Self-supervised Learning for Semi-supervised Time Series Classification

Shayan Jawed(✉), Josif Grabocka, and Lars Schmidt-Thieme

Information Systems and Machine Learning Lab, University of Hildesheim, Hildesheim, Germany
{shayan,josif,schmidt-thieme}@ismll.uni-hildesheim.de

Abstract. Self-supervised learning is a promising new technique for learning representative features in the absence of manual annotations. It is particularly efficient in cases where labeling the training data is expensive and tedious, naturally linking it to the semi-supervised learning paradigm. In this work, we propose a new semi-supervised time series classification model that leverages features learned from the self-supervised task on unlabeled data. The idea is to exploit the unlabeled training data with a forecasting task which provides a strong surrogate supervision signal for feature learning. We draw from established multi-task learning approaches and model forecasting as an auxiliary task to be optimized jointly with the main task of classification. We evaluate our proposed method on benchmark time series classification datasets in semi-supervised setting and are able to show that it significantly outperforms the state-of-the-art baselines.

Keywords: Self-supervised features · Semi-supervised classification · Auxiliary tasks · Convolutional Neural Networks

1 Introduction

Modern deep learning architectures have taken the fields of Computer Vision, Natural Language Processing and Recommender Systems by storm. Time series Classification is no stranger to Recurrent Neural Networks and Convolutional Neural Networks (ConvNets) too [6,19]. Although proven to learn high level features across a broad domain of time series classification problems, the success of ConvNets hinges on the availability of large amounts of labeled training data. In reality, however, there is a high cost associated in acquiring such labeled data. As a result, there have been efforts to utilize semi-supervised learning algorithms catered especially for time series classification [2,9,12,17,21,22]. The idea behind semi-supervised learning is to exploit unlabeled data for training purpose in the presence of only few labeled instances. The applicability of this learning paradigm naturally extends to time series data as plentiful of it can be acquired trivially. For example, a single polysomnography (sleep study) can generate up to

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12084, pp. 499–511, 2020.
https://doi.org/10.1007/978-3-030-47426-3_39

40,000 heartbeats but it takes the time and expertise of a cardiologist to annotate individual heartbeats [2]. Hence, effective methods for semi-supervised learning can lead to mining vast amounts of time series data for which only comparatively few labels might be available.

A related stream of works has been dedicated to learning high level ConvNets based representations that do not require any manual annotation of data. Self-supervised learning has emerged as a prominent learning paradigm among such, where the idea is to define an annotation-free pretext task that is inherent in the data itself. The task stands to provide a surrogate supervision signal for feature learning. Example tasks include classifying image rotations [7], colorizing images [23] solving Jigsaw puzzles [15] to learn transferable representations for high-level tasks such as object detection and semantic segmentation. Until so far, applications have been limited to the Computer Vision domain.

In the same spirit of learning generalizable representations, we now introduce Multi-task learning. Multi-task learning is an important paradigm in machine learning which builds upon the idea of sharing knowledge between different tasks [1]. A set of tasks is learned in parallel, aiming to improve performance over each task compared with learning one of these tasks in isolation. A multi-task learning problem can also be formulated with respect to main and auxiliary tasks. Auxiliary tasks are motivated by the intuition that for most problem settings, performance over one particular task is of primary importance. However, in order to still reap the benefits of multi-task learning, related tasks could be modeled as auxiliary tasks [16]. These exist solely for the purpose of learning an enriched representation that could increase prediction accuracy over the main tasks.

In our work, we bring together ideas from these high-impact research ideas of self-supervised learning and multi-task learning to propose an auxiliary forecasting task that is inherent in labeled and unlabeled time series data both. This auxiliary task stands to provide a strong surrogate supervision signal for feature learning which when learned in parallel with the main task of classification of time series boosts the performance of the classifier especially in semi-supervised setting. More specifically, we first define a sliding window function parametrized by hyper-parameters of stride and horizon to be forecasted. Next, we augment the training set with generated samples for the forecasting task by providing labeled and unlabeled samples as input to this function. The ConvNet model is trained jointly to classify the labeled samples and forecast future series values. This exploitation of the unlabeled samples leads to learning representations that help boost the classification accuracy. The intuition is that these unlabeled samples come from the same distribution and if the model learns the complex task of forecasting series values accurately, then the same latent representations could be leveraged for classification. In our experiments we show that is indeed the case and our proposed method excels in semi-supervised setting where only a few labeled instances might be available for the model to learn from.

To recap, our contributions are:

- A novel self-supervised task that is intuitive, requires close to no changes in the base network structure and provides a strong surrogate supervisory signal for feature learning in the realm of time series classification.
- A multi-task network which enables the forecasting and classification task to share latent representations and learns high-order interactions automatically.
- Extensive experimental evaluation of our self-supervised method in the domain of semi-supervised learning for time series classification and show that it outperforms state-of-the-art baselines.

2 Related Work

The problem of learning with both labeled and unlabeled data is of central importance in machine learning [25]. We specifically review works that have focused on time series classification. We note the seminal work in the field from Wei et al. [21]. They proposed a self-training approach based on a nearest neighbor classifier. The work from [2] later improved the method significantly by proposing a new meta-feature based distance. In [14] a clustering approach was proposed combined with self-training. Another SSL algorithm in [12] also is in essence a clustering based method. The authors of [22] proposed a graph theoretic SSL algorithm that constructs graphs relating all samples based on different distance functions and consequently propagates labels. The current state-of-the-art method in the field [17] is based on shapelet learning [8] on both labeled and unlabeled time series data.

On the other hand, we note recent works [4,5,7,15,23] which showed that strong supervision could be leveraged by describing a task that is inherent in the data itself (requires no manual annotation). We consider the pioneering work by [4] which leveraged spatial context in an image for self-supervised learning by predicting relative location of one sampled patch to another. Similar self-supervised tasks were image colorization [23], solving jigsaw puzzles [15] and classifying image rotations [7]. More closely related to our work is a multi-task self-supervised network [5]. The work firstly tries to compare how the representations learned from recent proposed self-supervised approaches like above compare with each other, and then shows that combining these tasks even in a bare-bones multi-task network without catering for any controlled parameter sharing lifted the accuracy compared with the single-task networks compared before. Moreover, we also note the works that cater for temporal structure. Such temporal structure is inherent in video data, work in [13] proposed a sequential verification task to determine whether a sequence of frames was in correct order. It was shown that with this simple but intuitive task, the ConvNet captures temporally varying information such as human poses and ultimately lifted the accuracy on benchmark action recognition datasets. Another closely related example is [20] where the task was to recognize whether the video is playing forwards or backwards.

With motivations behind our method set from the literature review, we now draw the following insights: firstly there exist semi-supervised learning approaches similar to ours that learn from unlabeled data, most notably current state-of-the-art shapelet learning approach [17] if we consider shapelets to be similar to convolutional filters. However, with our work we exploit deep learning based methods which solve an auxiliary self-supervised task of forecasting which forces the network to learn filters to solve this particular complex task. Secondly, there have been a plethora of works that proposed novel self-supervised tasks, however to the best of our knowledge, there are no examples for the time series domain and neither that cast a self-supervised task as an auxiliary task.

3 Method

Our aim is to learn a ConvNet model that can estimate a forecasting function $f(.)$ and a classification function $g(.)$ jointly. To put it in concrete terms, we have a set of univariate time series samples $X = \{X_1, X_2, ..., X_n\}$ with their respective labels $Y = \{Y_1, Y_2, ..., Y_n\}$. We randomly split X to artificially construct $X^U = \{X_1^U, X_2^U, ..., X_k^U\}$ a set of unlabeled samples, and a labeled set comprising of $X^L = \{X_1^L, X_2^L, ..., X_l^L\}$ and $Y^L = \{Y_1^L, Y_2^L, ..., Y_l^L\}$. Note that $k + l = n$ and total series length is T. Furthermore, we define a sliding window function $w(.)$ which is parametrized by a stride s and horizon h. This function takes as input time series from X, and segments each in forecasting samples for e.g., $X_{11}^F = \{x_{1,t=p}^1, x_{1,t=p+1}^1, ..., x_{1,t=p+h}^1\}$ and $Y_{11}^F = \{y_{1,t=p+h+1}^1, y_{1,t=p+h+2}^1, ..., y_{1,t=p+2h}^1\}$ which denote the first sample i.e X_1's first window. The next sample is chosen with regard to $p = p + s$ and complete set of forecasting samples is given by $X^F = \{X_{11}^F, X_{12}^F, ..., X_{nm}^F\}$ and $Y^F = \{Y_1^F, Y_2^F, ..., Y_{nm}^F\}$. It is worth noting that these windows have a total length of $2h < T$ of which the later half consists of targets to be forecasted. And, the total number of forecasting samples, $m = n \times \lfloor (2 \times h + 1)/s \rfloor$ where $s > 0$. To fix ideas, our objectives are $Y^F = f(X^F)$ and $Y^L = g(X^L)$. The loss function for the objective with respect to $f(.)$:

$$L_f(X^F, \theta_f) = \frac{1}{n \times m \times h} \sum_{i}^{n} \sum_{j}^{m} \sum_{t}^{h} (y_{jt}^i - \hat{y}_{jt}^i)^2 \tag{1}$$

Specifically, we wish to learn the set of parameters θ_f that minimize the loss with respect to predictions, $\hat{Y}^F$. The model does multi-step predictions for the horizon h and the loss stated above captures this with the outer sum.

Moreover, for the classification task, the model outputs a probability distribution over all possible classes, C. In contrast to the forecasting samples, the input corresponds to the complete length, T. We also emphasize here the difference in parameters by denoting θ_c as the classification task parameters. The loss to be minimized with respect to predicting classes C for samples X^L:

$$L_c(X^L, \theta_c) = -\frac{1}{l} \sum_{i}^{l} log \left(\frac{e^{\hat{y}_{i=c}}}{\sum_j^C e^{\hat{y}_i}} \right) \tag{2}$$

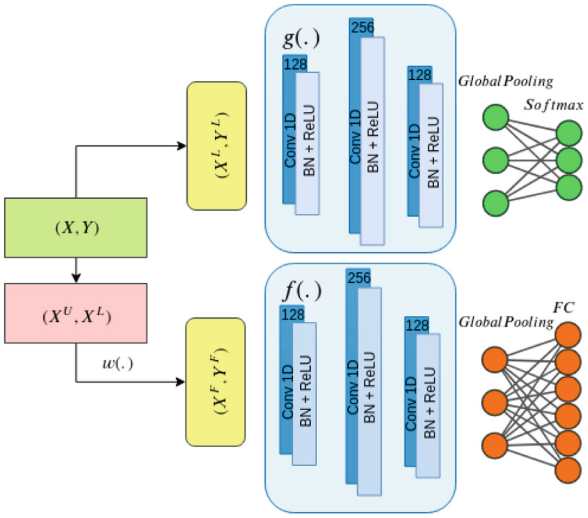


Fig. 1. The proposed multi-task model for joint forecasting and classification of time series. We adopt this architecture from [19] where it was shown to outperform variety of baselines on a majority of datasets. We reuse the same parameters for $f(.)$ up-to the last convolutional block, from where a dedicated linearly fully connected layer denoted by FC outputs for the horizon.

3.1 Forecasting as a Self-supervised Task

The core intuition to model forecasting as an auxiliary task is to force the ConvNet model to learn a set of rich hidden state representations from unlabeled but structured data. In the case of only few labeled instances being available as in semi-supervised setting, a fully-supervised approach can overfit on the training instances by learning a poor set of features which can hardly distinguish different classes. However, since training proceeds, by using the same set of features repeatedly the model can be more assuming of its predictions which would decrease the training loss in turn. In order to avoid this, a self-supervised task on unlabeled data could be leveraged that can learn comparatively more discriminative features for training and ultimately lead to a significant lift in accuracy on unseen data.

Additionally, forecasting is well-studied and easily formulated, but at the same time is complex enough which does not open any doors for cheating, as there are no trivial shortcuts for the model to exploit for solving the task [7]. Moreover, the task allows us flexibility in terms of data generation. By configuring the different values of the horizon and stride, h and s respectively, one could control the number of samples needed to configure an optimal balance between the classification and forecasting task samples.

3.2 Multi-task Learning Approach

Central to the theory of multi-task learning is the leveraging of hidden state representations from multiple tasks simultaneously in order to create a more robust model. This begs the question as to whether there exist tasks that could mutually benefit each other by sharing parameters between. Naturally, forecasting fits well with a classification task in a multi-task model as both tasks share the same input space. Moreover, the learned feature spaces are expected to be correlated in turn also [1].

However, designing a multi-task learning network poses two key challenges. Firstly, how to divide the feature space in shared and task-specific feature sets. Secondly, how to balance the weights between the different loss functions so as to distinguish between the main and auxiliary tasks. We rely on the hard-parameter sharing scheme, in which the learning parameters are all shared between the tasks up to the final fully connected layer in a layered architecture. From thereon, task-specific final layers output predictions for each task. This is illustrated in the Fig. 1 where we indicate shared parameters between the two tasks with same colored space. On the other hand, by adopting task specific weights we aim to cast the forecasting as an auxiliary task. We formulate the multi-task learning approach as an optimization process over the weighted sum of the two loss functions.

$$L_{MTL}(X^F, \theta_f, X^L, \theta_c) = L_c(X^L, \theta_c) + \lambda L_f(X^F, \theta_f) \tag{3}$$

λ is a hyper-parameter that controls parameter updates of the network relative to forecasting loss. It is thus crucial to tune for λ as too high of a value could bias the network weights for the forecasting task. On the other hand, if it is set too low, then the network would not learn for the forecasting task at all [1,10].

So far we have not drawn a link between the two feature sets of the tasks. In order to do so, consider that in a multi-task setting, the model is able to accurately forecast an unlabeled sample. The intuition is, if this unlabeled sample belongs to the same class as the very labeled sample the model is now trying to classify, and hence both are similar, then the latent features that were activated for the unlabeled sample could be leveraged to classify. Additionally, since the model is trained end-to-end, we also hypothesize that the model automatically learns to share latent representations between tasks and their corresponding high-order interactions based on this latent space.

4 Experiments

We compare our proposed multi-task model to multiple baselines on 13 real-world public time series datasets [3]. Since the data generating processes are completely different[1], the proposed method's performance can be judged without bias to similar data generating processes. Previously proposed methods were

[1] With exception to Lightning datasets.

compared on the same in [17]. A summary of the datasets is given in Table 1. All experiments were run with PyTorch and code[2] is published online to encourage reproducibility.

Table 1. Summary statistics of 13 real-world datasets from [3,17].

	Coffee	CBF	ECG	Face-Four	OSULf	Italy-Power	Light.2	Light.7	Gun-Point	Trace	Word-Syn	Olive-Oil	Star-Light
#	56	930	200	112	442	1096	121	143	200	200	905	60	9236
C	2	3	2	4	6	2	2	7	2	4	25	4	3
T	286	128	96	350	427	24	637	319	150	275	270	570	1024

4.1 Baselines

Wei's method [21] is based on self-training through which the classifier iteratively augments the labeled set by adding a sample from the unlabeled set. The choice as to which sample to add is based on the (nearest neighbour) classifier's prediction of which sample was the closest to any of its labeled counterpart in euclidean distance. The newly added sample is then given the same class as its closest neighbour.

DTW-D [2] is a meta-feature based distance. This distance was defined as the ratio of DTW to the euclidean distance. The intuition is to exploit the difference between the two distance's performance mainly the benefit of choosing DTW over the euclidean distance. Self-training is then carried out based on this distance.

SUCCESS [12] does constrained hierarchical clustering of the complete set of training samples, irrespective of labels. The distance metric utilized is DTW and all unlabeled samples are given the top-level seed's label.

Xu's method [22] is a graph theoretic SSL algorithm that constructs graphs relating all samples based on different distance functions such as DTW or Wavelet Transform. A probabilistic method optimally combines these various graphs after which a well studied harmonic Gaussian field based method [24] is adopted for label propagation.

Bag-of-words [18] leverages a sliding window procedure to generate local segments from time series data. These local segments are used to create histograms to train an SVM model for classification. It is worth noting that this method differs from above as it uses only labeled samples.

[2] https://github.com/super-shayan/semi-super-ts-clf.

SSSL [17] is the current state-of-the-art method in the field. It uses shapelets to classify unlabeled samples thereby producing pseudo-labels. A coordinate descent solver wraps the optimization process by iteratively solving for the classification of labeled samples, pseudo-labels and shapelets respectively.

Base [19] is a single-task variant of our proposed method that is only trained on the labeled samples to do classification.

Π-*Model* [11] is well-known semi-supervised learning method for image classification task. The basic idea is rooted in incorporating stronger regularization via ensembling. The method relies on dropout and asks the network being trained to output consistent labels for the same input. The input albeit goes through different dropout conditions leading to stochastic outputs. This makes it a well-defined task to exploit especially for unlabeled data. As the training proceeds, it is expected for the network's self-ensembled predictions to converge to the same labels for both labeled and unlabeled data. We sandwiched dropout layers after the batch-normalization layers and trained with dropout values of 20% and 40%.

Transfer Learning is common in the regime of self-supervised learning based methods [4,7]. Following these works, we train a non-linear classifier on top of each of a network's layers trained particularly for forecasting the datasets under consideration. The forecasting network in question is composed of stacking 6 convolutional layers in successive order with filter numbers 8, 16, 32, 64, 128, 256 respectively. Moreover, we sandwich maxpooling layers between halving the input in temporal dimension after each convolutional layer. Next, flattening and training 2 non-linear fully connected layers with dimensions 200 and 100. The very final layer's dimensionality corresponds to the horizon. This network is the result of an extensive grid search over multiple forecasting tasks from concurrent work. As we motivated, this network is geared towards forecasting in sharp contrast to the network in Fig. 1 adopted for the classification task. We trained this network with a grid search in $s \times h$ where, $s \in \{0.05, 0.1, 0.2\}$ and $h \in \{0.1, 0.2\}$[3], and used the configuration resulting with the least loss in Eq. 1.
This baseline serves to evaluate the self-supervised learned features from the forecasting task, by measuring classification accuracy that they achieve when we train a classifier on top of them without any fine-tuning [7]. This classifier has two non-linear layers corresponding to dimensions of 200 and 100 respectively. We hypothesize that if the features do correlate between the classification and forecasting task, then this non-linear classifier is expected to perform well.

5 Results

We begin this section by shedding light on the evaluation protocol. We randomly split each dataset into train and test splits with 80% and 20% of the samples.

[3] We overload the notation, and use s and h as percentages of the series length T.

Secondly, we split the train split further into labeled and unlabeled sets by randomly discarding 90% of the labels. This evaluation protocol is kept in line with the previous published methods, so as to report a direct fair comparison. The metric of evaluation is classification accuracy throughout the experiments. Given that the initial splits can bias the maximum achievable accuracy, similar to the works before ours, we report the maximum achieved accuracy on the test split by running the experiments 10 times with different hyper-parameters altogether.

Table 2 shows the comparison of accuracies for the proposed method and the baselines. The results are also stated for the best performing transfer learning scheme and the Π-model. A number of interesting observations can be drawn from these results. Firstly, we observe that our proposed method is able to outperform all other methods by a wide margin across almost all benchmark datasets considered. This is only made possible because of the exploitation of the unlabeled data better than other methods. Given that we consider here only 10% labeled samples, the difference between the performance of the methods boils down to how these cater for the unlabeled samples. By leveraging the forecasting task, the model is able to pick up useful representations that directly effect the final accuracy.

We observe that the proposed model fails to correctly model the underlying distribution of the *WordSynonyms* dataset. Firstly, we observe that in contrast to other datasets, the number of classes is the highest and hence we hypothesize that due to intra-class variances the model simply needs more labeled samples to model the underlying distribution. Moreover, we plot samples from the dataset together with our forecasts for last window of each of these samples in Fig. 2. Our second intuition is that given the dataset is based on handwriting, it can be a difficult task to forecast it correctly. The mean squared error that we achieve also hints in this direction, though exceptionally for these handwriting samples it does not convey explicit meaning.

Perhaps surprisingly, the Base method, is able to come at a close second. With exception to *WordSynonyms* and the *Lightning7* datasets, it is able to beat the current state-of-the-art by a considerable margin. We posit that this relates to the powerful non-linear modeling of the ConvNet in contrast to other algorithms. Also worth considering is that this architecture is the result of an extensive search over a wide variety of benchmark datasets as evident in [19]. Hence because of these reasons, even with few labeled samples available it is able to lead over rest of the baselines.

Additionally, we notice that the Π-model does not lead to fruitful results. Although the underlying architecture is the same as in *Base* model, we observed that the model was unfortunately not able to properly cater for the consistency regularization term for the unlabeled data. As a result, the multi-task loss it optimizes for diverges resulting in poor performance over the test splits not outperforming the *base* model. Despite initially considering to orthogonally integrate the Π-model with our proposed multi-task model, we refrained because of poor performance for this standard configuration.

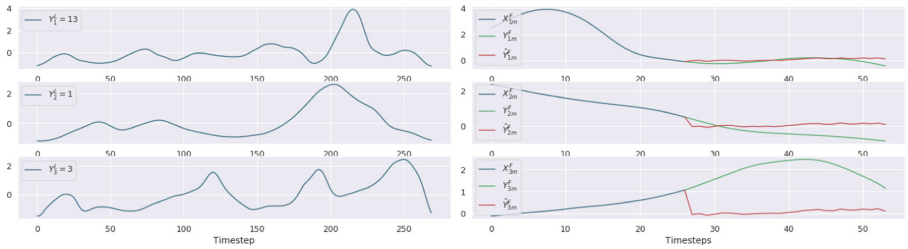
We also summarize the results for the transfer learning based approach which serves to quantify the usefulness of features learned purely for the self-supervised forecasting task. We can see that without any fine-tuning, by learning a non-linear classifier on top of the layers provides useful results. Although, results reported here are generalized over the layers, by reporting only the maximum possible accuracy regardless of the layer, it still serves as a validation of our initial hypothesis that feature spaces correlate heavily among the forecasting and classification tasks.

To point out the effect of hyper-parameters on our proposed multi-task model, we state the results with respect to different stride and horizon values in Table 3. It can be noted that for a subset of datasets the search was fruitful. Indeed, we find out that performance varies considerably with respect to the size of forecasting samples generated with particular stride and horizon, as it has direct effect on the learned representations.

We also wish to highlight further observations here though briefly. We observed that the final forecasting loss (Eq. 1) was consistent across s and h configurations albeit least at the extremities of stride and horizon values i.e those that lead to maximum possible forecasting samples. Also, we observed that network was robust to the different values of λ altogether. More importantly, as we posited earlier, the network performance was indeed biased to labeled samples resulted from random splits.

Table 2. The proposed method vs. baselines.

Datasets	Results verbatim from table in [17]						Proposed			
	Wei.	DTW-D	SUC.	Xu.	BoW	SSSL	Base	Π	Tr.	MTL
Coffee	0.571	0.601	0.632	0.588	0.620	0.792	**1.0**	**1.0**	**1.0**	**1.0**
CBF	0.995	0.833	0.997	0.921	0.873	**1.0**	**1.0**	**1.0**	0.784	**1.0**
cre ECG	0.763	0.953	0.775	0.819	0.955	0.793	0.9	0.875	0.9	**0.975**
FaceFour	0.818	0.782	0.800	0.833	0.744	0.851	0.913	0.913	0.739	**0.957**
OSULf	0.468	0.701	0.534	0.642	0.685	0.835	0.977	0.977	0.460	**0.978**
ItalyPower	0.934	0.664	0.924	0.772	0.813	0.941	0.986	0.986	0.959	**0.991**
Light.2	0.658	0.641	0.683	0.698	0.721	0.813	0.92	0.84	0.88	**0.92**
Light.7	0.464	0.503	0.471	0.511	0.677	0.796	0.758	0.689	0.482	**0.828**
GunPoint	0.925	0.711	0.955	0.729	0.925	0.824	**1.0**	**1.0**	0.825	**1.0**
Trace	0.950	0.801	**1.0**	0.788	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
WordSyn	0.590	0.863	0.618	0.639	0.795	**0.875**	0.497	0.491	0.342	0.519
OliveOil	0.633	0.732	0.617	0.639	0.766	0.776	0.916	**1.0**	0.833	**1.0**
StarLight	0.860	0.743	0.800	0.755	0.851	0.872	0.982	0.983	**1.0**	0.991



(a) First 3 samples from WordSynonyms (b) Predictions for last window

Fig. 2. We plot here the qualitative results for the forecasting task. We observe that the network is able to model the underlying distribution, albeit not perfectly.

Table 3. This table reports maximum accuracy for our proposed approach when marginalizing out horizon and stride from all runs and possible λ values.

Datasets	*s:*	0.05		0.1		0.2	
	h:	0.1	0.2	0.1	0.2	0.1	0.2
Coffee		**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
CBF		**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
ECG		0.950	0.9	0.925	0.9	**0.975**	0.875
FaceFr.		0.913	0.913	0.870	**0.957**	**0.957**	0.913
OSULf.		0.966	0.966	**0.978**	0.955	**0.978**	0.955
ItalyPower		0.986	**0.991**	0.986	0.986	0.982	**0.991**
Light.2		0.840	**0.920**	0.840	0.880	0.880	0.880
Light.7		**0.828**	**0.828**	0.759	0.759	0.793	0.759
GunPoint		**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Trace		**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
WordSyn.		0.497	**0.519**	0.508	0.497	0.503	0.508
OliveOil		**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
StarLight		0.983	0.983	0.99	0.97	0.983	**0.991**

5.1 Conclusion

We proposed a novel semi-supervised learning algorithm for time series classification based on a self-supervised feature learning task. We trained a ConvNet model that jointly classified and did auxiliary forecasting by sharing latent representations and learning high-order interactions end-to-end. As a result of exploiting the unlabeled data more effectively, our method was able to outperform state-of-the-art baselines. Future work includes extending our method to multivariate time series and researching additional ways to incorporate consistency regularization, which might yield better performance.

Acknowledgements. This work was co-funded by Volkswagen Financial Services through the Data-driven Mobility Services project.

References

1. Caruana, R.: Multitask learning. Mach. Learn. **28**(1), 41–75 (1997)
2. Chen, Y., Hu, B., Keogh, E., Batista, G.E.: DTW-D: time series semi-supervised learning from a single example. In: Proceedings of the 19th ACM SIGKDD, pp. 383–391. ACM (2013)
3. Chen, Y., et al.: The UCR time series classification archive (2015)
4. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE ICCV, pp. 1422–1430 (2015)
5. Doersch, C., Zisserman, A.: Multi-task self-supervised visual learning. In: Proceedings of the IEEE ICCV, pp. 2051–2060 (2017)
6. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.-A.: Deep learning for time series classification: a review. Data Min. Knowl. Disc. **33**(4), 917–963 (2019). https://doi.org/10.1007/s10618-019-00619-1
7. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728 (2018)
8. Grabocka, J., Schilling, N., Wistuba, M., Schmidt-Thieme, L.: Learning time-series shapelets. In: Proceedings of the 20th ACM SIGKDD, pp. 392–401. ACM (2014)
9. Grabocka, Josif, Schmidt-Thieme, Lars: Invariant time-series factorization. Data Min. Knowl. Disc. **28**(5), 1455–1479 (2014). https://doi.org/10.1007/s10618-014-0364-z
10. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE CVPR, pp. 7482–7491 (2018)
11. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242 (2016)
12. Marussy, K., Buza, K.: SUCCESS: a new approach for semi-supervised classification of time-series. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013, Part I. LNCS (LNAI), vol. 7894, pp. 437–447. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38658-9_39
13. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part I. LNCS, vol. 9905, pp. 527–544. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_32
14. Nguyen, M.N., Li, X.L., Ng, S.K.: Positive unlabeled learning for time series classification. In: Twenty-Second International Joint Conference on Artificial Intelligence (2011)
15. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part VI. LNCS, vol. 9910, pp. 69–84. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_5
16. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)
17. Wang, H., Zhang, Q., Wu, J., Pan, S., Chen, Y.: Time series feature learning with labeled and unlabeled data. Pattern Recogn. **89**, 55–66 (2019)

18. Wang, J., Liu, P., She, M.F., Nahavandi, S., Kouzani, A.: Bag-of-words representation for biomedical time series classification. Biomed. Signal Process. Control **8**(6), 634–644 (2013)
19. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: a strong baseline. In: 2017 international joint conference on neural networks (IJCNN), pp. 1578–1585. IEEE (2017)
20. Wei, D., Lim, J.J., Zisserman, A., Freeman, W.T.: Learning and using the arrow of time. In: Proceedings of the IEEE CVPR, pp. 8052–8060 (2018)
21. Wei, L., Keogh, E.: Semi-supervised time series classification. In: Proceedings of the 12th ACM SIGKDD, pp. 748–753. ACM (2006)
22. Xu, Z., Funaya, K.: Time series analysis with graph-based semi-supervised learning. In: 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 1–6. IEEE (2015)
23. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part III. LNCS, vol. 9907, pp. 649–666. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_40
24. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the 20th International conference on Machine learning (ICML-2003), pp. 912–919 (2003)
25. Zhu, X.J.: Semi-supervised learning literature survey. University of Wisconsin-Madison Department of Computer Sciences, Technical report (2005)