USA, Athens, Georgia, bgp@uga.edu; WenZhan Song, The University of Georgia, USA, Athens, Georgia, wsong@uga.edu.

2474-9567/2024/9-ART124
https://doi.org/https://doi.org/10.1145/3678514

1 INTRODUCTION

Bed occupancy detection, which determines whether a person is present on the bed, stands as the key step for developing sleep monitoring systems. Ensuring accurate detection of bed occupancy is essential for authenticating subsequent tasks, such as monitoring heart rate, respiration rate, and sleep posture recognition [3, 11, 15, 25, 30, 32, 34, 38]. Without reliable bed occupancy detection, the system may inaccurately calculate sleep parameters when no individual is present on the bed, compromising user experience and individual-specific parameters' accuracy. This underscores the importance of research in bed occupancy detection.

Existing systems for bed occupancy detection can be categorized according to the types of sensors they applied, such as cameras [39], mat pressure sensors [20, 27], and accelerometers [1, 18]. However, using cameras may raise privacy concerns, especially during sleep. Mat pressure sensors can be challenging to install and may cause discomfort for individuals during sleep. At the same time, accelerometers may lack sensitivity to capture heartbeat signals in a contactless manner [26], given their typical usage as wearable devices [16]. In contrast, seismic sensors offer a privacy-friendly, easy-to-install, and comfortable monitoring solution, with sufficient sensitivity to detect mini-vibrations caused by heartbeat [11, 29].

Some existing methods for bed occupancy detection are limited to identifying only the 'moment of bed entry' and 'moment of bed exit' [1, 4, 18, 27], lacking continuous monitoring capability. Others using threshold-based approaches [1, 20, 26, 27] not only require extensive labeled data for optimal threshold determination but are also limited to the single environment setting. Seismic sensors exhibit excellent sensitivity in capturing heartbeat signals contactless manner [4, 11, 12, 26], making them a preferred choice for sleep monitoring. However, their high sensitivity also makes them susceptible to picking up background noise from external sources such as nearby footsteps or air conditioning sounds[11, 19, 26, 29], which presents challenges for bed occupancy detection. Moreover, changes in background noise distribution across different environments may impact the model's generalization ability. These challenges emphasize the necessity of developing a method that can effectively learn robust representations from a limited dataset and handle generalization issues resulting from changes in background noise distribution.

In this paper, we introduce SeismoDot, a novel framework that leverages self-supervised learning and spectrum-temporal feature fusion techniques to achieve continuous and accurate bed occupancy detection across diverse environments, even with limited labeled training data. Self-supervised learning aids in robust representation learning by leveraging inherent data structure to generate training signals, enhancing model adaptability to varied data distributions [23, 24, 40]. The prevailing self-supervised learning method involves pre-training on the entire unlabeled dataset followed by fine-tuning, where the pre-trained feature extractor is frozen, and only the final layers are trained for classification. However, there is no guarantee that the representations learned during pre-training are highly relevant to downstream tasks. In contrast, our proposed self-supervised learning module operates as an auxiliary objective, concurrently optimized with the primary bed occupancy detection task, which ensures that the learned representations are relevant to downstream tasks.

In sensor-related tasks such as human activity recognition, atrial fibrillation detection, and occupancy detection, incorporating both spectral and temporal features is essential due to their complementary characteristics. While some methods rely solely on temporal domain features [6, 17, 23, 37] and others exclusively on spectral domain features [40]. Integrating spectral and temporal features for bed occupancy is necessary because (1) solely focusing on the time domain can lead to false detections due to periodic background noise, and (2) exclusive emphasis on the frequency domain may result in errors when dominant frequency ranges overlap between on-bed and off-bed signals. The proposed self-supervised learning and spectral-temporal feature fusion modules leverage spectral and temporal domain features to enhance bed occupancy detection performance. However, they operate differently. The self-supervised learning module expands feature space through data augmentation and multi-class learning on concatenated spectral and temporal features. In contrast, the spectral-temporal feature fusion module

maximizes information diversity between temporal and spectral feature spaces, facilitating the effective fusion of representations between the two domains. Both modules enhance feature information but in different spaces. By combining both modules, the performance degradation due to changes in background noise distribution across environments could be effectively addressed.

We outline the **key contributions** of this paper as follows:

(1) We proposed SeismoDot, a bed occupancy detection framework that leverages temporal and spectral features instead of using temporal features alone. SeismoDot has two key components: a self-supervised learning module and a spectral-temporal feature fusion module. SeismoDot exhibits outstanding performance using only 20% (4 days) of the labeled training data, outperforming end-to-end fully supervised learning employing only temporal data trained on the entire dataset (23 days).
(2) We propose a spectrum-temporal feature fusion module that leverages information from both the frequency and time domains, demonstrating its effectiveness through experiments.
(3) Our comprehensive evaluation, conducted across various users and diverse environments, yields an average accuracy of 98.49% and an F1 score of 98.08%.
(4) Our proposed system could be deployed in real-world environments, enabling continuous real-time bed occupancy detection.

2 RELATED WORKS

2.1 Bed Occupancy Detection

Existing bed occupancy detection methods, such as [1, 20, 27], calculate a baseline threshold when the bed is unoccupied, which is then compared to incoming signals to determine occupancy status. While this method is intuitive and easy to implement, it requires re-calibration for each new environment, which may not be practical in real-world applications. [20] employs an accelerometer and capacitive proximity sensors to perform bed occupancy detection; the accelerometer can only detect the 'moment of bed exit' and 'moment of bed entry'; capacitive proximity sensors have to be placed in touch with the human body [2]. Similarly, [1, 27] detects the 'moment of bed exit' and 'moment of bed entry' but using a mat pressure sensor. However, this type of sensor is not easy to install [12].

In [18], bed occupancy detection uses an accelerometer with feature extraction and Long Short-Term Memory (LSTM) techniques. Their method employs three distinct classifiers: one for detecting stage changes, another for identifying intervals between stage changes, and a third for making corrections based on the outputs of the first two classifiers, achieving good performance. However, the involvement of multiple stages in classification and feature extraction may improve accuracy at the cost of increased complexity during training. Additionally, their testing was limited to the same environmental conditions, involving only seven patients. While accelerometer data can detect bed occupancy, its capability to capture heartbeat signals remains unexplored in [18]. This is critical, as bed occupancy detection aims to enable subsequent tasks such as heart rate estimation, necessitating high-quality heartbeat signals.

The bed occupancy methods outlined in [4, 14, 26] utilize seismic sensors, which are straightforward to set up and sufficiently sensitive to capture heartbeat signals. However, their approaches have certain limitations. For instance, [4] introduced a novel feature fusion method by combining Spectral Entropy, Kurtosis, and Teager Energy Operator (TEO) to identify bed entry and exit events, achieving high accuracy on their dataset. Nevertheless, their method exclusively focuses on these events, which may result in cascading errors if either detection fails. While both [14] and [26] continuously monitor bed occupancy, they rely on threshold-based signal processing methods, which may lack robustness compared to data-driven methods like deep learning. [14] could experience a significant performance drop when tested in more complex environments, as it utilizes an autocorrelation function to differentiate between 'on bed' and 'off bed' signals, which are sensitive to noisy periodic signals. Although this method can achieve high accuracy, another periodic signal besides a heartbeat could lead to false

detection. [26] introduced a signal processing method that consistently merges zero crossing rate and kurtosis to detect on-bed and off-bed states with remarkable accuracy. However, it requires back correction since movement could potentially influence off-bed detection, and determining optimal thresholds may demand a substantial amount of labeled data.

In contrast, SeismoDot distinguishes itself in the following ways: (1) It performs bed occupancy detection continuously using a deep learning model without the need for back corrections. (2) It does not demand extensive training data to adapt to various environments. While threshold-based or statistical signal processing methods do not require training data, they often require exposure to significant data volumes to fine-tune thresholds for optimal performance. (3) It demonstrates robustness to environmental changes and requires no re-initialization when deployed in new settings.

2.2 Self-supervised Representation Learning for Time-Series

Contrastive learning, as described by [17], aims to learn invariant representations by maximizing the mutual information between past and future representations. This approach involves comparing similar and dissimilar data points and encouraging the network to produce similar representations for the former and dissimilar representations for the latter. By maximizing the mutual information between these pairs, contrastive learning enables the network to learn robust and generalizable features that can be used for a range of downstream tasks.

In recent works by Zhang et al. [37], a novel model'Crossformer', built upon the transformer architecture, is introduced. What sets this model apart is its ability to simultaneously capture both temporal dynamics and the interplay between different modalities. Their experiments showcased its impressive performance, and while the primary aim of their paper was to enhance forecasting results, it also contributes to representation learning—a fundamental aspect, as many existing studies leverage forecasting for representation learning.

Eldele et al. [6] proposed a contrastive learning framework focused on learning representations. They incorporate both temporal and contextual contrast aspects into their approach. In addition, they propose a cross-view module to enhance representation learning further. [23] introduced a multi-task self-supervised learning paradigm. This paradigm transforms raw data into various views and conducts binary classification on each transformation to learn robust representations. Sarkar et al. [24] build upon a similar idea and apply it to different tasks. Yue et al. [33] delve into self-representation learning for time-series data. They achieve this by generating diverse data views using masking and random cropping of time steps facilitated by a hierarchical temporal contrast mechanism. Kiyasseh et al. [13] propose a contrastive learning framework to learn invariant representations within individual patients. While promising for personalized health monitoring, its ability to generalize across different individuals may be limited. Tonekaboni et al. [28] tackle the challenge of estimating stationary temporal window sizes using the Augmented Dickey-Fuller (ADF) statistical test. They consider issues with naive negative sampling, which may include false negatives and adversely affect embedding learning, addressing this through Positive Unlabeled learning. Fan et al. [7] explore the relationships between different samples in the time domain and within the same sample but across different time segments. Lastly, Zhang et al. [35] propose a modified contrastive learning method known as Skip-Step Contrastive Predictive Coding, which includes a skip step to create a temporal gap between future and past representations.

[5] introduced a novel contrastive learning framework designed for multi-modality sensors. They considered sensor data from different modalities but at the same time-step as positive pairs. In contrast, data from the same modality but at different time steps were treated as negative pairs. [10] proposed a device selection algorithm aimed at multi-modality contrastive learning on sensor data. Their method selects the device whose data sample has the least Maximum Mean Discrepancy (MMD) [8] to the anchor device as the positive device. Data samples from the same time-step of the positive device pair are considered positive pairs. In contrast, weights are assigned to all devices (except the anchor device) for negative device selection based on the reciprocal of MMD to the

anchor device. Samples from these devices that are not time-aligned with the anchor data sample are treated as negative pairs. A key distinction between the two methods lies in negative sample selection: while [10] considers samples from different devices but with different time steps as negative samples, [5] considers samples from the same device at different time steps as negative samples. However, both approaches require multi-modality sensors rather than a single sensor. It is important to note that multi-modality differs from the concept of spectral and temporal domains of the same sensor proposed in this paper.

These approaches above contribute to self-supervised learning and representation learning in various domains. However, the literature discussed thus far predominantly focuses on utilizing temporal information, where spectral information might not be relevant to temporal features. Nonetheless, it is worth highlighting that spectral information remains crucial in tasks like occupancy detection.

On the contrary, [36] took into account both temporal and spectral domains in their approach. They introduced a contrastive learning method to enhance data representations by maximizing the similarity between original and augmented data in both the time and frequency domains. Furthermore, they projected time and frequency representations into a shared latent space to ensure alignment. Although this method proved effective and achieved high performance across various datasets by leveraging features from both temporal and spectral domains, their technique for frequency domain augmentation may not be suitable for bed occupancy. This is because they directly perturb the frequency domain by adding one or multiple random dominant frequency bands, which could destroy the distribution of the frequency domain. Moreover, their focus on time and frequency feature alignment rather than fusion might result in a loss of information. Furthermore, the pre-training plus fine-tuning strategy cannot ensure the learned representation is highly relevant to downstream tasks.

[31] suggested the utilization of dropout on input data as an instance-level augmentation technique. Their experiments indicated that dropout augmentation outperformed various other augmentation techniques. Furthermore, they introduced a bilinear temporal-spectral fusion module aimed at iteratively refining representations. They employed contrastive learning to maximize the alignment between temporal and spectral features. However, our approach differs from theirs, as we focus on promoting diversity between temporal and spectral features rather than making them homogeneous.

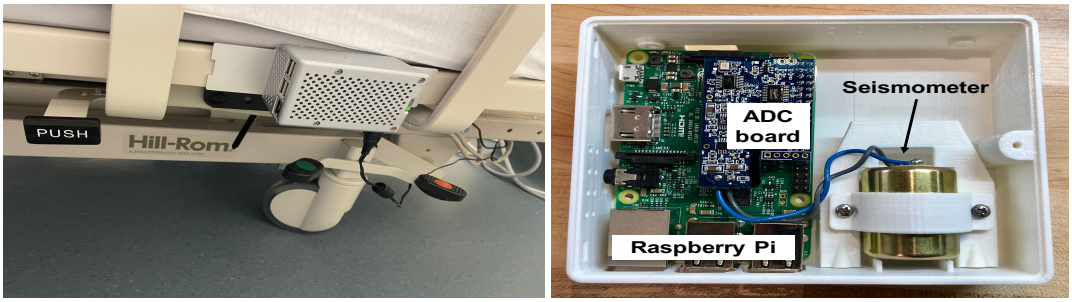


Fig. 1. Installation of SeismoDot on the side of bed frame(left), Design of SeismoDot(right).

3 SYSTEM DESIGN AND METHODOLOGY

In this section, we provide insights into our hardware platform, followed by a comprehensive overview of the SeismoDot methodology framework. Next, we discuss pre-processing techniques applied to raw data and data augmentation on both time and spectral domains. We then delve into the self-supervised learning module, which utilizes multi-class classification on concatenated temporal and spectral features. Furthermore, we explore the spectral-temporal feature fusion module. Finally, we offer a step-by-step guide to the training and inference procedure.

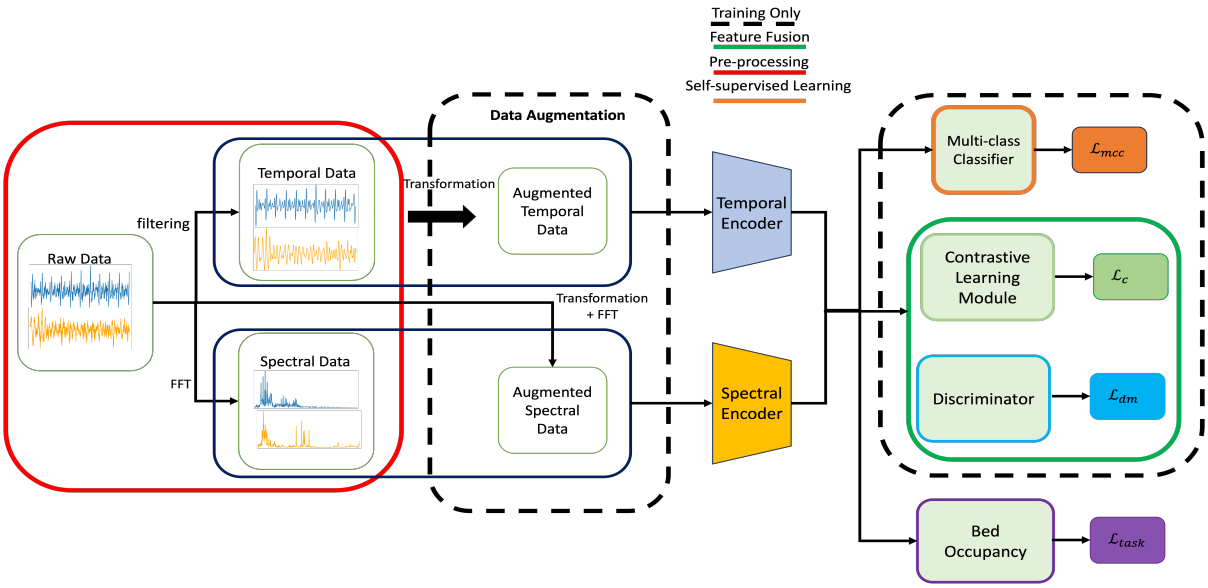


Fig. 2. Workflow of SeismoDot.

3.1 Hardware Platform

Figure 1 visually presents our system's design and critical components. SeismoDot comprises a Raspberry Pi serving as the onboard computer, an Analog Digital Converter (ADC) board that efficiently digitizes data at a sampling rate of 100 Hz, and a vertical seismic sensor (seismometer) [22] designed to detect vibrations. This seismic sensor incorporates a magnetic element surrounded by wire coils, and as the magnetic element moves within these coils, it generates an electrical signal. The seismic sensor exhibits a remarkable sensitivity to micro-vibrations stemming from human heartbeat, enabling SeismoDot to capture real-time heartbeat without requiring physical contact. Furthermore, SeismoDot has a magnet mounted on it, which makes it easy to install as shown in the left of figure 1.

3.2 Overview of SeismoDot

Figure 2 provides an overview of the operational flow in SeismoDot. It begins with pre-processing data in the time domain using a band-pass filter. Subsequently, data augmentation is applied to input for both the time-domain and spectral domain of data, combined with multi-class classification, to generate efficient representations. Following this, the representations are being further refined via the spectrum-temporal feature fusion module. Meanwhile, the model is under supervised training using labeled training data to drive representations toward task-related feature space.

3.3 Data Pre-processing

We implemented a band-pass filter on the time domain data with a cutoff frequency between 2Hz and 10Hz. This choice is motivated by the fact that on-bed signals tend to predominantly occupy this frequency range, thus enhancing the visibility of on-bed features.

The spectral analysis of on-bed and off-bed signals reveals distinct patterns in the frequency components. To capture a more comprehensive representation, we conduct spectrum analysis on the raw data rather than the filtered data. This choice is motivated by the fact that raw data preserves richer patterns in the high-frequency

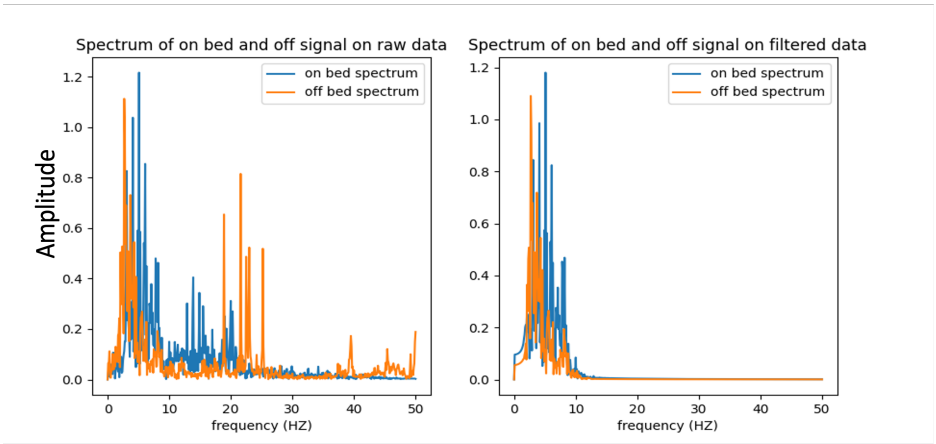


Fig. 3. Spectrum comparison between on-bed and off-bed data before and after filtering.

part, which contains valuable information for distinguishing between on-bed and off-bed instances, as depicted on the left side of figure 3.

3.4 Self-Supervised Representation Learning

We employed a Multi-class classification module to distinguish the data from diverse perspectives. These different perspectives were generated using various data augmentation techniques. Both data augmentation techniques and multi-class classification help enhance the diversity of the embedding features space, which is subsequently leveraged for the primary task. Moreover, it serves as an auxiliary task that is co-optimized alongside the primary task loss to ensure that the features learned through self-supervised learning are directly relevant to the downstream task. The prevailing approach to self-supervised learning involves the 'pre-train + fine-tuning' method, Which raises two concerns: (1) How to ensure that the representations learned during the pre-training stage are highly relevant to the downstream task. (2) Distribution shifts between training and testing datasets can impact the performance of all machine learning models. This issue is especially pronounced in self-supervised learning (SSL), where models are pre-trained unsupervised on unlabeled data. The inherent sensitivity of SSL models to these distribution divergences can pose significant challenges.

3.4.1 Data Augmentation. : We propose applying six data augmentations on temporal data for self-supervised learning to unveil general features within the temporal and spectral domains. Specifically, by using data augmentations on a filtered temporal input signal, they are denoted as $\hat{x}_i$, thereby generating six variations of $\hat{x}_i$, each coupled with corresponding variants of $FFT(x_i)$ within the spectral domain on raw temporal data. Subsequently, features from the temporal and spectral encoders are concatenated, forming seven categories for the subsequent multi-class classification task. Furthermore, both original and augmented data are used for bed occupancy detection during training. This strategy exposes the model to a diverse data space, making the training stage more challenging. Moreover, This model is trained concurrently with the primary (downstream) task, ensuring that the learned representation remains robust across various transformations on both the spectral and temporal domains. It reinforces its relevance to the specific downstream task, in this case, the bed occupancy detection. The data augmentation details for bed occupancy are summarized below:

(1) **Jitter:** Introducing random noise with zero mean and random standard deviation between 0 and 1 into the original signal serves as a means to simulate diverse background noise.

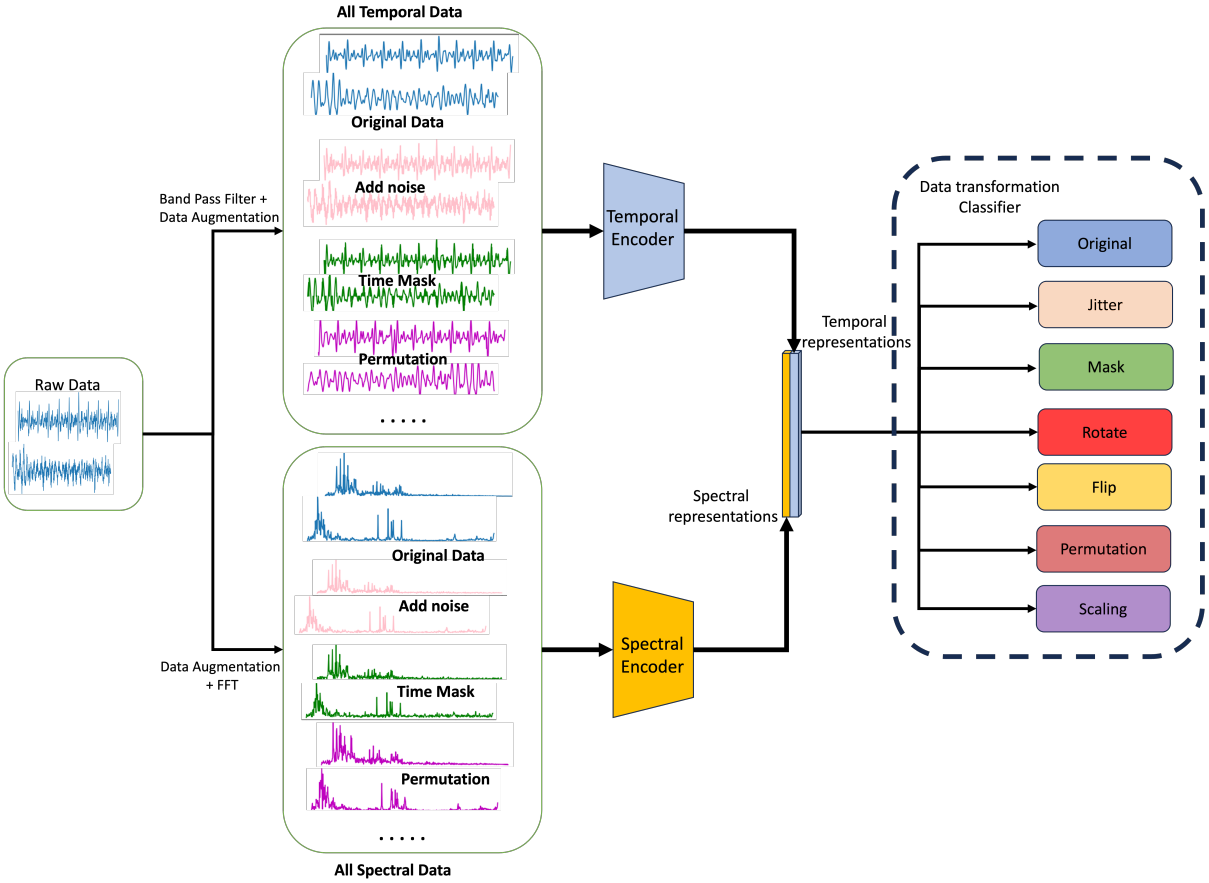


Fig. 4. Multi-class classification for distinguishing which data augmentation techniques have been applied. Each data pair represents an on-bed signal (up) and an off-bed signal (bottom).

(2) **Mask:** We have incorporated a dropout layer with a dropout rate 0.1 within the model architecture, positioned between the input temporal data and the model itself. This dropout layer introduces random masking within the time domain, yielding a temporally masked signal with a different view.
(3) **Rotation:** This transformation inversely adjusts the sample signs when the sensor is upside down.
(4) **Flip:** This transformation entails reversing the order of samples along the time dimension, effectively generating a mirrored original signal in the opposite time direction.
(5) **Permutation:** Dividing the signal into a variable number of segments at a maximum of N, and subsequently shuffling these segments randomly whose visualization of this data augmentation.
(6) **Scaling:** Alters the amplitude of the segmented signal by applying a random scalar factor ranging from 0.5 to 2.

3.4.2 Multi-class Classification for Expanding Feature Space. : In this approach, as shown in figure 4, we employ multi-class classification on six different views of the raw data, optimizing the model to distinguish between data in various augmented perspectives. This, in turn, enables the model to learn efficient and general representations.

Meanwhile, it leads to a larger distance in embedding space among all domains, including augmented domains and original domains.

An alternative approach proposed by [23] is to use multi-task classification. However, our choice of multi-class classification over multi-task classification is driven by the desire to reduce computational overhead as suggested in [21, 23], as multi-class classification requires only a single classifier, unlike multi-task classification, which necessitates multiple classifiers.

Data augmentation involves applying K data augmentations to filtered temporal data in the time domain. Regarding spectral data, data augmentations on raw time domain data must be performed first. Then, it is transformed into the corresponding frequency domain using FFT. This results in augmented data in both the time domain, denoted as $A_k(\hat{x_i})$, and the frequency domain, represented as $FFT(A_k(x_i))$. Here, A_k is the kth data augmentation technique applied to the temporal data, $\hat{x_i}$ is filtered data in the temporal domain, x_i is the raw temporal data, and $FFT(A_k(x_i))$ is the augmented spectral data.

The temporal encoder, denoted as $T_{\theta_t}(\cdot)$, takes in the augmented temporal input $A_k(\hat{x_i})$, while the spectral encoder, $S_{\theta_s}(\cdot)$, handles the augmented spectral input $FFT(A_k(x_i))$, producing compact representations, t_i^k and s_i^k, for the temporal and spectral domains, respectively. A classifier, $MC(\cdot)$, is integrated at the top of both encoders and takes in concatenated representations of the temporal and spectral features, resulting in $MC([t_i^k, s_i^k])$. This classifier aims to determine the probability of a given pair of temporal and spectral representations associated with a particular data augmentation technique, including the original data category, resulting in seven distinct categories. The label set for this auxiliary task is represented as $\mathcal{Y}^A$, while $\hat{\mathcal{X}}$ encompasses the space comprising both the original data and augmented data for the temporal domain. Furthermore, $FFT(\mathcal{X})$ denotes the whole spectral domain space. For the self-supervised learning task, we have employed cross-entropy loss to facilitate the learning of the mapping between the combined data spaces $[\mathcal{X}, FFT(\mathcal{X})]$ and the label space $\mathcal{Y}^A$. The objective function for the multi-class classification module is defined as:

$$\mathcal{L}_{mcc}\left(T_{\theta_t}, S_{\theta_s}; \hat{\mathcal{X}}, FFT(\mathcal{X}), \mathcal{Y}^A\right) = -\mathbb{E}_{([\hat{x_i}, FFT(x_i)] \in [\hat{\mathcal{X}}, FFT(\mathcal{X})], y_i^a \in \mathcal{Y}^A)} \sum_{k=0}^{K} y_i^a \log \delta_k \left[MC([t_i^k, s_i^k])\right] \tag{1}$$

where $t_i^k = T_{\theta_t}(A_k(\hat{x_i}))$, $s_i^k = S_{\theta_s}(FFT(A_k(x_i)))$, δ represents the softmax function and y_i corresponds to the category of data augmentation technique. It is important to note that when $i = 0$, data augmentation is essentially the original data itself.

3.5 Spectral-temporal feature fusion

We perform a fusion within the embedding space to harness the combined advantages of temporal and spectral features for bed occupancy detection. Nevertheless, before concatenating these features, we must ensure that the features we aim to fuse possess heterogeneous rather than homogeneous information. This is crucial because if the features from both spectral and temporal domains are similar, fusion becomes unnecessary, and we could effectively use either the spectral or temporal domain features to accomplish the downstream task. Hence, our focus is on maximizing the diversity of the learned representations between the spectral and temporal domains, differentiating the features in each space from one another to facilitate meaningful fusion.

In this pursuit, we emphasize maximizing the separation between the spectral and temporal feature spaces. However, we aim to prevent these features from becoming overly disparate since spectral features represent an alternative view of temporal features. As a result, we introduce constraints that preserve some overlap between the temporal and spectral features, ensuring that they remain connected within the embedding space to keep shared features across both domains rather than being entirely separated. The process of fusing spectral and temporal features is illustrated in figure 5.

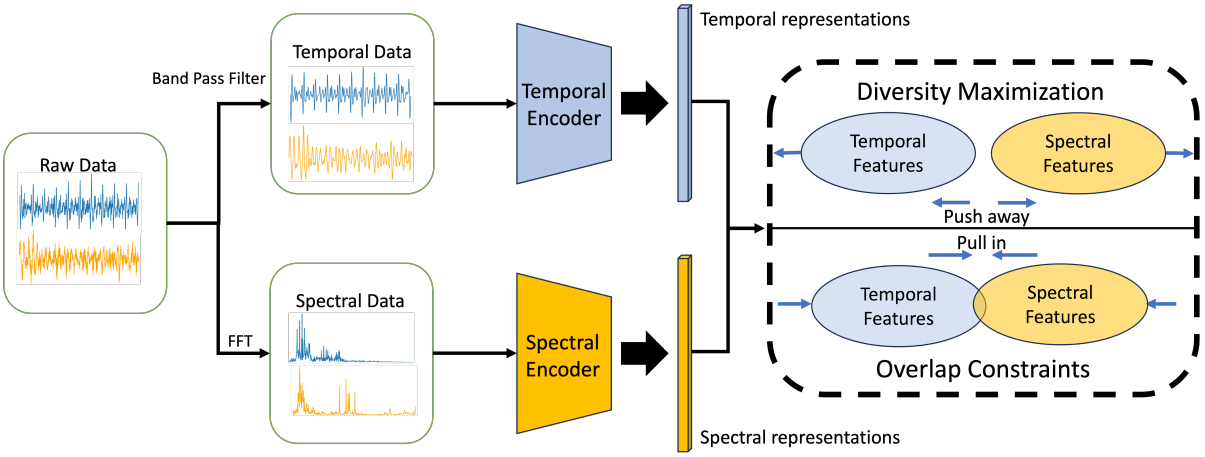


Fig. 5. Workflow of feature fusion module. Each data pair represents an on-bed signal (up) and an off-bed signal (bottom).

3.5.1 Diversity Maximization: To maximize the diversity between temporal and spectral features, we work to maximize the distance between these two feature sets. This objective is equivalent to minimizing the following loss, denoted as $\mathcal{L}_{dm}$, which is defined as:

$$\mathcal{L}_{dm} = -\operatorname{dist}(\mathcal{T}, \mathcal{S}) \tag{2}$$

Here, $\mathcal{L}_{dm}$ represents the feature diversity maximization loss, $\mathcal{T}$ corresponds to the domain of temporal features, and $\mathcal{S}$ refers to the domain of spectral features. The function $\operatorname{dist}(\cdot)$ calculates the distance between these two feature domains. Maximizing $\operatorname{dist}(\mathcal{T}, \mathcal{S})$ is equivalent to minimizing $-\operatorname{dist}(\mathcal{T}, \mathcal{S})$.

Inspired by [21], one way to quantify the distance between $\mathcal{T}$ and $\mathcal{S}$ is to consider it as the error associated with constructing a binary classifier tasked with distinguishing between these two domains. Consequently, maximizing the diversity between the temporal and spectral domains corresponds to minimizing the error of this binary classifier, specifically designed to distinguish between these domains. To achieve this, we initially employ separate projectors on top of both the temporal and spectral encoders for projecting temporal and spectral representations, respectively. Subsequently, we introduce a domain discriminator on top of these projectors, explicitly designed to differentiate between the temporal and spectral domains. By minimizing the error of the domain discriminator, we effectively enlarge the gap between the two domains, enabling their differentiation.

3.5.2 Overlap Constraints: While the objective is to introduce diversity into both temporal and spectral features rather than homogeneity, it is equally essential to maintain the overlap between temporal and spectral feature domains and avoid pushing them too far apart, which may lead to loss of shared information across both domains, considering that spectral features represent an alternative view of temporal features. Therefore, it is reasonable to assume that some degree of overlap should exist between the spectral and temporal domains, as depicted in figure 5, to prevent complete separation.

We employ contrastive learning to encourage an overlap between the temporal and spectral features to achieve this. It treats each pair of features from a given sample as positive pairs, and temporal and spectral features from different samples as negative pairs. Both the temporal and spectral data are passed through their respective encoders, extracting representations denoted as t_i^k and s_i^k for the temporal and spectral domains, respectively. Subsequently, these representations are mapped into new embedding pairs using two projectors, each consisting

of a fully connected layer added on top of the encoders. Finally, a contrastive loss function is applied to these pairs of projected embeddings, zt_i^k and zs_i^k. This function aims to maximize the mutual information between the temporal and spectral embeddings to maintain some overlap between the two types of embedding while minimizing the mutual information between embeddings from different data samples. The loss function is defined as follows:

$$\ell_i^{tc} = -\log \frac{\exp\left(\text{sim}\left(zt_i, zs_i\right)/\tau\right)}{\sum_{j=1}^{N} \mathbb{1}_{i\neq j} \exp\left(\text{sim}\left(zt_i, zt_j\right)/\tau\right) + \sum_{j=1}^{N} \mathbb{1}_{i\neq j} \exp\left(\text{sim}\left(zt_i, zs_j\right)/\tau\right)} \tag{3}$$

$$\ell_i^{sc} = -\log \frac{\exp\left(\text{sim}\left(zs_i, zt_i\right)/\tau\right)}{\sum_{j=1}^{N} \mathbb{1}_{i\neq j} \exp\left(\text{sim}\left(zs_i, zs_j\right)/\tau\right) + \sum_{j=1}^{N} \mathbb{1}_{i\neq j} \exp\left(\text{sim}\left(zs_i, zt_j\right)/\tau\right)} \tag{4}$$

$$\mathcal{L}_c = \frac{1}{2N} \sum_{i=1}^{N} (\ell_i^{tc} + \ell_i^{sc}) \tag{5}$$

where $\text{sim}(u, v) = u^T v/(\|u\| \cdot \|v\|)$, For each sample, the temporal view and spectral view of the same data sample are considered as a positive pair. Conversely, for any given sample, its temporal view zt_i and the temporal view of other samples zt_j, its spectral view zs_i and the spectral view of other samples zs_j, and cross views of other samples (zt_i, zs_j, and zs_i, zt_j), where $i \neq j$, are treated as negative pairs. In this context, τ denotes a temperature parameter, and $sim(u, v)$ represents the cosine similarity between vectors u and v. ℓ_i^{tc} is the temporal contrastive loss, ℓ_i^{sc} is the spectral contrastive loss, and $\mathcal{L}_c$ is the overall overlap constraints loss.

3.6 Training and Inference:

We present the training and inference process for SeismoDot in the context of bed occupancy detection. As illustrated in figure 2, the training phase begins with applying a band-pass filter to all temporal data. Subsequently, we employ FFT on the raw temporal data to obtain spectral features. Furthermore, we introduce six data augmentation techniques on the filtered temporal data, resulting in augmented temporal data. Similarly, we apply these augmentation methods to the raw temporal data and FFT to produce augmented spectral data. Both temporal and spectral data are passed to dedicated temporal and spectral encoders to extract embedding representations. The multi-class classification module uses the concatenated spectral and temporal representations pair from the original and augmented datasets, employing a multi-class classifier to learn general and efficient representations and expand data space.

Meanwhile, the feature fusion module utilizes discriminators to maximize the diversity between projected spectral and temporal features while maintaining some overlap using contrastive learning. The spectral and temporal representations are concatenated and fed into the bed occupancy classifier, ensuring that the learned representations space is highly relevant to the primary task. The overall loss function during the training stage is defined as follows:

$$\mathcal{L} = \mathcal{L}_{dm} + \lambda \mathcal{L}_c + \mathcal{L}_{mcc} + \mathcal{L}_{task} \tag{6}$$

where $\mathcal{L}_{task}$ represents the loss associated with the primary task, which, in our case, is bed occupancy. For the task loss, we employ the cross-entropy loss function.

The trained model is applied to new test data with filtered temporal and raw spectral features during inference to provide bed occupancy predictions. Data augmentation, multi-class classification, and feature fusion optimizations are not included during the inference stage.

4 EXPERIMENTS SET UP AND EVALUATIONS

4.1 Data Collection

SeismoDot can be installed using a mounted magnetic, allowing easy deployment on standard beds or seats without requiring any special modifications. Our experiments have been approved by IRB. The data collection process comprised two phases. During Phase 1, data were collected at a controlled hospital environment (hospital research data collection center) designated for multi-research purposes, including bed occupancy detection. Phase 2 was dedicated to evaluating the stability of our bed occupancy algorithm exclusively in real-world scenarios. For Phase 1, we gathered approximately 291 hours of data from 100 individuals, with 147 hours collected on hospital beds and 144 hours on foam beds. However, for training purposes, we only utilized the data from the first 50 individuals on a twin-size hospital bed (we denote as lab1), totaling around 75 hours, and the data from the last 50 individuals on a twin-size foam bed (we denoted as lab2), totaling around 67 hours. This selection was made to prevent overlap among patients in our training dataset. In Phase 2, we collected a total of about 401 hours of data on another 11 environments, including eight home environments (home1-home8), two hospital environments (hospital1, hospital2), and one office environment (office). The total data samples used during the training stage was 195478, with 93821 on-bed data and 101657 off-bed data. Each data sample is 10 seconds long, and there is no segment overlap among all data samples, and no same patients between the training and testing datasets. Figure 6 depicts class distribution across all environments.

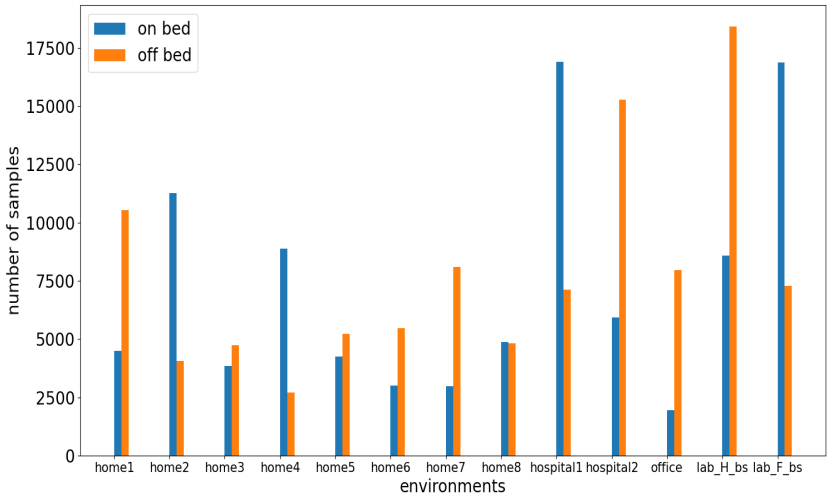


Fig. 6. Distribution of classes across all environments.

4.1.1 Phase 1 Data Collection. In the first stage, we collected data in a controlled hospital environment room with two types of beds: a twin-size hospital bed and a twin-size foam bed in this environment. Each bed had one of our devices positioned beneath the chest region. We gathered data from 100 participants who were instructed to lie on a hospital bed in four postures on the back, right, left, and stomach, with each posture for approximately 5 minutes. In addition, participants were instructed to perform specific activities between back posture and right posture, including moving their right leg, left leg, right arm, left arm, head to the left, and head to the right. Subsequently, participants were instructed to exit the bed and walk on a treadmill for approximately 5 minutes before returning to lie on the hospital bed for another 5 minutes. Following this, participants were encouraged to take a short nap lasting approximately one hour on the foam bed to simulate real-life sleep scenarios. The ages of the participants varied around a mean of 49 ± 18 years, consisting of 42 males and 58 females. On average, their heights measured 167.29 ± 8.65cm, with average weights of 80.81 ± 19.08 kg. In addition, we add another device

on the left side of the hospital bed, as shown in figure 1 for data of the last 25 people to further check how sensor position affects our proposed algorithms.

4.1.2 Phase 2 Data Collection. In the second stage, SeismoDot underwent further testing across 11 new environments, comprising eight home environments, two hospital environments, and one office environment, each involving different participants. The detailed information of beds for each environment is listed in table 1. The objective of phase 2 was to evaluate the performance of the bed occupancy algorithm in real-world scenarios. Therefore, we did not collect data on individual participants' weights, ages, or BMI during this phase. In Phase 2, participants were not given specific instructions and were encouraged to continue their usual daily activities.

Table 1. Environmental settings. Home4 has two people sleeping simultaneously.

Environment	Bed Size	Bed Type	Frame Material
Home1	Queen	Box spring	Metal
Home2	Queen	Standard	Wood
Home3	King	Standard	Metal
Home4	King	Standard	Metal
Home5	King	Standard	Wood
Home6	Queen	Box spring	None
Home7	Queen	Standard	Metal
Home8	Twin XL	Standard	Metal
Hospital1	Twin	Hospital bed	Standard
Hospital2	Twin	Standard	Wood
Office	Sofa bed	Sofa bed	None

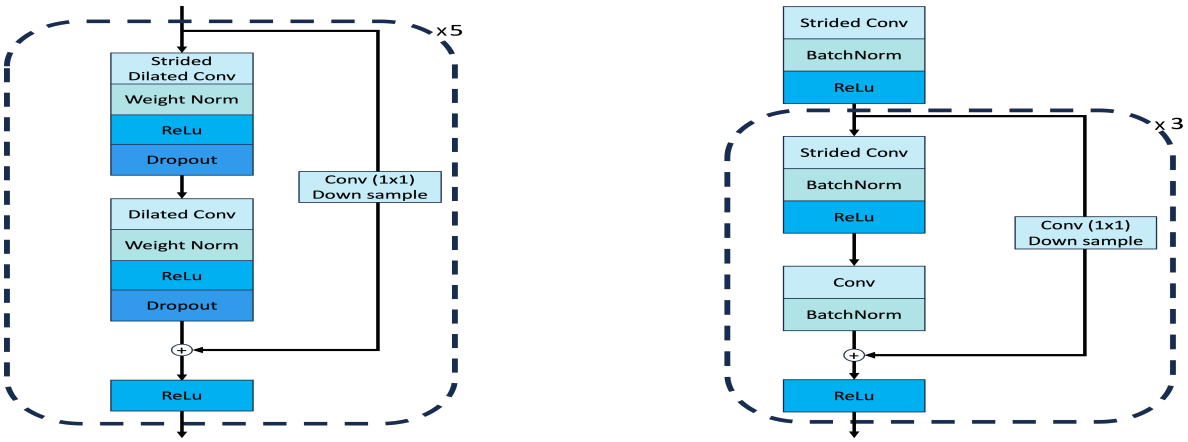
4.2 Implementation Details

In the training stage, we utilized an NVIDIA GTX 1080 GPU. The learning rate was set to 0.001, and we applied L2 regularization with 0.001 to the convolutional layers. We chose a value of 0.1 for the hyperparameter λ to balance the trade-off effectively.

The encoders for temporal data consist of five temporal residual blocks, illustrated in figure 7a. Our temporal input data has dimensions of 1×1000, where '1' denotes a single sensor as we only need one sensor installed under each bed, and '1000' represents the length of the input signal, which corresponds to 100 Hz data over 10 seconds. The output representations generated by the temporal encoders are of dimensions 256 with a temporal length of 31. The spectral data encoders consist of three standard residual blocks [9], along with an additional convolution layer preceding the three residual blocks, following the standard ResNet design [9]. The input spectral data has dimensions of 1×501, where 501 represents the length of the spectral signal. The output representations generated by the spectral encoders are the same as temporal encoders. The overall architecture for the spectral encoder is depicted in figure 7b.

4.3 Preliminary Results

In this section, we present our preliminary experiments, which guided the development of our current solutions. Initially, we explored threshold-based methods, which require manual selection of thresholds and may not be optimal, as discussed in section 4.3.1. Subsequently, we employed "feature extraction + machine learning" methods, such as Random Forest, to facilitate the selection of optimal thresholds. However, feature extraction may result in information loss and introduce domain shift, as illustrated in section 4.3.2.



(a) Temporal encoder (b) Spectral encoder

Fig. 7. Temporal and spectral encoders

4.3.1 Threshold Based Preliminary Results. Our initial investigations were inspired by strategies outlined in [14] and [26]. While [14] proved effective in scenarios where on-bed signals exhibited clear heartbeat patterns with a high Signal-to-Noise Ratio (SNR), it struggled with noisy off-bed signals containing periodic noise. The algorithm in [14] relied on the Auto-correlation Function (ACF) to directly identify periodic patterns in the data, thus relying on the periodicity of on-bed signals for detection. An example illustrating the limitations of this approach is depicted at the top of figure 8. For better visualization, the ground truth labels for on-bed (orange line) and off-bed (green line) periods are also plotted (the y-axis amplitude is not applicable for on-bed and off-bed; for example, the orange line is up means on-bed period regardless of the value of y-axis). Typically, the number of ACF peaks for on-bed periods is less than 5, while it varies between 0 and 17 for off-bed periods. Setting simple thresholds, such as 5, often leads to false detection of on-bed states when it is off-bed. Similar challenges were encountered, as depicted at the bottom of figure 8, with the method proposed in [26], which leverages zero crossing rate to distinguish between on-bed and off-bed signals. Compared to the method in [14], [26] performs better regarding average accuracy and F1 score. Therefore, we present results of [26] as one of the baseline methods.

4.3.2 Machine Learning Approach. We conducted observations on the collected data, and in conjunction with the results presented earlier, we extracted eight features and used the random forest as the classifier. Specifically, we extracted eight features as follows: (1) Primary frequency of raw data (2) Max value of a signal minus min value of that signal (3) Dominant frequency after band pass filter denoised signal (4) Zero crossing rate (5) Variance of mean from segmented ACF (6) Dominant Frequency of ACF (7) Max amplitude of spectrum calculated from ACF (8) Amplitude of max peak on ACF.

A significant challenge in bed occupancy detection arises from the distribution shift between the source environment (training data) and a new environment (testing data), particularly for off-bed data. This distribution shift leads to a performance drop when a model is trained on the source environment data and validated on data from a new environment using feature extraction + machine learning. To illustrate this issue, we present examples in figure 9, which shows the distribution shift between the training data (source environment) and the testing data (new environment) on the left side of figure 9. Furthermore, the distribution of both on-bed and off-bed data in training and testing exhibits an overlap, as depicted on the right side of figure 9. This overlap in distribution poses a challenge for the random forest model in accurately distinguishing between on-bed and off-bed data,

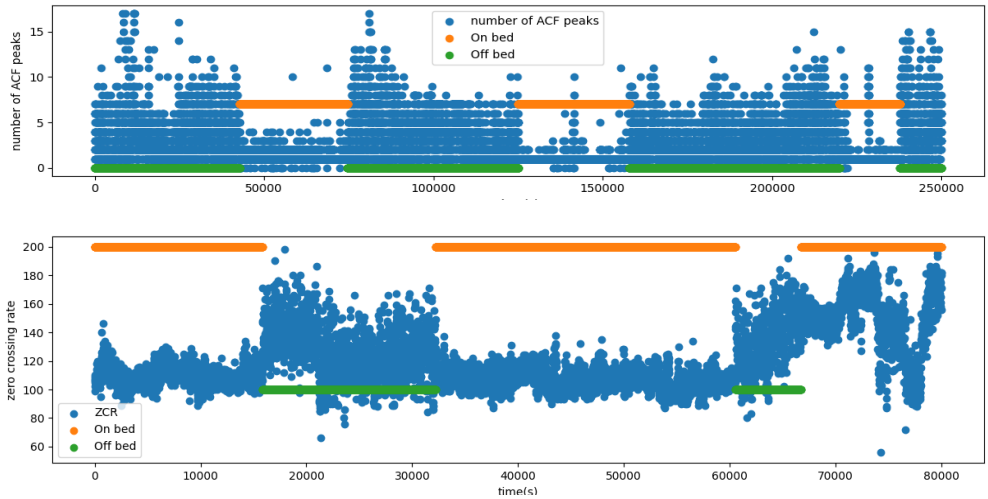


Fig. 8. Example results of [14] (top), Example results of [26] (bottom).Orange lines are on-bed period, and green lines are off-bed period.

especially in low data availability cases, contributing to the complexity of the bed occupancy detection task in diverse environments.

We selected Random Forest as one of our baseline models, given its robust performance and popularity in competitions like Kaggle. While we also experimented with other machine learning algorithms such as KNN, XGBoost, and SVM, SVM required significantly more training time than RF, KNN, and XGBoost. Consequently, we excluded SVM from our considerations. All results of tested machine learning algorithms are provided in table 2. Overall, Random Forest is selected as one of the baseline methods, presented in section 4.4 in terms of F1 score.

Table 2. Classification accuracy (%) and F1 score (%) for different ML Algorithms

METHOD	Metric	Home1	Home2	Home3	Home4	Home5	Home6	Home7	Home8	Hospital1	Hospital2	office	Lab1	Lab2	Ave
RF	**Acc**	72.69	**93.28**	93.25	**77.96**	93.31	**96.59**	94.80	88.49	**64.35**	90.60	**86.03**	92.93	95.20	87.65
	F1	73.83	**93.49**	93.18	**79.54**	93.26	**96.57**	94.91	88.41	**65.64**	90.21	**86.76**	92.94	95.25	**88.00**
KNN	**Acc**	57.56	83.99	47.92	55.00	77.84	64.78	89.21	98.53	58.74	89.50	69.89	88.71	97.02	75.28
	F1	57.67	75.70	43.72	44.04	75.15	65.16	82.62	98.54	56.13	88.10	50.82	85.80	96.82	70.79
XGBoost	**Acc**	**87.31**	91.67	**95.10**	65.01	**98.39**	94.06	**99.83**	**99.01**	56.78	**92.17**	85.63	**97.87**	**96.30**	**89.16**
	F1	**81.45**	90.87	**95.43**	48.61	**98.24**	95.04	**99.69**	**99.01**	36.90	**91.33**	80.20	**97.27**	**96.72**	85.44

4.4 Comparison between Baseline Methods and SeismoDot

We established three baseline methods for comparison in our study: (1) [4], a seismic sensor-based system that leverages the fusion of spectral energy, kurtosis, and Teager energy operator. (2) [14] another seismic sensor-based system that directly applies ACF on raw data and counts the peak on ACF. (3) The Digital Signal Processing (DSP) method: leverage zero crossing rate as suggested in [26]. (4) Feature extraction + Random Forrest: this method involves feature extraction, followed by a Random Forest classifier. (5) **Vanilla:** This baseline method is regular supervised learning with only temporal data as input. It uses the same temporal architecture as SeismoDot. Our

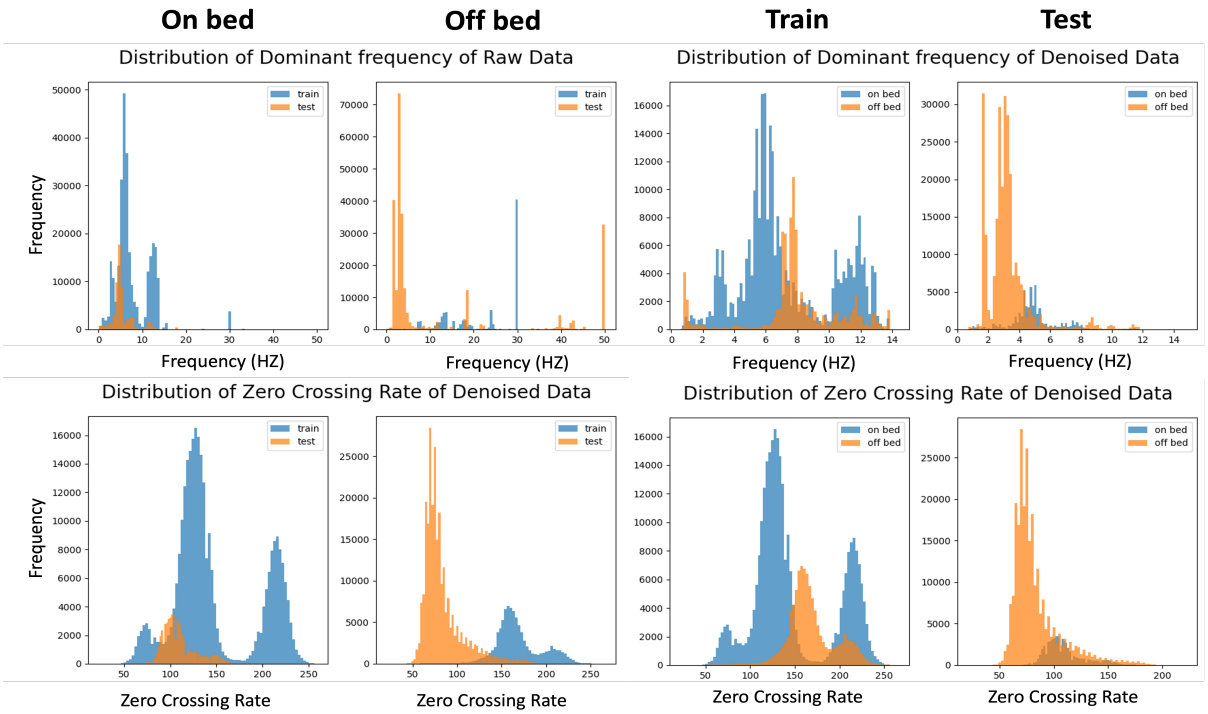


Fig. 9. Distribution of zero crossing rate and dominant frequency of raw data of both on-bed and off-bed signal on training and testing dataset

proposed method is used for comparison in this section with both temporal and spectral data as input, and it includes the feature fusion and self-supervised learning modules during the training stage.

We conducted tests in a total of 13 different environments. Initially, there were 12 distinct environments, and the one in 'controlled hospital environment' was divided into two separate environments due to two different beds, as explained in section 4.1.1. We employed a leave-one-out cross-validation approach to assess the model's performance for these 13 environments. We report both balanced accuracy and 'macro' F1 score for bed occupancy detection. The evaluation results are summarized in table 3.

As shown in table 3, our proposed method consistently outperforms all baseline methods across all 13 environments regarding F1 score and accuracy. Furthermore, our approach exhibits remarkable stability compared to the baseline methods, as it succeeds in all environments. In contrast, other baseline methods experience failures in one or more environments. For example, the method proposed by [26] failed in 'home1' and 'hospital2' environments, with mediocre performance in 'home2'. The 'Feature Extraction + Random Forest' method exhibits instability in 'home1' and 'hospital1'. Meanwhile, the 'Vanilla' method fails in 'home8'. In addition, our proposed method consistently attains an accuracy exceeding 92%. It approaches 90% for the F1 score even in challenging scenarios such as 'home4', where the device is attached to the side of the bed frame, and two people are sleeping on the bed.

4.5 Investigation of Generalizability Across Patients

We did a 5-fold cross-validation (20 people for each fold) on data collected from Phase I. Because we have collected data from each person on two beds (hospital bed and Foam bed), we employed 5-fold cross-validation on each

Table 3. Classification accuracy (%) and F1 score (%) comparison between baseline methods and proposed methods on 13 environments (best performance of each method for each environment is **bold**)

METHOD	Metric	Home1	Home2	Home3	Home4	Home5	Home6	Home7	Home8	Hospital1	Hospital2	Office	Lab1	Lab2	Ave
[4]	**Acc**	78.77	92.81	80.40	49.38	50.86	95.89	49.32	91.02	5.11	74.88	49.24	66.61	43.58	63.68
	F1	70.96	91.35	80.50	23.29	36.94	95.46	22.52	91.00	6.37	65.63	17.27	66.62	26.53	53.42
[14]	**Acc**	78.00	75.09	95.48	69.67	97.98	87.34	98.94	96.65	80.26	80.05	78.21	97.43	85.42	86.19
	F1	69.32	75.32	95.28	54.95	97.89	88.36	99.27	96.66	82.37	75.09	68.06	97.18	81.84	83.20
[26]	**Acc**	30.02	69.88	98.59	76.71	98.83	99.76	99.91	89.24	99.01	28.56	83.68	93.79	93.17	78.92
	F1	13.98	60.46	98.59	78.63	98.83	99.76	99.89	89.11	99.01	15.89	84.94	93.73	93.17	81.63
FEATURE EXTRACTION + RF	**Acc**	72.69	93.28	93.25	77.96	93.31	96.59	94.80	88.49	64.35	90.60	86.03	92.93	95.20	87.65
	F1	73.83	93.49	93.18	79.54	93.26	96.57	94.91	88.41	65.64	90.21	86.76	92.94	95.25	88.00
VANILLA	**Acc**	92.78	92.79	96.87	79.07	91.96	90.61	93.09	53.67	94.24	90.66	89.30	97.86	97.56	89.26
	F1	90.95	91.29	96.84	74.63	91.72	89.06	90.53	42.69	93.40	87.85	79.94	97.70	97.51	86.47
PROPOSED	**Acc**	**98.63**	**97.48**	**98.83**	**92.53**	**99.61**	**99.82**	**99.92**	**99.36**	**99.57**	**97.56**	**98.47**	**99.10**	**99.53**	**98.49**
	F1	**98.38**	**96.83**	**98.82**	**89.77**	**99.58**	**99.81**	**99.90**	**99.36**	**99.48**	**96.95**	**97.59**	**99.05**	**99.52**	**98.08**

bed. The results are shown in table 4. According to table 4, our proposed model shows its ability to generalize across different people.

Table 4. 5 Fold Cross Validation on Phase I on 2 Beds

Beds	Metric	Fold1	Fold2	Fold3	Fold4	Fold5	Ave
HOSPITAL BED	**Acc**	99.15	99.61	99.35	99.03	98.06	99.04
	F1	99.05	99.47	99.17	99.10	98.15	98.99
FOAM BED	**Acc**	98.72	99.49	99.88	99.71	99.46	99.45
	F1	98.77	99.58	99.90	99.71	99.53	99.50

4.6 Ablation Study of Proposed Method

We conduct quantitative analysis to explore the contribution of each module in SeismoDot. We compared the complete SeismoDot with five variants: VANILLA: This variant uses only temporal data as input. VANILLA + F: This variant employs both temporal and spectral data as input, with the same architecture as SeismoDot. However, it does not have a spectral-temporal feature fusion module and self-supervised learning module during the training stage. VANILLA + F + SSL: In this variant, the self-supervised learning module is included during the training stage, using both temporal and spectral data as input. VANILLA + F + FEATURE FUSION: This variant incorporates the spectral-temporal feature fusion module during the training stage, using both temporal and spectral data as input. VANILLA + F + FEATURE FUSION + SSL This variant includes all modules proposed in this paper.

The results are presented in table 5. When comparing VANILLA and VANILLA + F, it is evident that adding spectral features to the input data significantly improves bed occupancy detection performance in terms of both accuracy and F1 score. This improvement is particularly notable in the 'home8' environment, where accuracy increases by 42.39% and F1 score by 53.37%. On average, across all 13 environments, the addition of spectral features leads to a 5.9%

The inclusion of the self-supervised learning module (VANILLA + F + SSL) further enhances performance, resulting in an average improvement of 2.19% in accuracy and 2.55% in F1 score across all environments over VANILLA + F. This illustrates the positive impact of the self-supervised learning module on overall performance. Similarly, replacing self-supervised learning with the spectral and temporal feature fusion module (VANILLA + F + FEATURE FUSION) still leads to improvements in classification accuracy and F1 score, with an average increase of

Table 5. Ablation Study (best performance of each method for each environment is **bold**)

METHOD	Metric	Home1	Home2	Home3	Home4	Home5	Home6	Home7	Home8	Hospital1	Hospital2	Office	Lab1	Lab2	Ave
Vanilla	**Acc**	92.78	92.79	96.87	79.07	91.96	90.61	93.09	53.67	94.24	90.66	89.30	97.86	97.56	89.26
	F1	90.95	91.29	96.84	74.63	91.72	89.06	90.53	42.69	93.40	87.85	79.94	97.70	97.51	86.47
Vanilla + F	**Acc**	93.41	95.51	97.14	84.26	93.58	98.43	97.27	96.06	97.06	93.86	93.82	97.84	98.82	95.16
	F1	92.56	94.49	97.12	80.72	93.38	98.27	97.07	96.06	96.56	92.23	89.69	97.77	98.79	94.21
Vanilla + F + SSL	**Acc**	96.72	97.21	97.56	90.98	97.25	99.43	99.59	96.58	**99.71**	96.59	95.68	98.91	99.32	97.35
	F1	96.19	96.49	97.54	88.22	97.04	99.38	99.48	96.58	**99.65**	95.86	93.25	98.85	99.31	96.76
Vanilla + F + feature fusion	**Acc**	97.56	97.33	98.40	90.74	98.51	99.60	99.59	98.16	99.49	96.03	95.15	98.94	99.26	97.66
	F1	97.13	96.63	98.39	87.97	98.42	99.56	99.48	98.16	99.39	95.22	92.43	98.88	99.24	97.12
Vanilla + F + feature fusion + SSL	**Acc**	**98.63**	**97.48**	**98.83**	**92.53**	**99.61**	**99.82**	**99.92**	**99.36**	99.57	**97.56**	**98.47**	**99.10**	**99.53**	**98.49**
	F1	**98.38**	**96.83**	**98.82**	**89.77**	**99.58**	**99.81**	**99.90**	**99.36**	99.48	**96.95**	**97.59**	**99.05**	**99.52**	**98.08**

2.5% and 2.91%, respectively, over Vanilla + F. This highlights the effectiveness of both modules in bed occupancy detection.

When all modules are combined in SeismoDot, it achieves the best overall performance as illustrated in table 5, outperforming the other variants in terms of accuracy and F1 score for nearly all environments, except 'hospital1'. Even for 'hospital1', the performance of SeismoDot is very close to the best, with just a 0.14% difference in accuracy and a 0.17% difference in F1 score. The proposed method (Vanilla + F + feature fusion + SSL) demonstrates significant improvement compared to the baseline (Vanilla), with a 9.23% increase in accuracy and 11.6% increase in F1 score.

4.7 Investigation of Model Performance in Different Availability of Training Data

To evaluate the robustness of our proposed method in scenarios with limited training data, we conducted experiments to compare it with other baseline methods. We varied the amount of training data, using percentages of 100%, 80%, 60%, 40%, and 20%, in leave-one-environment-out cross-validation. The total duration of collected data for training is around 543 hours (approximately 23 days). As we did cross-validation for each environment, the average training data duration for each validation in the 20% case is 100.25 hours (approximately four days). It is worth noting that we excluded the method proposed by [26] from this comparison, as it is not a data-driven method. Importantly, we ensured that the same training data was consistently used for evaluation for all methods under the same percentage of data cases. The summarized results can be found in table 6.

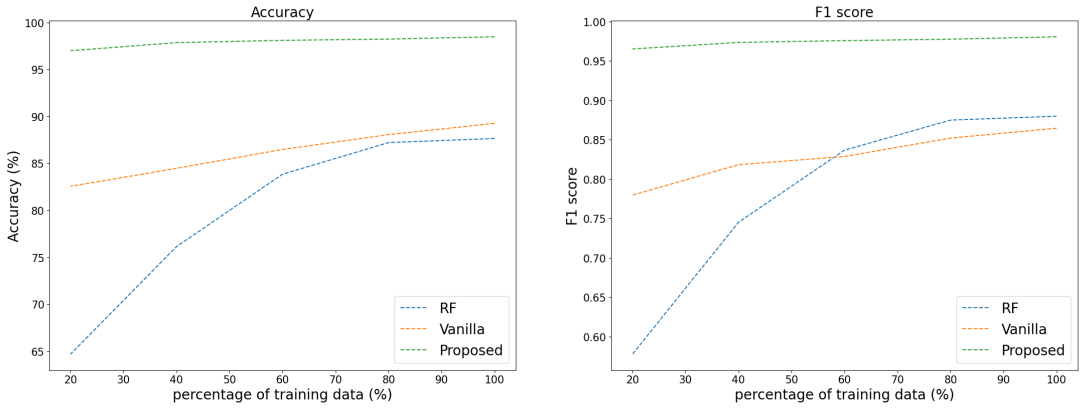


Fig. 10. Results of baseline methods and proposed method with different percentages of training data.

Table 6. Classification accuracy (%) and F1 score (%) comparison between baseline methods and Proposed Methods on 13 environments with different percentages of data (best performance of each method for each environment is **bold**)

Percentage	METHOD	Metric	Home1	Home2	Home3	Home4	Home5	Home6	Home7	Home8	Hospital1	Hospital2	Office	Lab1	Lab2	Ave
100	FEATURE EXTRACTION + RF	**Acc**	72.69	93.28	93.25	77.96	93.31	96.59	94.80	88.49	64.35	90.60	86.03	95.20	92.93	87.65
		F1	73.83	93.49	93.18	79.54	93.26	96.57	94.91	88.41	65.64	90.21	86.76	95.25	92.94	88.00
	VANILLA	**Acc**	92.78	92.79	96.87	79.07	91.96	90.61	93.09	53.67	94.24	90.66	89.30	97.86	97.56	89.26
		F1	90.95	91.29	96.84	74.63	91.72	89.06	90.53	42.69	93.40	87.85	79.94	97.70	97.51	86.47
	PROPOSED	**Acc**	**98.63**	**97.48**	**98.83**	**92.53**	**99.61**	**99.82**	**99.92**	**99.36**	**99.57**	**97.56**	**98.47**	**99.10**	**99.53**	**98.49**
		F1	**98.38**	**96.83**	**98.82**	**89.77**	**99.58**	**99.81**	**99.90**	**99.36**	**99.48**	**96.95**	**97.59**	**99.05**	**99.52**	**98.08**
80	FEATURE EXTRACTION + RF	**Acc**	70.81	94.83	93.55	80.21	93.91	95.34	95.37	81.05	68.07	90.03	85.86	94.67	90.11	87.22
		F1	72.02	94.95	93.48	81.45	93.86	95.30	95.45	80.67	69.21	89.67	86.60	94.72	90.16	87.50
	VANILLA	**Acc**	93.45	91.81	96.13	76.17	87.15	90.15	91.57	51.56	93.29	89.64	89.23	97.63	97.05	88.06
		F1	91.85	90.24	96.05	73.63	87.01	88.45	87.59	41.49	92.36	85.73	79.03	97.45	97.00	85.22
	PROPOSED	**Acc**	**98.40**	**97.39**	**98.69**	**91.88**	**99.21**	**99.75**	**99.84**	**99.13**	**99.55**	**97.19**	**97.74**	**99.05**	**99.35**	**98.24**
		F1	**98.10**	**96.72**	**98.68**	**88.99**	**99.17**	**99.73**	**99.79**	**99.13**	**99.46**	**96.58**	**96.43**	**98.99**	**99.33**	**97.78**
60	FEATURE EXTRACTION + RF	**Acc**	78.90	94.81	90.38	81.98	96.61	96.24	87.24	84.41	67.81	90.96	88.89	51.86	79.86	83.84
		F1	79.77	94.93	90.21	82.80	96.60	96.21	87.83	84.10	69.17	90.74	89.49	46.23	79.83	83.69
	VANILLA	**Acc**	88.22	90.78	95.18	72.92	86.84	86.69	91.22	51.51	92.06	87.28	87.48	97.32	96.76	86.48
		F1	84.28	89.07	95.08	70.55	86.70	83.91	87.58	38.97	91.01	81.84	74.69	97.12	96.71	82.89
	PROPOSED	**Acc**	**98.12**	**97.32**	**98.62**	**91.31**	**99.13**	**99.74**	**99.76**	**99.05**	**99.46**	**96.99**	**97.62**	**99.01**	**99.31**	**98.11**
		F1	**97.78**	**96.61**	**98.61**	**87.92**	**99.07**	**99.72**	**99.69**	**99.05**	**99.36**	**96.31**	**96.33**	**98.95**	**99.30**	**97.59**
40	FEATURE EXTRACTION + RF	**Acc**	72.02	95.24	85.47	77.20	85.76	96.45	58.86	66.58	60.29	90.44	89.62	37.69	74.09	76.13
		F1	68.90	95.34	84.91	78.85	85.66	96.43	59.88	62.56	61.75	90.45	89.81	20.63	73.68	74.53
	VANILLA	**Acc**	84.71	88.37	90.98	70.25	84.22	84.86	89.43	49.61	90.88	86.08	85.44	96.87	96.43	84.47
		F1	83.81	86.63	90.73	68.20	84.12	81.35	84.90	38.38	89.80	79.87	83.14	96.62	96.37	81.84
	PROPOSED	**Acc**	**97.86**	**97.25**	**98.40**	**90.98**	**98.86**	**99.72**	**99.76**	**97.86**	**99.15**	**96.88**	**97.21**	**98.99**	**99.31**	**97.86**
		F1	**97.78**	**96.61**	**98.61**	**87.92**	**99.07**	**99.72**	**99.69**	**99.05**	**99.36**	**96.31**	**96.33**	**98.95**	**99.30**	**97.59**
20	FEATURE EXTRACTION + RF	**Acc**	70.98	79.76	55.19	81.61	81.32	93.93	33.74	49.77	57.93	28.00	84.09	37.69	86.62	64.66
		F1	59.85	75.97	39.26	82.76	80.50	93.86	25.26	33.08	59.46	12.25	81.52	20.63	86.65	57.77
	VANILLA	**Acc**	81.00	85.97	90.48	66.57	82.08	81.71	86.67	49.15	87.87	83.14	87.80	96.02	94.80	82.56
		F1	71.33	84.10	90.47	65.04	82.02	76.55	79.72	37.31	86.77	74.32	75.63	95.67	94.74	77.97
	PROPOSED	**Acc**	**97.08**	**96.09**	**97.72**	**90.51**	**98.16**	**99.14**	**99.67**	**94.95**	**97.19**	**95.92**	**96.75**	**98.93**	**98.99**	**97.01**
		F1	**96.53**	**96.07**	**97.72**	**87.35**	**98.07**	**99.14**	**99.59**	**94.95**	**97.16**	**95.87**	**94.63**	**98.92**	**98.99**	**96.54**

Our results consistently demonstrate that SeismoDot outperforms all other baseline methods in terms of both F1 score and accuracy across various levels of data availability. As expected, as the amount of training data decreases, all methods experience a decline in accuracy overall. This trend is reflected in the average F1 score and accuracy across all 13 environments. However, SeismoDot exhibits remarkable resilience to diminishing training data. Even when provided with only 20% of the data, our proposed method experiences only a marginal reduction of 1.54% in F1 score and 1.48% in accuracy.

In contrast, the other baseline methods exhibit more significant performance drops in low data availability scenarios. For instance, RF experiences a substantial decrease of 24.36% in accuracy and 31.63% in F1 score under similar conditions, while VANILLA shows a reduction of 6.70% in accuracy and 8.50% in F1 score. These findings highlight the effectiveness of our approach in handling limited data availability.

Moreover, as illustrated in figure 10, our method consistently outperforms RF and VANILLA across all data availability scenarios, maintaining high accuracy and F1 score with minimal degradation. In contrast, RF and VANILLA exhibit substantial performance drops, particularly in scenarios with limited training data. It is worth noting that the performance of our proposed method with only 20% data availability even surpasses that of other baseline methods with 100% data availability.

In summary, these observations emphasize the effectiveness and robustness of our approach when addressing scenarios with limited data availability in bed occupancy detection. This capability can potentially reduce the requirement for extensive data collection in bed occupancy detection applications.

4.8 Robustness of Models to Sensor Position

In this section, we aim to investigate how the position of sensors may impact the performance of our proposed method and other baseline methods. Additionally, we explore how different methods perform in this scenario when the amount of available data is reduced. We evaluate three scenarios using a total of seven methods,

including the method proposed by [26], FEATURE EXTRACTION + RF under 100% and 20% data availability, VANILLA under 100% and 20% data availability and PROPOSED under 100% and 20% data availability.

'home4' is being re-used as the sensor is attached on the right side of the bed instead of under the bed. 'lab3' scenario used the same bed as 'lab1' but with a sensor attached on the left side of the bed as discussed in section 4.1. To establish a direct comparison with 'lab3,' the 'lab1' scenario is re-evaluated since they share the same bed. This allows us to easily assess how sensor position affects model performance.

During this evaluation, we exclude the 'lab1' and 'lab2' scenarios from the training dataset, as 'lab1,' 'lab2,' and 'lab3' all take place in similar environments, although 'lab2' uses a different bed from 'lab1' and 'lab3.' Furthermore, 'home4' is excluded from the training dataset, as it serves as the target dataset for evaluation. Therefore, we utilize data from all environments except 'lab1,' 'lab2,' 'lab3,' and 'home4' for training to ensure a fair comparison.

The results are presented in table 7. Notably, the proposed method with 100% training data (PROPOSED (100%)) achieves the best performance across all environments, regardless of different sensor positions. PROPOSED (100%) attains an accuracy of 96.97% and an F1 score of 96.96% on 'lab3,' as well as 86.72% accuracy and an 87.11% F1 score on 'home4,' indicating the robustness of our model to variations in sensor position. The VANILLA (100%) approach also performs well on 'lab3' and delivers an acceptable result on 'home4.' The results from [26] achieve similar performance to VANILLA (100%). FEATURE EXTRACTION + RF (100%) demonstrates acceptable performance on both 'lab3' and 'home4' but not as good as VANILLA (100%) and [26] on 'lab3'.

A direct comparison between 'lab1' and 'lab3' shown in figure 11 reveals that PROPOSED (100%) experiences a minor drop of 1.43% in accuracy and 1.33% in F1 score under different sensor position. In contrast, VANILLA (100%) experiences a 5.15% drop in accuracy and a 5% drop in F1 score, while FEATURE EXTRACTION + RF (100%) shows a significant drop of 13.87% in accuracy and 13.89% in F1 score. [26] yields different results than other methods, performing better on 'lab3' than 'lab1' because it is a non-data-driven method, which does not bias toward sensor position in the training dataset. In summary, PROPOSED (100%) consistently outperforms all other baseline methods under varying sensor positions. Sensor position does not significantly affect PROPOSED (100%) when compared to other baseline methods.

Table 7. Classification accuracy (%) and F1 score (%) comparison between baseline methods and Proposed Methods for different sensor positions

METHOD	Metric	Lab1	Lab3	Home4
[26]	**Acc**	89.24	93.52	76.71
	F1	89.11	93.53	78.63
FEATURE EXTRACTION + RF (20%)	**Acc**	62.31	73.87	60.74
	F1	47.84	73.12	63.22
FEATURE EXTRACTION + RF (100%)	**Acc**	96.22	82.35	77.77
	F1	96.19	82.21	79.41
VANILLA (20%)	**Acc**	96.30	88.27	66.86
	F1	95.99	88.27	65.04
VANILLA (100%)	**Acc**	97.63	92.48	78.81
	F1	97.46	92.46	75.23
PROPOSED (20%)	**Acc**	98.07	95.07	85.61
	F1	97.94	95.07	86.04
PROPOSED (100%)	**Acc**	**98.40**	**96.97**	**86.72**
	F1	**98.29**	**96.96**	**87.11**

The performance of the proposed method with 20% training data (PROPOSED (20%)) closely resembles that of PROPOSED (100%) with only minor differences. PROPOSED (20%) exhibits a 1.9% difference in accuracy and a

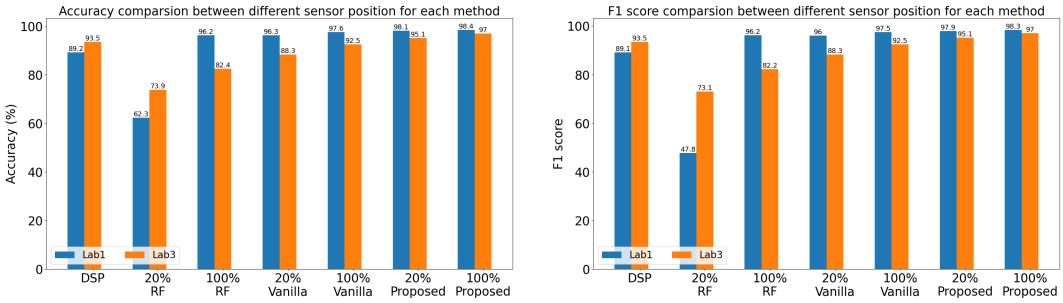


Fig. 11. Comparison between lab1 and lab3 to see how sensor position affects performance for each method (Digital signal processing (DSP) refers to the method in [26]).

1.89% difference in F1 score on 'lab3,' and a 1.11% difference in accuracy and a 1.07% difference in F1 score on 'home4.' Overall, Proposed (20%) achieves the second-best method across all comparisons. However, Vanilla (20%) shows a 4.21% drop in accuracy and a 4.19% drop in F1 score compared to Vanilla (100%) on 'lab3,' and a significant 11.95% drop in accuracy and a 10.19% drop in F1 score on 'home4.' feature extraction + RF (100%) exhibits a substantial 13.87% drop in accuracy and a 13.89% drop in F1 score on 'lab3,' and an even greater 17.03% drop in accuracy and 16.19% drop in F1 score on 'home4.' This highlights the robustness of our proposed method under low data availability conditions, in contrast to the significant performance drops observed in other baseline methods in 20% training data availability. Furthermore, Proposed (20%) exhibits only a 3% performance drop in 'lab3' compared to 'lab1', suggesting that the proposed algorithm remains robust to sensor position even under conditions of low data availability. This analysis suggests the adaptability and effectiveness of the proposed method across different sensor positions and varying data availability scenarios.

4.9 Spectral-Temporal Feature Fusion vs Spectral-Temporal Feature Alignment

This section compares proposed spectral-temporal feature fusion with spectral-temporal feature alignment. We conducted two comparison experiments: for the first experiment, we performed spectral-temporal alignment on Vanilla + F denote as Vanilla + F + Alignment, to compare with Vanilla + F + feature fusion. For the second experiment, we delved into the compatibility of feature alignment with self-supervised representation learning (Vanilla + F + Alignment + SSL). This investigation aimed to assess how feature alignment performs when integrated with self-supervised learning, performs compared to the proposed Vanilla + F + Feature Fusion + SSL. This comparison helps us understand the impact of each approach in the context of self-supervised representation learning.

The comparison between Vanilla + F + Feature Fusion and Vanilla + F + Alignment is presented in table 8. It is observed that Vanilla + F + Feature Fusion outperforms Vanilla + F + Alignment in 10 out of 12 environments. On average, there is an improvement of 1.01% in accuracy and 1.08% in F1 score across all 13 environments. These findings underscore the efficacy of the feature fusion approach, suggesting its superiority over feature alignment in enhancing model performance across diverse environments.

Table 8. Feature Fusion vs Feature Alignment

METHOD	Metric	Home1	Home2	Home3	Home4	Home5	Home6	Home7	Home8	Hospital1	Hospital2	Office	Lab1	Lab2	Ave
Vanilla + F + Alignment	**Acc**	95.79	96.30	97.96	86.04	98.35	99.40	**99.68**	**98.61**	96.32	95.25	98.85	98.86	95.01	96.65
	F1	95.58	95.30	97.65	83.90	98.29	99.05	**99.59**	**98.58**	95.24	93.01	98.50	98.83	95.00	96.04
Vanilla + F + feature fusion	**Acc**	**97.56**	**97.33**	**98.40**	**90.74**	**98.51**	**99.60**	99.59	98.16	**99.49**	**96.03**	**95.15**	**98.94**	**99.26**	**97.66**
	F1	**97.13**	**96.63**	**98.39**	**87.97**	**98.42**	**99.56**	99.48	98.16	**99.39**	**95.22**	**92.43**	**98.88**	**99.24**	**97.12**

The comparison between Vanilla + F + Feature Fusion + SSL and Vanilla + F + Alignment + SSL is presented in table 9. The results reveal that Vanilla + F + Feature Fusion consistently outperforms Vanilla + F + Alignment in all 13 environments, demonstrating an improvement of 1.21% in both accuracy and F1 score. This suggests that the proposed feature fusion approach is more effective when combined with self-supervised representation learning (SSL) than feature alignment combined with SSL. Nevertheless, it is essential to note that Vanilla + F + Alignment + SSL still demonstrates strong compatibility with SSL.

Table 9. Feature Fusion vs Feature Alignment after adding SSL

METHOD	Metric	Home1	Home2	Home3	Home4	Home5	Home6	Home7	Home8	Hospital1	Hospital2	Office	Lab1	Lab2	Ave
Vanilla + F + Alignment + SSL	**Acc**	97.13	97.24	98.20	84.69	99.15	99.41	99.57	99.43	97.39	96.02	99.01	99.07	98.35	97.28
	F1	96.87	96.13	98.14	82.85	99.17	99.46	99.38	99.42	95.84	94.43	98.92	99.08	98.35	96.77
Vanilla + F + feature fusion + SSL	**Acc**	**98.63**	**97.48**	**98.83**	**92.53**	**99.61**	**99.82**	**99.92**	**99.57**	**97.51**	**98.47**	**99.10**	**99.53**	**99.36**	**98.49**
	F1	**98.38**	**96.83**	**98.82**	**89.77**	**99.58**	**99.81**	**99.90**	**99.48**	**96.95**	**97.59**	**99.05**	**99.52**	**99.36**	**98.08**

4.10 Comparison with State-of-the-Art SSL Model

We conducted a comparative analysis between our proposed method and TF-C [36], which also incorporates both time and spectral domains and is suitable to our case. Our proposed method demonstrates superior performance over TF-C across 11 environments in terms of accuracy and 12 environments in terms of F1 scores. On average, our method exhibits a 2.54% and a 2.62% improvement in both accuracy and F1 scores.

Table 10. Classification accuracy (%) and F1 score (%) comparison between TF-C and proposed methods on 13 environments

METHOD	Metric	Home1	Home2	Home3	Home4	Home5	Home6	Home7	Home8	Hospital1	Hospital2	office	Lab1	Lab2	Ave
TF-C	**Acc**	92.63	93.79	**98.90**	90.29	96.87	97.99	98.38	96.69	94.03	**97.74**	96.50	95.93	97.55	95.95
	F1	93.16	92.53	**98.95**	88.67	96.73	98.11	98.68	95.86	92.05	96.33	96.82	95.48	97.55	95.46
Proposed	**Acc**	**98.63**	**97.48**	98.83	**92.53**	**99.61**	**99.82**	**99.92**	**99.36**	**99.57**	97.56	**98.47**	**99.10**	**99.53**	**98.49**
	F1	**98.38**	**96.83**	98.82	**89.77**	**99.58**	**99.81**	**99.90**	**99.36**	**99.48**	**96.95**	**97.59**	**99.05**	**99.52**	**98.08**

5 DISCUSSION AND FUTURE WORK

5.0.1 Testing on more complicated environments. : It is important to acknowledge that anticipated external factors may occasionally result in false detections. However, these factors can only be fully understood and addressed through more testing. For instance, during a test, a participant dropped his phone on the bed while it was playing music, which led to approximately 30 consecutive seconds of false on-bed detection by our system. Because the signal generated from music was a periodic signal, similar to a heartbeat signal in the time domain. The system was using a time-domain signal only at that time. This led us to enhance our system by incorporating both temporal and spectral domain features, as the signal generated from that music on the spectral domain differed from the heartbeat signal. Although such occurrences are rare, they underscore the importance of thorough testing to uncover and address unforeseen external factors that may compromise system performance in real-world scenarios.

5.0.2 Potential to transfer from bed to seat scenarios. : Testing in-seat scenarios is more complicated than in-bed scenarios. Seat scenarios such as smart office typically include more activity events and more sources of interference, including conversations, computer typing, and handling objects. Implementing smart seats in office environments could facilitate monitoring daily activities to detect stress levels, offering a potentially critical application in the future.

We did some preliminary tests on seat scenarios. We employ well-trained models from section 4.8 without additional training, as these models are trained explicitly on 'sensor under the bed' data. This approach allows us

to directly observe the performance of models when transferring from the 'sensor under the bed' scenario to seat scenarios.

Data is collected from three environments, each using a different seat for data collection with devices attached underneath. 'seat1' is in the same environment as 'lab1,' 'lab2,' and 'lab3'. 'seat2' takes place in another lab environment with two participants engaged in a casual conversation. This aims to test the model's ability to detect seat occupancy with continuous talking interference. 'seat3' is conducted in a home environment with a participant working using a computer and phones, resembling a daily office scenario. The seats from the three scenarios are made of different materials: ' seat1' is a polyurethane medium bench, ' seat2' is an office seat with cushion, and 'seat3' is a game chair. According to table 11, our proposed method consistently achieves high performance across all three seat scenarios, showcasing the remarkable potential for seamless transfer from bed to seat.

Table 11. Classification accuracy (%) and F1 score (%) of Proposed Methods for seat scenarios

METHOD	Metric	Seat1	Seat2	Seat3	Ave
Proposed	**Acc**	97.36	99.65	86.76	94.59
	F1	95.91	99.74	86.87	94.17

5.0.3 Potential for other physiological signals. : Our proposed method pipeline could be applied directly to other physiological signals. For example, we tested our pipeline of the proposed method on a dataset for VAs (i.e., Ventricular Fibrillation and Ventricular Tachycardia) detection from single-lead (RVA-Bi) IEGM recordings. With the feature fusion module, we can improve the performance from 93.82% and 95.98% to 97.26% and 98.70% in terms of both accuracy and F-beta score. In addition, we discovered that if we replace STFT (short-time Fourier transform) with FFT to obtain spectral features, the performance will be further improved in most cases. However, there is a trade-off between computational cost&memory cost and performance improvement by replacing STFT with FFT. Computational cost is during the data pre-processing stage, and memory cost is during the training stage. Many research works use 2D-CNN instead of 1D CNN on STFT, which might significantly increase the computational cost. Our suggestion is using 1D-CNN instead of 2D-CNN would be good enough.

5.0.4 Automatically Adapt to New Environment. : (1) It is better to adapt the model to a new environment by leveraging unlabelled data from new environments or by incorporating labeled data provided by users. (2) As the volume of labeled data grows, it becomes essential to find methods that prevent the need for training the model from scratch each time new data arrives. Furthermore, it is crucial to address the challenge of preventing the model from overfitting to the new data and potentially "forgetting" the knowledge acquired from the old data.

5.0.5 Collecting more data for two people scenarios. : Among the 13 different environments tested, one scenario involved two people, where our system showed promising results. However, to confidently affirm that our method works effectively in two-person scenarios, we need to collect and test more data specific to such situations.

5.0.6 Expanding Capabilities of SeismoDot. : The bed occupancy detection of SeismoDot serves as the foundation for subsequent tasks such as vital signs monitoring, sleep apnea detection, and sleep posture classification. Additional functions can be developed and integrated to enhance the system. Our system already includes vital signs monitoring, such as heart rate estimation and respiration rate estimation. And our next step is to incorporate sleep posture classification into the existing framework.

6 CONCLUSION

This paper presents SeismoDot, a user-friendly, cost-effective system designed for continuous bed occupancy monitoring. SeismoDot is a foundation for various subsequent tasks, including monitoring vital signs, sleep

apnea, and classification of sleep posture. The system leverages both temporal and spectral features and expands the diversity of data space through self-supervised learning in both domains. In addition, to enable meaningful spectral-temporal feature fusion, SeismoDot maximizes diversity between the temporal and spectral domains to facilitate rich information for both domains while implementing an overlap constraint. This constraint prevents the temporal and spectral spaces from being pushed too far apart, ensuring they keep shared features. SeismoDot has been comprehensively evaluated across diverse environments, achieving an accuracy rate of 98.49% and an F1 score of 98.08% in bed occupancy detection across 13 different scenarios. These results underscore the adaptability and robust performance of the proposed algorithm across various environmental conditions. Even when trained with just 20% of the available training data (4 days), SeismoDot maintains comparable performance with models trained on the complete dataset. Furthermore, the experiments reveal that the proposed algorithm is resilient to changes in sensor position, is promising for two-person-in-bed scenarios, and can seamlessly transition from bed to seat scenarios. SeismoDot exhibits promising potential for commercial applications in hospital settings, smart homes, and smart offices. Moreover, the system's flexibility enables the seamless integration of new functions.

REFERENCES

[1] Andreas Braun, Martin Majewski, Reiner Wichert, and Arjan Kuijper. 2016. Investigating low-cost wireless occupancy sensors for beds. In *Distributed, Ambient and Pervasive Interactions: 4th International Conference, DAPI 2016, Held as Part of HCI International 2016, Toronto, ON, Canada, July 17-22, 2016, Proceedings 4*. Springer, 26–34.

[2] Andreas Braun, Reiner Wichert, Arjan Kuijper, and Dieter W Fellner. 2015. Capacitive proximity sensing in smart environments. *Journal of Ambient Intelligence and Smart Environments* 7, 4 (2015), 483–510.

[3] Lili Chen, Jie Xiong, Xiaojiang Chen, Sunghoon Ivan Lee, Daqing Zhang, Tao Yan, and Dingyi Fang. 2019. LungTrack: Towards Contactless and Zero Dead-Zone Respiration Monitoring with Commodity RFIDs. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 79 (sep 2019), 22 pages. https://doi.org/10.1145/3351237

[4] Jose Clemente, Maria Valero, Fangyu Li, Chengliang Wang, and WenZhan Song. 2020. Helena: Real-time contact-free monitoring of sleep activities and events around the bed. In *2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 1–10.

[5] Shohreh Deldari, Hao Xue, Aaqib Saeed, Daniel V Smith, and Flora D Salim. 2022. Cocoa: Cross modality contrastive learning for sensor data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (2022), 1–28.

[6] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, Xiaoli Li, and Cuntai Guan. 2023. Self-supervised contrastive representation learning for semi-supervised time-series classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[7] Haoyi Fan, Fengbin Zhang, and Yue Gao. 2020. Self-supervised time series representation learning by inter-intra relational reasoning. *arXiv preprint arXiv:2011.13548* (2020).

[8] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. 2006. A kernel method for the two-sample-problem. *Advances in neural information processing systems* 19 (2006).

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[10] Yash Jain, Chi Ian Tang, Chulhong Min, Fahim Kawsar, and Akhil Mathur. 2022. Collossl: Collaborative self-supervised learning for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 1 (2022), 1–28.

[11] Zhenhua Jia, Musaab Alaziz, Xiang Chi, Richard E Howard, Yanyong Zhang, Pei Zhang, Wade Trappe, Anand Sivasubramaniam, and Ning An. 2016. HB-phone: a bed-mounted geophone-based heartbeat monitoring system. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 1–12.

[12] Zhenhua Jia, Amelie Bonde, Sugang Li, Chenren Xu, Jingxian Wang, Yanyong Zhang, Richard E Howard, and Pei Zhang. 2017. Monitoring a person's heart rate and respiratory rate on a shared bed using geophones. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. 1–14.

[13] Dani Kiyasseh, Tingting Zhu, and David A Clifton. 2021. Clocs: Contrastive learning of cardiac signals across space, time, and patients. In *International Conference on Machine Learning*. PMLR, 5606–5615.

[14] Fangyu Li, Maria Valero, Jose Clemente, Zion Tse, and Wenzhan Song. 2020. Smart sleep monitoring system via passively sensing human vibration signals. *IEEE Sensors Journal* 21, 13 (2020), 14466–14473.

[15] Chen Liu, Jie Xiong, Lin Cai, Lin Feng, Xiaojiang Chen, and Dingyi Fang. 2019. Beyond Respiration: Contactless Sleep Sound-Activity Recognition Using RF Signals. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 96 (sep 2019), 22 pages. https://doi.org/10.1145/3351254

[16] Yunyoung Nam, Yeesock Kim, and Jinseok Lee. 2016. Sleep monitoring based on a tri-axial accelerometer and a pressure sensor. *Sensors* 16, 5 (2016), 750.

[17] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[18] Madhurananda Pahar, Igor Miranda, Andreas Diacon, and Thomas Niesler. 2023. Accelerometer-based bed occupancy detection for automatic, non-invasive long-term cough monitoring. *IEEE Access* 11 (2023), 30739–30752.

[19] Jaeyeon Park, Hyeon Cho, Rajesh Krishna Balan, and JeongGil Ko. 2020. Heartquake: Accurate low-cost non-invasive ecg monitoring using bed-mounted geophones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 3 (2020), 1–28.

[20] Melanie Pouliot, Vilas Joshi, Rafik Goubran, and Frank Knoefel. 2012. Bed occupancy monitoring: Data processing and clinician user interface design. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 5810–5814.

[21] Xin Qin, Jindong Wang, Shuo Ma, Wang Lu, Yongchun Zhu, Xing Xie, and Yiqiang Chen. 2023. Generalizable Low-Resource Activity Recognition with Diverse and Discriminative Representation Learning. *arXiv preprint arXiv:2306.04641* (2023).

[22] Racotech. 2024. *RGI-4.5Hz Geophone*. http://www.racotech.biz/parameter/RGI-4.5Hz%20Geophone.pdf

[23] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. 2019. Multi-task self-supervised learning for human activity detection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 2 (2019), 1–30.

[24] Pritam Sarkar and Ali Etemad. 2020. Self-supervised ECG representation learning for emotion recognition. *IEEE Transactions on Affective Computing* 13, 3 (2020), 1541–1554.

[25] Narayan Schütz, Hugo Saner, Angela Botros, Bruno Pais, Valérie Santschi, Philipp Buluschek, Daniel Gatica-Perez, Prabitha Urwyler, René M Müri, Tobias Nef, et al. 2021. Contactless sleep monitoring for early detection of health deteriorations in community-dwelling older adults: Exploratory study. *JMIR mHealth and uHealth* 9, 6 (2021), e24666.

[26] Yingjian Song, Bingnan Li, Dan Luo, Zaipeng Xie, Bradley G. Phillips, Yuan Ke, and Wenzhan Song. 2024. Engagement-Free and Contactless Bed Occupancy and Vital Signs Monitoring. *IEEE Internet of Things Journal* 11, 5 (2024), 7935–7947. https://doi.org/10.1109/JIOT.2023.3316674

[27] Matthew Taylor, Theresa Grant, Frank Knoefel, and Rafik Goubran. 2013. Bed occupancy measurements using under mattress pressure sensors for long term monitoring of community-dwelling older adults. In *2013 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. IEEE, 130–134.

[28] Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. 2021. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750* (2021).

[29] Maria Valero, Jose Clemente, Fangyu Li, and WenZhan Song. 2021. Health and sleep nursing assistant for real-time, contactless, and non-invasive monitoring. *Pervasive and mobile computing* 75 (2021), 101422.

[30] Tianben Wang, Daqing Zhang, Yuanqing Zheng, Tao Gu, Xingshe Zhou, and Bernadette Dorizzi. 2018. C-FMCW Based Contactless Respiration Detection Using Acoustic Signal. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 170 (jan 2018), 20 pages. https://doi.org/10.1145/3161188

[31] Ling Yang and Shenda Hong. 2022. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *International Conference on Machine Learning*. PMLR, 25038–25054.

[32] Shichao Yue, Yuzhe Yang, Hao Wang, Hariharan Rahul, and Dina Katabi. 2020. BodyCompass: Monitoring Sleep Posture with Wireless Signals. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 66 (jun 2020), 25 pages. https://doi.org/10.1145/3397311

[33] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. 2022. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8980–8987.

[34] Camellia Zakaria, Gizem Yilmaz, Priyanka Mary Mammen, Michael Chee, Prashant Shenoy, and Rajesh Balan. 2023. SleepMore: Inferring Sleep Duration at Scale via Multi-Device WiFi Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4, Article 193 (jan 2023), 32 pages. https://doi.org/10.1145/3569489

[35] Kexin Zhang, Qingsong Wen, Chaoli Zhang, Liang Sun, and Yong Liu. 2022. Time Series Anomaly Detection using Skip-Step Contrastive Predictive Coding. In *NeurIPS 2022 Workshop: Self-Supervised Learning-Theory and Practice*.

[36] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. 2022. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems* 35 (2022), 3988–4003.

[37] Yunhao Zhang and Junchi Yan. 2022. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*.

[38] Langcheng Zhao, Rui Lyu, Qi Lin, Anfu Zhou, Huanhuan Zhang, Huadong Ma, Jingjia Wang, Chunli Shao, and Yida Tang. 2024. mmArrhythmia: Contactless Arrhythmia Detection via mmWave Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 1, Article 30 (mar 2024), 25 pages. https://doi.org/10.1145/3643549

[39] Yang Zhao, Peter Tu, and Ming-Ching Chang. 2019. Occupancy sensing and activity recognition with cameras and wireless sensors. In *Proceedings of the 2nd Workshop on Data Acquisition to Analysis*. 1–6.

[40] Hao Zhou, Taiting Lu, Yilin Liu, Shijia Zhang, and Mahanth Gowda. 2022. Learning on the Rings: Self-Supervised 3D Finger Motion Tracking Using Wearable Sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022), 1–31.