Systematic Evaluation of Self-Supervised Learning Approaches for Wearable-Based Fatigue Recognition

Tamás Visy[1] TAVISY@STUDENT.ETHZ.CH
Rita Kuznetsova[1] MKUZNETSOVA@ETHZ.CH
Christian Holz[1] CHRISTIAN.HOLZ@INF.ETHZ.CH
Shkurta Gashi[1,2] SHKURTA.GASHI@AI.ETHZ.CH
[1] *Department of Computer Science, ETH Zürich, Switzerland.*
[2] *ETH AI Center, ETH Zürich, Switzerland.*

Abstract

Fatigue is one of the most prevalent symptoms of chronic diseases, such as Multiple Sclerosis, Alzheimer's, and Parkinson's. Recently researchers have explored unobtrusive and continuous ways of fatigue monitoring using mobile and wearable devices. However, data quality and limited labeled data availability in the wearable health domain pose significant challenges to progress in the field. In this work, we perform a systematic evaluation of self-supervised learning (SSL) tasks for fatigue recognition using wearable sensor data. To establish our benchmark, we use Homekit2020, which is a large-scale dataset collected using Fitbit devices in everyday life settings. Our results show that the majority of the SSL tasks outperform fully supervised baselines for fatigue recognition, even in limited labeled data scenarios. In particular, the domain features and multi-task learning achieve 0.7371 and 0.7323 AUROC, which are higher than the other SSL tasks and supervised learning baselines. In most of the pre-training tasks, the performance is higher when using at least one data augmentation that reflects the potentially low quality of wearable data (e.g., missing data). Our findings open up promising opportunities for continuous assessment of fatigue in real settings and can be used to guide the design and development of health monitoring systems.

Data and Code Availability. This paper uses the Homekit2020 dataset (Merrill et al., 2023) collected as part of the Home Testing of Respiratory Illness Study by Evidation Health and described in Synapse. The study was conducted in partnership with the Biomedical Advanced Research and Development Authority (BARDA), an existing office of the U.S. Department of Health and Human Services, and Audere, a non-profit digital health technology corporation. We provide the code repository here.

Institutional Review Board (IRB). The study that collected the data used in this paper was approved by the Western Institutional Review Board (WIRB, Puyallup, WA, USA) and the University of Washington IRB (Study #1271380).

1. Introduction

Fatigue, defined as *a decrement in mental and/or physical performance caused by cognitive overload, physical exertion, sleep deprivation, circadian phase/circadian rhythm disruption, or illness* (Adão Martins et al., 2021), is one of the most prevalent symptoms of neurodegenerative diseases including Alzheimer's, Parkinson's, and Multiple Sclerosis. It impacts people's daily functioning and the overall quality of life. There is a pressing need for automated and frequent fatigue monitoring and management approaches (Adão Martins et al., 2021; Luo et al., 2020; Antikainen et al., 2022; Rao et al., 2023).

Mobile and wearable devices have emerged as promising alternatives for monitoring different aspects of health such as physical activity (Guan and Plötz, 2017), stress (Sano and Picard, 2013; Matton et al., 2023) and sleep stages (Gashi et al., 2022) as well as fatigue (Adão Martins et al., 2021; Antar et al., 2023; Rao et al., 2023). Such devices have the potential to revolutionize healthcare by enabling continuous and unobtrusive monitoring of different health aspects and parameters like heart rate, heart rate variability, respiration rate, and more.

Nevertheless, the deployment of such devices for health monitoring in the real world has been ham-

pered by two fundamental challenges: the scarcity of labeled data and data quality (Plötz, 2021). The limited availability of labeled datasets hinders the training and validation of machine learning algorithms. To address these challenges and advance the field of wearable health monitoring, researchers leveraged self-supervised learning (SSL) techniques (Deldari et al., 2022b,a; Saeed et al., 2021; Merrill and Althoff, 2023). These techniques have proven effective in other fields, like patient monitoring in the Intensive Care Unit (Yèche et al., 2021; Kuznetsova et al., 2023). However, a critical gap remains in understanding their applicability in the context of fatigue recognition, which is the focus of this paper.

We address this issue by presenting a comprehensive benchmark for the newly released Homekit2020 dataset (Merrill et al., 2023). This is one of the very few large-scale, real-world, and high-resolution datasets collected with wearable devices that is available to other researchers. It contains physiological and behavioral data like *heart rate*, *number of steps*, and *sleep stage*, as well as self-reports related to *fatigue state*. We aim to show the significance of SSL methods in recognizing fatigue using this data.

Nevertheless, learning from noisy and incomplete data is a challenging and open problem. These aspects of wearable devices pose obstacles to developing accurate and robust models not only for fatigue but overall for human behavior recognition. To tackle this issue, we explore data augmentation techniques designed to capture the potential noise and incompleteness inherent in wearable sensor data.

The main contributions of this work are as follows: (1) We provide a comprehensive benchmark of SSL tasks for fatigue recognition from physiological and behavioral data collected using wearable devices. We find that the domain features and multi-task learning achieve higher results than the other SSL tasks. (2) We investigate the importance of data augmentations for fatigue recognition using both SSL and supervised learning pipelines. The data augmentations reflect real-world problems with wearable device data. (3) We use a large-scale, real-world dataset collected from 5034 participants over 4 months using wearable devices.

The paper is organized as follows. Section 2 presents an overview of related work in wearable health monitoring and SSL techniques for wearable health. In Section 3 we provide details about the data analysis pipeline. We describe the dataset used to run our benchmark in Section 4. Our experiments and results are described and discussed in Section 5 and Section 6. Section 7 presents concluding remarks.

2. Related Work

Wearable Health Monitoring. Several researchers investigated automated methods for health and well-being monitoring using mobile and wearable sensors. Examples include the use of inertial signals for human activity recognition (Guan and Plötz, 2017) and eating episodes detection (Bedri et al., 2017), physiological and behavioral signals for sleep monitoring (Gashi et al., 2022) and stress detection (Sano and Picard, 2013; Matton et al., 2023) and more (Starner et al., 2004). In contrast to these studies, we focus on the automatic assessment of fatigue using data collected with wearable devices.

Fatigue is one of the most prevalent symptoms in patients with chronic diseases such as Multiple Sclerosis, Alzheimer's, and Parkinson's disease. It impacts people's mood, sleep quality, and overall quality of life (Lobentanz et al., 2004; Stanton et al., 2006). Robust and automatic recognition of fatigue would enable both patients and clinicians to continuously monitor patients' fatigue over the long term, and this is the focus of this paper.

Only a few researchers investigate using data from wearable devices to assess fatigue (Luo et al., 2020; Adão Martins et al., 2021; Antikainen et al., 2022; Rao et al., 2023; Moebus et al., 2024). Antikainen et al. (2022), for instance, shows that objective physiological measures are significantly correlated to fatigue. The majority of these studies compare the performance of supervised learning (Rao et al., 2023; Antar et al., 2023; Moebus et al., 2024) or unsupervised learning methods (Luo et al., 2020) on handcrafted features. In addition, existing approaches use very small datasets or datasets collected in controlled, laboratory environments, which might not generalize to other, larger datasets collected in real-world settings, as shown in a comprehensive literature review by Adão Martins et al. (2021). Although there are several studies on supervised learning in this domain, there is a scarcity of studies focused on SSL.

Self-Supervised Learning for Wearable Health Monitoring. SSL techniques for time series can be divided into three main categories: *generative-based*, *contrastive-based* and *adversarial-based* (Liu et al., 2021; Zhang et al., 2023). Several researchers propose new SSL methods for health time-series data modeling (Yèche et al., 2021;

Table 1: An overview of existing self-supervised learning approaches in wearable health sensing. For a more comprehensive literature review, we refer the reader to Liu et al. (2023) and Deldari et al. (2022a).

Paper	Fatigue Task	Physiological Signals	Data Aug.	SSL Method	Backbone
Luo et al. (2020)	✓	✓	✗	✗	✗
Rao et al. (2023)	✓	✓	✗	✗	✗
Antar et al. (2023)	✓	✓	✗	✗	✗
Moebus et al. (2024)	✓	✓	✗	✗	✗
Merrill et al. (2023)	✓	✓	✗	✗	CNN Transformer
Saeed et al. (2019)	✗	✗	✓	Multi-task	TCN
Haresamudram et al. (2020)	✗	✗	✗	Generative	Transformer
Yuan et al. (2022)	✗	✗	✓	Contrastive	ResNet-V2
Jain et al. (2022)	✗	✗		Contrastive	CNN
Matton et al. (2023)	✗	✓	✓	Contrastive	CNN
Deldari et al. (2022b)	✗	✓	✓	Contrastive	N/A
Xu et al. (2021)	✗	✗	✗	Generative	BERT
Saeed et al. (2021)	✗	✓	✓	Contrastive Transformation Domain features Generative	TCN
Deldari et al. (2023)	✗	✓	✓	Generative	CNN
Merrill and Althoff (2023)	✓	✓	✗	Same user Domain features Generative	CNN Transformer
Our work	✓	✓	✓	Contrastive Same user Domain features Generative Multi-task	CNN Transformer

Kiyasseh et al., 2020; Tipirneni and Reddy, 2022). Deldari et al. (2022a) and Liu et al. (2023) provide a comprehensive literature review of such studies. In contrast to this work, we focus on wearable health time series used for fatigue monitoring. In this context, the work most closely related to ours is the study by Merrill and Althoff (2023). The authors propose a representation learning approach combining CNNs and a transformer architecture. They investigate three SSL tasks and find that using the prediction of hand-crafted features as a pre-training task increases the performance of several tasks including fatigue recognition. We expand the methodology of Merrill and Althoff (2023) and incorporate contrastive-based and multi-task learning SSL techniques that have shown promising results in other tasks like human activity recognition (Haresamudram et al., 2021), but have not previously been explored for fatigue recognition. In addition, we investigate the impact of data augmentations that reflect the potential quality of physiological and behavioral sensor data.

Summary of Related Work. Table 1 presents an overview of existing approaches for wearable health monitoring that are relevant to this work. From this literature review, we find that the prevailing focus in related work using SSL predominantly revolves around human activity recognition

tasks (Jain et al., 2022; Yuan et al., 2022; Haresamudram et al., 2020), stress and sleep analysis (Saeed et al., 2021; Matton et al., 2023) and do not consider fatigue, which is a prevalent and understudied health aspect. Several studies investigate only one type of SSL task, for instance, multi-task (Saeed et al., 2019) or contrastive learning (Matton et al., 2023; Yuan et al., 2022; Deldari et al., 2023). While many studies employ conventional architectures like Convolutional Neural Networks (CNNs) (Deldari et al., 2023) as foundational models, we adopt a unique strategy, first introduced by Merrill and Althoff (2023), by employing CNNs for efficient feature extraction and Transformers to capture the sequential dependencies within our dataset, thereby enhancing the robustness and effectiveness of our approach. Our work further confronts challenges inherent in wearable data, such as missing data, through an extensive exploration of data augmentation techniques tailored to address these issues.

3. Methods

3.1. Model Architecture

The model we used in this work consists of a Convolutional Neural Network (CNN) (LeCun et al., 2015), a Transformer (Vaswani et al., 2017), and a linear projection head. We adopt this model from Merrill and Althoff (2023) for several reasons. First, the CNN encoder learns hierarchical feature representation and reduces the dimensionality of longitudinal sensor data. Second, the transformer learns relationships between these features extracted from temporal sensor data. Lastly, this architecture provided the highest performance for the majority of the tasks explored by Merrill and Althoff (2023).

3.2. Data Augmentations

While there are different types of data augmentations available in the literature, our selection criteria were simulating the low data quality scenario in wearable devices, which is a common challenge when working with this type of data (Plötz, 2021). We implement four types of data augmentations, namely, *noise*, *masking*, *permutation*, and *swapping*. Noise refers to applying Gaussian noise to sensor data. Masking is the application of a Dropout layer (Hinton et al., 2012) before the encoder. To implement the permutation and swap data augmentations, we split the week-long data into seven equal segments, each corresponding to a day. Permutation shuffles these segments randomly. Swapping switches two neighboring segments to simulate cases when sensor data streams of a day are swapped with another day. We investigated the impact of each of these data augmentations for fatigue recognition.

3.3. Self-Supervised Pre-training Tasks

Although Zhang et al. (2023) demonstrated a diverse set of SSL tasks for time series, our benchmark focuses on SSL methods that exhibit promising performance on wearable sensor data. These methods have not been extensively investigated for the fatigue recognition. Figure 1 and Table 2 present an overview of the SSL tasks explored in this work and their corresponding loss functions. For each of these tasks, we use the same encoder described in Section 3.1.

Contrastive. Similar to SimCLR's approach (Chen et al., 2020), we formulate this task by training a model to distinguish between the positive and negative examples derived from the sensor data. Contrastive learning has shown promising results in related work for human activity recognition (Jain et al., 2022; Yuan et al., 2022) and stress detection (Matton et al., 2023). To create the positive pairs, we randomly select a data sample from the train set and apply a transformation two times on the original sample, creating two similar, transformed samples - the anchor and the positive sample. We then randomly select another data sample and transform it, creating a negative sample. We transform the negative sample to ensure that the model learns representations that distinguish positive and negative pairs instead of representations related to whether the transformation was applied or not. To optimize the model's parameters during the pre-training, we experiment with two types of loss functions: `triplet margin loss` (Baltas et al., 2016) and `infoNCE contrastive loss` (Oord et al., 2018). The triplet loss function minimizes the distance between the anchor and the positive sample, simultaneously maximizing the distance between the negative sample from the anchor and the positive sample. Contrastive loss uses categorical cross-entropy loss to distinguish between a positive sample and a set of negative samples. With these approaches, we investigate the feasibility of learning general-purpose representations by extracting similarities between features of the original and transformed sample and differences with other examples.

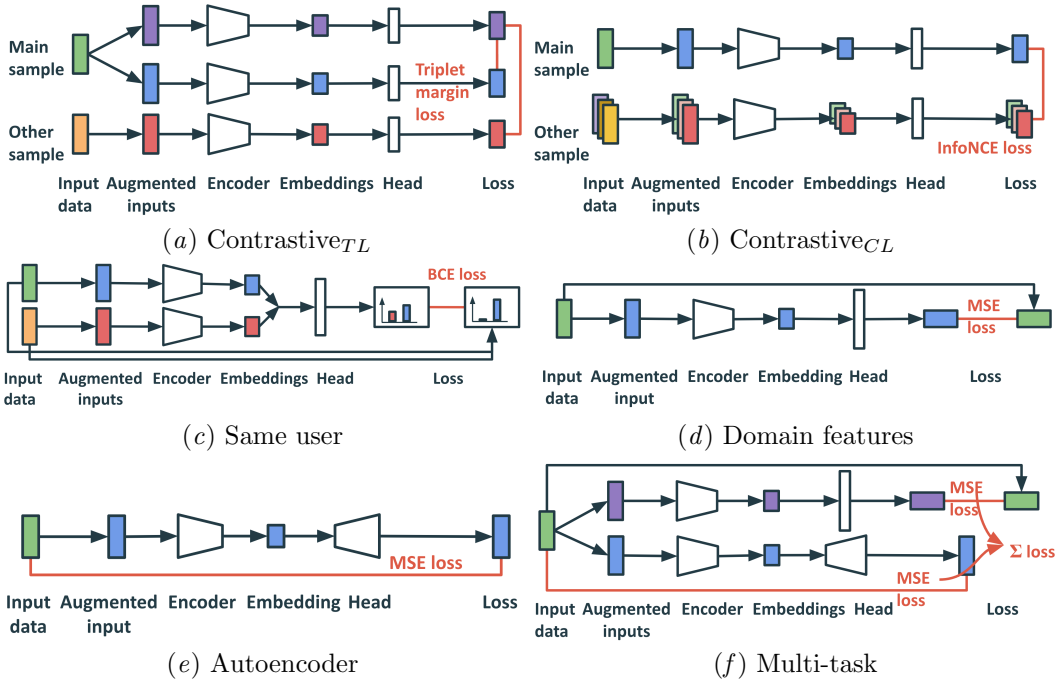

Figure 1: An overview of the SSL methods explored in this work. *Contrastive* refers to pretraining using Triplet or InfoNCE loss. *Same user* refers to distinguishing between the data of the same or different users. *Domain features* predicts hand-crafted features. *Autoencoder* reconstructs the provided input. *Multi-task* is a combination of the two SSL techniques above by using their respective heads and losses. Unless specifically noted, tasks use the same head as input and the same encoder explained in Section 3.1.

Table 2: A summary of the loss functions used for each SSL task. TL refers to the triplet loss function (Chechik et al., 2010). CL refers to InfoNCE loss function Oord et al. (2018).

SSL Task	**Loss Function**	**Formalism**
Contrastive$_{TL}$	$\max\{d(a_i, p_i) - d(a_i, n_i) + m, 0\}$	where `a` - anchor, `n` - negative and `p` - positive and `m` - minimum offset between distances of similar vs dissimilar pairs.
Contrastive$_{CL}$	$\mathrm{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right]$	Given a set of N random samples containing one positive sample from $p\,(x_{t+k}, c_t)$ and $N-1$ negative samples from the 'proposal' distribution $p\,(x_{t+k})$.
Same User	$-(y \log(p) + w(1-y)\log(1-p))$	`log` - the natural log, `y` - binary indicator (0 or 1) `p` - probability observation is of positive class.
Domain Features	$\sum_{i=1}^{D}(x_i - y_i)^2$	`x` - inputs and `y` - targets
Autoencoder	$\sum_{i=1}^{D}(\hat{x}_i - x_i)^2$	where $\hat{x}$ is a copy of x corrupted by a form of noise.
Multi-task	$L_{SSL1}(x, y) + L_{SSL2}(x, y)$	where L_{SSL1} and L_{SSL2} refer to the loss functions of SSL tasks that perform the best on inputs x and targets y.

Throughout the paper, we refer to these two methods as $\texttt{Contrastive}_{TL}$ and $\texttt{Contrastive}_{CL}$, respectively.

Same User. For this pre-training task, we randomly pair up samples of data from the same or different users and considered positive if they belong to the same user, and negative otherwise similar to Merrill and Althoff (2023). We used the Binary Cross Entropy (BCE) loss function to measure the difference between the predicted binary outcomes and actual labels. To counteract the imbalance in the number of positive and negative samples, we apply a weight to the negative part when calculating the loss.

Domain Features. This pretraining task consists of multiple regression tasks to predict the daily features extracted by the FitBit device from the raw, minute-level resolution sensor data, proposed by Merrill and Althoff (2023). The domain features consist of the 95th and 50th percentile of resting heart rate (HR), the standard deviation of resting HR, the 95th percentile of HR while awake, the 95th and 50th percentile of continuous steps, the number of minutes spent in bed, the number of minutes spent asleep, the total number of steps and indicators of missing HR, sleep, steps, and all data.

Autoencoder. The autoencoder consists of an encoder and a decoder, the latter mirrors the architecture of the encoder. This task reconstructs the input of the network. We experiment with both a *denoising autoencoder* and *normal autoencoder*. The denoising autoencoder adds noise to the input series to corrupt the data and reconstructs the original, unperturbed data and the autoencoder reconstructs the perturbed data. With this approach, we investigate the capability of the network to learn essential representations of the signal and avoid possible noise. The autoencoder provided promising results in other wearable health tasks (Saeed et al., 2021).

Multi-task. This task jointly trains the model on two tasks that share the data and encoder, but use separate prediction heads. For the optimization objective, we sum up the losses of subtasks as shown in Table 2. We hypothesize that training compatible pretext tasks together in a multi-task learning setting enhances domain generalization performance compared to training each task individually, as also shown by Albuquerque et al. (2020) in a computer vision task.

4. Dataset

We use the Homekit2020 dataset (Merrill et al., 2023), which consists of Fitbit data collected from 5034 participants over four months. The dataset contains two types of data: *wearable sensor data* and *self-reports*. Wearable data refers to minute-level measurements of the number of steps, average heart rate, and binary flags related to the sleep state (e.g., *sleep*, *awake*, or *in bed*). Self-reports refer to daily ratings of participants' fatigue levels during the day.

Fatigue Task. Participants reported their daily fatigue level on a scale from 1 to 4. From this scale, Merrill et al. (2023) define the task of fatigue based on the question *"Will the participant report severe fatigue today?"*. Severe fatigue refers to an answer equal to three or more and low fatigue otherwise. Homekit2020 suffers from severe class imbalance with a ratio of 1:78 of positive to negative labels. For further information regarding the dataset, we refer the reader to Merrill et al. (2023).

Data Exploration. We observe that the dataset contains outliers, which deviate significantly from the true distribution of the data. This is in particular evident when considering potential step counts per minute corresponding to various activity states such as resting, walking, or running. To be able to replicate the results by Merrill and Althoff (2023), we performed the experiments on the original data. We include the distribution of the data in Figure **??** in Appendix A.

5. Experimental Setting

The main goal of the study is to achieve strong fatigue recognition performance considering the possible few labeled data and low data quality. Thus, we explore existing SSL approaches to identify a good initialization for the encoder. We further evaluate how the data augmentations that mirror possible noise and missing data in wearable devices impact the overall performance of pre-training tasks. We examine the performance of both SSL tasks and data augmentations for fatigue recognition.

Model. We used an architecture introduced by Merrill and Althoff (2023), which consists of a 1-D CNN with 3 convolutional blocks each with a Rectified Linear Unit (ReLU) activation function. The Transformer consists of 2 blocks each with 4 attention heads and a feed-forward layer. A linear layer is used as the classification head. To train the model, we use

Adam optimizer and cross-entropy loss. We provide further details about the model in Appendix B.

Self-supervised Tasks. When performing SSL tasks, we train the model consisting of the encoder and the defined objective for each task. After pre-training, we reuse the weights of the encoder for the supervised learning task, but discard the head trained on the SSL task and initialize an appropriate one randomly. The Contrastive$_{TL}$ initially provided poor performance, so we explored methods to improve it and decided to use batch normalization for regularization. We provide more details about the parameters of the contrastive approach in Appendix C.

Data Augmentations (DA). We explore three types of DA explained in Section 3. We perform additional experiments by applying the three augmentations simultaneously or selecting only one at random for each batch of data. We refer to such experiments as *All DA* and *Random DA*, respectively. We fine-tune the parameters of the above DA only for the fully supervised learning pipeline. Then we apply the best-performing hyperparameters of each augmentation method on the pre-training methods. This way, for each pre-training method we only evaluate five configurations, which consist of three for each DA and two for their combination. For details on how we select the parameters of DAs refer to Appendix B.

Baseline. To compare the performance of SSL tasks, we use the same encoder explained in Section 3 and train it end-to-end with supervised learning, as a common baseline in the literature (Deldari et al., 2023; Saeed et al., 2021). To explore the impact of DAs, we further evaluate the performance with and without DA.

Metrics. To evaluate the performance of SSL tasks and DAs, we use the area under the precision-recall curve (AUPRC) and the area under the receiver operating curve (AUROC) similar to Merrill and Althoff (2023). AUPRC shows the tradeoff between precision and recall for different thresholds. It is an informative metric for extremely imbalanced datasets (Saito and Rehmsmeier, 2015) as in our case. The AUROC curve shows the true positive rate against the false positive rate.

Procedure. To evaluate our approach we use the *temporal split* and *user split* similar to Merrill et al. (2023). Temporal split utilizes the first half of the data from a user for training the model and the remaining half of the data for testing. This technique verifies the ability of the model to generalize to distribution shifts over time. User split partitions the data into two participant-independent splits and uses one split for training the model and the other for testing. This validation procedure assesses the model's capability to generalize to new, unseen users.

6. Results and Discussion

In what follows we report the results obtained by applying the steps described in Section 3 and Section 5 to distinguish between severe fatigue and low fatigue. We first compare the performance between the SSL and fully supervised learning pipelines. Following that, we investigate the impact of DAs by comparing fatigue recognition results with and without augmentation. Then, we discuss the results obtained in a limited labeled data regime. Lastly, we evaluate the performance of the temporal and user splits.

Comparison of SSL and Supervised Learning. We evaluate the ability of pre-training tasks to initialize the parameters of the encoder and their performance for fatigue recognition. We compare the results of these pre-training tasks with the fully supervised learning baseline. Table 3 shows the average AUPRC and AUROC metrics for both the pre-training methods and supervised baseline using temporal split. Overall, we find that all the pretraining tasks, except the autoencoder, outperform the baseline. The AUROC for the `Multi-task`, `Domain features` and `Denoising Autoencoder` tasks in validation set is 0.7365, 0.7391, and 0.7197, respectively, which are 1-4 percentage points higher than the baseline. We observe similar results for the AUPRC metric. Overall, our results provide a new benchmark for fatigue recognition using wearable devices, representing a significant contribution to the field. The SSL tasks explored in this work improve the performance of fatigue recognition in comparison to the original benchmark by Merrill et al. (2023). We could not replicate the results by Merrill and Althoff (2023) even by using their codebase. For this reason, the results are not directly comparable to them. **Takeaway**: Self-supervised pretraining outperforms supervised learning for fatigue recognition.

Comparison of SSL Tasks. We then compare the performance of SSL methods explored in this work (explained in Section 3.3). Table 3 presents the AUPRC and AUROC for each SSL task for fatigue recognition using temporal split. We find that the Domain Features task, which trains the model to estimate the handcrafted features, outperforms the other tasks significantly, which is in line with the results of

Table 3: Comparison of performance of self-supervised pre-training tasks used in this work for fatigue recognition using the temporal split. Class balance: 1:78. We used random data augmentations for contrastive learning methods. We highlight the top two performing SSL methods in **bold**. The multi-task approach combines Domain features and Denoising Autoencoder as they provided the highest results in the validation set. The table reports the average (std) of the AUROC and AUPRC scores and their statistical significance according to t-test with `p<0.01` (*) and `p<0.001` (**).

Method	**AUPRC**$_{Test}$	**AUROC**$_{Test}$	**AUPRC**$_{Val}$	**AUROC**$_{Val}$
Contrastive$_{CL}$	0.0377 (0.0019)	0.7076 (0.0064)	0.0504 (0.0034)	0.7168 (0.0094)*
Contrastive$_{TL}$	0.0278 (0.0085)	0.7299 (0.0039)	0.0276 (0.0026)	0.7029 (0.0053)
Same user	0.0314 (0.0040)	0.7065 (0.0132)	0.0296 (0.0012)	0.6971 (0.0044)
Domain features	**0.0552 (0.0019)**	**0.7371 (0.0005)**	**0.0550 (0.0035)**	**0.7391 (0.0021)****
Autoencoder	0.0278 (0.0049)	0.6820 (0.0103)	0.0385 (0.0037)	0.6952 (0.0097)
Denoising Autoencoder	0.0443 (0.0023)	0.7203 (0.0033)	0.0545 (0.0011)	0.7197 (0.0093)*
Multi-task	**0.0489 (0.0017)**	**0.7323 (0.0007)**	**0.0578 (0.0033)**	**0.7365 (0.0073)****
No pretraining	0.0299 (0.0011)	0.6978 (0.0037)	0.0347 (0.0013)	0.7030 (0.0024)

Merrill and Althoff (2023). As opposed to their findings, the Multi-task learning method provides comparable results to the Domain Features. The AUPRC and AUROC for this task are 0.0578 and 0.7365, respectively, which are higher by 2-3 and 1-3 percentage points than the other SSL tasks. **Takeaway**: Domain features and multitask learning outperform other pretraining methods for recognizing fatigue.

Performance of Data Augmentations. To better understand the impact of DAs explained in Section 3.2, we investigate their performance on the downstream task. Table 4 shows the AUPRC and AUROC for each SSL task and the DA used to generate positive pairs (e.g., for contrastive methods) or to modify the original data (e.g., for Denoising Autoencoder). We fine-tuned the parameters of DAs and selected the one that provided the best performance for recognizing fatigue. We refer the reader to Table 5 in Appendix B for the results of other parameters. Table 4 reveals that in 7 out of 8 cases, data augmentations lead to improvements in both AUROC and AUPRC across both validation and test sets. Therefore, we conclude that data augmentation enhances the performance of SSL tasks in most of the cases, with the exception being AUROC of Domain Features in the test set. Similarly, the performance of the supervised baseline improved by a large margin when applying the DA. These findings suggest that the DAs explored in this work are effective for distorting physiological and behavioral data used for fatigue recognition. We believe this is because DAs produce samples from overlapping but different distributions (Bengio et al., 2011), which results in improved generalization by training with diverse samples. These results support previous findings on the impact of DA for increasing the generalizability of deep learning models for wearable data tasks (Alawneh et al., 2021; Yang et al., 2022). **Takeaway**: Data augmentations enhance the performance of fatigue recognition.

Performance on Limited Labeled Data. While collecting sensor data from wearable devices is straightforward, acquiring labels associated with fatigue state is challenging due to a decline in users' compliance with data collection over time. To replicate this scenario, in this set of experiments, we evaluate the robustness of pretraining in scenarios when limited samples of labeled data are available for fine-tuning and large unlabeled datasets are available for pre-training. Figure 2 reports the AUROC of Domain Features, Denoising Autoencoder and Multi-task pretraining tasks as well as the model with no pretraining using fractions of fatigue labels. The Multi-task and Domain Features tasks consistently outperform Denoising Autoencoder and end-to-end training for fatigue recognition when the number of labels is reduced. The performance of these two SSL methods is robust even with only 25% of labeled data available as well as when all the labeled data is used. We observe similar results for the AUPRC scores, which we provide in Figure 4 in Appendix D. These results indi-

Table 4: Performance of the data augmentations explored in this work. We measure the performance through the area under the precision-recall curve (`AUPRC`) and the area under the receiver operating curve (`AUROC`) for both validation (`Val`) and test (`Test`) sets. The table reports the average (standard deviation) of the AUROC and AUPRC scores. The best results are highlighted in **bold**. Note that Contrastive, Denoising Autoencoder, and Multi-task learning methods require data augmentations to be executed. For this reason, we add the results without data augmentations only for Same User, Domain Features, and Autoencoder.

Method	**DA**	**AUPRC**$_{Test}$	**AUROC**$_{Test}$	**AUPRC**$_{Val}$	**AUROC**$_{Val}$
Contrastive$_{TL}$	Noise	0.0182 (0.0080)	0.6445 (0.0099)	0.0117 (0.0007)	0.6469 (0.0006)
	Mask	0.0278 (0.0085)	**0.7299 (0.0039)**	**0.0276 (0.0026)**	**0.7029 (0.0053)**
	Swap	0.0207 (0.0096)	0.6608 (0.0357)	0.0201 (0.0115)	0.6616 (0.0353)
	Random DA	0.0223 (0.0132)	0.5828 (0.0673)	0.0094 (0.0029)	0.5773 (0.0572)
	All DA	**0.0347 (0.0167)**	0.6684 (0.0978)	0.0223 (0.0105)	0.6503 (0.0823)
Contrastive$_{CL}$	Noise	0.0368 (0.0012)	**0.7127 (0.0036)**	0.0410 (0.0002)	0.7072 (0.0026)
	Mask	**0.0377 (0.0019)**	0.7076 (0.0064)	**0.0504 (0.0034)**	**0.7168 (0.0094)**
	Swap	0.0351 (0.0031)	0.6928 (0.0102)	0.0376 (0.0048)	0.6995 (0.0044)
	Random DA	0.0340 (0.0165)	0.6547 (0.0725)	0.0292 (0.0154)	0.6540 (0.0652)
	All DA	0.0291 (0.0151)	0.6524 (0.0709)	0.0264 (0.0149)	0.6448 (0.0588)
Same User	Noise	0.0335 (0.0011)	0.7154 (0.0112)	0.0351 (0.0086)	0.7024 (0.0091)
	Mask	0.0350 (0.0020)	0.7050 (0.0048)	0.0334 (0.0012)	0.6998 (0.0046)
	Swap	0.0404 (0.0015)	0.7124 (0.0094)	0.0355 (0.0040)	0.7049 (0.0018)
	Random DA	0.0374 (0.0027)	0.7166 (0.0068)	0.0353 (0.0039)	0.7057 (0.0093)
	All DA	**0.0406 (0.0032)**	**0.7203 (0.0074)**	**0.0371 (0.0045)**	**0.7117 (0.0090)**
	None	0.0314 (0.0040)	0.7065 (0.0132)	0.0296 (0.0012)	0.6971 (0.0044)
Domain Features	Noise	0.0543 (0.0049)	0.7292 (0.0050)	0.0494 (0.0021)	0.7332 (0.0036)
	Mask	0.0522 (0.0018)	0.7210 (0.0067)	0.0586 (0.0017)	0.7362 (0.0008)
	Swap	**0.0598 (0.0033)**	0.7285 (0.0020)	0.0559 (0.0001)	0.7415 (0.0049)
	Random DA	0.0579 (0.0030)	0.7306 (0.0025)	0.0568 (0.0022)	0.7391 (0.0032)
	All DA	0.0566 (0.0011)	0.7285 (0.0021)	**0.0627 (0.0011)**	**0.7470 (0.0032)**
	None	0.0552 (0.0019)	**0.7371 (0.0005)**	0.0550 (0.0035)	0.7391 (0.0021)
Autoencoder	Noise	0.0344 (0.0011)	0.7029 (0.0053)	0.0428 (0.0019)	0.7098 (0.0031)
	Mask	0.0425 (0.0033)	**0.7157 (0.0053)**	0.0455 (0.0028)	0.7149 (0.0020)
	Swap	0.0386 (0.0038)	0.7009 (0.0068)	0.0452 (0.0043)	0.7031 (0.0051)
	Random DA	0.0395 (0.0016)	0.7070 (0.0154)	0.0482 (0.0088)	0.7098 (0.0074)
	All	**0.0457 (0.0022)**	0.7114 (0.0043)	**0.0501 (0.0034)**	**0.7238 (0.0017)**
	None	0.0278 (0.0049)	0.6820 (0.0103)	0.0385 (0.0037)	0.6952 (0.0097)
Denoising Autoencoder	Noise	0.0360 (0.0040)	0.7026 (0.0076)	0.0468 (0.0047)	0.7079 (0.0094)
	Mask	0.0323 (0.0036)	0.6992 (0.0052)	0.0435 (0.0072)	0.7067 (0.0194)
	Swap	0.0369 (0.0008)	0.7001 (0.0069)	0.0412 (0.0046)	0.7015 (0.0069)
	Random DA	**0.0443 (0.0023)**	0.7203 (0.0033)	**0.0545 (0.0011)**	**0.7197 (0.0093)**
	All	0.0434 (0.0039)	**0.7226 (0.0009)**	0.0483 (0.0038)	0.7186 (0.0078)
Multi-task	Noise	**0.0579 (0.0022)**	0.7261 (0.0057)	0.0535 (0.0054)	0.7316 (0.0098)
	Mask	0.0492 (0.0014)	0.7318 (0.0008)	0.0535 (0.0014)	0.7304 (0.0015)
	Swap	0.0565 (0.0049)	0.7178 (0.0018)	0.0528 (0.0030)	0.7258 (0.0046)
	Random DA	0.0523 (0.0032)	0.7289 (0.0045)	0.0507 (0.0008)	**0.7385 (0.0021)**
	All DA	0.0489 (0.0017)	**0.7323 (0.0007)**	**0.0578 (0.0033)**	0.7365 (0.0073)
No pretraining	Noise	0.0378 (0.0003)	0.7140 (0.0038)	0.0394 (0.0008)	0.7128 (0.0022)
	Mask	**0.0409 (0.0037)**	0.7151 (0.0037)	**0.0508 (0.0009)**	**0.7282 (0.0011)**
	Swap	0.0322 (0.0015)	0.7051 (0.0044)	0.0370 (0.0007)	0.7100 (0.0027)
	Random DA	0.0367 (0.0013)	**0.7169 (0.0037)**	0.0476 (0.0017)	0.7201 (0.0039)
	All DA	**0.0409 (0.0034)**	0.7140 (0.0083)	0.0464 (0.0014)	0.7263 (0.0007)
	None	0.0299 (0.0011)	0.6978 (0.0037)	0.0347 (0.0013)	0.7030 (0.0024)

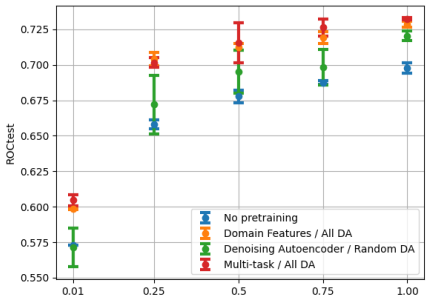

Figure 2: Average AUROC scores for fatigue recognition under limited labeled data using the pre-training tasks with the highest performance and temporal split. The X-axis shows the amount of available labeled data.

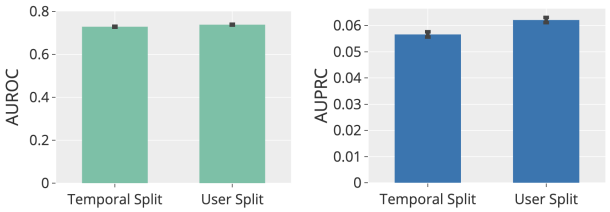

Figure 3: Average AUROC and AUPRC scores for fatigue recognition using `temporal` and `user` split as well as using the Domain Features and data augmentations.

cate that Multi-task and Domain Features tasks learn robust features from unlabeled data that generalize to limited labeled data scenarios. The model pre-trained with Contrastive$_{TL}$ performs worse than the others. One explanation for this poor performance is the random selection of positive and negative pairs for this task. Investigating informed techniques of positive and negative pairs selection is an interesting direction for future work. **Takeaway**: SSL tasks show improved performance over supervised training on fewer amount of labels.

Performance on New Users. Figure 3 presents the AUROC and AUPRC metrics using Domain Features, DAs, and the two evaluation techniques described in Section 5. The AUROC and AUPRC using temporal split are 0.7285 and 0.0566 and using user split are 0.7380 and 0.0621. These results show the performance of the SSL tasks with DAs is comparable between the two validation techniques. These results imply that the SSL task and DAs learn robust features that can be generalized to the data of new, unseen users. They further suggest that classifying the data from the future in the temporal split task is as challenging as classifying the data of a new user. **Takeaway**: SSL tasks show robust performance on new, unseen users.

7. Conclusions

In this paper, we presented the first benchmark of SSL tasks for fatigue recognition using data collected from wearable devices. We further examined the impact of data augmentations that reflect the quality of wearable sensor data collected in real-world settings. Our benchmark is systematic and comprehensive in the types of SSL techniques explored in comparison to related work. We evaluated our benchmark on a large-scale, real-world dataset collected from 5034 participants over four months. Our main findings suggest that the majority of pretraining tasks reach the performance of or outperform the fully supervised baseline, which is in line with previous findings in other types of tasks. We further find that in the majority of the cases, data augmentations contribute significantly to enhancing the performance of fatigue recognition models. This aligns with the broader understanding that data augmentations are suitable for training more robust and accurate machine learning models for wearable data.

While our approach shows promising results, it is crucial to acknowledge the limitations of our work. One limitation stems from relying on one dataset, which might not fully capture the diversity of datasets. However, given that the Homekit2020 dataset was collected from thousands of participants over months, we believe that the results are more representative than using such datasets which have been collected from less than a hundred participants over 1 or 2 weeks. We plan to investigate how these methods generalize to other datasets e.g., (Luo et al., 2020; Gashi et al., 2024). Another limitation pertains to the interpretability of the model, as complex algorithms may make it challenging to fully understand the decision-making process. We plan to explore the impact of different features in future work. Lastly, the self-reported nature of the fatigue state and the subjectivity of fatigue in the dataset that we used for our analysis, are other important limitations to acknowledge when interpreting our results.

Despite these limitations, we believe that our results demonstrate the potential of SSL to learn meaningful representations from wearable data for fatigue recognition. Furthermore, they demonstrate the advantages of incorporating data augmentations, both within the SSL and supervised learning pipelines.

Our research contributes to the application of machine learning in the analysis of wearable sensor data drawing inspiration from deep learning methods. We believe that our findings will inspire future work on SSL for fatigue monitoring using wearable devices, with the ultimate goal of informing healthcare systems and decision-making.

References

Neusa R Adão Martins, Simon Annaheim, Christina M Spengler, and René M Rossi. Fatigue Monitoring Through Wearables: A State-of-the-art Review. *Frontiers in physiology*, page 2285, 2021.

Luay Alawneh, Tamam Alsarhan, Mohammad Al-Zinati, Mahmoud Al-Ayyoub, Yaser Jararweh, and Hongtao Lu. Enhancing human activity recognition using deep learning and time series augmented data. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–16, 2021.

Isabela Albuquerque, Nikhil Naik, Junnan Li, Nitish Keskar, and Richard Socher. Improving out-of-distribution generalization via multi-task self-supervised pretraining. *arXiv preprint arXiv:2003.13525*, 2020.

Anindya Das Antar, Anna Kratz, and Nikola Banovic. Behavior modeling approach for forecasting physical functioning of people with multiple sclerosis. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT 2023)*, 7(1):1–29, 2023.

Emmi Antikainen, Haneen Njoum, Jennifer Kudelka, Diogo Branco, Rana Zia Ur Rehman, Victoria Macrae, Kristen Davies, Hanna Hildesheim, Kirsten Emmert, Ralf Reilmann, et al. Assessing fatigue and sleep in chronic diseases using physiological signals from wearables: A pilot study. *Frontiers in Physiology*, page 2380, 2022.

Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Bmvc*, volume 1, page 3, 2016.

Abdelkareem Bedri, Richard Li, Malcolm Haynes, Raj Prateek Kosaraju, Ishaan Grover, Temiloluwa Prioleau, Min Yan Beh, Mayank Goel, Thad Starner, and Gregory Abowd. Earbit: using wearable sensors to detect eating episodes in unconstrained environments. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 1(3):1–20, 2017.

Yoshua Bengio, Frédéric Bastien, Arnaud Bergeron, Nicolas Boulanger-Lewandowski, Thomas Breuel, Youssouf Chherawala, Moustapha Cisse, Myriam Côté, Dumitru Erhan, Jeremy Eustache, et al. Deep learners benefit more from out-of-distribution examples. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 164–172. JMLR Workshop and Conference Proceedings, 2011.

Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(3), 2010.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Shohreh Deldari, Hao Xue, Aaqib Saeed, Jiayuan He, Daniel V Smith, and Flora D Salim. Beyond just vision: A review on self-supervised representation learning on multimodal and temporal data. *arXiv preprint arXiv:2206.02353*, 2022a.

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

Shohreh Deldari, Dimitris Spathis, Mohammad Malekzadeh, Fahim Kawsar, Flora Salim, and Akhil Mathur. Latent masking for multimodal self-supervised learning in health timeseries. *arXiv preprint arXiv:2307.16847*, 2023.

Shkurta Gashi, Lidia Alecci, Elena Di Lascio, Maike E Debus, Francesca Gasparini, and Silvia Santini. The role of model personalization for sleep

stage and sleep quality recognition using wearables. *IEEE Pervasive Computing*, 21(2):69–77, 2022.

Shkurta Gashi, Pietro Oldrati, Max Moebus, Marc Hilty, Liliana Barrios, Firat Ozdemir, PHRT Consortium, Veronika Kana, Andreas Lutterotti, Gunnar Rätsch, and Christian Holz. Modeling Multiple Sclerosis using Mobile and Wearable Sensor Data, January 2024. URL https://doi.org/10.5281/zenodo.10497826.

Yu Guan and Thomas Plötz. Ensembles of deep LSTM learners for activity recognition using wearables. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 1(2):1–28, 2017.

Harish Haresamudram, Apoorva Beedu, Varun Agrawal, Patrick L Grady, Irfan Essa, Judy Hoffman, and Thomas Plötz. Masked reconstruction based self-supervision for human activity recognition. In *Proceedings of the 2020 ACM International Symposium on Wearable Computers*, pages 45–49, 2020.

Harish Haresamudram, Irfan Essa, and Thomas Plötz. Contrastive predictive coding for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–26, 2021.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors, 2012.

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

Dani Kiyasseh, Tingting Zhu, and David A Clifton. CLOCS: Contrastive learning of cardiac signals. *arXiv preprint arXiv:2005.13249*, 2020.

Rita Kuznetsova, Alizée Pace, Manuel Burger, Hugo Yèche, and Gunnar Rätsch. On the importance of step-wise embeddings for heterogeneous clinical time-series. In *Machine Learning for Health (ML4H)*, pages 268–291. PMLR, 2023.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering*, 35(1):857–876, 2021.

Ziyu Liu, Azadeh Alavi, Minyi Li, and Xiang Zhang. Self-supervised contrastive learning for medical time series: A systematic review. *Sensors*, 23(9):4221, 2023.

IS Lobentanz, S Asenbaum, K Vass, C Sauter, G Klösch, H Kollegger, W Kristoferitsch, and Josef Zeitlhofer. Factors influencing quality of life in multiple sclerosis patients: disability, depressive mood, fatigue and sleep quality. *Acta Neurologica Scandinavica*, 110(1):6–13, 2004.

Hongyu Luo, Pierre-Alexandre Lee, Ieuan Clay, Martin Jaggi, and Valeria De Luca. Assessment of fatigue using wearable sensors: a pilot study. *Digital biomarkers*, 4(1):59–72, 2020.

Katie Matton, Robert Lewis, John Guttag, and Rosalind Picard. Contrastive learning of electrodermal activity representations for stress detection. In *Conference on Health, Inference, and Learning*, pages 410–426. PMLR, 2023.

Mika A Merrill and Tim Althoff. Self-supervised pretraining and transfer learning enable flu and covid-19 predictions in small mobile sensing datasets. In *Conference on Health, Inference, and Learning*, pages 191–206. PMLR, 2023.

Mike A Merrill, Esteban Safranchik, Arinbjörn Kolbeinsson, Piyusha Gade, Ernesto Ramirez, Ludwig Schmidt, Luca Foshchini, and Tim Althoff. Homekit2020: A benchmark for time series classification on a large mobile sensing dataset with laboratory tested ground truth of influenza infections. In *Conference on Health, Inference, and Learning*, pages 207–228. PMLR, 2023.

Max Moebus, Shkurta Gashi, Marc Hilty, Pietro Oldrati, and Christian Holz. Meaningful digital biomarkers derived from wearable sensor to predict daily fatigue in multiple sclerosis patients and healthy controls. *iScience*, 2024.

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

Thomas Plötz. Applying machine learning for sensor data analysis in interactive systems: Common pitfalls of pragmatic use and ways to avoid them. *ACM Computing Surveys (CSUR)*, 54(6): 1–25, 2021.

Chaitra Rao, Elena Di Lascio, David Demanse, Nell Marshall, Monika Sopala, and Valeria De Luca. Association of digital measures and self-reported fatigue: a remote observational study in healthy participants and participants with chronic inflammatory rheumatic disease. *Frontiers in Digital Health*, 5:1099456, 2023.

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

Aaqib Saeed, Victor Ungureanu, and Beat Gfeller. Sense and learn: Self-supervision for omnipresent sensors. *Machine Learning with Applications*, 6: 100152, 2021.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

Akane Sano and Rosalind W Picard. Stress recognition using wearable sensors and mobile phones. In *2013 Humaine association conference on affective computing and intelligent interaction*, pages 671–676. IEEE, 2013.

BR Stanton, F Barnes, and E Silber. Sleep and fatigue in multiple sclerosis. *Multiple Sclerosis Journal*, 12(4):481–486, 2006.

T Starner, S Mann, A Pentland, AK Dey, D Salber, GD Abowd, M Futakawa, D Cottet, J Grzyb, T Kirstein, et al. Wearable systems for health care applications. *Methods of information in medicine*, 43(03):232–238, 2004.

Sindhu Tipirneni and Chandan K Reddy. Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6):1–17, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Huatao Xu, Pengfei Zhou, Rui Tan, Mo Li, and Guobin Shen. LIMU-BERT: Unleashing the potential of unlabeled data for imu sensing applications. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 220–233, 2021.

Huiyuan Yang, Han Yu, and Akane Sano. Empirical evaluation of data augmentations for biobehavioral time series data with deep learning. *arXiv preprint arXiv:2210.06701*, 2022.

Hugo Yèche, Gideon Dresdner, Francesco Locatello, Matthias Hüser, and Gunnar Rätsch. Neighborhood contrastive learning applied to online patient monitoring. In *International Conference on Machine Learning*, pages 11964–11974. PMLR, 2021.

Hang Yuan, Shing Chan, Andrew P Creagh, Catherine Tong, David A Clifton, and Aiden Doherty. Self-supervised learning for human activity recognition using 700,000 person-days of wearable data. *arXiv preprint arXiv:2206.02909*, 2022.

Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Zhang, Yuxuan Liang, Guansong Pang, Dongjin Song, et al. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *arXiv preprint arXiv:2306.10125*, 2023.

Appendix A. Homekit2020 Dataset

The Homekit2020 dataset was collected for 4 months continuously. Participants completed self-reports about their fatigue daily and wore the Fitbit device which collected minute-level sensor data. Merrill et al. (2023) provide a summary of detailed statistics regarding the dataset. In particular, the authors show that the completion rate of daily self-reports was 85% over the whole study. The average number of days of data provided by each user is 114. In addition, the mean percentage of missing data per day was 9.8 %, which corresponds to 21.6 hours of data per day. These statistics show that overall the fraction of missing data in the Homekit2020 dataset was

low. For a further description of the dataset, we refer the reader to Merrill et al. (2023).

Appendix B. Hyperparameter Tuning

The convolutional encoder consists of 3 blocks as described above, with kernel sizes *[5, 5, 2]*, number of output channels as *[8, 16, 32]*, and stride sizes of *[5, 3, 2]* respectively. Two transformer blocks are stacked after the encoder, each having 4 attention heads and a dropout rate of 0.4 in the residual block. The dimension of embeddings produced by the convolutional encoder and the transformer blocks is 32. We do not use positional encoding. For training, we use 20 warmup steps, disable *val_bootstraps* by setting it to 0 to avoid a memory leak, and limit trains to 20 epochs at most. This limit is rarely reached, and best-performing runs do not need that many epochs neither during pre-training nor during regular training. Additionally, we enable early stopping based on validation AUROC (or, if not applicable, validation loss) and stop once the chosen metric has not improved in 2 consecutive epochs.

To choose the parameters of data augmentation approaches, we evaluate the methods and pick the highest mean AUPRC.

For the triplet margin loss, we use a margin of 1.0 (default parameter in PyTorch). For the cross-entropy loss of the same user approach, we scale the loss of negative samples by 0.1.

Overall, we reused the hyperparameters that were found to be optimal by Merrill et al. (2023).

Appendix C. Other Experiments

C.1. Contrastive Learning

Contrastive learning with triplet loss initially showed very poor performance, providing worse results than a randomly initialized network, taking a considerable amount of training time to even reach the performance of an untrained network. To improve the performance, we explored different regularization methods such as weight decay or batch normalization. The performance of batch normalization was higher in the validation set for this reason we decided to proceed with this regularization technique. While more stable, this approach still suffered from huge variance, often resulting in worse than randomly initialized performance.

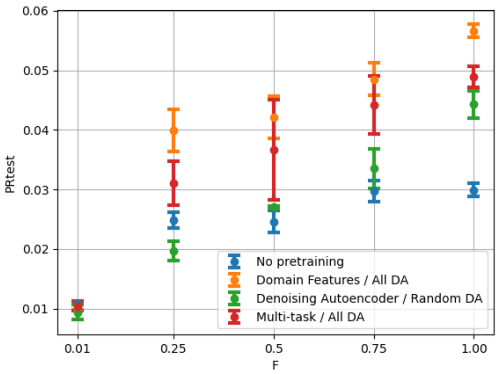

Figure 4: Average AUPRC scores for fatigue recognition under limited labeled data using Domain Features, Denoising Autoencoder and Multi-task pre-training tasks and temporal split.

C.2. Multi-task Learning

We conducted new experiments to assess the impact of weighting tasks within the multi-task learning setting. The AUPRC is 0.0502 (0.019) and 0.0541 (0.049) across test sets and validation sets and the AUROC is 0.7330 (0.0045) and 0.7265 (0.0083), which are comparable with the Multitask approach without weighting AUPRC - 0.0489 (0.0017) and 0.0578 (0.0033) as well as AUROC - 0.7323 (0.0007) and 0.7365 (0.0073) for test and validation set respectively. We believe this is because of the relatively modest disparity between the top-performing SSL tasks (domain features and denoising autoencoder).

Appendix D. Results

Table 5: Performance of data augmentations for fatigue recognition.

DA	**Parameter**	**AUPRC**$_{Test}$	**AUROC**$_{Test}$	**AUPRC**$_{Val}$	**AUROC**$_{Val}$
Noise	0.05	0.0314 (0.0018)	0.6973 (0.0008)	0.0367 (0.0003)	0.7056 (0.0011)
	0.1	0.0314 (0.0012)	0.6980 (0.0030)	0.0375 (0.0008)	0.7082 (0.0022)
	0.2	0.0378 (0.0003)	0.7140 (0.0038)	**0.0394 (0.0008)**	0.7128 (0.0022)
	0.4	**0.0409 (0.0018)**	**0.7142 (0.0019)**	0.0379 (0.0020)	**0.7134 (0.0016)**
	0.6	0.0375 (0.0033)	0.7083 (0.0013)	0.0350 (0.0009)	0.7095 (0.0048)
Mask	0.05	0.0347 (0.0013)	0.7044 (0.0011)	0.0401 (0.0014)	0.7119 (0.0017)
	0.1	0.0359 (0.0031)	0.7087 (0.0038)	0.0433 (0.0013)	0.7168 (0.0017)
	0.2	0.0364 (0.0030)	0.7217 (0.0000)	0.0438 (0.0020)	0.7242 (0.0005)
	0.4	0.0369 (0.0031)	**0.7219 (0.0028)**	0.0455 (0.0020)	0.7210 (0.0003)
	0.6	**0.0409 (0.0037)**	0.7151 (0.0037)	**0.0508 (0.0009)**	**0.7282 (0.0011)**
Reorder	Swap	**0.0322 (0.0015)**	**0.7051 (0.0044)**	**0.0370 (0.0007)**	**0.7100 (0.0027)**
	Permutation	0.0297 (0.0014)	0.6656 (0.0059)	0.0172 (0.0006)	0.6661 (0.0058)
	Baseline	0.0299 (0.0011)	0.6978 (0.0037)	0.0347 (0.0013)	0.7030 (0.0024)