

Deep Semi-supervised Learning for Time Series Classification

Jann Goschenhofer^{1,2}, Rasmus Hvingelby², David Ruegamer¹, Janek Thomas¹, Moritz Wagner², Bernd Bischl^{1,2}

LMU Munich, Munich, Germany¹

Fraunhofer Institute for Integrated Circuits (IIS), Erlangen, Germany²

jann.goschenhofer@stat.uni-muenchen.de

Abstract—While deep semi-supervised learning has gained much attention in computer vision, limited research exists on its applicability in the time series domain. In this work, we investigate the transferability of state-of-the-art deep semi-supervised models from image to time series classification. We discuss the necessary model adaptations, in particular an appropriate model backbone architecture and the use of tailored data augmentation strategies. Based on these adaptations, we explore the potential of deep semi-supervised learning in the context of time series classification by evaluating our methods on large public time series classification problems with varying amounts of labeled samples. We perform extensive comparisons under a decidedly realistic and appropriate evaluation scheme with a unified reimplementation of all algorithms considered, which is yet lacking in the field. We find that these transferred semi-supervised models show significant performance gains over strong supervised, semi-supervised and self-supervised alternatives, especially for scenarios with very few labeled samples.

Index Terms—Semi-supervised Learning, Time Series Classification, Data Augmentation

I. INTRODUCTION

Time series classification (TSC) spans many real-world applications in domains from healthcare [1] over cybersecurity [2] to manufacturing [3]. Several algorithms for TSC have been proposed over the years [4] [5].

In many real-world scenarios, time series data can be collected easily, but acquiring labels for this data is costly. For instance, in disease monitoring, sensor data are collected with low effort but the labelling of this data requires time-consuming work by medical experts [6]. Semi-supervised learning (SSL) addresses this by leveraging large amounts of unlabeled data in combination with a small amount of labeled data when training machine learning (ML) models.

Especially in computer vision, the advances in deep neural networks and the promised label efficiency of SSL have lead to the introduction of several innovative approaches for image data [7]. While there is much work on classical semi-supervised models for TSC, research on the use of neural network-based SSL algorithms for TSC is still limited.

This motivates our main research question that we approach holistically in this work: *Can we transfer well established deep semi-supervised models from the image to the time series domain?* More specifically, we answer this question for the most prominent state-of-the-art SSL approaches, by proposing adaptions for MixMatch [8], Virtual Adversarial Training [9], the Mean Teacher [10] and the Ladder Net [11]. These include

the modification of a suitable backbone architecture as well as adaptions of an appropriate data augmentation strategy to account for the domain transfer of these models. For demonstration of the efficacy of our proposed frameworks we adhere to best practices for realistic evaluation of semi-supervised models and provide a fair and reliable model comparison with a high degree of practicality [12].

A. Related Work

a) Time Series Classification: Over the past years, a variety of methods has been developed for TSC. A detailed overview on classical ML methods that were specifically developed for TSC [13], [14], [15] is provided in [4]. An alternative approach towards TSC consists in the extraction of statistical features from the raw time series as the basis for training any strong classifier for tabular data [16]. Also in deep learning, specific methods for time series classification have been developed [17], [18], [19]. A comprehensive overview on these recent developments can be found in [5].

b) Semi-Supervised Learning: There exists a plethora of different concepts that extract additional information from unlabeled data via semi-supervision. These range from the extension of supervised ML methods such as the semi-supervised Support Vector Machine [20] or semi-supervised Boosting [21] to inherently semi-supervised methods such as Label Propagation [22], Manifold Regularization [23] or Co-Training [24]. [25] provide a detailed overview on these semi-supervised approaches. There is also growing research on deep semi-supervised learning, mainly driven by the computer vision community. A recent overview and taxonomy on these developments are provided by [7]. Amongst these are graph-based methods such as Deep Label Propagation [26], SNTG [27] or the extension of pseudo-labelling for deep learning [7]. Further, there is growing research on regularization-based approaches following the rationale of adding an additional unsupervised regularization loss term to the initial supervised loss. The Mean Teacher [10] and its predecessors, Temporal Ensembling and the Π -Model [28], employ a consistency loss over the unlabeled samples to reward similar predictions for differently augmented versions of the same unlabeled sample. To overcome one drawback of those methods, the need for domain-dependent data augmentation strategies, Virtual Adversarial Training (VAT) [9] adds small perturbations to the input data to create an auxiliary unsupervised training

target. MixMatch [8] in turn combines different regularization strategies in one common framework. These regularization-based approaches yield state-of-the-art performance on image classification benchmarks.

c) SSL for TSC: Different classical semi-supervised models have been developed for TSC. In their foundational work, [29] propose an approach that combines pseudolabelling with a nearest-neighbor model for imbalanced, binary TSC tasks. This cluster-then-label [7] rationale for labeled and unlabeled time series via custom distance metrics is also employed in approaches such as DTW-D [30], SUCCESS [31] or LCLC [32]. Graph-based label propagation [22] is combined with time-series-specific distance metrics by [33] and [34] introduced the shapelet-based SSSL.

d) Deep SSL for TSC: There has been recent developments on neural net-based approaches. A customized version of the LadderNet [11] based on the FCN architecture [17] was applied by [35] on three multivariate human activity recognition (HAR) datasets. They report relative gains of the semi-supervised model over the supervised baselines for small amounts of labeled samples. To the best of our knowledge, [35] are the first to evaluate SSL methods on large, multivariate TSC datasets. A self-supervised approach, where the model is jointly trained on an auxiliary forecasting task over the whole dataset next to the initial supervised classification task on the labeled data only, was introduced by [36]. They build upon the benchmark of [34] on a subset of smaller, univariate TSC datasets from the UCR repository [3] and report state-of-the-art performance compared to the majority of above methods as well as a customized variant of the II–Model [28] that works on time series problems. In alignment with [35], they report particularly strong model performance for the deep supervised baseline FCN [17] trained on few labeled samples only reporting it to outperform all above mentioned classical semi-supervised models. This deep learning baseline outperforms all of the classical semi-supervised models and almost always beats the II–Model. We include this approach as a self-supervised baseline in our experiments.

e) Limitations: All existing model comparisons for semi-supervised TSC, despite the work of [35], are limited to univariate time series datasets with a maximal size of 1000 training samples. In contrast to computer vision research on SSL [7], these model comparisons are conducted for one fixed relative amount of labeled samples in the vast majority of experiments, making it hard to deduce general information for different data situations. They also do not align with the guidelines established by [12] for SSL on image data and do not include repeated model runs to account for randomness in the selection of labeled samples. Another issue is the lack of publicly accessible implementations of the classical approaches to semi-supervised TSC, making it impossible to validate against these approaches. This in turn leads to the problem that model comparisons with existing methods solely rely on values reported in former work for the same datasets with partially opaque dataset splits and unlabelling procedures.

Our **main contributions** can be summarized as follows:

- 1) We propose four new deep SSL algorithms for TSC and describe tuning parameters and meaningful data augmentation strategies.
- 2) We investigate the applicability of deep SSL in the domain of TSC and provide insights in which settings the proposed methods work well and how they compare to existing approaches.
- 3) Through these experiments we are able to identify two out of our four proposed methods that notably improve over existing approaches.

II. FROM IMAGES TO TIME SERIES

A. Problem Formulation

We define an equidistant time series as $x^{(i)} = \{\{x_{1,1}^{(i)}, \dots, x_{1,t}^{(i)}\}, \dots, \{x_{c,1}^{(i)}, \dots, x_{c,t}^{(i)}\}\}$, where t describes the length and c the amount of covariates such that $x^{(i)} \in \mathcal{X} \subseteq \mathbb{R}^{c \times t}$. For $c = 1$ the time series is called univariate and for $c > 1$ multivariate. Next to the input space \mathcal{X} , we use $y^{(i)} \in \mathcal{Y}$ to denote a categorical variable in the target space \mathcal{Y} . The goal of SSL is to train a prediction model $f : \mathcal{X} \mapsto \mathcal{Y}$ on a dataset $\mathcal{D} = (\mathcal{D}^l, \mathcal{D}^u)$ which consists of a labeled dataset $\mathcal{D}^l = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n_l}$ and an unlabeled dataset $\mathcal{D}^u = \{x^{(i)}\}_{i=n_l+1}^n$ where $n = n_l + n_u$. We consider the case where $n_l \ll n_u$, as usual in SSL. Further, we define one batch of data as $\mathcal{B} \subset \mathcal{D}$, where $\mathcal{B}^l \subseteq \mathcal{D}^l$ contains the labeled samples and $\mathcal{B}^u \subseteq \mathcal{D}^u$ the unlabeled samples in that batch such that $\mathcal{B} = (\mathcal{B}^l, \mathcal{B}^u)$.

B. Backbone Architecture

A basic building block in deep learning for images is a 3-dimensional tensor, whereas time series can be represented as 2-dimensional tensors with channels corresponding to the number of covariates. The extension of building blocks of powerful image classification architectures to TSC is thus straightforward, yet the right choice of a backbone architecture is crucial. We propose the use of the Fully Convolutional Network (FCN) [17] as a backbone architecture as it was shown to outperform a variety of models on 44 different TSC problems and is used in related work on semi-supervised TSC [36]. In all regularization-based semi-supervised methods discussed in Section II-D, except for the Ladder Net [11], the network architecture can be decoupled from the model training strategy. This allows us to replace the backbone architecture of many of the established SSL methods from image classification with the FCN. In case of the Ladder Net, we design the decoder as a mirrored version of the FCN encoder (see Section II-D).

C. Data Augmentation

One crucial component of regularization-based semi-supervised methods is the injection of random noise into the model. Data augmentation strategies $g(x^{(i)}), g : \mathcal{X} \mapsto \mathcal{X}$ should be designed such that they perturbate the input $x^{(i)}$ of a sample while preserving the meaning of its label $y^{(i)}$. This can be achieved by utilizing inherent invariances in the data, e.g., rotations of images usually preserve the meaning of an image. For images, invariances can be easily understood visually. In the time series domain, such invariances are not

straightforward to understand, rendering the design of reasonable data augmentation strategies in this domain challenging. A set of data augmentation strategies for multivariate time series classification was introduced by [37] and evaluated on one HAR task. They show that the majority of strategies are beneficial, but some can deteriorate the model performance. To overcome the additional burden of choosing the right strategy, we propose the use of the RandAugment strategy [38] which removes the need for a separate search phase. For each training batch, N augmentation strategies are randomly chosen out of a set of K possible policies. Next to N , a *magnitude* hyperparameter is introduced which controls the augmentation intensity of the selected policies. We use the following set of augmentation policies [37]: warping in the time dimension, warping the magnitude, addition of Gaussian Noise and random rescaling. We use RandAugment in this context following the rationale that even if a augmentation strategy is (not) label preserving, training with RandAugment with $N = 1$ will still produce correct model updates in at least $\frac{K-1}{K}$ of the forward passes. Early experiments in a fully supervised setting showed that the application of this data augmentation strategy improves model performance across all datasets used in our experiments.

D. Methods

The Mean Teacher [10] is the successor of a series of consistency-regularization-based models such as Temporal Ensembling or the Π -Model [28] for SSL and was empirically shown to outperform its predecessors [12]. Thereby, a teacher model, that is an average of the consecutive student models, is used to enforce consistency in model predictions over the course of model training.

Virtual Adversarial Training (VAT) [9] also focuses on consistency regularization. Similar to adversarial examples [39], a small data perturbation is learned such that its addition to the initial data point is expected to yield the maximum change in the model's prediction. These perturbed model predictions are used as auxiliary labels for the unlabeled samples within a regularization term to enable model training on the whole data set. This approach is particularly interesting for the time series domain where visual inspection of the appropriateness of data augmentation policies is difficult, as it does not rely on data augmentation techniques.

In MixMatch, various semi-supervised techniques such as data augmentation for consistency regularization, Mixup training [40] and pseudo-labeling are combined within one holistic approach [8]. It was empirically shown to perform well on image data, motivating our use of it in this work [8].

The Ladder Net by [11] is a reconstruction-based SSL model and is inspired by denoising autoencoders [41]. In its core, it extends a supervised encoder model with a corresponding decoder network which allows for the calculation of an unsupervised reconstruction loss over the unlabeled samples enabling training on the whole dataset. The Ladder Net was previously extended to TSC problems [35] and is thus also part of this study.

III. EXPERIMENTAL DESIGN

A. Baseline Models

Next to shapelet- and distance-based methods [4], fitting standard ML methods on hand-crafted statistical features has been a widely used approach for TSC before the introduction of specific deep learning architectures for TSC [17] [18]. We include a Random Forest and a Logistic Regression trained on features, extracted via the tsfresh framework [16] from the time series, as baselines.

In addition, we train the FCN architecture [17] on the labeled samples \mathcal{D}^l based on the cross entropy loss as a supervised deep learning baseline model for our experiments. To ensure a fair and reliable model comparison, we explicitly use the same architecture of this supervised baseline model as the backbone for all SSL approaches. We also use the performance of a supervised FCN trained on the fully labeled datasets as an estimated upper bound for the model performance.

Furthermore, we evaluate the performance of the self-supervised approach that was recently introduced for TSC by [36]. Thereby, an auxiliary forecasting task from the time series data \mathcal{D} is created and combined with the initial classification task as a surrogate supervision signal allowing the use of unlabeled data in model training. The model is then jointly trained on both tasks simultaneously. Next to its re-implementation, we further extend their approach for multivariate TSC by increasing the amount of neurons in the surrogate model head accordingly. The direct comparison with this self-supervised approach is of special interest as it was shown to outperform classical semi-supervised approaches in a set of experiments on smaller TSC datasets [36].

B. Data Sets

We evaluate the performance of the above described semi-supervised models on 6 publicly available datasets. In contrast to previous work [33], [34], [36], we explicitly focus on large datasets with at least 1000 observations. Their main characteristics are described in Table I.

TABLE I: Characteristics of the used data sets where c refers to the amount of covariates, *Size* to the size of the whole training data set and *Length* to the length of the time series.

Name	Classes	Size	Length	c	Balanced
Crop	24	7,200	46	1	✓
ElectricDevices	7	8,926	96	1	✗
FordB	2	3,636	500	1	✓
Pamap2	13	11,313	100	6	✗
WISDM	6	10,727	80	3	✗
Balanced SITS	6	35,064	46	1	✓

With Crop, ElectricDevices and FordB we include three of the largest datasets from the UCR Time Series Classification Repository [3]. In addition, we use the two multivariate HAR datasets Pamap2 [42] and WISDM [43]. We also evaluate the models on a class-balanced version of the Satellite Image Time Series (SITS) dataset [44].

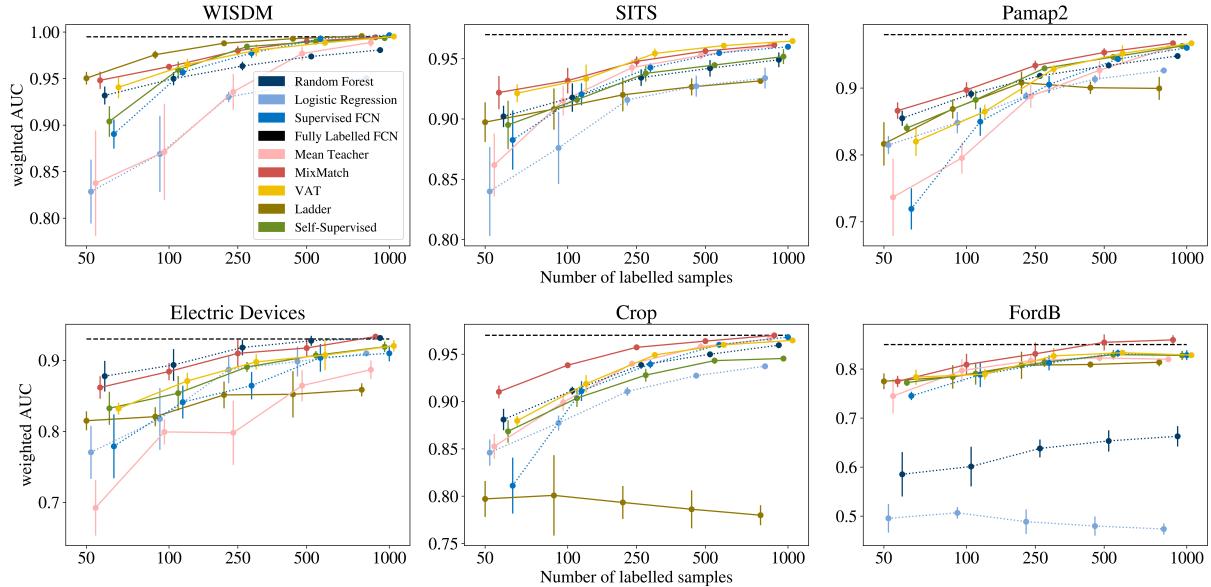


Fig. 1: Performance of all models on the 6 different datasets over various n_l as presented in Table II in the appendix. The horizontal line marks the performance of the fully labeled baseline, i.e. the supervised FCN model trained on the fully labeled dataset. Dots represent the mean wAUC and the vertical lines the standard deviation over 5 repeated *unlabeling* steps. The performance of the baseline models are depicted as dotted, those of the semi-supervised models as solid lines. Semi-supervised models clearly outperform the baseline models in settings with few labeled samples $n_l \in \{50, 100\}$ on all but the Electric Devices dataset.

C. Evaluation, Tuning and Implementation

Due to special factors, such as the selection of the labeled data points, an unbiased and fair model comparison is particularly crucial to get a realistic perspective on the performance of the semi-supervised models [7]. We adhere to the guidelines for realistic evaluation of semi-supervised models by [12] to guarantee reliable and fair experimental results. For performance evaluation of SSL models, the standard procedure is to split a fully labeled dataset \mathcal{D} into labeled and unlabeled datasets \mathcal{D}^l and \mathcal{D}^u via artificial *unlabeling* of n_u randomly drawn samples [7]. This way, semi-supervised data settings for different amounts of labeled samples l are simulated. We unlabel in a stratified manner to retain the datasets' label distributions. For the following experiments, we split the evaluation of one model f on one data set D in two distinct phases.

a) Tuning Phase: In the tuning phase, we tuned the model f with one fixed amount of labeled samples to yield an optimal set of hyperparameters θ^* . Thereby, f was trained on a training dataset $\mathcal{D}_{train} = (\mathcal{D}_{train}^l, \mathcal{D}_{train}^u)$, where we fixed $|\mathcal{D}_{train}^l| = 500$, and validated on a labeled holdout validation set \mathcal{D}_{val} . The choice of the size of \mathcal{D}_{val} is subject to recent discussions [11], [12], [45]. Large \mathcal{D}_{val} are expected to yield stable results for model tuning, which is important for many hyperparameter-sensitive semi-supervised models, but stands in contrast to the promised practicality of these models in settings with few labeled data. First insights on this trade-off are given by [12] and [45], which empirically show in smaller experiments $|\mathcal{D}_{val}| = 1000$ to be a vali-

dation set size where variance in the performance estimates is still low enough to allow for reasonable model selection. Following this, we set the size of the labeled validation set to $|\mathcal{D}_{val}| = 1000$ which is rather small compared to recent literature where $|\mathcal{D}_{val}| \geq 4000$ [9], [28], [10]. A separate labeled test set \mathcal{D}_{test} with $|\mathcal{D}_{test}| = 2000$ is kept aside for the evaluation phase. Hyperband [46] with random sampling as implemented in the Optuna framework [47] was used for tuning, with a fixed budget of 100 GPU hours for each deep learning model and dataset. We measure model performance in terms of weighted Area under the Curve (wAUC) to account for model calibration and class imbalance.

b) Evaluation Phase: In the evaluation phase, we train $f(\theta^*)$ on \mathcal{D}_{train} with varying amounts of $n_l \in \{50, 100, 250, 500, 1000\}$ for a maximum of 25000 model update steps, assuming θ^* is also a suitable hyperparameter set for amounts of labels $n_l \neq 500$ on which the model was not specifically tuned. This evaluation scheme is in line with previous work on SSL for image data [8], [12]. Model performance is tracked on \mathcal{D}_{val} and the model checkpoint with the best validation performance is used for inference on the holdout \mathcal{D}_{test} . The selection of especially (un-)informative labeled samples can have a major effect on the model performance, especially for small n_l . To account for potentially (un-)lucky selection of \mathcal{D}_{train}^l in the *unlabelling* split of $\mathcal{D}_{train} = (\mathcal{D}_{train}^l, \mathcal{D}_{train}^u)$, we repeat this *unlabelling* step 5 times. In case of the ML baseline models, we use a Random Search with a budget of 100 model evaluations for the tuning phase and evaluate them on the same set of values for n_l .

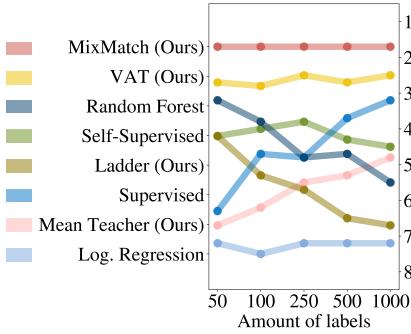


Fig. 2: Average ranks of all models based on the wAUC over the 6 datasets for varying n_l . Models are sorted by their strongest performance on $n_l = 50$ and plotted with decreasing rank as indicated on the right vertical axis.

in the evaluation phase. See Table IV in the appendix for the specific ranges. All deep learning models were implemented in a unified codebase¹ and trained using the Adam optimizer [48] with all parameters set to default values except the learning rate and weight decay. We implemented all deep learning models from scratch in one unified framework and validated our implementations based on performance metrics reported on image classification tasks..

IV. EXPERIMENTAL RESULTS

Experimental findings are visualized in Figure 1 and Table II in the appendix. The ranking of the various models for different n_l , averaged over the datasets, is shown in Figure 2 and Table III in the appendix.

a) Semi-supervised models outperform supervised baselines: Overall, our results show that semi-supervised models outperform baseline models especially for small amounts of labeled data. This relative performance gain of semi-supervised over supervised models is decreasing with an increase in n_l and we find that all models benefit from more labeled samples in most cases. This is in line with literature on SSL [7].

b) Deep SSL translates well to TSC: Following our experimental results in Figure 1, we deduct that *transferring well-established semi-supervised models from the image to the time series domain is indeed possible*. We find that the deep semi-supervised models, especially the transferred MixMatch and VAT, show impressive performance gains over the deep supervised baseline model over all datasets up to $n_l = 500$, even reaching the performance of the fully labeled baseline in few cases. For instance, the Mixmatch model exceeds the deep supervised baseline by 0.16 wAUC on the Pamap2 and by 0.10 wAUC on the Crop dataset for $n_l = 50$. These findings again encourage our proposed transfer.

c) Strong baselines are crucial: We find the use of strong baselines crucial for a realistic perspective on semi-supervised learning performance. For instance, the Mean Teacher shows weak performance on the majority of datasets, often performing even worse than the supervised baseline. This is in line

with results of [36]. The strong performance of the Random Forest for small n_l on the other hand also stresses the need for realistically strong supervised baselines.

d) Proposed methods outperform existing semi-supervised approaches: While our results on the Ladder Net outperforming other supervised methods align with those of [35], we also observe that the Ladder Net is notably worse compared to alternative SSL algorithms we propose. This varying performance might be grounded in the large amount of hyperparameters of the Ladder Net and its sensitivity to different settings of those.

e) Proposed methods outperform self-supervised modeling: Similar to [36], we find their self-supervised approach to perform better or at least equally well compared to the deep supervised baseline model. Additionally, we are able to show that our extension towards multivariate time series also works well on the two multivariate datasets, WISDM and Pamap2. The proposed approaches MixMatch and VAT furthermore consistently outperform this self-supervised approach across different amounts of labels on all 6 datasets.

f) Ranking of model performance similar to image domain: In terms of model performance ranking, literature suggests that MixMatch performs better than VAT which again outperforms the Mean Teacher and the Ladder Net [8], [12]. When ranking the algorithms across the datasets in Figure 2, we confirm this ranking in the TSC setting.

Our results show that the promised label efficiency of modern, deep semi-supervised model approaches translates well to TSC problems. Furthermore, these findings suggest the use of strong semi-supervised models from the image domain as these transferred models show stronger performance than the currently existing semi- and self-supervised approaches tailored towards TSC. We believe that this work, also thanks to a strong focus on a fair and reliable model comparison, can serve as the basis for future research advances in semi-supervised learning for time series classification.

ACKNOWLEDGEMENTS

This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy through the Center for Analytics – Data – Applications (ADA-Center) within the framework of BAYERN DIGITAL II (20-3410-2-9-8) as well as the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A.

REFERENCES

- [1] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, *et al.*, “Scalable and accurate deep learning with electronic health records,” *NPJ Digital Medicine*, vol. 1, no. 1, p. 18, 2018.
- [2] G. A. Susto, A. Cenedese, and M. Terzi, “Time-series classification methods: Review and applications to power systems data,” in *Big data application in power systems*, pp. 179–220, Elsevier, 2018.
- [3] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, “The ucr time series archive,” 2019.
- [4] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, “The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances,” *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 2017.

¹<https://github.com/Goschjann/ssltsc>

- [5] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, “Deep learning for time series classification: a review,” *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.
- [6] J. Goschenhofer, F. M. Pfister, K. A. Yuksel, B. Bischl, U. Fietzek, and J. Thomas, “Wearable-based parkinson’s disease severity monitoring using deep learning,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 400–415, Springer, 2019.
- [7] J. E. Van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.
- [8] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, “Mixmatch: A holistic approach to semi-supervised learning,” in *Advances in Neural Information Processing Systems 32*, pp. 5049–5059, 2019.
- [9] T. Miyato, S.-i. Maeda, S. Ishii, and M. Koyama, “Virtual adversarial training: a regularization method for supervised and semi-supervised learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2019.
- [10] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *Advances in Neural Information Processing Systems 30*, pp. 1195–1204, 2017.
- [11] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, “Semi-supervised learning with ladder networks,” in *Advances in Neural Information Processing Systems 28*, pp. 3546–3554, 2015.
- [12] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, “Realistic evaluation of deep semi-supervised learning algorithms,” in *Advances in Neural Information Processing Systems 31*, pp. 3235–3246, 2018.
- [13] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, “Learning time-series shapelets,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 392–401, 2014.
- [14] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, “Time-series classification with cote: the collective of transformation-based ensembles,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2522–2535, 2015.
- [15] R. J. Kate, “Using dynamic time warping distances as features for improved time series classification,” *Data Mining and Knowledge Discovery*, vol. 30, no. 2, pp. 283–312, 2016.
- [16] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, “Time series feature extraction on basis of scalable hypothesis tests (tsfresh—a python package),” *Neurocomputing*, pp. 72–77, 2018.
- [17] Z. Wang, W. Yan, and T. Oates, “Time series classification from scratch with deep neural networks: A strong baseline,” in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1578–1585, 2017.
- [18] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, “Inceptiontime: Finding alexnet for time series classification,” *Data Mining and Knowledge Discovery*, vol. 34, pp. 1936–1962, 2020.
- [19] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *CoRR*, 2018.
- [20] V. Vapnik, *Statistical learning theory*. Wiley New York, 1998.
- [21] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu, “Semiboost: Boosting for semi-supervised learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 11, pp. 2000–2014, 2008.
- [22] X. Zhu and Z. Ghahramani, “Learning from labeled and unlabeled data with label propagation,” tech. rep., 2002.
- [23] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: A geometric framework for learning from labeled and unlabeled examples,” *Journal of Machine Learning Research*, vol. 7, no. Nov, pp. 2399–2434, 2006.
- [24] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100, 1998.
- [25] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-supervised learning*. MIT Press, 2006.
- [26] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, “Label propagation for deep semi-supervised learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5070–5079, 2019.
- [27] Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang, “Smooth neighbors on teacher graphs for semi-supervised learning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8896–8905, 2018.
- [28] S. Laine and T. Aila, “Temporal ensembling for semi-supervised learning,” in *5th International Conference on Learning Representations, ICLR 2017*, 2017.
- [29] L. Wei and E. Keogh, “Semi-supervised time series classification,” in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 748–753, 2006.
- [30] Y. Chen, B. Hu, E. Keogh, and G. E. Batista, “Dtw-d: time series semi-supervised learning from a single example,” in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 383–391, 2013.
- [31] K. Marusy and K. Buza, “Success: a new approach for semi-supervised classification of time-series,” in *International Conference on Artificial Intelligence and Soft Computing*, pp. 437–447, 2013.
- [32] M. N. Nguyen, X.-L. Li, and S.-K. Ng, “Positive unlabeled learning for time series classification,” in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, , IJCAI-11*, 2011.
- [33] Z. Xu and K. Funaya, “Time series analysis with graph-based semi-supervised learning,” in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–6, 2015.
- [34] H. Wang, Q. Zhang, J. Wu, S. Pan, and Y. Chen, “Time series feature learning with labeled and unlabeled data,” *Pattern Recognition*, vol. 89, pp. 55–66, 2019.
- [35] M. Zeng, T. Yu, X. Wang, L. T. Nguyen, O. J. Mengshoel, and I. Lane, “Semi-supervised convolutional neural networks for human activity recognition,” in *2017 IEEE International Conference on Big Data (Big Data)*, pp. 522–529, 2017.
- [36] S. Jawed, J. Grabocka, and L. Schmidt-Thieme, “Self-supervised learning for semi-supervised time series classification,” in *Proceedings of the 24th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2020*, pp. 499–511, 2020.
- [37] T. T. Um, F. M. J. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulic, “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*, pp. 216–220, 2017.
- [38] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “RandAugment: Practical automated data augmentation with a reduced search space,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.
- [39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, pp. 2672–2680, 2014.
- [40] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in *6th International Conference for Learning Representations, ICLR 2018*, 2018.
- [41] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion..,” *Journal of Machine Learning Research*, vol. 11, no. 12, pp. 3371–3408, 2010.
- [42] A. Reiss and D. Stricker, “Introducing a new benchmarked dataset for activity monitoring,” in *2012 16th International Symposium on Wearable Computers*, pp. 108–109, 2012.
- [43] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.
- [44] F. Petitjean, J. Inglada, and P. Gancarski, “Satellite image time series analysis under time warping,” *IEEE transactions on geoscience and remote sensing*, vol. 50, no. 8, pp. 3081–3095, 2012.
- [45] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, “S4I: Self-supervised semi-supervised learning,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1476–1485, 2019.
- [46] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, “Hyperband: A novel bandit-based approach to hyperparameter optimization,” *Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6765–6816, 2017.
- [47] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631, 2019.
- [48] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference for Learning Representations, ICLR 2015*, 2015.

APPENDIX

A. Model Performance

TABLE II: Results for models over datasets with varying numbers of labels n_l . Performance is measured as weighted AUC. The best results for each n_l -dataset-combination are emphasized in bold with standard deviations over 5 replications in brackets.

Number of labels Dataset Model	50						100					
	Crop	Electric Devices	FordB	Pamap2	SITS	WISDM	Crop	Electric Devices	FordB	Pamap2	SITS	WISDM
Ladder	0.797 (0.019)	0.815 (0.013)	0.775 (0.016)	0.816 (0.033)	0.897 (0.017)	0.95 (0.007)	0.801 (0.042)	0.821 (0.013)	0.784 (0.024)	0.869 (0.014)	0.908 (0.017)	0.976 (0.004)
Logistic Regression	0.846 (0.014)	0.771 (0.037)	0.496 (0.029)	0.815 (0.013)	0.84 (0.037)	0.829 (0.034)	0.877 (0.008)	0.817 (0.043)	0.507 (0.012)	0.848 (0.016)	0.876 (0.03)	0.869 (0.041)
Mean Teacher	0.853 (0.013)	0.692 (0.039)	0.745 (0.036)	0.737 (0.058)	0.862 (0.026)	0.838 (0.057)	0.899 (0.009)	0.799 (0.018)	0.797 (0.023)	0.795 (0.023)	0.915 (0.012)	0.871 (0.052)
MixMatch	0.910 (0.007)	0.862 (0.016)	0.775 (0.012)	0.861 (0.013)	0.922 (0.014)	0.948 (0.009)	0.938 (0.003)	0.884 (0.012)	0.809 (0.022)	0.897 (0.011)	0.932 (0.011)	0.963 (0.003)
Random Forest	0.881 (0.011)	0.878 (0.022)	0.585 (0.045)	0.855 (0.011)	0.902 (0.009)	0.932 (0.01)	0.911 (0.004)	0.893 (0.022)	0.601 (0.04)	0.891 (0.006)	0.918 (0.012)	0.95 (0.007)
Self-Supervised	0.868 (0.012)	0.832 (0.023)	0.772 (0.007)	0.84 (0.007)	0.895 (0.02)	0.904 (0.016)	0.904 (0.009)	0.854 (0.024)	0.79 (0.022)	0.882 (0.014)	0.916 (0.003)	0.959 (0.009)
Supervised	0.811 (0.03)	0.779 (0.045)	0.745 (0.009)	0.719 (0.031)	0.883 (0.025)	0.89 (0.016)	0.911 (0.01)	0.841 (0.023)	0.788 (0.025)	0.85 (0.021)	0.921 (0.009)	0.957 (0.004)
VAT	0.88 (0.005)	0.832 (0.009)	0.783 (0.016)	0.82 (0.022)	0.921 (0.007)	0.941 (0.012)	0.919 (0.009)	0.871 (0.012)	0.789 (0.011)	0.865 (0.01)	0.933 (0.012)	0.965 (0.006)

Number of labels Dataset Model	250						500					
	Crop	Electric Devices	FordB	Pamap2	SITS	WISDM	Crop	Electric Devices	FordB	Pamap2	SITS	WISDM
Ladder	0.793 (0.017)	0.851 (0.019)	0.808 (0.028)	0.908 (0.017)	0.92 (0.014)	0.988 (0.001)	0.786 (0.02)	0.852 (0.033)	0.809 (0.007)	0.901 (0.009)	0.927 (0.007)	0.993 (0.001)
Logistic Regression	0.911 (0.005)	0.887 (0.019)	0.489 (0.025)	0.888 (0.006)	0.916 (0.005)	0.93 (0.006)	0.927 (0.002)	0.898 (0.021)	0.48 (0.019)	0.913 (0.006)	0.927 (0.009)	0.944 (0.004)
Mean Teacher	0.94 (0.004)	0.798 (0.045)	0.817 (0.011)	0.888 (0.017)	0.943 (0.007)	0.936 (0.019)	0.958 (0.004)	0.864 (0.022)	0.823 (0.007)	0.927 (0.014)	0.953 (0.003)	0.977 (0.008)
MixMatch	0.957 (0.003)	0.910 (0.019)	0.831 (0.023)	0.934 (0.007)	0.948 (0.004)	0.980 (0.005)	0.964 (0.003)	0.917 (0.013)	0.854 (0.016)	0.953 (0.006)	0.956 (0.002)	0.990 (0.003)
Random Forest	0.939 (0.004)	0.918 (0.011)	0.638 (0.018)	0.918 (0.005)	0.934 (0.007)	0.963 (0.005)	0.950 (0.003)	0.928 (0.007)	0.653 (0.022)	0.934 (0.004)	0.942 (0.007)	0.974 (0.001)
Self-Supervised	0.928 (0.007)	0.891 (0.007)	0.814 (0.01)	0.938 (0.001)	0.938 (0.007)	0.984 (0.001)	0.943 (0.003)	0.907 (0.006)	0.829 (0.01)	0.947 (0.004)	0.945 (0.002)	0.990 (0.002)
Supervised	0.939 (0.004)	0.864 (0.019)	0.812 (0.015)	0.905 (0.013)	0.943 (0.004)	0.977 (0.005)	0.960 (0.002)	0.904 (0.019)	0.832 (0.008)	0.943 (0.004)	0.955 (0.001)	0.993 (0.004)
VAT	0.949 (0.002)	0.898 (0.011)	0.827 (0.017)	0.929 (0.006)	0.954 (0.004)	0.98 (0.006)	0.960 (0.003)	0.907 (0.022)	0.833 (0.005)	0.952 (0.012)	0.961 (0.001)	0.989 (0.002)

Number of labels Dataset Model	1000					
	Crop	Electric Devices	FordB	Pamap2	SITS	WISDM
Ladder	0.78 (0.011)	0.858 (0.009)	0.814 (0.008)	0.899 (0.017)	0.932 (0.001)	0.996 (0.001)
Logistic Regression	0.937 (0.001)	0.91 (0.003)	0.474 (0.011)	0.926 (0.004)	0.934 (0.009)	0.950 (0.003)
Mean Teacher	0.966 (0.001)	0.887 (0.013)	0.82 (0.006)	0.96 (0.002)	0.96 (0.002)	0.989 (0.005)
MixMatch	0.970 (0.001)	0.933 (0.003)	0.859 (0.010)	0.967 (0.004)	0.961 (0.002)	0.994 (0.001)
Random Forest	0.959 (0.001)	0.932 (0.004)	0.663 (0.021)	0.948 (0.003)	0.949 (0.006)	0.981 (0.001)
Self-Supervised	0.945 (0.001)	0.919 (0.007)	0.828 (0.010)	0.963 (0.004)	0.952 (0.006)	0.994 (0.001)
Supervised	0.968 (0.001)	0.910 (0.011)	0.828 (0.010)	0.960 (0.003)	0.960 (0.002)	0.997 (0.001)
VAT	0.964 (0.001)	0.920 (0.008)	0.828 (0.006)	0.967 (0.004)	0.965 (0.002)	0.995 (0.001)

B. Model Ranking

TABLE III: The average rank of all models based on the wAUC over the 6 different datasets for various amounts of labels n_l . Lower rank indicates stronger model performance. Ranks are shown with decimals due to averaging over datasets.

	Number of labels				
	50	100	250	500	1000
MixMatch	1.7	1.7	1.7	1.7	1.7
VAT	2.7	2.8	2.5	2.7	2.5
MeanTeacher	6.7	6.2	5.5	5.3	4.8
Self-supervised	4.2	4.0	3.8	4.3	4.5
Ladder	4.2	5.3	5.7	6.5	6.7
Supervised	6.3	4.7	4.8	3.7	3.2
Random Forest	3.2	3.8	4.8	4.7	5.5
Logistic Regression	7.2	7.5	7.2	7.2	7.2

C. Hyperparameters

TABLE IV: Hyperparameter ranges used for tuning of the different models. Deep Learning models were tuned via Hyperband as described in Section 3 while the Random Forest and the Logistic Regression were tuned via Random Search with a budget of 100 model evaluations each.

Parameter	Range	Scale
Shared		
Weight decay	[$1e^{-6}$; $1e^{-2}$]	log
Learning rate	[$1e^{-5}$; $1e^{-2}$]	log
Rampup length	[5000; 25000]	linear
Magnitude (RandAug)	[1; 10]	linear
N (RandAug)	[1; 6]	linear
VAT		
ϵ	[0.1; 10.0]	linear
α	[0.1; 5.0]	linear
MixMatch		
α	[0.5; 1.0]	linear
λ_u	[0.0; 150.0]	linear

Parameter	Range	Scale
Self-Supervised Learning		
λ	[0.1; 10]	log
horizon h	[0.1, 0.2, 0.3]	discrete
stride s	[0.05, 0.1, 0.2, 0.3]	discrete
Ladder Net		
Noise ratio	[0.1, 0.3, 0.45, 0.6]	discrete
Loss weights	[0.1; 10.0]	log
Mean Teacher		
α_{ema}	[0.9; 1.0]	log
v_{max}	[0; 10]	linear
Random Forest		
Number of trees	[100; 1000]	linear
Max. tree depth	[3; 25]	linear
Logistic Regression		
Regularization term	[None, L_1 , L_2]	discrete