Feature-Informed Self-Supervised Learning for Time Series Understanding

Parv Thacker*†
Indian Institute of Technology
Gandhinagar, Gujarat, India
parv.thacker@iitgn.ac.in

Ayush Shrivastava*
Indian Institute of Technology
Gandhinagar, Gujarat, India
shrivastavaayush@iitgn.ac.in

Nipun Batra
Indian Institute of Technology
Gandhinagar, Gujarat, India
nipun.batra@iitgn.ac.in

Abstract

Self-supervised learning (SSL) learns representations from unlabeled data by constructing a supervisory signal from the data itself. Most existing methods build this signal through augmentation, applying image-inspired transformations to the input and training the model to be invariant to them, often via a contrastive objective. In the time series domain, these include jittering, flipping, scaling, and temporal permutation. However, these transformations can produce signals that are no longer physically coherent in the time domain, encouraging invariances that discard information essential for downstream tasks. A second paradigm, masked modeling, reconstructs withheld segments of the signal, but this assumes missing values can be recovered from local context, an assumption that often breaks down in non-stationary domains such as biomedical sensing. We argue that time series SSL should instead capture the signal's intrinsic structure. We train an encoder to predict statistical, temporal, and spectral descriptors extracted from the raw signal as a direct supervisory target. This yields an interpretable objective that requires neither augmentation nor contrastive learning. We evaluate the approach across two sensing datasets, four backbone architectures, multiple label fractions, and a range of hyperparameters. Feature-based supervision consistently ranks among the top methods while requiring the least compute, and it is competitive with, or even outperforms, conventional SSL under frozen finetuning. These results suggest that time series descriptors are an effective yet underexplored source of self-supervision.

CCS Concepts

• **Computing methodologies** → **Learning paradigms**; *Neural networks*; *Learning latent representations.*

ACM Reference Format:
Parv Thacker, Ayush Shrivastava, and Nipun Batra. 2026. Feature-Informed Self-Supervised Learning for Time Series Understanding. In *Proceedings of Mining and Learning from Time Series, SIGKDD International Conference on Knowledge Discovery and Data Mining 2026 (KDD MILETS Workshop '26).* ACM, New York, NY, USA, 8 pages. 10.1145/nnnnnnn.nnnnnnn

*Both authors contributed equally to this research.
†Corresponding Author

KDD MILETS Workshop '26, Jeju, Korea

ACM ISBN 978-1-4503-XXXX-X/2018/06
10.1145/nnnnnnn.nnnnnnn

1 Introduction

Time series data are prevalent across many domains, including healthcare [24], remote sensing [10], industrial and household monitoring [18], finance [32], among others [17]. Recent advances in deep learning and sensing techniques have enabled large-scale signal collection across various systems [30].

Despite the abundant availability of raw data, the volume of signals frequently exceeds what can be practically labeled by human annotators [9]. Hence, supervised learning approaches are often unable to utilize the majority of available time series data, motivating the development of self-supervised learning (SSL) methods capable of learning meaningful representations directly from unlabeled signals [6, 12]. Augmentation-based SSL methods, such as SimCLR [4], Barlow Twins [31], and BYOL [11], have shown strong performance on computer vision tasks [16] and have been utilized extensively for time series tasks [26]. These methods generate transformed views of the input samples and train the representations such that representations of related views remain similar, whereas those of unrelated views remain separated [12].

In computer vision, approaches such as flips, masks, crops, color augmentations, and geometric transformations are used to create consistent yet diverse views [4]. On the other hand, in time series, we use jittering, scaling, masking, signal reversals, etc., for augmentations [15, 25].

While these augmentations provide variance for SSL tasks, they may not always be semantically accurate to the real-world signals [28]. Jittering, for instance, forces invariance to noise, effectively training the encoder to suppress high-frequency variations that may encode meaningful events. In applications such as physiological, industrial, or environmental monitoring, even small signal fluctuations can correspond to meaningful events, anomalies, or system states [25].

Time series data, unlike images, are often causal in nature and hence encode temporal progression and frequency structure, as well as physical constraints or amplitude-dependent information. In many applications, this information (ordering, continuity, etc.) is crucial to the meaning of the signal or the task itself. For example, reversing the signals, scaling, or aggressively permuting a signal might destroy the necessary signals or may create physically impossible samples. Hence, the derived augmentation strategies may distort the meaning of the signals or miss crucial information during the representation learning process. Therefore, there is a growing need for self-supervised learning techniques that capture the semantics of time series data.

Handcrafted features, on the other hand, have played a crucial role in classical time series analysis. Frameworks such as TSFEL [2] and TSFRESH [5] provide a rich collection of temporal, spectral,

and statistical features that encode the meaningful properties of sequential signals. While modern DL techniques replace handcrafted features, these features still encode valuable low-level information.

In this paper, we investigate a TSFEL-based SSL framework for time series analysis. We examine whether handcrafted time series descriptors can improve not only performance but also robustness to optimization choices and compute efficiency across various tasks. We perform a large-scale evaluation across multiple datasets, architectures, and SSL baselines, and compare them via a normalized rank-based evaluation framework.

Across 1600 experiments spanning two wearable sensing benchmarks (HHAR activity recognition [3] and PPG-Dalia heart rate estimation [21]), four backbone architectures (ResNet-18 [14], MLP, MLP-Mixer [23], and PatchTST [19]), multiple label fractions, frozen and unfrozen finetuning regimes, and a range of hyperparameter settings, we evaluate feature-based self-supervised learning against six conventional SSL baselines. Methods are compared via a rank-based evaluation framework that aggregates performance across all configurations and avoids best-case reporting. We find that feature-based supervision consistently ranks among the strongest-performing approaches while incurring the lowest computational cost and remains stable across finetuning regimes and hyperparameter choices. These findings suggest that utilizing domain-informed time series features in self-supervised objectives offers a robust and efficient alternative to augmentation-based representation learning methods

2 Related Work

Using handcrafted signal properties as self-supervised targets has been explored in various modalities. In vision, tasks involved predicting objectives such as image colorization [33] or spatial patch arrangements [20]. MaskFeat [27] showed that predicting HOG features, a handcrafted descriptor, outperforms pixel-level reconstruction as a masked pretraining objective. data2vec [1] generalized this idea across speech, vision, and language modalities, using contextualized representations rather than raw inputs as prediction targets. Feature-based pretraining has been largely missing from the SSL literature for time series, and TSFEL [2] and TSFRESH [5] have demonstrated that statistical, temporal, and spectral descriptors capture task-relevant signal descriptions.

A major challenge in SSL research is that results are highly sensitive to the evaluation protocol. Prior work has shown that rankings between SSL methods change substantially depending on whether the encoder is finetuned, the amount of labeled data available, and the choice of hyperparameters [8], cautioning against conclusions drawn from a single evaluation setting. For time series, Haresamudram et al. [13] conducted a large-scale empirical study evaluating wearable HAR in various sensor positions, activity types, and label availability. The authors showed that method behavior changes significantly between settings, motivating more systematic comparison protocols. Motivated by the issues, rather than reporting performance under a single favorable setting, we aggregate ranks across many experimental configurations to measure performance consistency.

We address these gaps by introducing feature-targeted supervision as a simple, augmentation-free self-supervised alternative for time series representation learning, and we evaluate it across a wide range of backbones, fine-tuning regimes, hyperparameters, and data fractions using a rank-based comparison framework that aggregates performance across all configurations to avoid best-case reporting.

3 Methodology

3.1 Overview of the Proposed Framework

We propose a self-supervised learning (SSL) framework for time series representation learning based on predicting handcrafted multi-domain time series descriptors. Rather than using supervisory signals generated by contrastive objectives, signal augmentations, etc., the proposed approach uses time series features as targets during pretraining.

The proposed framework consists of three stages. First, the fixed-length time series windows are processed using the TSFEL feature extraction library to generate a set of statistical, temporal, and spectral descriptors. These features are then used as self-supervised learning targets. Second, an encoder is trained to predict the extracted handcrafted feature representations directly from the raw input signal.

3.2 Time Series Feature-Based Self-Supervised Learning

Following the predictive SSL paradigm established by MaskFeat and data2vec (Section 2), the supervisory signal is derived directly from the input, requiring no human annotation. The core idea of the proposed framework is to use interpretable multi-domain time series descriptors as supervisory signals for representation learning. Rather than relying on techniques such as masking, contrastive objectives, or reconstruction losses, we train a neural network to predict a set of time series characteristics directly from the input.

Given an input time series segment $\mathbf{X} \in \mathbb{R}^{T \times C}$, where T is the window length and C is the number of channels, an engineered time series feature extractor obtains a target feature vector $\mathbf{f}(\mathbf{X}) \in \mathbb{R}^{d}$. The feature extractor is applied independently to each channel of $\mathbf{X}$, and the resulting per-channel descriptors are concatenated to form the target vector, where $d = n_f \times C$ and n_f is the number of selected features per channel. The extracted descriptors comprise descriptors from all three: statistical, temporal, and spectral characteristics of the signal. Statistical descriptors summarize the distributional properties of a dataset, such as the mean, variance, skewness, etc. Temporal descriptors capture signal behavior through features such as zero-crossing rate, autocorrelation, and temporal entropy. Spectral descriptors characterize frequency-domain behavior using features such as the spectral centroid, dominant frequency, and spectral entropy.

To evaluate how the number of features used affects the models, multiple feature subsets were evaluated, consisting of 15, 30, 45, and 90 signal descriptors, each selected from an equal number of descriptors across domains. These were chosen to span from compact representations to richer descriptor sets, while remaining divisible by three to ensure equal domain coverage across statistical, temporal, and spectral descriptors.

During self-supervised pretraining, the encoder network, $E(\cdot)$, maps the input segment to a latent representation $\mathbf{z} \in \mathbb{R}^{D}$,

Table 1: Experimental grid structure: permutation factors and configurations per backbone–method–dataset pair.

Group	Factor	Values	Configs
LR Robustness	Finetuning regime	Frozen, Unfrozen	12
	Pretraining LR	10^{-3}, 10^{-2}	
	Finetuning LR	10^{-5}, 10^{-4}, 10^{-3}	
Label Scarcity	Label fraction	5%, 100%	4
	Finetuning regime	Frozen, Unfrozen	
Head Augmentation	Prediction head	Linear, MLP	4
	Pooling strategy	Mean, Max	
Per backbone–method–dataset pair			20
Total (4 backbones × 10 methods × 2 datasets × 20 configs)			1600

$$\mathbf{z} = E(\mathbf{X}), \quad \mathbf{z} \in \mathbb{R}^D \tag{1}$$

A lightweight prediction head, $P(\cdot)$, then projects the latent representation into the time series feature space,

$$\hat{\mathbf{f}}(\mathbf{X}) = P(\mathbf{z}), \quad \hat{\mathbf{f}} \in \mathbb{R}^d \tag{2}$$

The network is trained to minimize the difference between the feature vector $\hat{\mathbf{f}}(\mathbf{X})$ and the target feature vector $\mathbf{f}(\mathbf{X})$. Feature descriptors are standardized at the dataset level before training. The self-supervised objective is defined as the mean squared error (MSE) between predicted and target feature values,

$$L_{\text{feat}} = \frac{1}{d}\sum_{i=1}^{d}(\hat{f}_i - f_i)^2 \tag{3}$$

where $d = n_f \times C$ denotes the dimensionality of the target feature vector, n_f is the number of selected features per channel, and C is the number of input channels.

The exact feature configurations for each subset are provided in the project repository for reproducibility.

3.3 Backbone Architectures

To evaluate the generality of the proposed SSL architecture, the experiments were conducted across multiple backbones, including multilayer perceptron (MLP), convolution, and transformer-based architectures. The convolution architecture consists of a 1D ResNet-18 encoder [14]. To evaluate the transformer-based architecture, we conducted experiments using PatchTST [19], which represents time series signals as sequences of temporal patches. Two MLP-based architectures were additionally considered. The first is a fully connected encoder that provides a lightweight baseline. The second is an MLP-Mixer [23], which combines token-mixing and channel-mixing operations to learn temporal features.

For all backbones, the same time series feature prediction pretraining was employed, alongside the conventional SSL baselines during pretraining.

3.4 Experimental Setup

Experiments were conducted on two wearable sensing benchmarks, one for classification and one for regression. Together, these cover both discrete and continuous prediction targets, two distinct sensor modalities (inertial and photoplethysmography), and multiple subjects and device positions, providing a representative evaluation scope within the wearable sensing domain. Human activity recognition experiments were performed using the Heterogeneity Human Activity Recognition (HHAR) dataset [3], containing inertial sensor recordings collected during activities such as walking, sitting, standing, stair climbing, and cycling. The downstream task is multi-class activity classification.

Regression experiments were conducted using the PPG-Dalia dataset [21], a wearable physiological sensing benchmark that includes photoplethysmography (PPG), accelerometer, and additional sensor modalities. The downstream task is continuous heart-rate estimation. For this work, only PPG and accelerometer channels were retained, while other sensor modalities were excluded to focus on signals commonly available in consumer wearable devices.

Signals were segmented into fixed-length windows according to the predefined evaluation protocols of each benchmark [22] and standardized per channel using training-set statistics. Predefined train, validation, and test splits were used for the experiments. To evaluate the effects of label availability, the experiments were repeated with multiple training fractions generated by random sampling. For downstream adaptation, a common pipeline was implemented across all backbones and methods, with a prediction head and two fine-tuning regimes: one with the encoder frozen and only the prediction head trained, and another with all network parameters fine-tuned.

We compared the proposed framework against six representative self-supervised baselines spanning contrastive, redundancy-reduction, and masked-modeling paradigms. SimCLR [4] and BYOL [11] represent contrastive approaches, and Barlow Twins [31] represents redundancy reduction. TS2Vec [29] serves as a general time series representation learning baseline. For masked modeling, we included SimMTM [7] and MPM [19], the latter a PatchTST-based masked prediction method.

Table 2: Summary results on the PPG-Dalia heart-rate estimation task aggregated across all experimental units. Average Rank (↓) is the mean rank across experimental configurations, where each method is ranked from 1 (best) to 10 (worst). Mean Absolute Error (MAE, ↓) is reported in beats per minute. Best results per condition are marked in bold. *The proposed approach, TSFEL15 achieves the best average rank (3.95) and TSFEL30 the lowest MAE (11.58 BPM), with extracted time series features consistently outperforming all self-supervised methods.*

Method	Avg Rank ↓	SD	Avg MAE ↓	SD
Barlow	5.30	3.24	11.59	3.59
BYOL	5.35	3.66	17.06	20.31
MPM	6.15	3.72	14.11	10.73
SimCLR	5.97	3.01	21.13	25.62
SimMTM	6.75	3.53	21.27	21.31
TS2Vec	7.23	3.18	13.07	3.21
TSFEL15	**3.95**	2.45	11.99	6.37
TSFEL30	4.15	3.06	**11.58**	3.65
TSFEL45	6.62	3.76	14.41	16.16
TSFEL90	5.88	3.76	15.15	18.52

Table 3: Summary results on the HHAR activity recognition task aggregated across all experimental units. Average Rank (↓) is the mean rank across experimental configurations, where each method is ranked from 1 (best) to 10 (worst). Macro F1-score (↑) is reported as the primary classification metric. Best results per condition are marked in bold. *The proposed approach, TSFEL45 achieves both the best average rank (3.76) and highest F1-score (0.79), with self-supervised methods performing notably worse — TS2Vec, SimMTM, and MPM all rank in the bottom three.*

Method	Avg Rank ↓	SD	Avg F1 ↑	SD
Barlow	4.43	3.52	0.78	0.17
BYOL	5.11	2.93	0.76	0.21
MPM	8.55	2.19	0.56	0.31
SimCLR	4.43	3.99	0.75	0.19
SimMTM	8.00	1.54	0.72	0.17
TS2Vec	8.67	0.98	0.69	0.21
TSFEL15	4.64	2.26	0.78	0.15
TSFEL30	4.56	2.66	0.78	0.15
TSFEL45	**3.76**	2.57	**0.79**	0.15
TSFEL90	4.37	1.66	0.78	0.16

3.5 Evaluation Protocol

Each self-supervised learning method was evaluated across multiple combinations of datasets, backbone architectures, training fractions, encoder adaptation strategies, and optimization settings.

Experiments were conducted using ResNet18 [14], MLP, MLP-Mixer [23], and PatchTST [19] backbones. The full experimental grid was structured across three groups as detailed in Table 1, resulting in a total of 1600 experiments covering both tasks.

Performance was evaluated using task-specific metrics. For classification tasks, we report the macro F1-score, while for regression tasks, we report the mean absolute error (MAE). To ensure fair comparison, methods were ranked only within the experimental settings.

An experimental setting was defined by a unique combination of dataset, backbone architecture, training fraction, encoder adaptation strategy (frozen or unfrozen), along with the hyperparameters used (learning rates, batch sizes, etc.). Within each experimental setting, methods were ordered according to the primary evaluation metric, with Rank 1 assigned to the best-performing method, Rank 2 to the second-best method, and so on. For classification tasks, rankings were determined using the macro F1-score, whereas for regression tasks, rankings were determined using the MAE.

The reported average rank corresponds to the mean rank achieved by a method across all experimental settings. The lower average ranks indicate stronger and more consistent performance across the evaluation conditions. We also report rank standard deviation to capture performance stability.

The codebase and experimental framework are available https://github.com/parvthacker/SSL-for-Time-Series.

4 Results

4.1 Regression Task

Table 2 summarizes aggregate performance on the PPG-Dalia Heart Rate estimation experiments. Across the evaluated configurations, time series feature-based representations achieved equal or better performance than traditional SSL baselines in the majority of cases. In particular, TSFEL-15 and TSFEL-30 achieved the best ranks among the evaluated frameworks, and increasing the number of time series features slightly degraded performance.

The rank distribution shown in Fig. 1a shows these trends. Although several baseline SSL techniques achieve better performance in specific configurations, the time series feature-based self-supervised learning method consistently performs well across various configurations. The distributions suggest that feature-based supervision techniques maintain their performance across the evaluated optimization settings and pretraining configurations, with no increase in tuning effort.

While no method universally dominated all configurations, the time series feature-based self-supervised learning method demonstrated strong aggregate performance while requiring the least computational effort among the evaluated methods.

4.2 Classification Task

Table 3 summarizes aggregate performance on the HHAR activity recognition benchmark. Across the evaluated HHAR activity recognition experiments, time series feature-based SSL consistently ranked among the strongest performing approaches. Feature-based supervision occupied four of the six highest-ranked positions in the aggregate ranking, alongside SimCLR and Barlow Twins, while

Table 4: Effect of frozen and unfrozen fine-tuning on HHAR activity recognition performance, aggregated across all experimental configurations. Average Rank (↓) is the mean rank across experimental configurations, where each method is ranked from 1 (best) to 10 (worst). Macro F1-score (↑) is reported as the primary classification metric. *While unfrozen fine tuning substantially improves performance for weaker self-supervised methods, TSFEL features remain competitive under both conditions, suggesting that learned representation capture important time series structure.*

	Frozen		Unfrozen	
Method	**Rank ↓**	**F1 ↑**	**Rank ↓**	**F1 ↑**
Barlow	4.96	0.74	3.90	0.82
BYOL	4.02	0.72	6.21	0.79
MPM	9.67	0.33	7.44	0.79
SimCLR	4.75	0.70	4.10	0.81
SimMTM	7.83	0.65	8.17	0.80
TS2Vec	9.00	0.58	8.33	0.80
TSFEL15	4.05	0.74	5.23	0.81
TSFEL30	4.58	0.74	4.53	0.82
TSFEL45	2.68	0.77	4.83	0.81
TSFEL90	5.10	0.74	3.64	0.82

Table 5: Effect of frozen and unfrozen fine-tuning on PPG-Dalia heart rate estimation performance, aggregated across all experimental configurations. Average Rank (↓) is the mean rank across experimental configurations, where each method is ranked from 1 (best) to 10 (worst). MAE (↓) is reported in beats per minute. *Under frozen evaluation, TSFEL15 achieves the best rank, with baseline methods performing notably worse. Unfrozen fine-tuning improves TS2Vec and SimMTM, yet TSFEL30 retains the best MAE (7.11 BPM), suggesting that learned representation capture important time series structure.*

	Frozen		Unfrozen	
Method	**Rank ↓**	**MAE ↓**	**Rank ↓**	**MAE ↓**
Barlow	4.38	12.62	6.23	10.26
BYOL	4.88	20.92	5.81	13.21
MPM	6.80	16.82	5.51	11.40
SimCLR	5.56	26.07	6.38	16.19
SimMTM	7.60	23.62	5.91	18.93
TS2Vec	7.81	14.22	6.65	11.91
TSFEL15	3.46	12.68	4.45	11.30
TSFEL30	4.45	13.09	3.85	10.07
TSFEL45	6.59	21.34	6.65	7.47
TSFEL90	5.91	23.36	5.85	6.93

consistently achieving a better F1 score than SimCLR and Barlow Twins, by margins of 0.01 to 0.03. TSFEL variants achieved a consistent average F1 score of 0.78-0.79.

Unlike in Regression experiments, the separation between the methods is less pronounced. Traditional SSL approaches, such as SimCLR and Barlow Twins, achieved aggregate performance comparable to that of the time series feature-based self-supervised learning method, but showed more extreme ranks: performing well in some cases but ranking last in many configurations. Time series feature-based self-supervised learning methods displayed greater consistency in their performance across the evaluated backbones and training fractions. This observation is supported by Fig. 1b, where the feature-based supervision variants remain concentrated in lower-ranking regions, while the competing method displayed broader performance distribution.

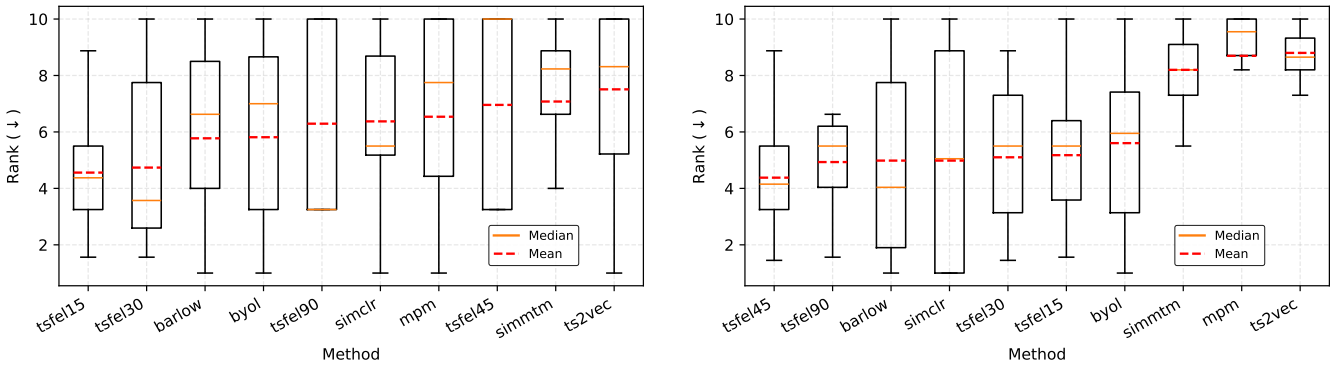
This behavior suggests that time series feature-based self-supervised learning is less dependent on hyperparameter choices and continues to perform consistently across the evaluated configurations.

4.3 Ablation Study

4.3.1 Effect of Finetuning Strategy. To better understand the behavior of time series representations under different downstream configurations, we evaluated all models under frozen and unfrozen finetuning regimes. Across both benchmarks, feature-guided representation learning methods show a strong performance in frozen finetuning. On classification tasks, as shown in Table 4, TSFEL45 and TSFEL90 achieved the lowest average rank, frozen (2.68) and unfrozen (3.64), respectively, whereas the lowest achieved by conventional SSL methods was 4.02 and 3.90 for frozen and unfrozen, respectively. Notably, the average performance of time series feature-based self-supervised learning methods improves very little between regimes, indicating that the representations learned are very informative from the start. In conventional SSL methods, on the other hand, we see a large jump when the encoders are finetuned, particularly with methods like Barlow Twins, where the average rank drops by more than 1, and F1 increases by 0.08. These results suggest that, on the evaluated benchmarks, conventional SSL methods benefit more from encoder fine-tuning, whereas representations learned through handcrafted time series feature supervision appear to retain a larger fraction of their downstream utility prior to adaptation.

A similar trend is observed in Table 5, showing the behavior on the regression task. Under the frozen regimen, time series approaches show the strongest aggregate ranks, with TSFEL15 achieving an average rank of 3.46. After unfrozen finetuning, the performance gap narrowed, in line with observations during the classification task, and SimMTM became competitive. *The strong performance under frozen evaluations and competitive performance in unfrozen cases suggest that such time series descriptors capture information that remains useful for downstream tasks on the evaluated benchmarks.*

4.3.2 Effect of Extracted Time Series Feature Set Size. To understand the sensitivity of time series feature targeted pretraining to the number of features selected, we evaluated four feature sets of sizes 15, 30, 45, and 90.



(a) PPG-Dalia Dataset **(b) HHAR Dataset**

Figure 1: Rank distributions for (a) PPG-Dalia and (b) HHAR datasets, where each method is ranked from 1 (best) to 10 (worst) per experimental configuration. The solid and dashed lines denote the median and mean rank, respectively. *The proposed approach, TSFEL15 and 30 cluster toward the lower end of the rank distribution on both datasets, with lower means and tighter spreads than most SSL baselines, indicating more consistent performance across experimental configurations.*

Across both benchmarks, the feature set size influence was not strictly monotonically increasing. Instead, the results suggest an inverted-U-shaped trend, in which intermediate feature set sizes achieved the strongest aggregate performance. TSFEL45 ranked highest on HHAR, while TSFEL15 and TSFEL30 led on PPG-Dalia. Larger feature sets showed degraded performance on both benchmarks.

These observations suggest that there exists a tradeoff between the richness of representations and feature redundancy for the evaluated feature sets. Smaller feature sets may fail to capture important information, whereas larger ones may introduce redundancies that complicate learning. However, while degradation was observed beyond a certain feature set size, the difference across all feature set sizes was modest relative to the performance differences between SSL methods.

4.3.3 Effect of Backbone Architecture. We examined per-backbone-normalized ranks to assess how feature-based supervision generalizes across models. On PPG-Dalia, TSFEL variants ranked first across all four backbones versus the closest competitor, with normalized ranks of 2.0 vs. 4.2, 3.4 vs. 5.8, 3.2 vs. 4.6, and 3.6 vs. 3.8 across MLP, MLP-Mixer, ResNet-18, and PatchTST, respectively, indicating that the gains are not architecture-specific. On HHAR, the results are more varied. SimCLR significantly outperformed TSFEL on MLP-Mixer (1.8 vs. 2.8), suggesting that contrastive objectives are more effective for the task evaluated with MLP-Mixer. On PatchTST, MPM was notably more competitive for regression (0.40 vs. 0.38 for TSFEL) than on any other backbone. Overall, feature-based supervision is broadly robust on the evaluated datasets, but its relative advantage is strongest on regression tasks, and the advantage reduces when a baseline is aligned with the backbone.

4.4 Computational Efficiency Analysis

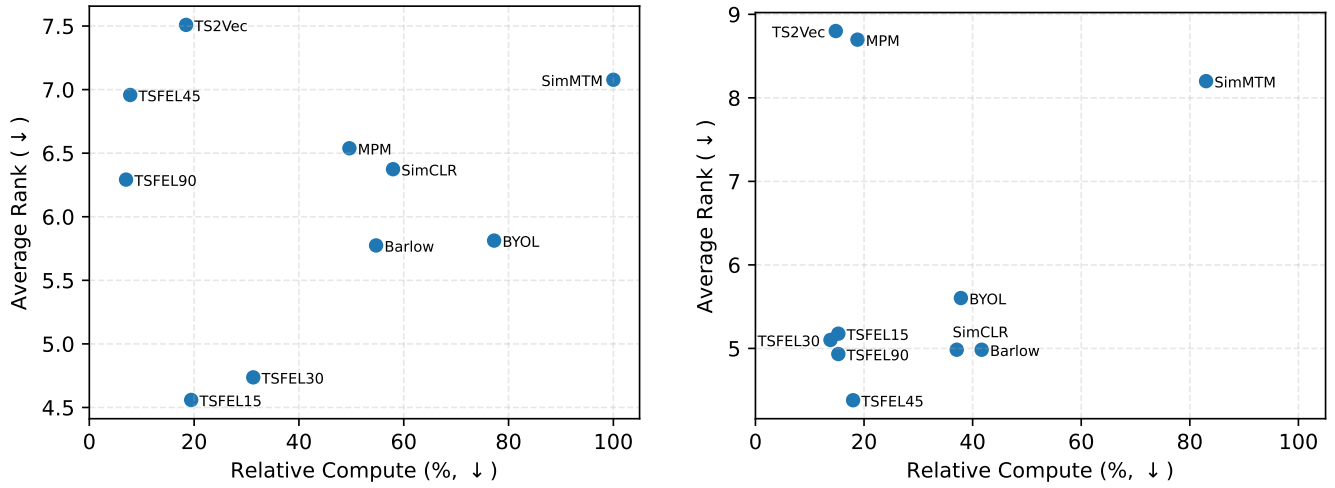
Fig. 2a and 2b show the relationship between average normalized rank (the average rank divided by 10) and average computational effort required to get the best performance of the method, for PPG-Dalia and HHAR, respectively. The computational effort for training is proportional to the number of network passes required for training the model.

Across the evaluated benchmarks, time-series-based approaches consistently occupied the most favorable, lower left side region of the plot, corresponding to strong aggregate performance and low computational cost. In particular, for Regression, TSFEL15 and TSFEL30 achieved strong aggregate performance while requiring the least computing. This trend was followed by TSFEL45 in the classification case. Additionally, an increase in computation does not necessarily translate into better performance. In particular, SimMTM often required higher computational effort, but rarely performed competitively with time-series-based methods.

Overall, the computational efficiency analysis shows that time series feature-based self-supervised learning provides a favorable balance between performance and efficiency on the evaluated wearable sensing benchmarks.

5 GenAI Disclosure

The authors used generative AI tools (large language models) to assist with code implementation and refactoring during the development of the experimental framework. This assistance was limited to localized contributions such as improving or generating individual functions, classes, and code snippets, and to suggesting incremental changes to existing code. Generative AI tools were also used to a limited extent for language editing of the manuscript, such as improving clarity, grammar, and phrasing in individual sentences, without altering the technical content. All AI-assisted code was reviewed, tested, and verified by the authors to ensure it met the intended specification and produced correct results, and all AI-assisted text was checked by the authors for accuracy and consistency with the reported experiments. No generative AI tools were used to design experiments, generate or analyze data, or produce the reported results. The study design, methodology, experimental



(a) PPG-Dalia Dataset **(b) HHAR Dataset**

Figure 2: Average rank (1 = best, 10 = worst) vs. relative compute cost for each method on (a) PPG-Dalia and (b) HHAR-ACT datasets. Methods in the lower-left are preferable, achieving better rank at lower computational cost.
Key takeaway: TSFEL15 and TSFEL30 occupy the lower-left region on both datasets, offering strong performance at a fraction of the compute required by SSL baselines. In contrast, SimMTM incurs a high computational cost and ranks poorly across both benchmarks.

analysis, and scientific conclusions are entirely the work of the authors.

6 Limitations

Our study has several limitations that bound its conclusions. The method is not free of inductive bias, only of augmentation: it replaces contrastive transformations with a fixed set of TSFEL descriptors, and the feature-set-size ablation shows this choice is consequential, with performance following an inverted-U trend and no principled criterion for selecting the subset. The evaluation also covers only two wearable sensing benchmarks within a single modality family of short fixed-length windows, leaving forecasting, long-context dependencies, and the non-stationary clinical signals that motivate the method untested. Finally, the analysis is empirical and does not isolate the underlying mechanism, as we neither disentangle the statistical, temporal, and spectral feature families nor probe which signal properties the encoder learns, leaving the central hypothesis supported by performance but not by direct analysis.

7 Future Work

Several directions follow from these limitations. The most immediate is to extend evaluation beyond short-window wearable sensing to forecasting, anomaly detection, and longer non-stationary clinical recordings such as polysomnography or electrocardiography, where the assumptions behind both augmentation and masked modeling are most strained. A second is to replace the manually selected descriptor sets with a principled feature-selection or feature-weighting procedure that recovers the inverted-U behavior in a data-driven manner and removes the dependence on a fixed subset size. A third is a mechanistic analysis that ablates the statistical, temporal, and spectral targets independently to test whether the gains stem from spectral structure, temporal ordering, or distributional summaries. A fourth is to investigate whether combining feature prediction with a contrastive or reconstruction term yields complementary gains, since empirical results suggest that the two paradigms encode different aspects of the signal, with augmentation-based SSL methods occasionally outperforming feature-based supervision.

8 Conclusion

We presented a feature-informed self-supervised learning framework that predicts handcrafted statistical, temporal, and spectral descriptors directly from the raw signal, requiring neither augmentation nor contrastive objectives. Across 1600 experiments spanning two wearable benchmarks, four backbones, multiple label fractions, two fine-tuning regimes, and a range of hyperparameters, evaluated through a rank-based protocol that avoids best-case reporting, feature-targeted supervision consistently ranked among the strongest methods while requiring the least compute. The advantage was most pronounced on regression and under frozen evaluation, where the representations retained most of their downstream utility without adaptation, while on classification the method matched the best contrastive baselines with tighter rank distributions. These results indicate that time series descriptors are an effective and underexplored source of self-supervision on the evaluated benchmarks, complementary to augmentation and reconstruction based approaches rather than universally superior, and a promising basis for the extensions outlined above.

References

[1] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. 2022. data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language. In *Proceedings of the 39th International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 1298–1312.

[2] Marília Barandas, Duarte Folgado, Letícia Fernandes, Sara Santos, Mariana Abreu, Patrícia Bota, Hui Liu, Tanja Schultz, and Hugo Gamboa. 2020. TSFEL: Time Series Feature Extraction Library. *SoftwareX* 11 (2020), 100456. doi:10.1016/j.softx.2020.100456

[3] Henrik Blunck, Sourav Bhattacharya, Thor Prentow, Mikkel Kjærgaard, and Anind Dey. 2015. Heterogeneity Activity Recognition. UCI Machine Learning Repository. DOI: 10.24432/C5689X. https://doi.org/10.24432/C5689X

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*. 1597–1607.

[5] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W. Kempa-Liehr. 2018. Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing* 307 (2018), 72–77. doi:j.neucom.2018.03.067

[6] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. 2015. Unsupervised Visual Representation Learning by Context Prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[7] Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. 2023. SimMTM: A Simple Pre-Training Framework for Masked Time-Series Modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 36.

[8] Linus Ericsson, Henry Gouk, and Timothy Hospedales. 2022. Why Do Self-Supervised Models Transfer? On the Impact of Invariance on Downstream Tasks. arXiv:2111.11398 [cs.CV] https://arxiv.org/abs/2111.11398

[9] Teodor Fredriksson, David Issa Mattos, Jan Bosch, and Helena Holmström Olsson. 2020. Data Labeling: An Empirical Investigation into Industrial Challenges and Mitigation Strategies. In *Product-Focused Software Process Improvement*, Maurizio Morisio, Marco Torchiano, and Andreas Jedlitschka (Eds.). Springer International Publishing, Cham, 202–216.

[10] Yingchun Fu, Zhe Zhu, Liangyun Liu, Wenfeng Zhan, Tao He, Huanfeng Shen, Jun Zhao, Yongxue Liu, Hongsheng Zhang, Zihan Liu, Yufei Xue, and Zurui Ao. 2024. Remote Sensing Time Series Analysis: A Review of Data and Applications. *Journal of Remote Sensing* 4 (2024), 0285. arXiv:https://spj.science.org/doi/pdf/10.34133/remotesensing.0285 doi:10.34133/remotesensing.0285

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 21271–21284.

[12] Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. 2024. A Survey on Self-supervised Learning: Algorithms, Applications, and Future Trends. arXiv:2301.05712 [cs.LG] https://arxiv.org/abs/2301.05712

[13] Harish Haresamudram, Irfan Essa, and Thomas Plötz. 2022. Assessing the State of Self-Supervised Human Activity Recognition Using Wearables. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 3, Article 116 (Sept. 2022), 47 pages. doi:10.1145/3550299

[14] 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 (CVPR)*. 770–778.

[15] Brian Kenji Iwana and Seiichi Uchida. 2021. An empirical survey of data augmentation for time series classification with neural networks. *PLOS ONE* 16, 7 (07 2021), 1–32. doi:10.1371/journal.pone.0254841

[16] Mingu Kang, Heon Song, Seonwook Park, Donggeun Yoo, and Sérgio Pereira. 2023. Benchmarking Self-Supervised Learning on Diverse Pathology Datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 3344–3354.

[17] Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. 2024. Foundation Models for Time Series Analysis: A Tutorial and Survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Barcelona, Spain) *(KDD '24)*. Association for Computing Machinery, New York, NY, USA, 6555–6565. doi:10.1145/3637528.3671451

[18] Yi Liu, Sahil Garg, Jiangtian Nie, Yang Zhang, Zehui Xiong, Jiawen Kang, and M. Shamim Hossain. 2021. Deep Anomaly Detection for Time-Series Data in Industrial IoT: A Communication-Efficient On-Device Federated Learning Approach. *IEEE Internet of Things Journal* 8, 8 (2021), 6348–6358. doi:10.1109/JIOT.2020.3011726

[19] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=Jbdc0vTOcol

[20] Mehdi Noroozi and Paolo Favaro. 2016. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. doi:10.1007/978-3-319-46466-4_5

[21] Attila Reiss, Ina Indlekofer, and Philip Schmidt. 2019. PPG-DaLiA. UCI Machine Learning Repository. DOI: 10.24432/C53890. https://doi.org/10.24432/C53890

[22] Attila Reiss, Ina Indlekofer, Philip Schmidt, and Kristof Van Laerhoven. 2019. Deep PPG: Large-Scale Heart Rate Estimation with Convolutional Neural Networks. *Sensors* 19, 14 (2019). doi:10.3390/s19143079

[23] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Peter Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. 2021. MLP-Mixer: An all-MLP Architecture for Vision. In *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.). https://openreview.net/forum?id=EI2KOXKdnP

[24] Latchezar Tomov, Lyubomir Chervenkov, Dimitrina Georgieva Miteva, Hristiana Batselova, and Tsvetelina Velikova. 2023. Applications of time series analysis in epidemiology: Literature review and our experience during COVID-19 pandemic. *World Journal of Clinical Cases* 11, 29 (2023), 6974–6983. doi:10.12998/wjcc.v11.i29.6974

[25] Terry T. Um, Franz M. J. Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 2017. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction* (Glasgow, UK) *(ICMI '17)*. Association for Computing Machinery, New York, NY, USA, 216–220. doi:10.1145/3136755.3136817

[26] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Chen Wang, Mingsheng Long, and Jianmin Wang. 2026. Deep Time Series Models: A Comprehensive Survey and Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2026), 1–20. doi:10.1109/TPAMI.2026.3690845

[27] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. 2022. Masked Feature Prediction for Self-Supervised Visual Pre-Training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 14668–14678.

[28] Tete Xiao, Xiaolong Wang, Alexei A Efros, and Trevor Darrell. 2021. What Should Not Be Contrastive in Contrastive Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=CZ8Y3NzuVzO

[29] 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.

[30] Zahra Zamanzadeh Darban, Geoffrey I. Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. 2024. Deep Learning for Time Series Anomaly Detection: A Survey. *ACM Comput. Surv.* 57, 1, Article 15 (Oct. 2024), 42 pages. doi:10.1145/3691338

[31] Jure Zbontar, Li Jing, Jean Ponce, Yann LeCun, and Stéphane Deny. 2021. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*. 12310–12320.

[32] Cheng Zhang, Nilam Nur Amir Sjarif, and Roslina Ibrahim. 2024. Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022. *WIREs Data Mining and Knowledge Discovery* 14, 1 (2024), e1519. arXiv:https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1519 doi:10.1002/widm.1519

[33] Richard Zhang, Phillip Isola, and Alexei A Efros. 2016. Colorful Image Colorization. In *ECCV*.