Improving machine learning-based weather forecast 1 post-processing with clustering and transfer learning

Xiaomeng Huang[1], Yuwen Chen[1], Yi Li[1], Yue Chen[1], Chi Yan Tsui[1], Xing Huang[1], Mingqing Wang[1], and Jonathon S Wright[1]

[1]Tsinghua University

November 21, 2022

Abstract

Machine learning has been widely applied in numerical weather prediction, but the incorporation of new observational sites into models trained on stations with long historical records remains a challenge. Here we propose a post-processing framework consisting of three machine learning methods: station clustering with K-means, temperature prediction based on decision trees, and transfer learning for newly-built stations. We apply this framework to post-processing forecasts of surface air temperature at 301 weather stations in China. The results show significant reductions (as much as 39.4%˜20.0%) in the root-mean-square error of operational forecasts at lead times as long as 7 days. Moreover, the use of transfer learning to incorporate new stations improves forecasts at the new site by 36.4% after only one year of data collection. These results demonstrate the potential for clustering and transfer learning to boost existing applications of machine learning techniques in weather forecasting.

Improving machine learning-based weather forecast post-processing with clustering and transfer learning

Yuwen Chen[1], Xiaomeng Huang[1,2,3], Yi Li[1,3], Yue Chen[1], Chi Yan Tsui[3], Xing Huang[1,3], Mingqing Wang[1,3], Jonathon S. Wright[1]

[1]Ministry of Education Key Laboratory for Earth System Modeling, Department of Earth System Science, Tsinghua University, Beijing 100084, China
[2]Laboratory for Regional Oceanography and Numerical Modeling, Qingdao National Laboratory for Marine Science and Technology, Qingdao, 266237, China
[3]National Supercomputing Center in Wuxi, Wuxi, 214011, China

Key Points:

- A post-processing framework comprising clustering, decision tree, and transfer learning methods is employed to improve weather forecasts.
- This framework reduces the root-mean-square error by 27.9% (0.81°C) compared to operational ECMWF forecasts.
- Transfer learning improves forecasts by 36.4% at new stations with only one year of data available, reducing barriers to network expansion.

Corresponding author: Xiaomeng Huang, hxm@tsinghua.edu.cn

Abstract

Machine learning has been widely applied in numerical weather prediction, but the incorporation of new observational sites into models trained on stations with long historical records remains a challenge. Here we propose a post-processing framework consisting of three machine learning methods: station clustering with K-means, temperature prediction based on decision trees, and transfer learning for newly-built stations. We apply this framework to post-processing forecasts of surface air temperature at 301 weather stations in China. The results show significant reductions (as much as 39.4%∼20.0%) in the root-mean-square error of operational forecasts at lead times as long as 7 days. Moreover, the use of transfer learning to incorporate new stations improves forecasts at the new site by 36.4% after only one year of data collection. These results demonstrate the potential for clustering and transfer learning to boost existing applications of machine learning techniques in weather forecasting.

Plain Language Summary

Statistical approaches have been used for decades to enhance and interpret numerical weather forecasts. Artificial intelligence models have greatly advanced this field but the extension of these models to newly-built sites remains a challenge. To address this, we design a framework that combines three machine learning methods: clustering to group similar stations, decision trees to classify the forecasts, and transfer learning to adapt the model to new stations. We apply this framework to real forecasts and evaluate it against measurements from hundreds of weather stations in China. Station clustering and transfer learning both substantially improve predictions for recently-built sites, demonstrating how these tools can supplement existing artificial intelligence techniques in weather forecasting.

1 Introduction

The skill of numerical weather prediction (NWP) has improved significantly in recent decades due to advances in numerical models, data assimilation, and observation systems (Bauer et al., 2015). Nevertheless, the accuracy of NWP is still limited by imperfect model physics, numerical schemes, and initial/boundary conditions (Bauer et al., 2015; Lynch, 2008). Following the pioneering work of Glahn and Lowry (1972), Model Output Statistics (MOS) have been used operationally for over forty years. Raw model forecasts are post-processed using statistical relationships between observations and NWP results. However, the volume and variety of observational and model output data are increasingly overwhelming conventional implementations of these methods (e.g., Agapiou, 2017; Overpeck et al., 2011).

The emergence of machine learning (ML) techniques has provided new perspectives in this field (e.g., Reichstein et al., 2019). The climate community has increasingly turned to such techniques for applications such as improving subgrid-scale parameterizations in numerical models (e.g., Gentine et al., 2018; Rasp et al., 2018; Schneider et al., 2017; Jiang et al., 2018), improving forecasts at very short or very long lead times (e.g., Shi et al., 2015; Ham et al., 2019; B. Pan et al., 2019), detecting extreme weather (Hwang et al., 2019), and identifying complex teleconnection patterns (e.g., Runge et al., 2019; Boers et al., 2019). ML techniques could also substantially improve the accuracy of NWP results (McGovern et al., 2017; Rasp & Lerch, 2018; Scher, 2018).

The success of ML relies heavily on the quality and quantity of training data. Unfortunately, observations are usually sparse, especially for newly-built weather stations. Essential questions therefore arise regarding whether and by what means models trained on data-rich stations can be reliably extended to newly-built stations with limited data records.

Clustering techniques are widely used to extract information hidden in complex spatio-temporal data (Bador et al., 2015). Stations classified within the same cluster often share similar meteorological features. This type of feature-based classification provides a natural foundation for transfer learning, a technique by which knowledge gained in completing one task is repurposed for a different but related task (S. J. Pan & Yang, 2010). These methods may permit models trained for data-rich stations to be rapidly fine-tuned for application to data-poor stations. To take full advantage of these techniques, we propose a new framework that combines three different ML methods: Clustering, Decision trees, and Transfer learning, or CDT for short. We apply CDT to surface air temperature forecasts as an illustrative validation of this framework and its applicability.

2 Data

NWP data are provided by The International Grand Global Ensemble (TIGGE) project of the European Centre for Medium-Range Weather Forecasts (ECMWF) (e.g., Bougeault et al., 2010; Swinbank et al., 2016). The numerical forecasts are initialized twice per day at 00 and 12 UTC with lead times ranging from 6 to 168 hours at 6-hour increments (for a total of 28 lead times). We use data for the period from 1 January 2013 to 31 December 2018. The sample size is therefore 4384 for each weather station and lead time. Five variables are selected: temperature and dew point temperature at 2 m height, surface pressure, and the zonal and meridional wind components at 10 m height.

Observations from weather stations in China are obtained from `www.meteomanz.com` for the same period (1 January 2013 through 31 December 2018). As too few data are available in Xizang and Qinghai, we omit these areas from the analysis. We select 301 weather stations with data covering at least half of the year 2018 (the testing period as introduced below). Four variables (surface air temperature, surface pressure, surface air relative humidity, and near-surface wind speed) are provided every 3 hours (00, 03, 06, 09, 12, 15, 18, and 21 UTC). Static information for each station is also used, including latitude, longitude, and elevation. Missing values are filled via linear interpolation in the time dimension.

The historical observations are processed to generate feature vectors with shapes defined by $(n_{\text{samples}}, n_{\text{steps}}, n_{\text{features}})$, where n_{samples} is the number of records for a specified station, n_{steps} is the number of time steps used for temporal pattern mining, and n_{features} is equal to 4 (i.e., the number of measurements to match at each time step). For example, the shape of the input vector for the Beijing station is $(4384, 25, 4)$ when three days of past observations are used. NWP data are interpolated to each station location using an inversion-distance weighting ¡IDW;¿(Myers, 1994) applied to forecast data from the four nearest model grid cells. The observational and NWP data are combined for input to the CDT framework.

3 Methods

The CDT framework consists of three individual ML modules: clustering, decision-tree, and transfer learning. The clustering module classifies the 301 stations into groups using the traditional K-means technique. Separate decision-tree-based post-processing modules are then developed for each cluster and each lead time. Each newly-built station is assigned to the best-fit existing cluster. The transfer learning module is then used to produce the final results.

3.1 Clustering Stations with K-means

The traditional K-means (Hastie et al., 2009) clustering technique is often used for climate data analysis (e.g., Bador et al., 2015; Bernard et al., 2013). Stations with similar features are categorized into K individual clusters by calculating the feature distance

between them. The features used in this study are the annual averages and standard deviations of surface air temperature, surface air relative humidity, near-surface wind speed, surface pressure, latitude, longitude, and elevation. Models are established and trained for each cluster instead of for each station to reduce the computational cost and enlarge the training sample for each model.

The clustering result is highly sensitive to the value of K. We use the Silhouette Coefficient (Rousseeuw, 1987) to identify the optimal value of K. This metric measures the consistency of samples within each cluster as the ratio between cluster tightness and cluster dissociation. A larger Silhouette Coefficient indicates an increase in the inter-cluster distance relative to the intra-cluster distance. The maximum coefficient thus marks the optimal clustering result according to this metric.

The average Silhouette Coefficient (ASC; Text S1 in the supporting information) varies with the number of clusters K (Fig. 1a). We use the ASC to reduce the number of candidate K values so that we do not need to train ML models for all possible values of K. Although the ASC is useful for identifying potential optimal values of K, a larger ASC does not necessarily translate to a better ML model result. We test clusters based on $K = 2$, $K = 4$, and $K = 8$, which each produce climatologically coherent station groups. The result for $K = 2$ divides stations into two main groups corresponding to northern and southern China (Fig. 1b), while that for $K = 4$ produces clusters corresponding to the Northeast, North, and South regions along with some scattered stations (Fig. 1c). The scattered stations in cluster 3 are grouped because they experience much larger wind speeds than their geographic neighbors. The result for $K = 8$ further distinguishes some sub-regions with distinct climatological characteristics, such as the northwestern region and Yunnan Province (Fig. 1d).

3.2 Temperature post-processing based on LightGBM

After clustering, we apply a decision-tree model (Quinlan, 1986) to characterize relationships between the NWP forecasts and observations, correct biases, and identify how different features affect the prediction results. Decision trees are tree-like graph models. Information is passed from the root (representing the raw data) and split into branches at each level. The splitting rule is typically set by the variable that best discriminates among the samples along each branch. Decision trees produce naturally explainable outputs and can provide valuable insight into hidden relationships uncovered by the algorithm. This method has been successfully employed in a wide variety of weather applications (McGovern et al., 2017).

Gradient Boosting Decision Tree (GBDT; e.g., Chen & Guestrin, 2016) is a popular decision tree approach that involves an ensemble of sequentially-trained decision trees and gains knowledge by fitting negative gradients. In this work we use LightGBM (Ke et al., 2017), a highly efficient and scalable GBDT algorithm, to explore the relationships between NWP forecasts and observations in each cluster. LightGBM has been applied to sorting, classification, and regression tasks in a number of big-data studies (e.g., Cao & Gui, 2019; Ju et al., 2019). Adopting a leaf-wise growth strategy with depth limitation and gradient-based one-side sampling, LightGBM seldom overfits on small training datasets (Ke et al., 2017). More details on the LightGBM model and its implementation in this study are provided in Text S2 and Fig. S2 of the supporting information.

3.3 Transfer Learning for Newly-built Stations

In practice, ML models may malfunction due to data deficiencies or over-fitting. Transfer learning helps to reduce the likelihood of these types of failures by transferring knowledge from a previously trained model. The transferred model is then fine-tuned using newly-added data. This approach has been widely applied, including for the pre-

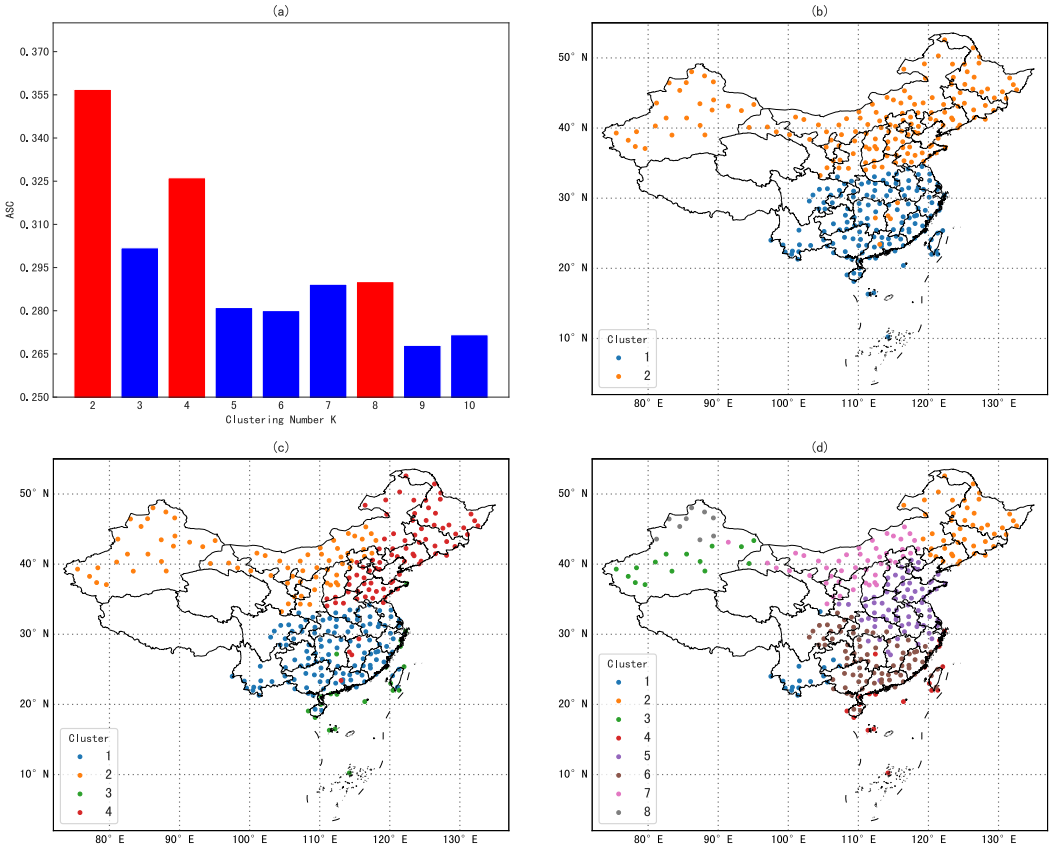

Figure 1. The effect of the number of clusters (K) on the clustering results. (a) The average Silhouette Coefficient (ASC, Text S1 in SI) as a function of K. Local maxima occur at $K = 2$, $K = 4$, and $K = 8$. (b) The spatial distribution of clusters for $K = 2$. (c) Same as (b) but for $K = 4$. (d) Same as (b) but for $K = 8$.

diction of wind speed (e.g., Hu et al., 2016; Qureshi & Khan, 2019). The LightGBM model for each cluster is taken as a pre-trained model, transferred and further trained on observations from newly-built stations identified as belonging to that cluster. The cluster to which each new station belongs is determined by static geolocation information along with the estimated annual means and standard deviations of key meteorological features (surface air temperature, pressure, wind speed, and relative humidity). The latter are IDW-interpolated from gridded NWP forecasts to accommodate the limited observational records at these stations. The refined LightGBM model is then applied to surface air temperature forecasts at the newly-built station.

4 Results

Data spanning the six-year period from 2013 to 2018 are divided into three parts. Data from 2013 to 2017 are used for training (80% of the data) and validation (the remaining 20%). All data for 2018 are used for testing. We construct a separate model to post-process ECMWF forecasts at each lead time (28 in all; Sect. 2) in each cluster. The benefits are most significant at short lead times, with error reductions as large as 39.4% (1.02°C) for 1-day forecasts (6∼24 h lead times; Table 1). Improvements decrease steadily to 20.0% (0.68°C) for 7-day forecasts (144∼168 h lead times). The average RMSE across all lead times is reduced by 0.81°C, corresponding to a 27.9% increase in accuracy. Clus-

tering improves the effectiveness of the decision tree algorithm, with the greatest error reduction achieved when stations are grouped into four clusters. Compared to models without clustering (i.e., a single model trained on all stations), the RMSE is reduced by 0.54% when two clusters are used ($K = 2$), 0.62% when $K = 4$, and 0.41% when $K = 8$. Since the $K = 4$ result produces the smallest RMSE, we adopt this model for all subsequent experiments. In addition to improving the overall forecast quality, clustering reduces the RMSE at 296 out of 301 individual stations (98.3%) when $K = 4$ (Fig. 2a).

Table 1 and Fig. 2 also show results for three alternative ML algorithms that are also widely used in meteorological applications (e.g., Gensler et al., 2017; Akram & El, 2016; Qing & Niu, 2018; Cao & Gui, 2019): linear regression (LR), artificial neural network (ANN), and long short-term memory (LSTM) with a fully-connected network (FCN). LR, ANN, and LSTM-FCN are used as control models to predict temperature using identical inputs. Detailed descriptions of the ANN and LSTM-FCN models are given in Text S3 and Figs. S3–S4 in the supporting information. The overall RMSE is reduced by 0.49°C (16.8%) under LR, 0.71°C (24.7%) under ANN, and 0.71°C (24.7%) under LSTM-FCN in the $K = 4$ scenario, including RMSE reductions at 211 stations under LR (Fig. 2b), 270 stations under ANN (Fig. 2c), and 272 stations under LSTM-FCN (Fig. 2d). LightGBM outperforms all three models, providing a further reduction of the RMSE for surface air temperature forecasts of 14.2% relative to LR, 3.8% relative to ANN, and 2.6% relative to LSTM-FCN, indicating that LightGBM is more effective for this application. LightGBM also takes less time for training (∼10 minutes) than ANN (∼20 minutes) or LSTM-FCN (∼40 minutes).

Table 1. RMSE of surface air temperature based on five different models for seven different lead times (Unit: °C). See text for details and definitions.

Lead Time	ECMWF	LightGBM	LR	ANN	LSTM-FCN
6∼24 h	2.59	**1.57**	1.94	1.63	1.60
30∼48 h	2.72	**1.83**	2.22	1.91	1.89
54∼72 h	2.83	**2.00**	2.37	2.10	2.09
78∼96 h	2.93	**2.15**	2.48	2.25	2.23
102∼120 h	3.05	**2.30**	2.60	2.40	2.39
126∼144 h	3.21	**2.49**	2.76	2.61	2.61
150∼168 h	3.41	**2.73**	2.95	2.85	2.95

Based on these findings, we conclude that LightGBM in combination with four clusters presents a substantial improvement over both the original operational forecasts and other ML-learning post-processing products. We therefore apply transfer learning to fine-tune the LightGBM model for extension to data-poor stations. To replicate the operational scenario, we randomly select 20% of the stations to serve as synthetic newly-built stations, using the remaining 80% stations to produce pre-trained models for each of the four clusters. We then fine-tune the pre-trained models using data covering between zero and 24 months at 2-month increments. The use of zero months of data corresponds to applying the pre-trained model directly without fine-tuning. We then evaluate the corrected forecasts for the 'new' stations using testing data from the year 2018. To validate the transfer learning results, we select seven lead times ranging from 24 h to 168 h at 24-h increments. The pre-trained models outperform the original NWP by 0.56°C (16.8%) even without fine-tuning (Fig. 3). The RMSE reduction continues to improve as the data span used for fine-tuning is extended, reaching 36.4% (1.23°C) when 12 months of data are used. Further improvements are negligible, indicating that the fine-tuning benefits plateau once the annual cycle is fully represented.

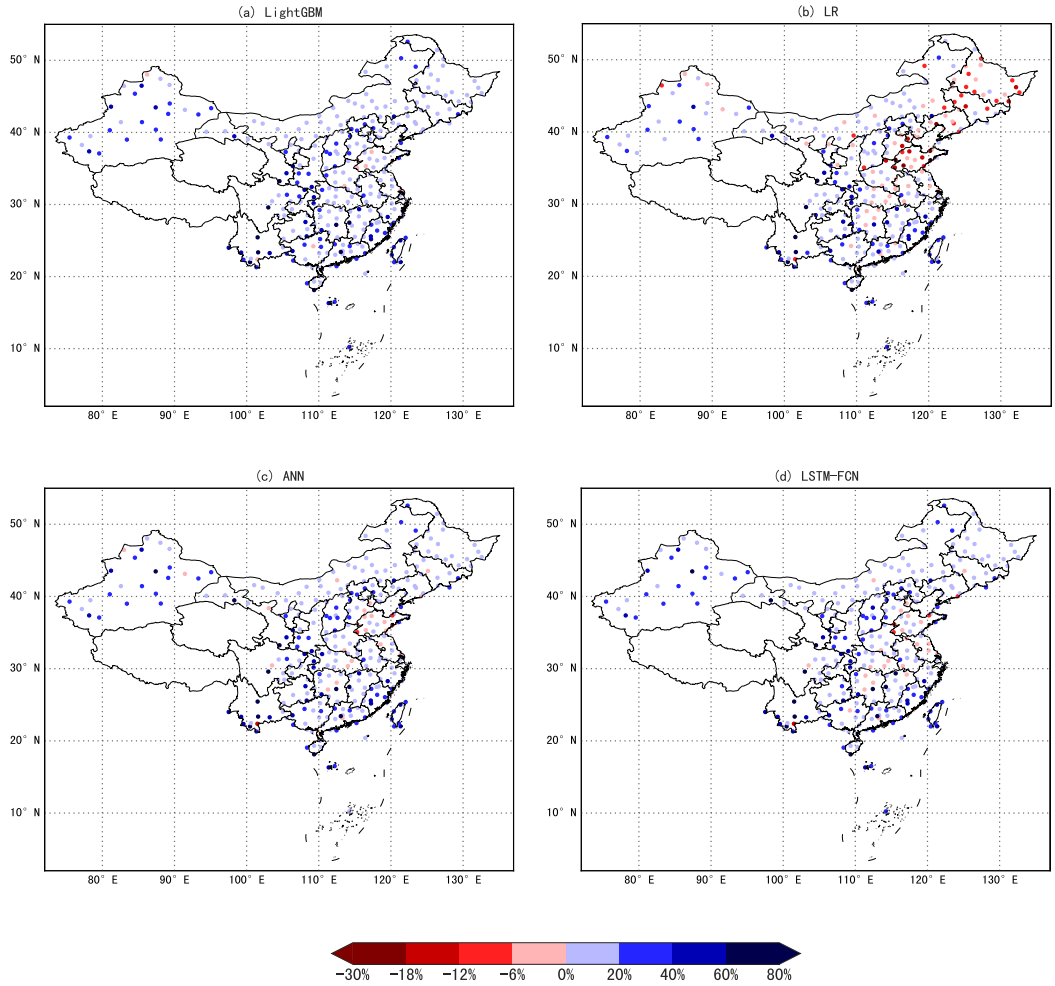

Figure 2. Model assessment for test data. (a) Spatial distribution of relative error reduction by the LightGBM model with four clusters. Blue colors indicate improvement; red colors indicate deterioration. (b) Same as (a) but for LR. (c) Same as (a) but for ANN. (d) Same as (a) but for LSTM-FCN.

LightGBM, as a GBDT variant, is a 'grey box' AI algorithm. Information gain, split times, and coverage rate can be calculated for each feature and used to explain the results (Gilpin et al., 2019). For example, the raw (NWP) surface air temperature forecast contributes the most information for most lead times and cluster members when $K = 4$ (Fig. 4). Temperature observations are the second most influential feature, but make only marginal contributions in most cases. For clusters where the RMSE of the operational ECMWF forecasts is already relatively small, such as cluster 2, the NWP forecasts account for a larger proportion of the overall influence. Conversely, observed temperatures play a larger role for clusters with larger RMSEs in the operational forecasts, such as cluster 4. The importance of the operational forecasts also increases as lead time increases, with concomitant reductions in the importance of the direct observations.

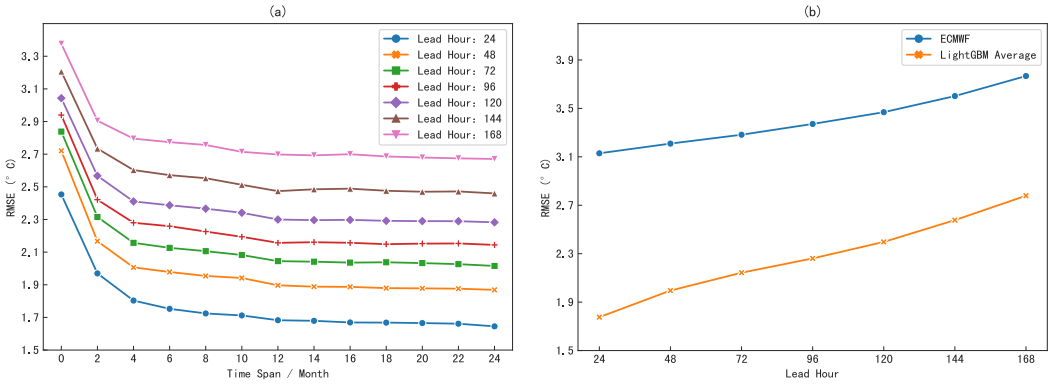

Figure 3. Results of transfer learning for the 60 sites randomly selected to serve as synthetic newly-built stations. The time span of training data used to fine-tune the model ranges from zero to 24 months, where zero months means the pre-trained model is used directly without fine-tuning. (a) RMSE values at seven different lead times using pre-trained models based on four clusters. (b) RMSE of the ECMWF forecasts and LightGBM post-processed results at seven different lead times. The LightGBM results reflect average RMSEs for training data time spans ranging from zero to 24 months.

5 Conclusion

ML algorithms show great potential for post-processing numerical weather forecasts, but their application is often restricted by the amount of available observations. In this paper we propose the CDT framework, based on clustering, decision tree, and transfer learning, and assess its performance in post-processing ECMWF forecasts of surface air temperature at lead times ranging from 6 to 168 h for 301 weather stations in China. The stations are first divided into two, four, and eight clusters, as these classifications produce climatologically and geographically meaningful station groupings. The CDT framework reduces the average RMSE of temperature forecasts at the 301 stations by up to 0.81°C (27.9%). These benefits are seen for all clustering scenarios and at all lead times, but the greatest improvements are for the 4-cluster scenario at 6–24 h lead times. Transfer learning aids the extension of models trained on data-rich stations to data-sparse stations within the same cluster. The RMSE at new stations is reduced by 16.8% (0.56°C) relative to the raw ECMWF forecasts even without fine-tuning, rising to 36.4% (1.23°C) once one year of observations is available for fine-tuning the algorithm. These improvements illustrate the great potential of the CDT framework for operational model post-processing, since newly-built sites typically suffer from short data records that restrict the application of AI techniques.

An attractive feature of decision tree-based models is that the results can be explained in terms of the contributions from each input feature. Here the main contribution is from the raw ECMWF forecast, especially at longer lead times. However, the station temperature observations are most important contributor for short lead times at stations in cluster 4, where the operational forecasts are less accurate than in other clusters. Overall, the CDT framework can help to correct prediction biases between NWP and observations, especially for newly-built stations or sites with sparse data records.

Acknowledgments

This work is based on data provided by the TIGGE project and `Meteomanz.com`. TIGGE (The Interactive Grand Global Ensemble) is an initiative of the World Weather Research

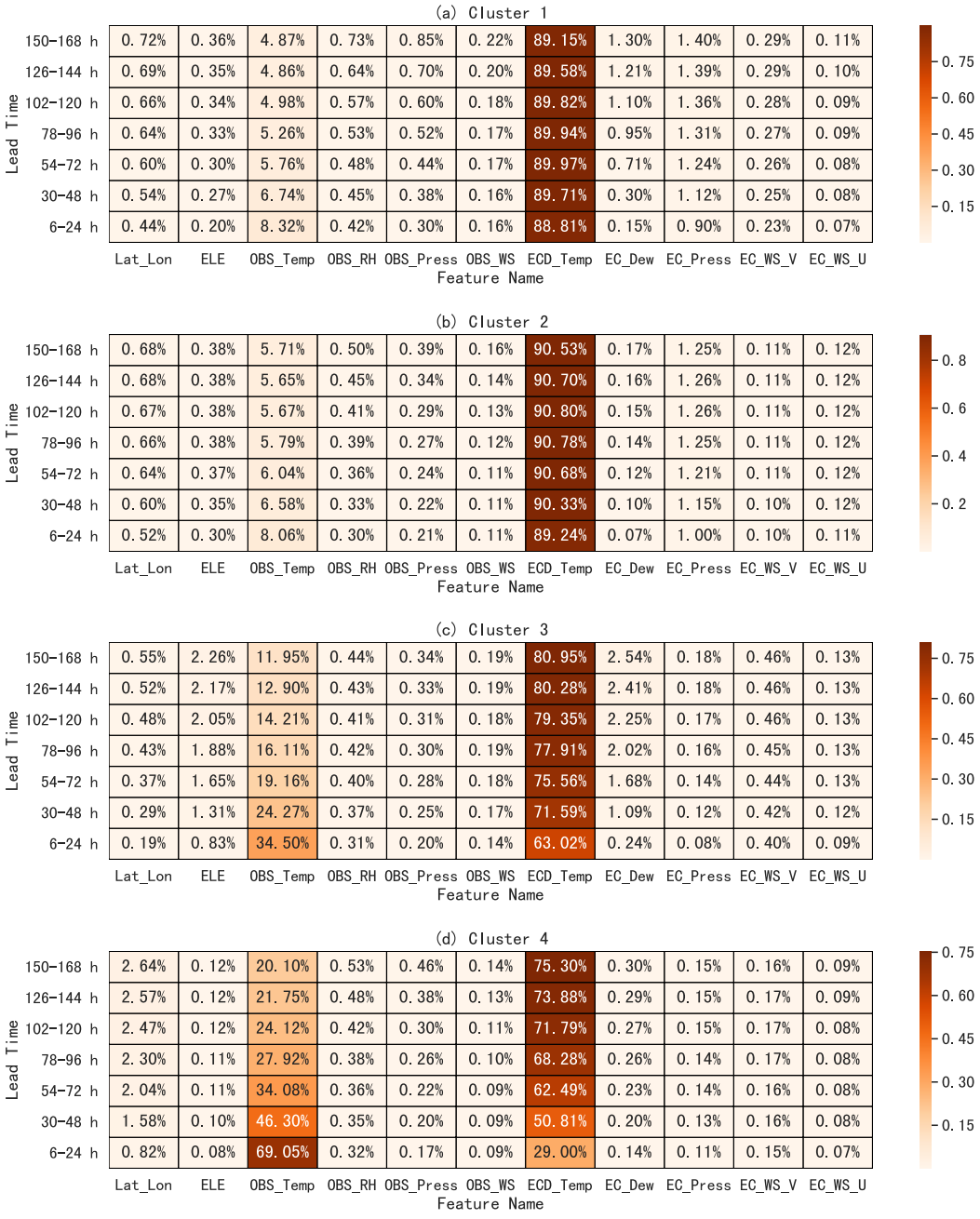

Figure 4. The relative importance of features at different lead times and for different clusters. The "EC" prefix indicates variables from the original ECMWF forecasts, while the "OBS" prefix indicates direct observations. Temp stands for temperature; RH for relative humidity; Press for surface pressure; WS for wind speed; dew for dew point temperature; WS_U and WS_V for the zonal and meridional components of wind speed, respectively; Lat_Lon for the latitude and longitude of the station; and ELE for the elevation of the station. The cluster numbers correspond to the $K = 4$ clustering result (Fig. 1c).

Programme (WWRP). `Meteomanz.com` collects observations released by official weather stations. The authors are sponsored by grants from the State's Key Project of Research and Development Plan (2016YFB0201100, 2017YFC1502200, 2018YFB0505000, 2018YFB1502800), the National Natural Science Foundation of China (41776010), and the Pilot National Laboratory for Marine Science and Technology (Qingdao)(QNLM2016ORP0108).

References

Agapiou, A. (2017). Remote sensing heritage in a petabyte-scale: satellite data and heritage Earth Engine© applications. *International Journal of Digital Earth*, *10*(1), 85–102. doi: 10.1080/17538947.2016.1250829

Akram, M., & El, C. (2016). Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks. *International Journal of Computer Applications*, *143*(11), 7–11. doi: 10.5120/ijca2016910497

Bador, M., Naveau, P., Gilleland, E., Castellà, M., & Arivelo, T. (2015). Spatial clustering of summer temperature maxima from the CNRM-CM5 climate model ensembles & E-OBS over Europe. *Weather and Climate Extremes*, *9*, 17–24. doi: 10.1016/j.wace.2015.05.003

Bauer, P., Thorpe, A., & Brunet, G. (2015). The quiet revolution of numerical weather prediction. *Nature*, *525*(7567), 47–55. doi: 10.1038/nature14956

Bernard, E., Naveau, P., Vrac, M., & Mestre, O. (2013). Clustering of maxima: Spatial dependencies among heavy rainfall in france. *Journal of Climate*, *26*(20), 7929–7937. doi: 10.1175/JCLI-D-12-00836.1

Boers, N., Goswami, B., Rheinwalt, A., Bookhagen, B., Hoskins, B., & Kurths, J. (2019). Complex networks reveal global pattern of extreme-rainfall teleconnections. *Nature*, *566*(7744), 373–377. doi: 10.1038/s41586-018-0872-x

Bougeault, P., Toth, Z., Bishop, C., Brown, B., Burridge, D., De Chen, H., ... Worley, S. (2010). The thorpex interactive grand global ensemble. *Bulletin of the American Meteorological Society*, *91*(8), 1059–1072. doi: 10.1175/2010BAMS2853.1

Cao, Y., & Gui, L. (2019). Multi-Step wind power forecasting model Using LSTM networks, Similar Time Series and LightGBM. *2018 5th International Conference on Systems and Informatics, ICSAI 2018*(Icsai), 192–197. doi: 10.1109/ICSAI.2018.8599498

Chen, T., & Guestrin, C. (2016, 3). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-Augu*, 785–794. doi: 10.1145/2939672.2939785

Gensler, A., Henze, J., Sick, B., & Raabe, N. (2017). Deep Learning for solar power forecasting - An approach using AutoEncoder and LSTM Neural Networks. *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings*(April), 2858–2865. doi: 10.1109/SMC.2016.7844673

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., & Yacalis, G. (2018). Could Machine Learning Break the Convection Parameterization Deadlock? *Geophysical Research Letters*, *45*(11), 5742–5751. doi: 10.1029/2018GL078202

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2019). Explaining explanations: An overview of interpretability of machine learning. *Proceedings - 2018 IEEE 5th International Conference on Data Science and Advanced Analytics, DSAA 2018*, 80–89. doi: 10.1109/DSAA.2018.00018

Glahn, H. R., & Lowry, D. A. (1972). *The Use of Model Output Statistics (MOS) in Objective Weather Forecasting* (Vol. 11) (No. 8). doi: 10.1175/1520-0450(1972)011⟨1203:tuomos⟩2.0.co;2

Ham, Y.-g., Kim, J.-h., & Luo, J.-j. (2019, 9). Deep learning for multi-year ENSO forecasts. *Nature*. doi: 10.1038/s41586-019-1559-7

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learn-*

ing (Vol. 27) (No. 2). New York, NY: Springer New York. doi: 10.1007/978-0-387-84858-7

Hu, Q., Zhang, R., & Zhou, Y. (2016). Transfer learning for short-term wind speed prediction with deep neural networks. *Renewable Energy*, *85*, 83–95. doi: 10.1016/j.renene.2015.06.034

Hwang, J., Orenstein, P., Cohen, J., Pfeiffer, K., & Mackey, L. (2019). Improving subseasonal forecasting in the western U.S. With machine learning. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2325–2335. doi: 10.1145/3292500.3330674

Jiang, G. Q., Xu, J., & Wei, J. (2018). A Deep Learning Algorithm of Neural Network for the Parameterization of Typhoon-Ocean Feedback in Typhoon Forecast Models. *Geophysical Research Letters*, *45*(8), 3706–3716. doi: 10.1002/2018GL077004

Ju, Y., Sun, G., Chen, Q., Zhang, M., Zhu, H., & Rehman, M. U. (2019). A model combining convolutional neural network and lightgbm algorithm for ultra-short-term wind power forecasting. *IEEE Access*, *7*(c), 28309–28318. doi: 10.1109/ACCESS.2019.2901920

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 3147–3155.

Lynch, P. (2008). The origins of computer weather prediction and climate modeling. *Journal of Computational Physics*, *227*(7), 3431–3444. doi: 10.1016/j.jcp.2007.02.034

McGovern, A., Elmore, K. L., Gagne, D. J., Haupt, S. E., Karstens, C. D., Lagerquist, R., ... Williams, J. K. (2017). Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bulletin of the American Meteorological Society*, *98*(10), 2073–2090. doi: 10.1175/BAMS-D-16-0123.1

Myers, D. E. (1994). Spatial interpolation: an overview. *Geoderma*, *62*(1-3), 17–28. doi: 10.1016/0016-7061(94)90025-6

Overpeck, J. T., Meehl, G. A., Bony, S., & Easterling, D. R. (2011, 2). Climate Data Challenges in the 21st Century. *Science*, *331*(6018), 700–702. doi: 10.1126/science.1197869

Pan, B., Hsu, K., AghaKouchak, A., & Sorooshian, S. (2019). Improving Precipitation Estimation Using Convolutional Neural Network. *Water Resources Research*, *55*(3), 2301–2321. doi: 10.1029/2018WR024090

Pan, S. J., & Yang, Q. (2010, 10). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359. doi: 10.1109/TKDE.2009.191

Qing, X., & Niu, Y. (2018). Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. *Energy*, *148*, 461–468. doi: 10.1016/j.energy.2018.01.177

Quinlan, J. R. (1986, 3). Induction of decision trees. *Machine Learning*, *1*(1), 81–106. doi: 10.1007/BF00116251

Qureshi, A. S., & Khan, A. (2019). Adaptive transfer learning in deep neural networks: Wind power prediction using knowledge transfer from region to region and between different task domains. *Computational Intelligence*, *35*(4), 1089–1113. doi: 10.1111/coin.12236

Rasp, S., & Lerch, S. (2018, 11). Neural Networks for Postprocessing Ensemble Weather Forecasts. *Monthly Weather Review*, *146*(11), 3885–3900. doi: 10.1175/MWR-D-18-0187.1

Rasp, S., Pritchard, M. S., & Gentine, P. (2018, 9). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, *115*(39), 9684–9689. doi: 10.1073/pnas.1810286115

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais,

366 N., & Prabhat. (2019, 2). Deep learning and process understanding
367 for data-driven Earth system science. *Nature*, *566*(7743), 195–204. doi:
368 10.1038/s41586-019-0912-1
369 Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and vali-
370 dation of cluster analysis. *Journal of Computational and Applied Mathematics*,
371 *20*(C), 53–65. doi: 10.1016/0377-0427(87)90125-7
372 Runge, J., Bathiany, S., Camps-Valls, G., Coumou, D., Deyle, E., Kretschmer,
373 M., ... Zscheischler, J. (2019). Inferring causation from time series
374 in Earth system sciences. *Nature Communications*, *10*(1), 2553. doi:
375 10.1038/s41467-019-10105-3
376 Scher, S. (2018). Toward Data-Driven Weather and Climate Forecasting: Approxi-
377 mating a Simple General Circulation Model With Deep Learning. *Geophysical*
378 *Research Letters*, *45*(22), 616–12. doi: 10.1029/2018GL080704
379 Schneider, T., Lan, S., Stuart, A., & Teixeira, J. (2017). Earth System Modeling 2.0:
380 A Blueprint for Models That Learn From Observations and Targeted High-
381 Resolution Simulations. *Geophysical Research Letters*, *44*(24), 396–12. doi:
382 10.1002/2017GL076101
383 Shi, X., Chen, Z., & Wang, H. (2015). Convolutional LSTM Network: A Machine
384 Learning Approach for Precipitation Nowcasting. *Nips*, 2–3. doi: 10.1007/978
385 -3-319-21233-3_6
386 Swinbank, R., Kyouda, M., Buchanan, P., Froude, L., Hamill, T. M., Hewson,
387 T. D., ... Yamaguchi, M. (2016). The TIGGE project and its achieve-
388 ments. *Bulletin of the American Meteorological Society*, *97*(1), 49–67. doi:
389 10.1175/BAMS-D-13-00191.1

Supporting Information for "Improving machine learning-based weather forecast post-processing with clustering and transfer learning"

Yuwen Chen[1], Xiaomeng Huang[1,2,3], Yi Li[1,3], Yue Chen[1], Chi Yan Tsui[3],

Xing Huang[1,3], Mingqing Wang[1,3], Jonathon Wright[1]

[1]Ministry of Education Key Laboratory for Earth System Modeling, Department of Earth System Science, Tsinghua University, Beijing 100084, China

[2]Laboratory for Regional Oceanography and Numerical Modeling, Qingdao National Laboratory for Marine Science and Technology, Qingdao, 266237, China

[3]National Supercomputing Center in Wuxi, Wuxi, 214011, China

Contents of this file

Introduction

Supporting information for the manuscript "Improving machine learning based weather forecast post-processing with clustering and transfer learning" is provided here. Text S1 introduces the calculation of Average Silhouette Coefficient (ASC), Text S2 introduces

details of the LightGBM as applied in this work, and Text S3 introduces the alternative machine learning frameworks used for comparison with the LightGBM results. Figure S1 shows a flow chart of our CDT framework, Figure S2 shows an example tree structure based on LightGBM, and Figure S3 and Figure S4 illustrate the structures of the LSTM-FCN and ANN models, respectively.

Text S1: Average Silhouette Coefficient

In this paper, we use the average Silhouette coefficient (ASC) as a guide to find viable candidates for the clustering number K. The ASC is calculated via the following steps:

(1) For data point i in cluster C_i, define

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j) \tag{1}$$

as the average distance from point i to the other points in cluster C_i, where $d(i, j)$ is the distance between point i and point j;

(2) Define

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j) \tag{2}$$

as the smallest average distance of point i to all points in any other cluster;

(3) Define the Silhouette of point i as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}; \tag{3}$$

(4) Define the ASC as the mean Silhouette of all points.

Text S2: LightGBM

We use the LightGBM model to post-process ECMWF temperature forecasts. Figure S2 provides an illustrative example of the LightGBM decision tree structure in the CDT

framework. Split features include ECMWF-predicted temperature (EC) and observations of temperature (T), pressure(p), wind speed (WS) and relative humidity (RH) measured at previous time steps. The path selects different branches in sequence depending on the split conditions, with the leaf value (ovals) returned as the final result. For the illustrated decision tree (Fig. S2), the value of leaf 10 (-0.176) is calculated using a series of criteria from the leftmost node $EC \leq 6.445$ to the rightmost node $T(t-1) \leq -0.450$). LightGBM provides the added advantages of rapid training speed, low memory usage, high accuracy, and parallel learning capacity.

LightGBM only supports two-dimensional structured datasets. Therefore, the observations are converted from the three-dimensional shape $(n_{\text{samples}}, n_{\text{steps}}, n_{\text{features}})$ to the two-dimensional shape $(n_{\text{samples}}, n_{\text{steps}} \times n_{\text{features}})$. When combined with the ECWMF data, latitude, longitude and elevation of stations, the final training data is organized in the shape $(n_{\text{samples}}, n_{\text{steps}} \times n_{\text{features}} + 8)$. LightGBM predictions represent the results of boosting all trees. The RMSE between the predicted result and observations collected at the valid forecast time is used to evaluate the prediction. To control overfitting, we tune the maximum tree depth to eight in this paper. This hyper-parameter provides the maximum depth that each tree is allowed to have. A smaller value indicates a weaker predictor.

Text S3: LSTM-FCN and ANN

Long-Short-Term-Memory (LSTM) models are widely used in time series prediction. In this paper we exploit temporal auto-correlation in observational time series in constructing a two-layer LSTM. The input is formatted as a three-dimensional array of the shape

($n_{\text{samples}}, n_{\text{steps}}, n_{\text{features}}$), and the LSTM output is merged with ECMWF forecasts using a four-layer, fully connected network (FCN). This deep neural network is then used as a control model. Fig. S3 shows the structure of our LSTM-FCN model.

As the most basic neural network frameworks, Artificial Neural Networks (ANNs) are also widely used for time series prediction. In this paper, we use a four-layer ANN as the control model. The input data to the ANN are the same as those provided to the LightGBM. Fig. S4 shows the structure of our ANN model.

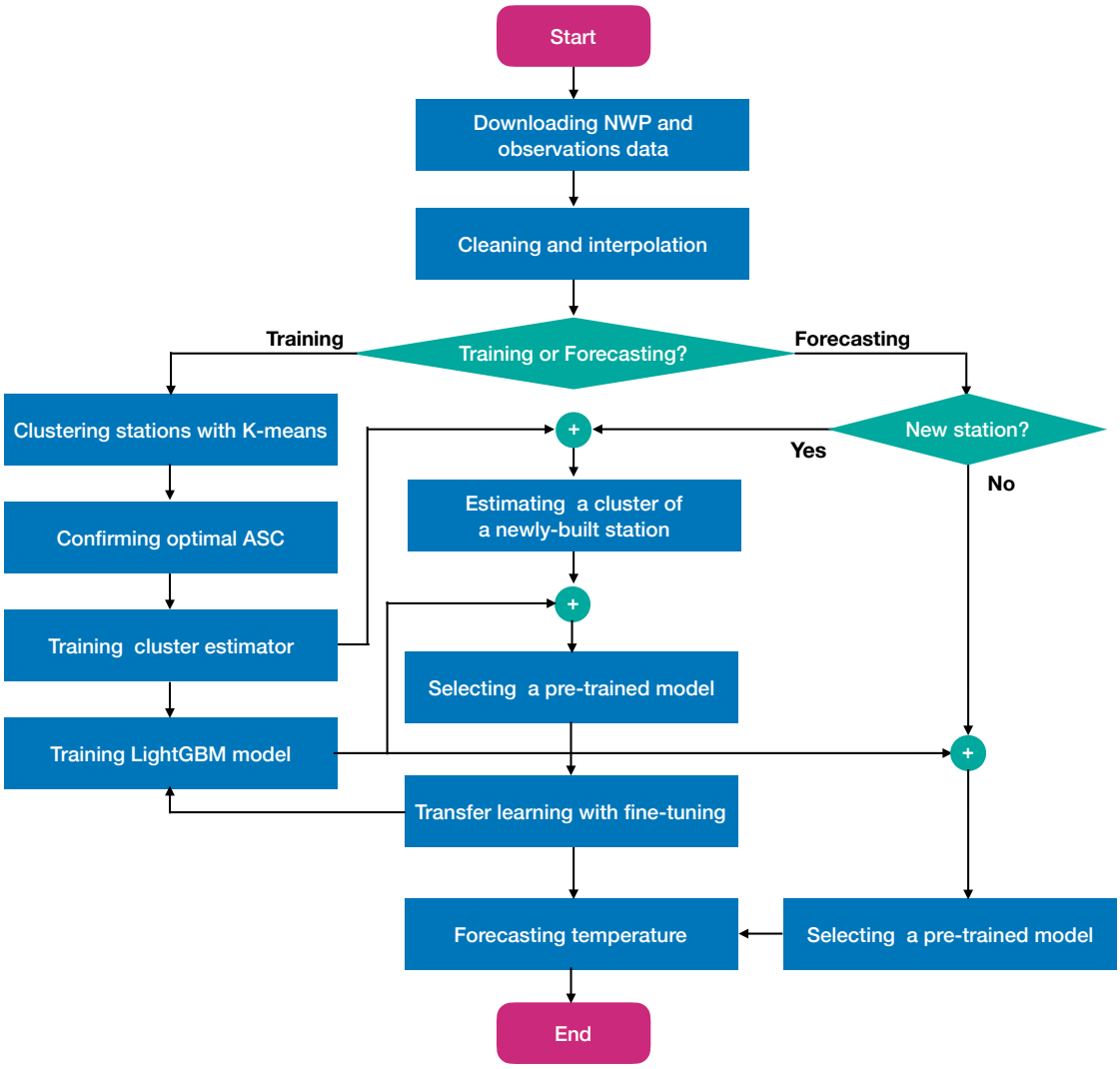

Figure S1. A flow chart of our CDT framework. Starting from the top, the NWP forecasts and observational data are downloaded and pre-processed. The training branch builds a 'cluster estimator' and groups the existing stations into clusters. Separate LightGBM models are then trained for each cluster and each lead time. Post-processed forecasts for existing stations are generated using the pre-trained model for the corresponding cluster. New stations are grouped into the best-fit existing cluster, after which the corresponding LightGBM model is fine-tuned to produce the final forecast.

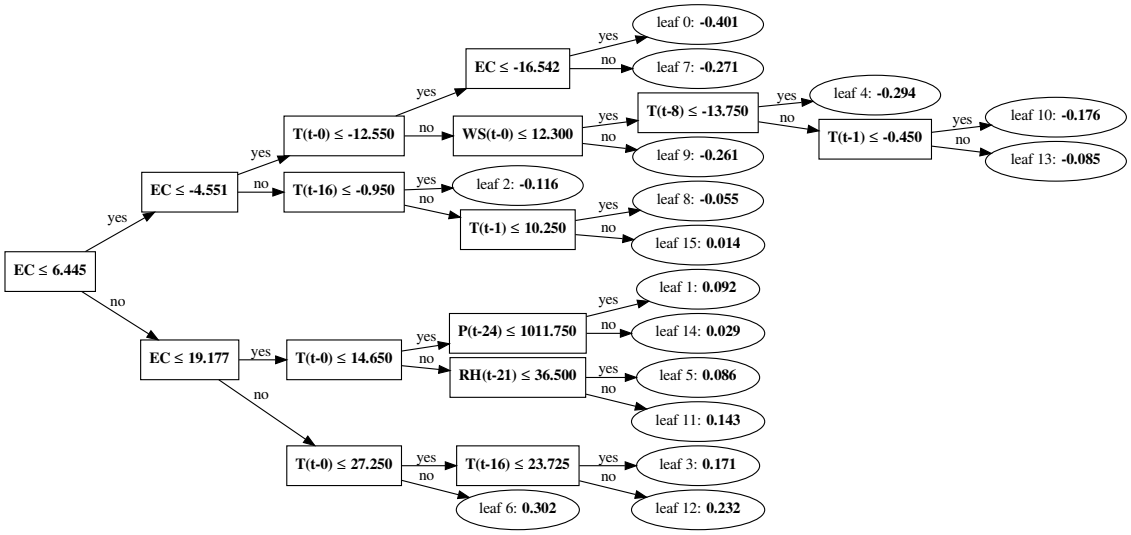

Figure S2. An example LightGBM decision tree. The features comprise the ECMWF result and observations of temperature (T), relative humidity (RH), pressure (p), and wind speed (WS) from a specified number of preceding time steps. Split features marked EC refer to the ECMWF prediction; $T(t-1)$ refers to the observed temperature one time step prior; $p(t-24)$ refers to the observed pressure 24 time steps prior; $\mathrm{WS}(t-0)$ refers to the observed wind speed at the current time; and $\mathrm{RH}(t-21)$ refers to the observed relative humidity 21 time steps prior. A LightGBM model consists of multiple such decision trees, and a LightGBM prediction is the result of boosting the returned leaf values from all trees.

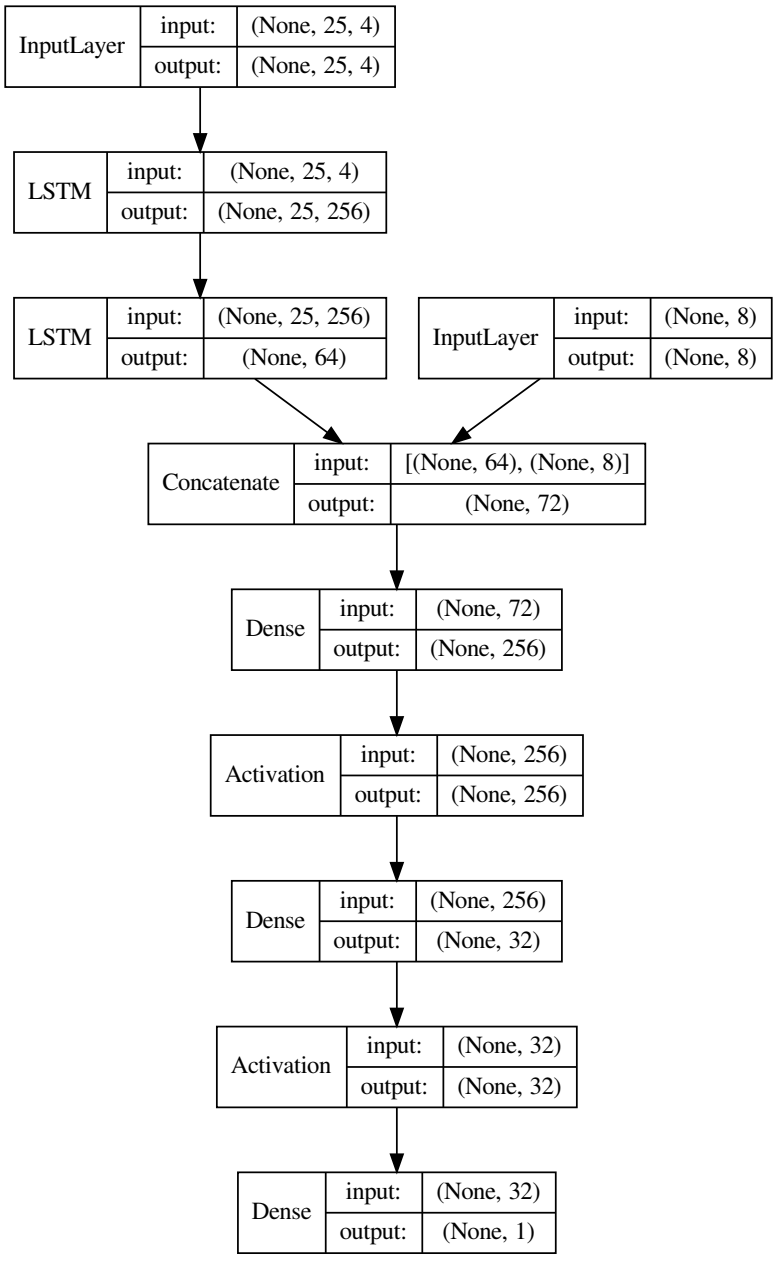

Figure S3. The structure of the LSTM-FCN model. The input consists of two parts, the time series of the observed data (left), and the inverse distance weighted (IDW) NWP result (right). The left input is in the shape of $(n_{\text{samples}}, n_{\text{steps}}, n_{\text{features}})$, where n_{steps} equals to 25, n_{features} equals to 4. The right input is in the shape of $(n_{\text{samples}}, 8)$, where the number 8 means 5 variables from ECMWF forecasts, and 3 location information of the stations (latitude, longitude and elevation).

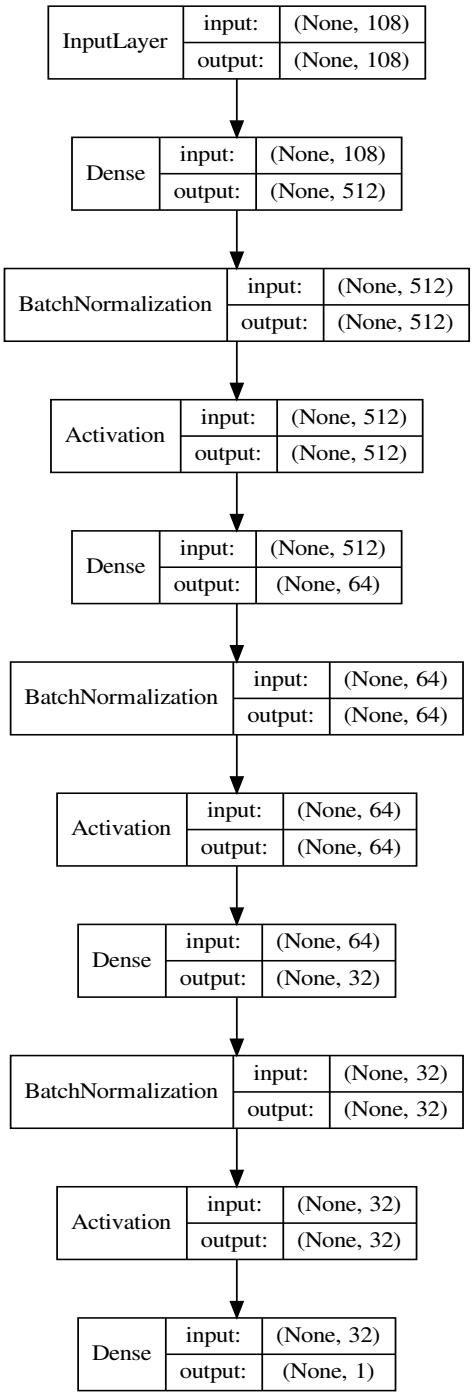

Figure S4. The structure of the ANN model.The input shape is $(n_{\text{samples}}, n_{\text{steps}} \times n_{\text{features}} + 8)$. Four FCN(Dense) layer with batch normalization function and Relu activation function are used to build this ANN model.