Deep shared representation learning for weather elements forecasting

Article *in* Knowledge-Based Systems · June 2019

DOI: 10.1016/j.knosys.2019.05.009

CITATIONS
13

READS
733

1 author:

Siamak Mehrkanoon
Maastricht University
89 PUBLICATIONS **525** CITATIONS

SEE PROFILE

Deep shared representation learning for weather elements forecasting

Siamak Mehrkanoon[1]

Department of Data Science and Knowledge Engineering, Maastricht University, Netherlands

Abstract

The accuracy and reliability of weather forecasting are of importance for many economic, business and management activities. This paper introduces novel data-driven predictive models based on deep convolutional neural networks (CNN) architecture for temperature and wind speed prediction in weather data. In particular, the proposed deep learning framework employs different upgrading versions of the convolutional neural networks i.e. 1d-, 2d- and 3d-CNN. The introduced models exploit the spatio-temporal multivariate weather data for learning shared representations using historical data and forecasting weather elements for a number of user defined weather stations simultaneously in an end-to-end fashion. The embedded feature learning component of the models as well as coupling the learned features of different input layers have shown to have a significant impact on the prediction task. The proposed models show promising results compared to the classical neural networks architecture used for modeling nonlinear systems. Two experimental setups have been considered based on a dataset collected from the Weather Underground website at six stations located in Netherlands and Belgium as well as a larger dataset with higher temporal resolution from the National Climatic Data Center (NCDC) at five stations located in Denmark. First, we focus on simultaneously predicting the temperature of two main stations of Amsterdam and Brussels for 1-10 days ahead. The second experiment concerns wind speed prediction at three weather stations located in Denmark for 6 and 12 hours ahead. The obtained numerical results show that learning new shared representations of the weather data by means of convolutional operations improves the prediction performance.

Key words: Deep learning, weather forecasting, convolutional neural networks, dimensionality reduction, representation learning

1. Introduction

Weather forecasting has recently gained the attention of many researchers due to its impact on the human life. The warning time for climate disasters may potentially save hundreds of lives every year. The importance of weather forecasting can also be seen in agriculture, e.g. for suitable planning of farm operations, transportation and storage of food grains among others. Knowledge of cyclones, tornados and heavy rains is necessary for life. If they are known a priori, many life losses will be saved. Adverse weather conditions and events can have direct and indirect effects on different transport or management sectors by for instance increasing transportation and time costs as well as accident risks. Weather forecasts can potentially facilitate efficient decisions making processes. Therefore, accurate or at least high-quality weather information is of benefit to operational short and long term plans. [25]. Weather forecasting primarily depends on the model-based methods such as simulation of dynamical systems governed by partial differential equations. Numerical weather prediction uses mathematical models derived from physical principles of the oceans and atmosphere to predict the weather variables based on current weather conditions. In particular, the atmosphere is modeled as a fluid. The present state of the atmosphere is sampled, and the future state is approximated by numerically solving the equations of fluid dynamics and thermodynamics [27]. The chaotic nature of these complex systems are represented by non-linear differential equations whose solutions are sought by numerical methods. The uncertainties in the initial conditions of the governed differential equations (i.e. measurement noise) and an incomplete understanding of complex atmospheric processes (i.e. process noise) may restrict the accuracy of weather forecast to a few days ahead and hence potentially become unreliable and computationally expensive for larger periods of time.

The use of machine learning techniques to address this data intensive challenge that involves inferences across time and space has recently gained a lot of attentions. As opposed to numerical weather forecast, instead data driven approaches are aiming at using historical weather observations to train a machine learning model that is able to map the exogenous input to a target output. Among machine learning models one can work with two types of architectures i.e. shallow and deep. Shallow machine learning techniques are limited in processing raw data and domain experts are required in transforming raw data into meaningful representations. On the other hand, deep learning based models have recently attracted many researchers due to their success in revolutionizing many application domains ranging from auditory to vision signal processing [2].

One of the advantages of deep learning models is that they engineer their own features during training. This is highly de-

[1]Corresponding author.
E-mail address:siamak.mehrkanoon@maastrichtuniversity.nl, siamak.mehrkanoon@esat.kuleuven.be, mehrkanoon2011@gmail.com

sirable, since one does not require domain expertise to be able to train an accurate model.

The superiority of deep architectures over the shallow ones in terms of accuracy in several application domains have been reported in the literature [2, 18, 23].

In particular, recent years have witnessed the emergence of convolutional neural networks (CNN) as a powerful model for addressing challenging tasks in computer vision. The history of CNN design goes back to LeNet-style models [19], which were stacks of convolutions for feature extraction and max-pooling operations for spatial sub-sampling. The authors in [14] refined these ideas and introduced the AlexNet architecture. Compared to conventional CNN architectures, AlexNet learns a richer feature representation at every spatial scale. AlexNet can also be considered as a turning point in view of CNN architecture as many researchers then followed a trend of making this type of network even deeper such as VGG architecture [31]. The emerging deep learning techniques together with the availability of massive weather observation data and the advancement of computer technology have motivated researches to explore hidden hierarchical patterns in the weather dataset. For instance the authors in [28] compared the prediction performance of deep learning architectures for the purpose of weather forecasting. A hybrid approach combining predictive models with a deep neural network that models the joint statistics of a set of weather-related variables is studied in [9].

It is the purpose of this paper to introduce new models for weather forecasting based on convolutional neural networks architectures. The proposed models can learn new feature representations of the given input data by exploiting its underlying spatio-temporal multi-modal characteristic. The first model learns new representations from historical data of individual weather stations using 1d-convolutional layers and afterwards fuse them before the output layers. This architecture can be regarded as multi-source convolutional NARX (Nonlinear AutoRegressive eXogenous) model. However, this approach might ignore learning features (representations) that could only be discovered by simultaneously considering historical data from all the available weather stations. Therefore, the second and third models explore the historical weather elements from multiple weather stations simultaneously and learn new predictive shared representations. To this end, 2d and 3d-convolutional operations have been employed. As opposed to the first model, the input of the second and third models preserve the spatio-temporal structure of the weather data and therefore new features are learned jointly by taking into account the underlying structure of the data.

This paper is organized as follows. A brief overview of the existing machine learning methodologies for weather forecasting is given in Section 2. The proposed deep convolutional neural network based architectures for weather forecasting are presented in section 3. The data collection process, the data description as well as the obtained experimental results are reported in Section 4. Conclusions and future works are drawn in Section 5.

2. Related works

Most of the research works in weather forecasting rely on the use of numerical methods for simulation of dynamical system describing the weather conditions [27, 13, 21]. Attempts have also been made in the literature to use machine learning models for weather forecasting. Among different methodologies, one can for instance mention time series analysis using autoregressive integrated moving average (ARIMA) models, shallow models based on support vector machines [26, 29] as well as artificial neural networks [4, 15, 22, 13]. The authors in [13] used a hybrid model based on neural networks to model the physics behind weather forecasting. Predicting weather conditions has been the focus of the work presented in [17]. Bayesian networks are also used to model and make weather predictions in [6]. Predicting severe weather conditions for a specific geographical location has been discussed in [1]. Attempts have also been made to predict different weather elements such as temperature, precipitation, speed of the wind among others [32, 15, 16].

Recently, deep learning techniques has become one of the most successful methodologies in a range of AI-related tasks such as computer vision, speech recognition and natural language processing. Authors in [34] proposed a convolutional LSTM model to predict the future rainfall intensity in Hong Kong over a relatively short period of time. A dynamic convolutional layer for short range weather prediction is introduced in [12]. Deep neural networks for ultra-short-term wind forecasting is discussed in [7]. The authors in [33] used Autoencoder for wind power prediction task. Motivated by the success of CNNs, here in this paper we introduce CNN based architectures and examine their performance in the context of temperature forecasting in weather data.

3. Proposed deep CNN architectures for weather forecasting

The weather datasets naturally follow spatio-temporal structure as each variable (weather element) is recorded in a specific time and location. One of the successful modeling strategies for multivariate time series analysis is NARX model where the next value of the dependent target signal is presented as a nonlinear function of previous values of the target as well as independent exogenous input signals as follows:

$$\begin{aligned} x(t+h) = & f(x(t), x(t-1), \ldots, x(t-n_x), \\ & u(t), u(t-1), \ldots, u(t-n_u)) \\ & + e(t+h), \end{aligned} \tag{1}$$

where $h \in 1, \ldots, H$ is the prediction horizon. Unlike the recursive strategy, this scheme does not accumulate errors as it does not use any predicted values for the subsequent predictions. Here, n_x and n_u are the lag parameters for the target and input signals respectively , i.e. the number of past observations in the time-series that are considered for the prediction task. The target and input signals at time t are denoted by $x(t)$ and

$u(t)$ respectively. Note that h also denotes the number of steps ahead to be predicted.

For weather forecasting applications, as per city there are several variables that are measured on a daily basis, therefore the input of the NARX model (1) for weather data will be a high dimensional vector. In addition, the regressor vector is potentially composed of some irrelevant variables to the target data. The key idea in this paper is to learn new shared representations of the weather data using convolutional neural network architectures by exploiting the weather data structure.

We assume that the number of weather stations is q, and the total number of weather elements (variables) is p. Furthermore, let $y_j^{s_i}(t)$ denotes the measurement corresponding to the j-th weather element of the i-th station at time t. If for instance we set the j-th weather element of the first station at time t as target variable, and also the lag parameter of both input and target signals to d, then one can construct the following regressor vector at time t:

$$\begin{aligned} z(t) =& [y_1^{s_1}(t-1), \ldots, y_p^{s_1}(t-1), \ldots, y_1^{s_1}(t-d), \ldots, y_p^{s_1}(t-d), \\ & \vdots \\ & y_1^{s_q}(t-1), \ldots, y_p^{s_q}(t-1), \ldots, y_1^{s_q}(t-d), \ldots, y_p^{s_q}(t-d)], \end{aligned} \tag{2}$$

which would be a vector of length $p \times q \times d$. Thus the problem is now reduced to finding a right mapping from the input vector $z(t)$ in (2) to the desired target variable $y_j^{s_1}(t)$ as follows:

$$y_j^{s_1}(t) = f(z(t)). \tag{3}$$

If one considers the available historical data, then the nonlinear mapping $f(\cdot)$ in (3) can be learned using training set which consists of several instances (see Figure 1).

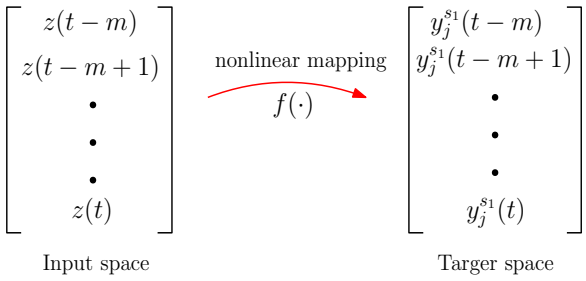

Figure 1: Nonlinear mapping from input weather data to a target data.

Remark 1. The above procedure explained the multi-input and single-output (MISO) learning framework. However, the same discussion also holds for multi-input and multi-output (MIMO) learning in the context of weather elements prediction. In particular, for MIMO learning the regressor vectors $z(t)$ can be used as it is, but the output vector i.e. the target space in Figure 1 should also include additional output variables.

In order to exploit the spatio-temporal structure of the input data, we first cast each regressor vector into a tensor with (stations, lags, variables) as (hight, width, channel), see Figure 2.

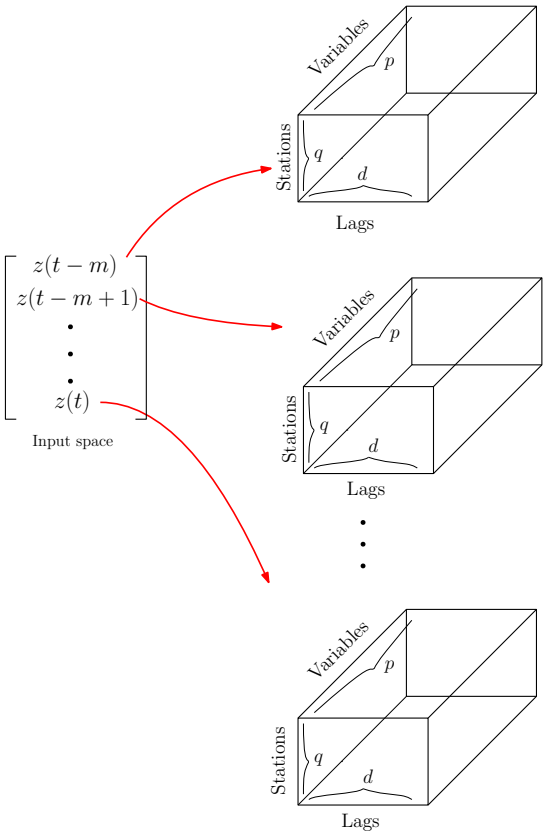

Figure 2: Tensorial input weather data to be used by the proposed CNN-based models.

In what follows, three convolutional neural network based architectures are proposed and examined for temperature as well as wind speed prediction using historical weather data. The first proposed model exploits the multiple-source characteristic of the weather data, i.e. the availability of measurements from different weather stations. If one extracts the corresponding data for a particular station form the tensorial data (Figure 2), then the data for each station can be casted in a vector of size $p \times d$. The model realizes new representations of the data by combining the information coming from different stations. In the second and third models, we directly work with the tensorial data shown in Figure 2, i.e. the input of the models are tensors with stations, lags and variables as their dimensions. The networks learn to extract shared predictive features from the given input data.

3.1. brief overview of convolutional layer

The main bulding block of a convolutional neural network (CNN) is Convolutional layer which aims at extracting the inherent feature in its previous layer by using convolutional operation. A convolution layer is composed of several convolution kernels (or filters) which are used to compute different feature maps. CNNs are a special type of classical feed-forward neural networks which are biologically inspired by the processing of the mammalian visual cortex. In particular, as opposed to fully connected neurons, in a CNN architecture each neuron of the feature map receives connections only from a subset of neurons

(also known as receptive field) in the previous layer. Therefore it is able to extract local spatial-temporal correlations from the input data. The new feature map is obtained by first convolving the input with a learned kernel and then applying an element-wise nonlinear activation function on the convolved results. In addition, to generate each feature map, the kernel is shared by all spatial locations of the input. Therefore, thanks to the weight sharing among neurons, CNNs are equipped with shift invariant property. Let $z^{\ell}_{i,j,k}$ denote the feature value at location (i, j) in the k-th feature map of the ℓ-th layer. Then $z^{\ell}_{i,j,k}$ can be calculated as follows:

$$z^{\ell}_{i,j,k} = f(w^{\ell}_{k}{}^{T} \; x^{\ell}_{i,j} + b^{\ell}_{k}) \tag{4}$$

where w^{ℓ}_{k} and b^{ℓ}_{k} are the kernel and bias term of the k-th filter in the ℓ-th layer respectively. Here $x^{\ell}_{i,j}$ is the input patch centered at location (i, j) of the ℓ-th layer and f is the nonlinear activation function. Commonly used activation functions in the literature are for instance sigmoid, hyperbolic tangent, Rectified Linear Units (ReLU) [24] and its variants, see [20], [10], [35] and [5].

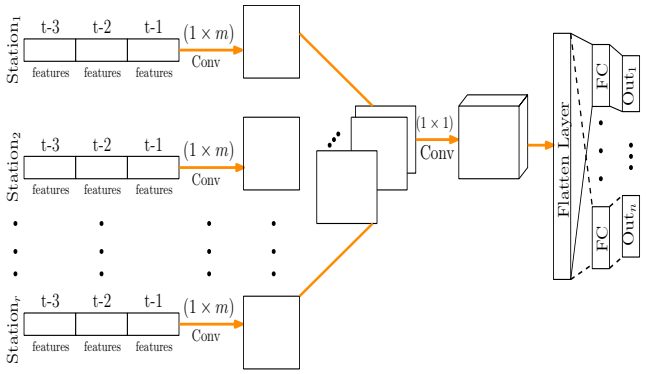

Figure 3: The 1d-convolutional neural network architecture corresponding to the proposed model in section 3.2.

3.2. *Multi-source one-dimensional CNN based model*

Considering that there are measurements from multiple cities, we aim at developing a model that can learn new representation of weather data from each city and leverage the fused learned features to predict the desired target. To this end, a multi-source convolutional neural networks architecture is designed. The network follows a multi-input, multi-output structure and a fusing layer is inserted to combine learned features from multiple inputs. A bank of one dimensional filters (kernels) are learned for each of the cities via applying 1d-convolutional layers. The learned feature maps are then merged via a fusing layer to foster sharing information coming from different sources. The output of this layer is fed to a (1×1)-convolutional layer, which applies the filters along the channel dimension to reduce the dimensionality of the merged features. The network is followed then by fully connected layers before reaching the targets, see Figure 3. The importance of having the convolutional layer at the beginning of the network is to find a new representation of the data which can potentially help in better describing the input-output relationship.

The number of filters, the dimension of the fully connected layers are hyper-parameters of the model which can for instance be tunned by cross-validation over a grid-search or alternatively one could also adopt a random search procedure [3]. Here, in order to save computational time, some of the parameters such as the depth of CNN were set fixed heuristically. Furthermore, the size of the input dimension depends on the number of weather stations, weather elements and the lag parameter (order of the model). Therefore as opposed to image analysis applications, in our case the input dimension of CNN is much smaller which drastically reduces the search space for (sub)-optimal filter size. Here we use 10-filters with length five in the convolutional layers for each weather station. In order to reduce the dimensionality of the concatenated learned maps obtained from convolutional layers, the fusing layer is followed by a two dimensional convolutional layer with filter size (1×1) applied along the channel dimension. The dimensions of the fully connected layers before the output layer are set to 50. In order to avoid over fitting the network is trained with an early stopping criterion. In this way, after maximum 20 epochs the training will stop if no improvement on the validation set is achieved. The layout of described network with lag=10 is given in Table 1.

Table 1: The outline of the proposed network architecture in section 3.2

Layer Type	**(Filter size) × # Filters**	**Output size**
Input		(1940, 50, 1)
Input		(1940, 50, 1)
Input		(1940, 50, 1)
Input		(1940, 50, 1)
Input		(1940, 50, 1)
Input		(1940, 50, 1)
Conv + ReLU	$(1, 5) \times 10$	(1940, 46, 10)
Conv + ReLU	$(1, 5) \times 10$	(1940, 46, 10)
Conv + ReLU	$(1, 5) \times 10$	(1940, 46, 10)
Conv + ReLU	$(1, 5) \times 10$	(1940, 46, 10)
Conv + ReLU	$(1, 5) \times 10$	(1940, 46, 10)
Conv + ReLU	$(1, 5) \times 10$	(1940, 46, 10)
Merge		(1940, 46, 60)
Conv + ReLU	$(1, 1) \times 2$	(1940, 46, 2)
Flatten		(1940, 920)
Dense + ReLU		(1940, 50)
Dense + ReLU		(1940, 50)
Dense (linear)		(1940, 1)
Dense (linear)		(1940, 1)
Total number of trainable parameters: 92, 576		

Remark 2. The proposed architecture in section 3.2. can be regarded as multi-source convolutional NARX model where the Hankel matrices corresponding to each of the weather sta-

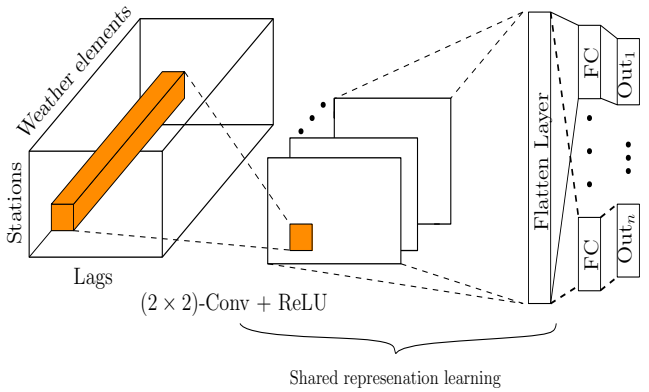

Figure 4: The 2d-convolutional neural network architecture corresponding to the proposed model in section 3.3.

tions are first constructed and then fed into multi-source 1d-convolutional neural networks. Thanks to the convolutional layer predictive features are learned from each source and are fused at the end of the network architecture before the output layer. However, it should be noted that this approach might ignore learning features that could only be discovered by simultaneously considering historical data from all the available weather stations.

3.3. Two and three dimensional CNN based models

Unlike the previously described model in section 3.2, here we present a model that learns a bank of two or three dimensional kernels that are applied on the tensorial data (see Figure 2) containing the measurements of all the weather stations. In the case of two dimensional convolutions, new shared representations of the data are obtained by convolving the learned kernels over hight (weather stations) and width (lags) dimensions. In this way, as the network takes into account the spatial relationship between different neurons with respect to the input weather data, it learns the local spatial and temporal correlations among the measurements.

In particular, the input data is fed to (2×2)-convolution layers with 10 filters and ReLU nonlinear activation function where the new shared features are learned. The obtained feature maps are then flattened and the network is followed by fully connected layers with ReLU and linear activation function respectively. The layout of described network with lag=10 is given in Table 2. It should be noted that the first layer shown in Figure 4 could be repeated many times in the network architecture using a stacking scheme. In the case that one have access to large data sets, this approach could potentially learn even more abstract representation of the weather data.

It should be noted as we have casted the Hankel matrix into a tensorial format with (hight=stations, width=lags, channel=variables), it is also possible to apply 3-dimensional convolutions to learn the new representations from data. In that case one can replace the 2d-convolution operations with 3d-convolutions. In 2d-convolution we do not slide filters along the channel dimension whereas 3-dimensional filters will be slided along the three dimensions. It is expected that given enough

Table 2: The outline of the proposed 2d-CNN network architecture described in section 3.3.

Layer Type	**(Filter size) × # Filters**	**Output size**
Input		(1940, 6, 10, 5)
Conv + ReLU	$(2, 2) \times 10$	$(5, 9, 10)$
Flatten		$(1940, 450)$
Dense + ReLU		$(1940, 100)$
Dense + ReLU		$(1940, 100)$
Dense (linear)		$(1940, 1)$
Dense (linear)		$(1940, 1)$
Total number of trainable parameters: $90,612$		

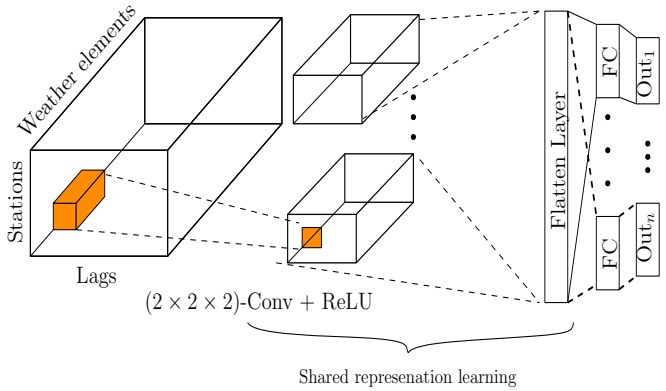

Figure 5: The 3d-convolutional neural network architecture corresponding to the proposed model in section 3.3 for weather elements forecasting

data, potentially the three dimensional CNN model learns existing correlations among channel space as well. The architecture of the proposed 3d-CNN model for weather forecasting is depicted in Figure 5. Here, the input data is fed to $(2 \times 2 \times 2)$-convolution layers with 10 filters followed by ReLU nonlinear activation function. The obtained feature maps are then flattened and the network is followed by fully connected layers with ReLU and linear activation functions respectively.

Remark 3. As previously mentioned in Remark 2, here the proposed architectures can also be regarded as two and three-dimensional convolutional NARX models where the constructed Hankel matrix corresponding to historical data of the weather stations is first casted into tensorial input. The introduced convolutional architectures explore the historical weather elements from multiple stations simultaneously and learns new predictive shared representations.

4. Numerical Experiments

In this section, we perform a set of experiments to evaluate the proposed models for temperature as well as wind speed prediction using historical weather elements. To this end, two datasets are collected. The first dataset which will be used for temperature prediction is collected in a daily basis from

Weather Underground website at six stations located in Netherlands and Belgium. The second dataset has a higher temporal resolution and is collected from the National Climatic Data Center (NCDC) in an hourly basis at five stations located in Denmark. Here the Mean Absolute Error (MAE) defined as:

$$\mathrm{MAE} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \left| z^i_{\text{true}} - z^i_{\text{prediction}} \right|$$

is used for evaluation of the performance of the forecasting models. The real and predicted target test variables are denoted by z^i_{true} and $z^i_{\text{prediction}}$ respectively and n_{test} is the number of test instances. A pre-processing step containing the following data scaling has been first employed to scale each dimension of the data between zero and one.

$$x_{\text{scaled}} = \frac{x - \min(x)}{\max(x) - \min(x)}.$$

4.1. Temperature Prediction

In the first experimental setup, we focus on predicting the future temperature, from one to ten days ahead, for Amsterdam and Brussels using available historical weather variables. In particular, the measurements from six weather stations located in Brussels, Liege, Antwerp, Amsterdam, Eindhoven and Groningen are used. In the experiments, we highlight the advantage of the convolutional layer in the network pipeline for learning more abstract representation of the weather variables as well as dimensionality reduction. Moreover, we compare the performance of the proposed models with NARX model and Long Short Term Memory (LSTM) networks (see Figure 6). The LSTMs are a special case of Recurrent Neural Networks, introduced in [11] and it is capable of learning long-term dependencies. For more details of LSTM networks and their variants one may refer to [30] and references therein.

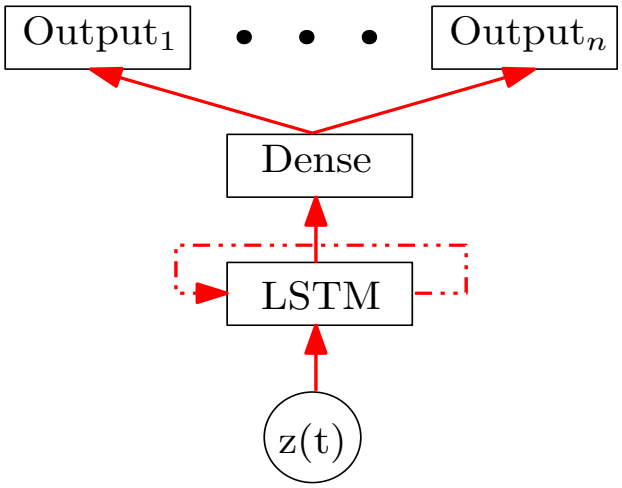

Figure 6: The LSTM architecture used in our analysis.

The data include real measurements of weather elements, for each day and for each city, in the period starting from 2009 till mid-November 2015. In total five weather elements including minimum and maximum temperature, dew point, precipitation, wind speed are measured. The weather stations at which the weather data has been recorded are shown in Figure 7.

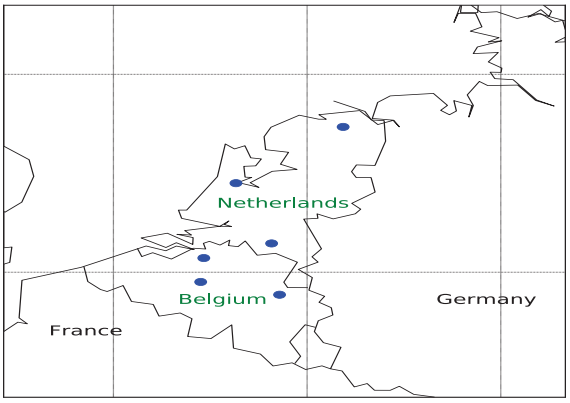

Figure 7: The location of the stations at which the weather data are collected.

For the temperature prediction task, in all the conducted experiments, the test sets used to evaluate the performance of the proposed models consist of the last 3% of the data. while the remaining percentage of the data is used for training the models. The test data corresponds to the time period from September, 2015 to mid-November, 2015. Furthermore, in our analysis for temperature prediction, the lag parameter d is chosen from the range $[4, 6, 8, 10]$ equally for both input and target signals. This means that for instance if the lag parameter is set to four then all the measured weather elements of the previous four days are considered in the estimation of the future temperature. We start by first preparing and casting the data into a tensorial format as shown in Figure 2. The tensorial representation of each sample is then fed into the proposed models. The flattened tensorial data is fed as input to the NARX model. As for the LSTM network the sequence length and the number of hidden units in the LSTM cell are set to two weeks of measurements and 200 respectively.

In the context of temperature prediction, MAE metric can be interpreted as the average difference between predictions and real values in terms of Celsius degree. The obtained MAEs over 1-10 days ahead predictions with lag parameters=4, 6, 8 and 10 of the proposed models as well as the NARX and LSTM models for Amsterdam and Brussels stations are tabulated in Table 3 and 4 respectively. The training and test computational time required by the studied models with lag parameter=10 are also depicted in Figure 8. The NARX model and the proposed 2d-CNN based model has the least training time compared to the other studied models.

From Table 3 and 4, one can observe that the 2d-CNN based model introduced in section 3.3 on average consistently outperforms the other models for almost all the considered lag parameters and across the two weather stations. In particular, the minimum MAE obtained by the studied models across different lag parameters are shown in Figure 9(a) and (b) for Brussels and Amsterdam respectively. Figure 9, illustrates the best em-

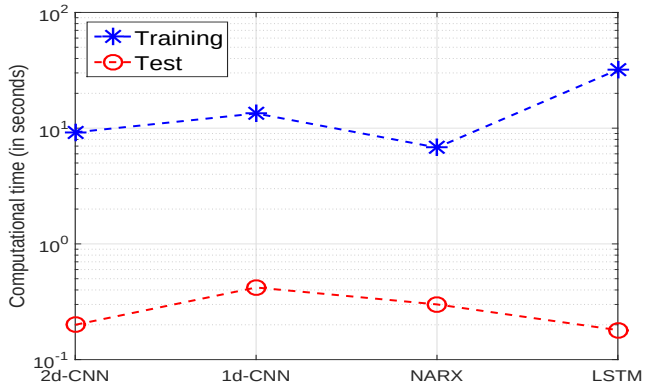

Figure 8: Training and test computational time of the proposed models as well as the NARX and LSTM models.

pirical result that one can achieve form each of the discussed models. In addition, from this figure, one can infer that the proposed 2d-CNN has the least MAEs almost for all the days ahead predictions for the two stations. Moreover, one can also observe that as the number of days ahead increases, the MAE also increases and this is common for all the models and across different lag parameters. In addition, it is hard to infer that a particular lag parameter is best for all the days ahead prediction, i.e. the optimal lag parameter could potentially varies over days ahead prediction.

The averaged MAEs over 1-10 days ahead temperature predictions obtained by the proposed 1d-CNN and 2d-CNN based models in section 3.2 and 3.3 as well as the NARX and LSTM models with lag=4, 6, 8 and 10 for Brussels and Amsterdam are depicted in Figure 10(a) and (c) respectively. The quantitative averaged MAEs over 1-10 days ahead of the four studied models are also shown in Figure 10(b) and (d). This figure indicates that the proposed 2d-CNN based model has the lowest averaged MAE compared to the other models over different lag parameters for both studied weather stations. In addition, Figure 10 shows that the proposed 1d-CNN and 2d-CNN based models have achieved their lowest averaged MAE with lag=10 for Brussels.

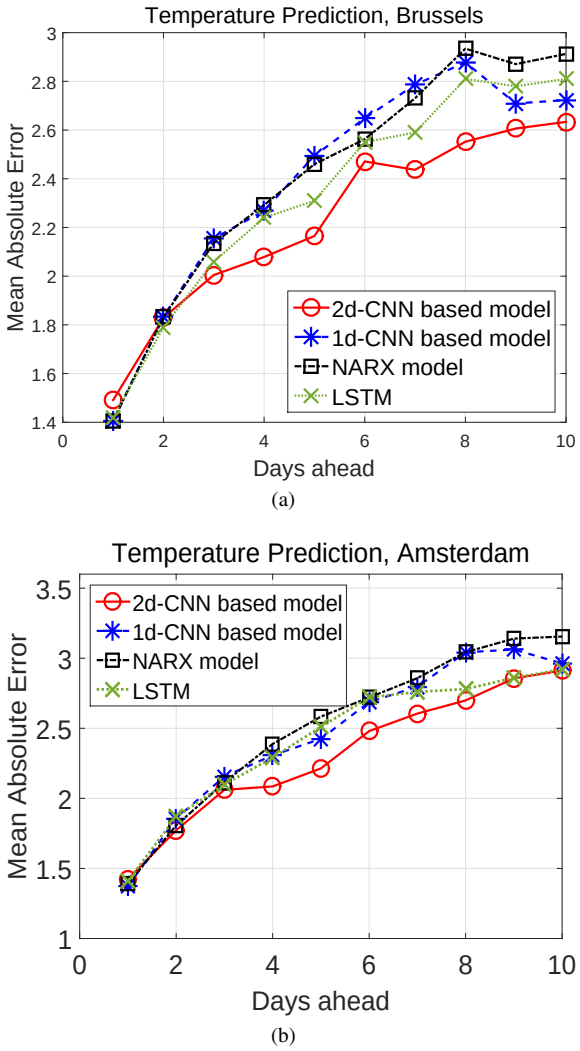

Figure 9: (a) Brussels: the minimum MAEs of discussed methods over four different lag parameters= 4,6,8 and 10. Amsterdam: the minimum MAEs of discussed methods over four different lag parameters= 4,6,8 and 10.

4.2. Wind Speed Prediction

Our second experiment concerns 6 and 12 hours ahead wind speed prediction for three weather stations: Esbjerg, Odense, Roskilde located in Denmark. Wind speed is often considered as one of the most difficult parameters to forecast because its underlying dynamics operates in an intermittent fashion therefore modeling its fluctuation is challenging. Here the hourly historical data which include four weather elements including temperature, pressure, wind speed and wind direction from 2000-2010 are used. The performance of the proposed 1d-, 2d- and 3d-convolutional neural networks models for wind speed prediction is compared with those of NARX and LSTM networks. For the wind speed prediction task, the test set consists of the last 10% of the data. while the remaining 90% percent of the data is used for training the models. In this case the test set includes the entire data of the last year, i.e. 2010, therefore covering different seasons. For this dataset, the sequence length and the number of hidden units in the LSTM cell are set to four days (96 hours) of measurements and 200 respectively.

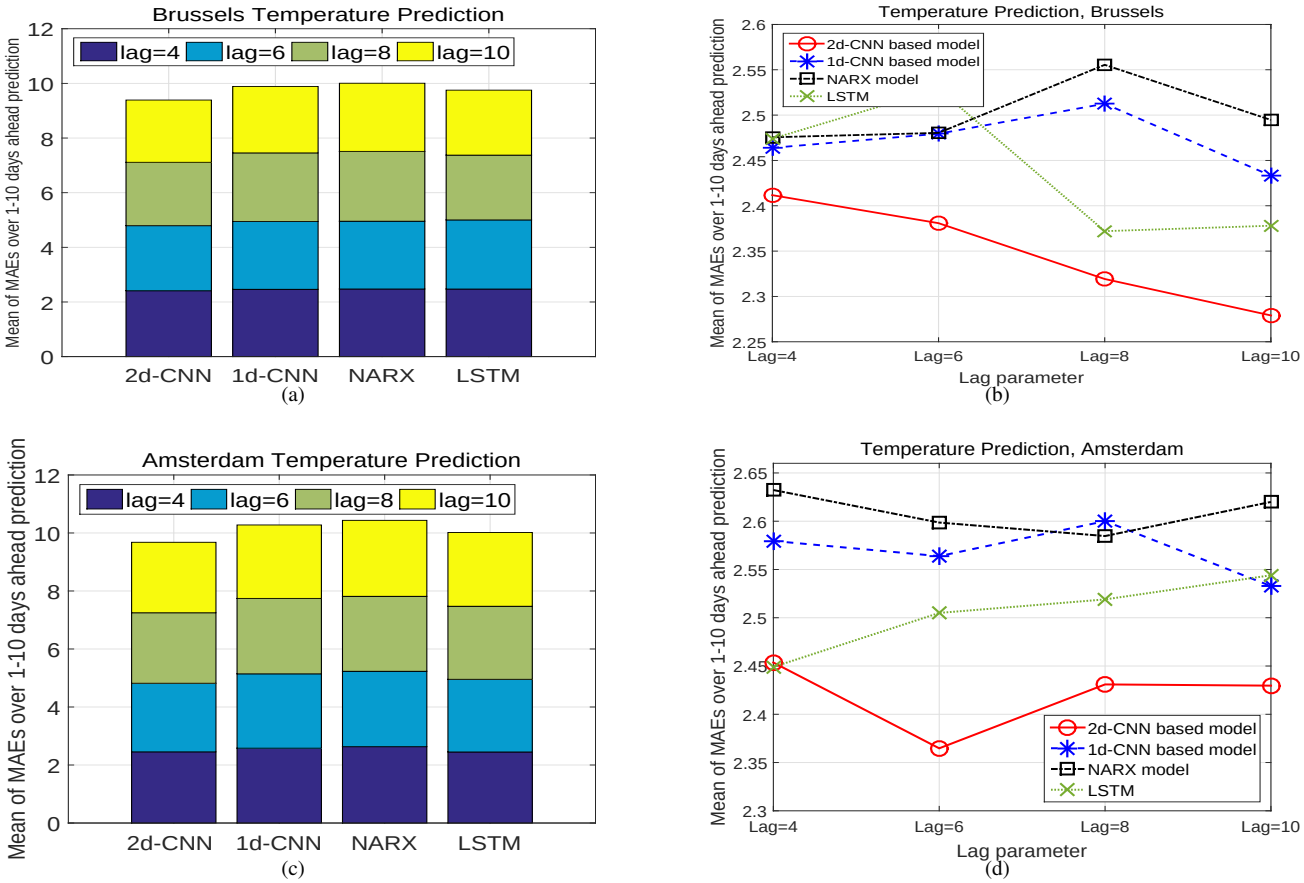

Figure 10: (a,b) Brussels: The averaged MAEs over 1-10 days ahead temperature predictions obtained by the proposed architectures in section 3.2 and 3.3 as well as the NARX and LSTM models when lag=4, 6, 8 and 10. (c,d) Amsterdam: The averaged MAEs over 1-10 days ahead temperature predictions obtained by the proposed architectures in section 3.2 and 3.3 as well as the NARX and LSTM models when lag=4, 6, 8 and 10.

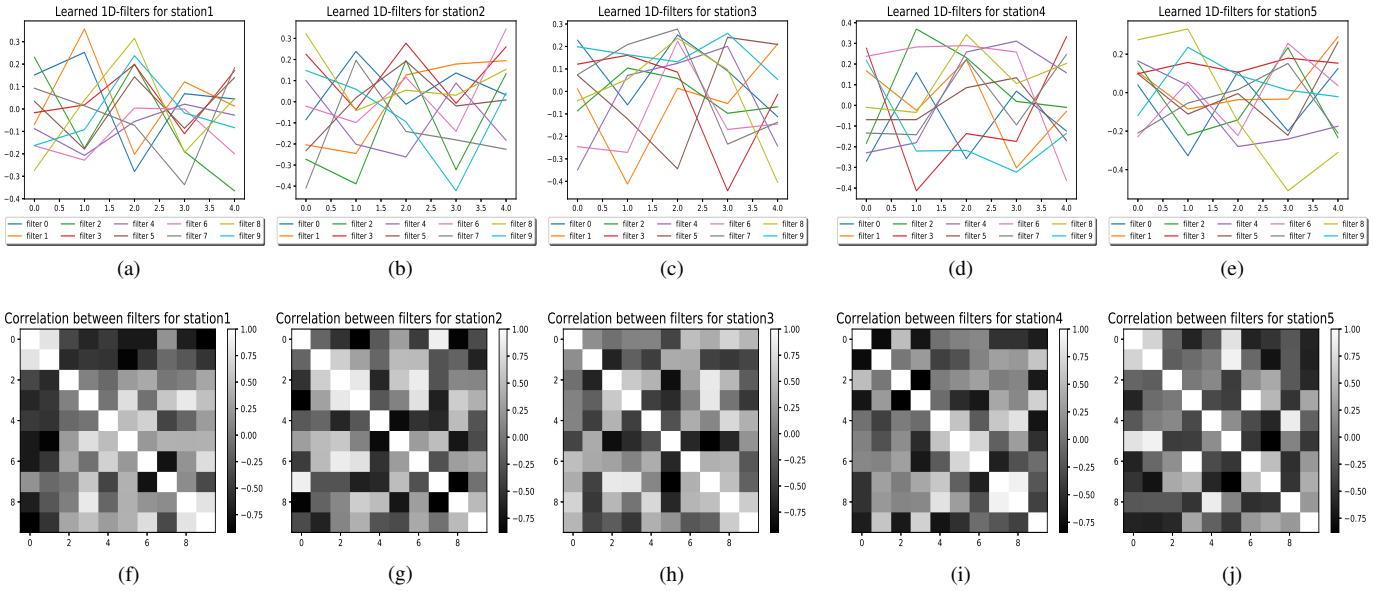

Figure 11: The ten learned filters (with length 5) of the 1d-CNN based model introduced in section 3.2 corresponding to 6 hours ahead wind speed prediction.

Table 3: The MAEs (mean absolute errors) of the proposed models as well the NARX and LSTM models for 1-10 days ahead temperature prediction of **Amsterdam**.

		Days ahead									
Lag	**Method**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
4	NARX	1.40	1.86	2.32	2.42	2.58	2.71	2.85	3.35	3.42	3.36
	LSTM	1.50	1.87	2.10	2.35	2.60	2.72	2.76	2.78	2.88	2.93
	1d-CNN	1.46	1.94	2.19	2.40	2.42	2.77	3.03	3.04	3.30	3.21
	2d-CNN	1.49	1.85	2.19	2.38	2.45	2.64	2.66	2.83	3.10	**2.91**
6	NARX	1.46	1.91	2.17	2.38	2.60	2.73	2.96	3.07	3.22	3.44
	LSTM	1.41	2.07	2.18	2.29	2.51	2.78	2.91	2.94	2.97	2.99
	1d-CNN	1.37	2.32	2.16	2.30	2.44	2.68	2.91	3.11	3.33	2.96
	2d-CNN	1.42	1.91	2.19	**2.08**	**2.21**	**2.48**	2.72	**2.69**	2.92	2.97
8	NARX	**1.39**	1.84	2.11	2.48	2.69	2.79	3.05	3.04	3.26	3.15
	LSTM	1.53	1.87	2.26	2.38	2.69	2.81	2.97	2.90	2.86	2.92
	1d-CNN	1.42	1.85	2.15	2.55	2.59	2.72	3.10	3.15	3.30	3.11
	2d-CNN	1.48	1.89	**2.06**	2.16	2.28	2.87	2.62	2.88	3.06	2.98
10	NARX	1.47	1.80	2.19	2.61	2.85	2.74	3.03	3.13	3.14	3.20
	LSTM	1.50	1.95	2.17	2.36	2.70	2.89	2.81	2.94	3.05	3.07
	1d-CNN	1.44	1.96	2.21	2.36	2.60	2.75	2.79	3.13	3.06	2.98
	2d-CNN	1.43	**1.77**	2.10	2.32	2.63	2.69	**2.60**	2.86	**2.85**	3.01
	The least MAE	1.39	1.77	2.06	2.08	2.21	2.48	2.60	2.69	2.85	2.91

The obtained MAEs of the discussed models with lag=4 are tabulated in Table 5. One can observe that in most of the cases the introduced 3d-CNN base model outperforms the other models. We have performed a paired t-tests [8] for comparison of the obtained mean absolute error of the 3d-CNN model with those of 1d-CNN, 2d-CNN, NARX and LSTM models. The obtained p-values of a pairwise t-test are tabulated in Table 6 which gives a clear evidence that the MAEs of the 3d-CNN based model is comparable to those obtained by other models and for some of the weather stations there is a statistically significant difference in the mean absolute errors.

In order to gain more insight on the nature of the learned filters we can visualize them. The learned filters of the 1d-CNN based model introduced in section 3.2 for each weather station and corresponding to 6 hours ahead prediction are depicted in Figure 11. As stated previously (see Table 1), in our analysis we used 10 filters with length 5. Interestingly, it can be seen from Figure 11 that some of the filters within a particular station follow the same pattern. Figure 11(f,g,h,i,j) show the correlations between learned filters of each stations. This analysis might potentially be useful for pruning the filters that are highly correlated in order to save time on the test set. The obtained 3d-CNN forecasts on a subset of the test wind speed dataset corresponding to 6 and 12 hours ahead at three stations are depicted in Figure 12. Note that the reported MAEs in Figure 12 correspond to the entire test dataset.

5. Conclusions

In this paper new models based on 1d-, 2d- and 3d-convolutional neural network architecture are introduced for multi-step ahead temperature and wind speed prediction using historical weather data. We showed how combining different types of convolutional layers in the network configuration can boost the prediction performance thanks to the representation learning as well as dimensionality reduction capabilities of the proposed networks. The designed networks leverage the shared representation learning layer to improve forecasting in multiple weather stations simultaneously. In particular 2d- and 3d-convolutional operations have shown promising results on the studied tasks.

Acknowledgments

This work was partially supported by the Postdoctoral Fellowship of the Research Foundation-Flanders (FWO: 12Z1318N). Siamak Mehrkanoon is an assistant professor in the Department of Data Science and Knowledge Engineering, Maastricht University, the Netherlands.

References

[1] Abramson, B., Brown, J., Edwards, W., Murphy, A., Winkler, R. L., 1996. Hailfinder: A bayesian system for forecasting severe weather. International Journal of Forecasting 12 (1), 57–71.

[2] Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence 35 (8), 1798–1828.

[3] Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. Journal of Machine Learning Research 13 (Feb), 281–305.

[4] Chen, L., Lai, X., 2011. Comparison between arima and ann models used in short-term wind speed forecasting. In: Power and Energy Engineering Conference (APPEEC), 2011 Asia-Pacific. IEEE, pp. 1–4.

[5] Clevert, D.-A., Unterthiner, T., Hochreiter, S., 2015. Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289.

Table 4: The MAEs (mean absolute errors) of the proposed models as well the NARX and LSTM model for 1-10 days ahead temperature prediction of **Brussels**.

		Days ahead									
Lag	**Method**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
4	NARX	1.41	1.84	2.13	2.29	2.45	2.56	2.73	3.20	3.09	3.01
	LSTM	1.65	1.82	2.23	2.32	2.48	2.70	2.71	2.92	2.93	2.98
	1d-CNN	1.42	1.88	2.17	2.39	2.52	2.73	2.80	2.87	2.93	2.87
	2d-CNN	1.67	1.83	2.19	2.29	2.46	2.77	2.58	2.75	2.84	2.69
6	NARX	1.44	1.83	2.15	2.30	2.50	2.60	2.80	2.93	3.06	3.14
	LSTM	1.51	1.92	2.24	2.29	2.45	2.56	3.64	2.81	2.91	2.96
	1d-CNN	1.45	1.98	2.19	2.32	2.49	2.64	2.93	2.94	3.06	2.74
	2d-CNN	1.49	1.90	2.31	2.28	2.36	**2.47**	2.70	2.74	2.74	2.77
8	NARX	1.45	1.88	2.16	2.49	2.66	2.94	3.06	2.96	2.99	2.92
	LSTM	1.50	1.83	2.06	2.24	2.31	2.69	2.59	2.81	2.86	2.83
	1d-CNN	1.48	1.83	2.15	2.50	2.55	2.74	2.97	2.99	3.01	2.87
	2d-CNN	1.52	**1.82**	**2.00**	2.19	**2.16**	2.78	**2.43**	2.71	2.78	2.75
10	NARX	**1.40**	1.83	2.27	2.44	2.64	2.69	2.90	2.95	2.87	2.91
	LSTM	1.42	**1.79**	2.21	2.29	2.33	2.55	2.71	2.89	2.78	2.81
	1d-CNN	**1.40**	1.83	2.23	2.26	2.58	2.84	2.78	2.93	2.70	2.72
	2d-CNN	1.49	1.83	2.13	**2.07**	2.28	2.56	2.60	**2.55**	**2.60**	**2.63**
	The least MAE	1.40	1.79	2.00	2.07	2.16	2.47	2.43	2.55	2.60	2.63

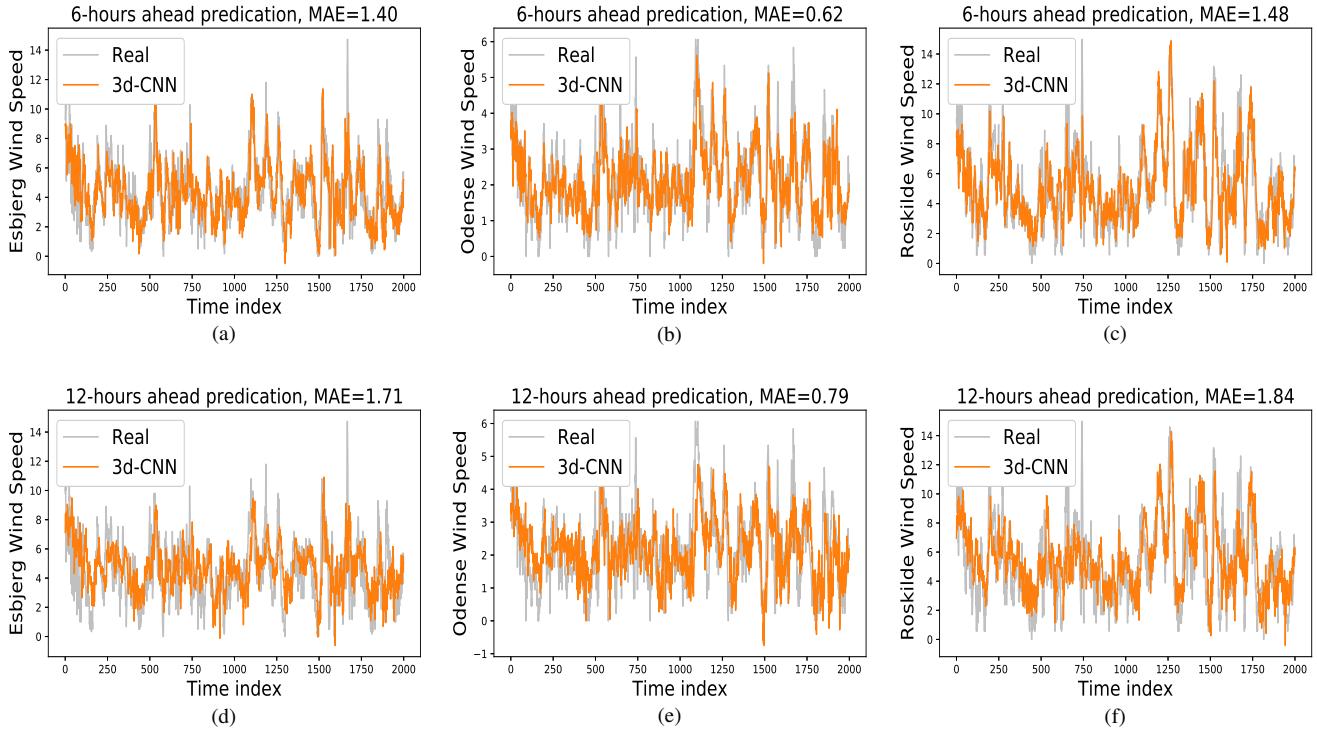

Figure 12: Illustrations of the obtained forecasts for a subset of test dataset. (a,b,c) The Obtained 6-hours ahead wind speed forecasts using the proposed 3d-CNN model for three stations. (e,f,g) The Obtained 12-hours ahead wind speed forecasts using the proposed 3d-CNN model for three stations. The reported MAEs correspond to the entire test dataset.

Table 5: The MAEs (mean absolute errors) of the proposed models, the NARX and LSTM models for (6 and 12)-hours ahead wind speed prediction of three stations located in Denmark.

		Method				
Hours ahead	**Station**	**3d-CNN**	**2d-CNN**	**1d-CNN**	**NARX**	**LSTM**
6	Esbjerg	1.40	1.42	1.44	1.59	1.54
	Odense	0.62	0.63	0.63	0.68	0.86
	Roskilde	1.48	1.50	1.52	1.56	1.49
12	Esbjerg	1.71	1.75	1.75	1.81	1.77
	Odense	0.79	0.80	0.82	0.86	1.05
	Roskilde	1.84	1.90	1.92	1.96	1.79

Table 6: P-values of a pairwise T-test on mean absolute error between 3d-CNN model and other models.

Hours ahead	**Station**	**2d-CNN**	**1d-CNN**	**NARX**	**LSTM**
6	Esbjerg	0.01	0.007	0.09	0.002
	Odense	0.09	0.08	0.09	0.001
	Roskilde	0.03	0.04	0.001	0.22
12	Esbjerg	0.01	0.02	0.01	0.001
	Odense	0.13	0.12	0.01	0.002
	Roskilde	0.02	0.03	0.008	0.27

[6] Cofino, A. S., Cano, R., Sordo, C., Gutierrez, J. M., 2002. Bayesian networks for probabilistic weather prediction. In: Proceedings of the 15th European conference on Artificial Intelligence. IOS Press, pp. 695–699.

[7] Dalto, M., Matuško, J., Vašak, M., 2015. Deep neural networks for ultra-short-term wind forecasting. In: Industrial Technology (ICIT), 2015 IEEE International Conference on. IEEE, pp. 1657–1663.

[8] David, H. A., Gunnink, J. L., 1997. The paired t test under artificial pairing. The American Statistician 51 (1), 9–12.

[9] Grover, A., Kapoor, A., Horvitz, E., 2015. A deep hybrid model for weather forecasting. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 379–386.

[10] He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034.

[11] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9 (8), 1735–1780.

[12] Klein, B., Wolf, L., Afek, Y., 2015. A dynamic convolutional layer for short range weather prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4840–4848.

[13] Krasnopolsky, V. M., Fox-Rabinovitz, M. S., 2006. Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. Neural Networks 19 (2), 122–134.

[14] Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105.

[15] Kuligowski, R. J., Barros, A. P., 1998. Localized precipitation forecasts from a numerical weather prediction model using artificial neural networks. Weather and forecasting 13 (4), 1194–1204.

[16] Kusiak, A., Zheng, H., Song, Z., 2009. Short-term prediction of wind farm power: a data mining approach. IEEE Transactions on energy conversion 24 (1), 125–136.

[17] Lai, L. L., Braun, H., Zhang, Q., Wu, Q., Ma, Y., Sun, W., Yang, L., 2004. Intelligent weather forecast. In: Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on. Vol. 7. IEEE, pp. 4216–4221.

[18] LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

[19] LeCun, Y., Jackel, L., Bottou, L., Cortes, C., Denker, J. S., Drucker, H., Guyon, I., Muller, U., Sackinger, E., Simard, P., et al., 1995. Learning algorithms for classification: A comparison on handwritten digit recognition. Neural networks: the statistical mechanics perspective 261, 276.

[20] Maas, A. L., Hannun, A. Y., Ng, A. Y., 2013. Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml. Vol. 30. p. 3.

[21] Marchuk, G., 2012. Numerical methods in weather prediction. Elsevier.

[22] Mcgovern, A., Supinie, T., Gagne, I., Collier, M., Brown, R., Basara, J., Williams, J., 2010. Understanding severe weather processes through spatiotemporal relational random forests. In: 2010 NASA conference on intelligent data understanding.

[23] Mehrkanoon, S., Suykens, J. A., 2018. Deep hybrid neural-kernel networks using random fourier features. Neurocomputing 298, 46–54.

[24] Nair, V., Hinton, G. E., 2010. Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 807–814.

[25] Nurmi, P., Perrels, A., Nurmi, V., 2013. Expected impacts and value of improvements in weather forecasting on the road transport sector. Meteorological Applications 20 (2), 217–223.

[26] Radhika, Y., Shashi, M., 2009. Atmospheric temperature prediction using support vector machines. International journal of computer theory and engineering 1 (1), 55.

[27] Richardson, L. F., 2007. Weather prediction by numerical process. Cambridge University Press.

[28] Salman, A. G., Kanigoro, B., Heryadi, Y., 2015. Weather forecasting using deep learning techniques. In: Advanced Computer Science and Information Systems (ICACSIS), 2015 International Conference on. IEEE, pp. 281–285.

[29] Sapankevych, N. I., Sankar, R., 2009. Time series prediction using support vector machines: a survey. IEEE Computational Intelligence Magazine 4 (2).

[30] Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural networks 61, 85–117.

[31] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[32] Steinhaeuser, K., Chawla, N. V., Ganguly, A. R., 2011. Complex networks as a unified framework for descriptive analysis and predictive modeling in climate science. Statistical Analysis and Data Mining: The ASA Data Science Journal 4 (5), 497–511.

[33] Tasnim, S., Rahman, A., Oo, A. M. T., Haque, M. E., 2017. Autoencoder for wind power prediction. Renewables: Wind, Water, and Solar 4 (1), 6.

[34] Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., Woo, W.-c., 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems. pp. 802–810.

[35] Xu, B., Wang, N., Chen, T., Li, M., 2015. Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853.