Self-Supervised Representations of Geolocated Weather Time Series - an Evaluation and Analysis

Arjun Ashok, Devyani Lambhate, Jitendra Singh

IBM Research India

arjun.ashok.psg@gmail.com, devyani.lambhate1@ibm.com, jitens@in.ibm.com

Abstract

Self-supervised learning (SSL) algorithms are gaining traction in various domains as a general paradigm of learning representations from data, largely outperforming supervised learning algorithms in tasks where labelled data is limited and costly to collect. In this work, we analyse existing self-supervised multivariate time series learning algorithms on their ability to learn representations of weather features, evaluating them on weather-driven downstream applications involving regression, classification and forecasting tasks. We experiment with a two-step protocol. In the first step, we employ an SSL algorithm and learn generic weather representations from multivariate weather data. Then, in the next step, we use these representations and train simple linear models for multiple downstream tasks. Through our experiments on air quality prediction tasks, we highlight the benefits of self-supervised weather representations. The benefits include improved performance across multiple tasks, the ability to generalize with limited in-task data, and a reduction in training time and carbon emissions. We highlight several areas of future work and the potential impact that such algorithms can have on real-world problems. We expect such a direction to be relevant in multiple weather-driven applications supporting climate change mitigation and adaptation efforts.

Introduction

The self-supervised learning (SSL) paradigm has gained popularity recently to extract effective representations from unlabeled data (Devlin et al. 2019; Brown et al. 2020). The idea is to *pretrain* models by defining learning tasks called *pretext tasks* on the unlabelled data itself without a specific target, and learning generic representations on the unlabelled data by training models to solve the pretext task. These representations are generic because the objectives they are trained to optimize are independent of any downstream task. These pre-trained models are then adapted/fine-tuned to a target task downstream and have often been observed to outperform their supervised counterparts on the downstream tasks. Further, in scenarios of limited labelled data, they showcase huge improvements (Chen et al. 2020c; Azizi et al. 2021). Self-supervised learning has already been demonstrated as an effective learning strategy in the domains of natural language processing (Devlin et al. 2019), computer vision (Chen et al. 2020a) and general time-series tasks (Zerveas et al. 2021). However, these techniques have not been aptly investigated in the context of weather data.

Weather data possess special characteristics such as being multivariate, geospatial, having non-linear relationships between weather attributes, etc. In the past few years, different types of weather data such as reanalysis, fore(hind)casts and observation data are available from multiple sources with a rich history. However, this readily available data is not effectively utilized in weather-driven applications, particularly for applications with limited labelled data. Such limited data scenarios arise in many domains: renewable energy generation forecasting when a new set of wind turbines are deployed on a farm, limited sensor observation for air pollution prediction, crop yield prediction with limited historical estimates available, and many more. Thus, self-supervised representations of weather data can be a powerful approach to address such practical problems across domains and applications. In this work, we focus on analysing the existing self-supervised techniques and evaluating the comparative benefits of self-supervised representations vs traditional end-to-end supervised approaches.

Related work

Self-Supervised Learning for Time Series

Self-supervised learning (SSL) has emerged as a general paradigm for training machine learning models. It is capable of adopting self-defined pseudo labels as supervision and use the learned representations for several downstream tasks, avoiding the cost of annotating large-scale datasets (Jaiswal et al. 2020). SSL has proven to achieve strong performance in the domains of computer vision (Chen et al. 2020b; Wang et al. 2021; Azizi et al. 2021), natural language processing (Gao, Yao, and Chen 2021; Logeswaran and Lee 2018) and speech recognition (Baevski et al. 2020; Xu et al. 2021). Recently, self-supervised learning approaches have been proposed for multivariate time series. SPIRAL (Lei et al. 2017) was the first to propose an unsupervised learning method with a simple objective, constraining the learned representations to preserve pairwise similarities in the time domain. Since then, the field has seen a lot more approaches (Malhotra et al. 2017; Wu et al. 2018). Some of these

methods (Franceschi, Dieuleveut, and Jaggi 2019; Tonekaboni, Eytan, and Goldenberg 2021; Eldele et al. 2021) assume transformation invariance properties and learn representations. TST (Zerveas et al. 2021) learns a transformer based model with a masked MSE loss. The TS2Vec objective(Yue et al. 2022) enforces the representation of a sub-series to be consistent in different augmented contexts, in both the instance-wise and temporal dimensions. The CoST objective(Woo et al. 2022) proposes to learn disentangled seasonal-trend representation using time-domain and a frequency-domain constrastive losses.

These techniques have so far been evaluated only on benchmark datasets, and have not been tested exclusively for learning representations of weather data, evaluating on real-world weather tasks. Such an investigation is necessary to realize the potential of self-supervised representation in weather-driven real-world problems.

Approach

We study self-supervised learning on weather data. Such a pretrained model can be later adapted to multiple downstream tasks. In theory, such representations can be adapted to any kind of task downstream. We experiment on regression and forecasting tasks. With any given weather dataset and corresponding set of tasks, we consider a two-stage protocol:

1. A pretraining phase: This is independent of any downstream task. A self-supervised time-series representation learning algorithm is trained on the weather features of the data such as temperature, humidity, precipitation, wind speed, wind direction, etc.
2. A finetuning phase: This is specific to every downstream task. The representations from the pre-trained model are utilized to train a simple model for the specific task(e.g. an SVM for classification, or a ridge regression model for a regression task).

Datasets

We conduct our initial experiments on tasks defined over two weather-driven problems: air quality prediction and air quality forecasting defined on the Beijing Multi-Site Air Quality Dataset (Zhang et al. 2017). The Beijing Multi-Site Air Quality Dataset (Zhang et al. 2017) contains 4 years of hourly air pollutants data from 12 nationally-controlled air-quality monitoring sites, matched with the nearest weather station. On the Beijing Multi-Site Air Quality Dataset (Zhang et al. 2017), we pretrain on 6 weather variables, and later, we define 4 supervised downstream tasks. The first two tasks are PM2.5 air quality regression and PM10 air quality regression. In these two tasks, the model is expected to output the target variable of a specific hour, given the weather features at the specific hour. The other two tasks are the *forecasting* variants of the same tasks. Here, the difference is that the model is expected to output the target variable for the next x hours, where x is set as 24, 48, 168, 336 and 720 in our experiments, denoting 1-day ahead, 2-days ahead, 1-week ahead, 2-weeks ahead, and 1-month ahead forecasting of the target variable respectively.

The Beijing Multi-Site Air Quality Dataset contains data from multiple sites in Beijing. We use the Changping site's data for our preliminary experiments. The data in the Beijing dataset is available from March 1st, 2013 to February 28th, 2017. We use a 60-20-20 train-validation-test split for our experiments. In our limited data experiments, we use the same splits for validation and test, but the train data is reduced to 50%, 40%, 30% and 20% where the X% data is taken from the end of the original training set. Therefore, in all our experiments, the train, validation and test splits are from consecutive dates.

The dataset contains weather variables such as temperature, pressure etc. as well as PM2.5 and PM10 air pollutant data. However, typically, given the geo-coordinates of a location, we can pull the weather data for that location from an external source. The weather variables and data distribution from this external source can be kept consistent across the finetuning tasks, removing the dependency of the model on distribution of the weather data available in the downstream tasks. Here, we use the geo-coordinates of the site to get multiple weather variables from ERA-5 reanalysis data (Hersbach et al. 2020) for the respective dates in the dataset. The 8 weather variables we use from ERA-5 are 10-meter-wind-towards-north, Atmospheric-water-content, 10-meter-wind-towards-east, Surface-pressure, Temperature, 100-meter-wind-towards-north, 100-meter-wind-towards-east, total-precipitation. Therefore, any SSL model is trained to take as input these 8 variables along with a 7-dimensional encoding of the date-time as covariates (therefore, 15 variables in total) and outputs a high-dimensional embedding. Similarly, a supervised model takes in the same 15 variables and is directly trained to output the specific target variable.

Implementation Details

For self-supervised learning/pretraining, we experimented with 2 algorithms from the literature: TS2Vec(Yue et al. 2022) and CoST(Woo et al. 2022). We use the official codebase of TS2Vec(Yue et al. 2022)[1] and CoST(Woo et al. 2022)[2] and for our experiments. Once the pretrained model is obtained, we use the representations from the model on the training data with a simple ridge regression model for every downstream task. For every downstream task, we perform a hyperparameter search for the regularization strength term α of the ridge regression model. We use an embedding dimension of 512 throughout our experiments.

For each downstream task, we benchmark our results against multiple supervised learning algorithms such as RNNs(Sherstinsky 2020), LSTMs(Hochreiter and Schmidhuber 1997), ResNet(He et al. 2016) and InceptionTime(Ismail Fawaz et al. 2020) models. We use the TSAI library (Oguiza 2022) which contains interfaces to many such time series models. For every supervised learning method, we perform an extensive hyperparameter search to ensure that the reported numbers are the best possible performances of each of the methods.

[1] https://github.com/yuezhihan/ts2vec

[2] https://github.com/salesforce/CoST

Methods	PM2.5 Forecasting					PM10 Forecasting				
	24-H	48-H	168-H	336-H	720-H	24-H	48-H	168-H	336-H	720-H
CoST (SSL)	**0.6914**	**0.7666**	**0.9301**	**0.9719**	**0.9954**	**0.6293**	**0.6951**	**0.8234**	**0.8594**	**0.8787**
TS2Vec (SSL)	0.7174	0.798	0.9571	0.9982	1.03	0.645	0.7145	0.8396	0.8733	0.9018
RNN (SL)	0.7623	0.7909	0.9737	1.017	1.018	0.68	0.7252	0.845	0.8881	0.9066
LSTM (SL)	0.7203	0.7809	0.9725	0.9999	0.9969	0.6566	0.7062	0.867	0.8717	0.9002
InceptionTime (SL)	0.7584	0.8606	0.9931	1.014	1.022	0.7153	0.7524	1.096	0.8719	0.8861
ResNet-18 (SL)	0.7658	0.8542	0.9755	1.013	0.9983	0.6744	0.7334	0.8812	0.8759	0.8894

Table 1: MSE on PM2.5 and PM10 Forecasting. SSL denotes self-supervised learning and SL denotes supervised learning.

Methods	PM2.5 Regression		PM10 Regression	
	MAE	MSE	MAE	MSE
CoST (SSL)	**0.5926**	**0.6788**	**0.5711**	**0.6199**
TS2VEC (SSL)	0.631	0.809	0.61	0.72
RNN (SL)	0.6635	0.8833	0.6262	0.8407
LSTM (SL)	0.6389	0.8484	0.6054	0.7364
InceptionTime (SL)	0.6625	0.9019	0.611	0.7401
ResNet-18 (SL)	0.662	0.9006	0.6405	0.7771

Table 2: Results on PM2.5 and PM10 regression tasks. SSL denotes self-supervised and SL denotes supervised.

The difference between training the self-supervised and supervised models is that in all our reported experiments with self-supervised models (marked SSL), only one model has been trained in common on the weather features. The representations from the same model have been used for any downstream task (eg. regression or forecasting at a horizon). This is in contrast to training a supervised model (marked SL) individually for each task. Hence, the total training time with SSL and N tasks is significantly less than N models independently trained over each of the tasks through an SL approach.

Results

Results on downstream tasks

Table 2 showcases the results of self-supervised and supervised learning approaches on the PM2.5 and PM10 regression tasks. Mean absolute error (MAE) and mean squared error (MSE) are used as metrics here. It can be clearly seen that CoST (Woo et al. 2022) outperforms all other methods in both the regression tasks. TS2Vec also outperforms all the supervised methods in terms of MSE in both the tasks, closely beating or equalling RNNs and LSTMs in the PM10 regression task. However, CoST (Woo et al. 2022) achieves much lesser MSE than other methods, and is consistently better than its self-supervised competitor TS2Vec (Yue et al. 2022).

Table 1 showcases the mean-squared error (MSE) on forecasting tasks, for various forecasting horizons. Figure 1 plots the MSE for the PM2.5 forecasting task across all considered horizons. The CoST (Woo et al. 2022) objective still outperforms all other models, across both tasks and across all forecasting horizons. TS2Vec (Yue et al. 2022), in most cases, outperforms the supervised baselines. The SSL models largely outperform the supervised counterparts in the short-range forecasting cases of 24 and 48 hour forecasting.

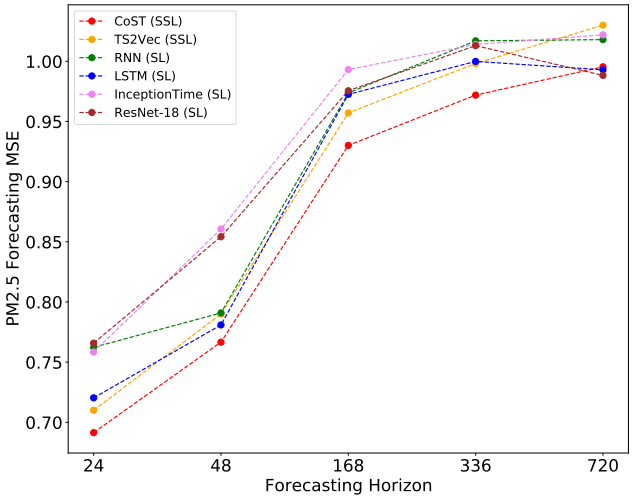

Figure 1: MSE on PM2.5 forecasting on various forecasting horizons

Varying the amount of labelled data in each downstream task

Next, we vary the amount of training data available in each task from 20% to 100%. When reducing the percentage of data available, we always sample from the end of the original 100% of training data. For instance, if the original training data is of 10 years from 2005 to 2015, then for an experiment with 80 percent training data, we use the data of 8 years from 2007 to 2015. The validation and test splits are kept consistent for all experiments throughout the paper.

Figure 2 showcases the MSE of all methods on the PM2.5 regression task, varying the amount of labelled data from 20% to 100%. It can be seen that, while, SSL methods also encounter drop in performances when the training data is brought down to certain levels (such as 40%), they still-incur a less sharper drop in performance, and outperform the supervised methods. Here, both TS2Vec and CoST consistently maintain strong performances under limited data conditions. The reason SSL methods maintain performance better is because of the pretraining stage where the initialized model itself offers strong representations on the data,

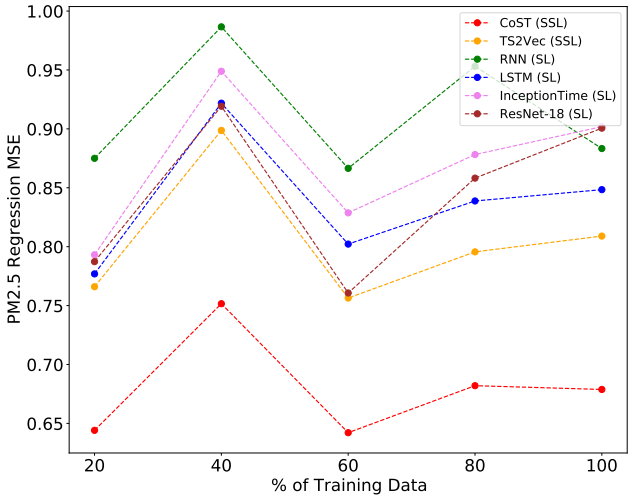

Figure 2: MSE on PM2.5 regression varying the amount of training data

hence the model is robust under limited data conditions downstream. The supervised models train from scratch on the downstream tasks and hence overfit in limited data setups.

Future Work

We find encouraging benefits of using self-supervised learning (SSL) to learn weather representations, evaluating them on weather-driven problems. We show how using SSL benefits and compares to supervised methods in typical weather tasks, and specifically under limited data conditions. As future work, we are investigating how existing time-series representations techniques generalize and perform in multi-geolocated weather data scenarios such as multi-site pollution and renewable energy generation prediction. An efficient representation of geo-locations and their characteristics in an efficient manner would be beneficial as the model would be able to leverage inter-location correlations during learning and inference.

References

Azizi, S.; Mustafa, B.; Ryan, F.; Beaver, Z.; Freyberg, J.; Deaton, J.; Loh, A.; Karthikesalingam, A.; Kornblith, S.; Chen, T.; et al. 2021. Big self-supervised models advance medical image classification. In *ICCV*, 3478–3488.

Baevski, A.; Zhou, Y.; Mohamed, A.; and Auli, M. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33: 12449–12460.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A Simple Framework for Contrastive Learning of Visual Representations. In III, H. D.; and Singh, A., eds., *ICML*, volume 119 of *Proceedings of Machine Learning Research*, 1597–1607. PMLR.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020b. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chen, T.; Kornblith, S.; Swersky, K.; Norouzi, M.; and Hinton, G. E. 2020c. Big self-supervised models are strong semi-supervised learners. *NeurIPS*, 33: 22243–22255.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Eldele, E.; Ragab, M.; Chen, Z.; Wu, M.; Kwoh, C. K.; Li, X.; and Guan, C. 2021. Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112*.

Franceschi, J.-Y.; Dieuleveut, A.; and Jaggi, M. 2019. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32.

Gao, T.; Yao, X.; and Chen, D. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hersbach, H.; Bell, B.; Berrisford, P.; Hirahara, S.; Horányi, A.; Muñoz-Sabater, J.; Nicolas, J.; Peubey, C.; Radu, R.; Schepers, D.; Simmons, A.; Soci, C.; Abdalla, S.; Abellan, X.; Balsamo, G.; Bechtold, P.; Biavati, G.; Bidlot, J.; Bonavita, M.; De Chiara, G.; Dahlgren, P.; Dee, D.; Diamantakis, M.; Dragani, R.; Flemming, J.; Forbes, R.; Fuentes, M.; Geer, A.; Haimberger, L.; Healy, S.; Hogan, R. J.; Hólm, E.; Janisková, M.; Keeley, S.; Laloyaux, P.; Lopez, P.; Lupu, C.; Radnoti, G.; de Rosnay, P.; Rozum, I.; Vamborg, F.; Villaume, S.; and Thépaut, J.-N. 2020. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730): 1999–2049.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Ismail Fawaz, H.; Lucas, B.; Forestier, G.; Pelletier, C.; Schmidt, D. F.; Weber, J.; Webb, G. I.; Idoumghar, L.; Muller, P.-A.; and Petitjean, F. 2020. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6): 1936–1962.

Jaiswal, A.; Babu, A. R.; Zadeh, M. Z.; Banerjee, D.; and Makedon, F. 2020. A survey on contrastive self-supervised learning. *Technologies*, 9(1): 2.

Lei, Q.; Yi, J.; Vaculin, R.; Wu, L.; and Dhillon, I. S. 2017. Similarity preserving representation learning for time series clustering. *arXiv preprint arXiv:1702.03584*.

Logeswaran, L.; and Lee, H. 2018. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*.

Malhotra, P.; TV, V.; Vig, L.; Agarwal, P.; and Shroff, G. 2017. TimeNet: Pre-trained deep recurrent neural network for time series classification. *arXiv preprint arXiv:1706.08838*.

Oguiza, I. 2022. tsai - A state-of-the-art deep learning library for time series and sequential data. Github.

Sherstinsky, A. 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404: 132306.

Tonekaboni, S.; Eytan, D.; and Goldenberg, A. 2021. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*.

Wang, X.; Zhang, R.; Shen, C.; Kong, T.; and Li, L. 2021. Dense contrastive learning for self-supervised visual pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3024–3033.

Woo, G.; Liu, C.; Sahoo, D.; Kumar, A.; and Hoi, S. 2022. CoST: Contrastive Learning of Disentangled Seasonal-Trend Representations for Time Series Forecasting. In *International Conference on Learning Representations*.

Wu, L.; Yen, I. E.-H.; Yi, J.; Xu, F.; Lei, Q.; and Witbrock, M. 2018. Random warping series: A random features method for time-series embedding. In *International Conference on Artificial Intelligence and Statistics*, 793–802. PMLR.

Xu, Q.; Baevski, A.; Likhomanenko, T.; Tomasello, P.; Conneau, A.; Collobert, R.; Synnaeve, G.; and Auli, M. 2021. Self-training and pre-training are complementary for speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3030–3034. IEEE.

Yue, Z.; Wang, Y.; Duan, J.; Yang, T.; Huang, C.; Tong, Y.; and Xu, B. 2022. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8980–8987.

Zerveas, G.; Jayaraman, S.; Patel, D.; Bhamidipaty, A.; and Eickhoff, C. 2021. A Transformer-Based Framework for Multivariate Time Series Representation Learning. KDD '21, 2114–2124. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383325.

Zhang, S.; Guo, B.; Dong, A.; He, J.; Xu, Z.; and Chen, S. X. 2017. Cautionary tales on air-quality improvement in Beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473.