2017 IEEE International Conference on Data Mining Workshops

A machine learning approach to non-uniform spatial downscaling of climate variables

Soukayna Mouatadid
Department of Computer Science
University of Toronto
Toronto, Canada
soukayna@cs.toronto.edu

Steve Easterbrook
Department of Computer Science
University of Toronto
Toronto, Canada
sme@cs.toronto.edu

Andre R. Erler
Aquanty Inc. / Department of Physics
University of Toronto
Toronto, Canada
aerler@atmosp.physics.utoronto.ca

***Abstract*—This study presents a scalable and robust approach to spatial downscaling in the context of climate downscaling. We explore the ability of four techniques to downscale a climate variable to a given location of interest. As an example, we focus on downscaling daily mean air temperature at twelve stations located across the topographically complex province of British Columbia, Canada. The techniques include multi-linear regression (MLR), artificial neural networks (ANN), extreme learning machines (ELM) and long-short term memory networks (LSTM). Our method based on LSTM generalizes well to different locations and leads to higher downscaling accuracy compared to MLR and ELM. The performance of the models is measured based on statistical metrics, including the coefficient of determination, and the root mean square error.**

***Index Terms*—Climate statistical downscaling, Reanalysis data, Artificial neural networks, Extreme learning machines, Long-short term memory networks**

I. Introduction

Complete and accurate climate datasets are not readily available in many regions around the world. They are especially lacking in the areas which are most sensitive to climate change [1]. One of the reasons for this lack of data is the complex topography of such regions, where it is difficult to install and maintain weather stations. As a result, some of the regions most affected by climate change are unable to obtain detailed climate data needed to understand impacts and develop adaptation plans for future anthropogenic climate change [2]. For regional climate studies, the ideal case would be a representative and homogeneous distribution of weather observing stations across the region under investigation. But often, the closest station to the region of interest is tens or hundreds of kilometers away, and the available data may not represent the climate of locations of interest [3]. For example, in Canada, regions in the central and northern parts of the country, as well as remote mountain ranges in northern BC, maintaining climate stations is challenging due to the lack of accessibility and harshness of the terrain and climate. These conditions have resulted in significant gaps in spatial coverage and continuity of records [4].

To address this problem, scientists often rely on *gridded reanalysis products* as a replacement for pure observational data [4]. These datasets are produced by constraining a physics-based simulation with available station observations, so as to generate a gridded data product, where missing data points are filled in by the simulation in a physically consistent manner [5]. However, gridded products for remote areas are typically coarse resolution, and do not capture small-scale climatic characteristics associated with regional topographic features, such as mountain ranges or lakes. For this reason, it is usually necessary to re-process these data sets to a finer scale, in a way that accounts for such features, but does not introduce additional errors and biases. This process is referred to as *downscaling*.

In this work, we present a novel downscaling method that learns from gridded reanalysis data and local station data, and outperforms traditional Multiple Linear Regression (MLR) techniques, in terms of downscaling accuracy. Our method can be used for locations with available historical observational data as well as new locations where no historical observational data is available, a case where traditional downscaling methods perform poorly. Our method offers four key advantages:

- Our method learns non-linear relationships and seasonal dependencies on given grid nodes across time series. This minimizes the manual feature engineering required to capture complex, location-dependent behavior.
- Our method learns from non-uniform data fields as it uses both low-resolution gridded data and scattered weather stations observations; a major improvement over downscaling methods in the literature that learn solely from gridded data.
- Our method can be applied to any region, as it doesn't need to be adapted for different topographies. We have tested it for locations across the province of British Columbia, which is characterized by a highly complex topography (i.e., mountains, proximity to large bodies of water, etc), for which traditional downscaling methods fail. The method can be applied without modification to any region in the world, no matter how complex the topography.
- Our method is flexible as it can be used with different types of neural networks. In this study, we compare the use of ANN, ELM and LSTM to traditional MLR.

The paper is organized as follows. In Section II, we introduce the downscaling task in more detail, as well as previous

2375-9259/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDMW.2017.49

work in the literature. In Section III, we present our novel downscaling method based on non-uniform data use along with the theoretical background behind the models. In Section IV, we evaluate our method on a real-word dataset of daily mean air temperatures in British Columbia, Canada. Section V covers a discussion of our results.

II. Background

Downscaling has received significant attention in the climate science community and has a wide range of applications [6]. Downscaling provides detailed climate and weather data at specific locations—a prerequisite for modeling environmental processes at the regional and local scale [7] and the sustainable management of natural resources [8]. Downscaling is also used on projections of future climate change from Global Climate Models (GCMs), as these models use relatively coarse grids, and localized climate change projections are necessary for detailed impact studies.

Unlike data-driven models in machine learning, climate models are physics-based simulations, based on scientific first principles from several fields such as meteorology, oceanography, and geophysics [9]. These GCMs simulate the dynamics of the atmosphere including the effects of cloud formation, rainfall, wind, ocean currents, radiative heat transfer through the atmosphere, etc. [10]. Climate scientists and meteorologists use GCMs for a variety of tasks, including creating gridded reanalysis products from observational data of the recent past, and for projecting future climate change under various scenarios for likely human greenhouse gas emissions.

Typically, the output from a GCM is a time series (with a timestep of a few hours), for each climate variable, at each point on a global three dimensional grid, for several decades or centuries of simulation. These models are computationally demanding, requiring high performance computing platforms, and generate very large data volumes (a single model run produces many terabytes). For feasibility, global models compute key climate variables on relatively coarse grids, typically 100km-250km to a side).

Downscaling is used to interpolate the output of a global climate model to a finer grid with resolutions of the order of a few kilometers, while accounting for local topography and climate characteristics. There are two major forms of downscaling [11]:

- Dynamical downscaling involves the use of a high-resolution limited-area model in order to generate simulations at the desired resolution over the area of interest. While physically consistent, this approach is computationally very expensive, and can introduce large biases. Furthermore, the limited-area models typically still do not reach the desired resolution, so that another downscaling step employing statistical methods is still necessary [12].
- Statistical downscaling is relatively inexpensive (computationally), and inherently includes bias-correction. It relies on statistical or empirical relationships between the large-scale predictor fields from the model simulations and the variables of interest, at the location of interest. These relationships have to be derived from historical observational data.

Statistical downscaling is challenging where there is insufficient historical data to derive robust relationships. The coverage of datasets from satellite measurements, station observations, and reanalysis products, which incorporate both, varies widely by region [2]. Several recent review papers describe the spatial interpolation methods used for downscaling in meteorology and climatology [13, 14, 15]. These techniques include nearest neighbor methods, splines, regression, kriging and cokriging but also machine learning techniques such as ANNs and SVMs [16, 17, 18, 19, 20, 21]. More recent studies have explored the use of super resolution convolutional neural networks (SRCNN) for statistical downscaling of climate variables. One study used an SRCNN for downscaling of ocean remote sensing data and found it to outperform bicubic interpolation [22], while another study applied a stacked SRCNN to a precipitation downscaling task and showed that it outperformed bias correction spatial disagregation methods as well as traditional SVM and ANN models [2]. These SRCNN-based methods require a complete gridded dataset to train and evaluate the downscaling model. However, for cases where complete climate data fields are not available, our proposed approach can operate on sparse and irregular point data. We refer the reader to [23] for a review of single-image super-resolution techniques.

For air temperature records, several studies have focused on the interpolation of minimum and maximum temperatures [24, 25, 26, 14, 27]. All these studies, except for [14] and [27] were for regions characterized by a relatively uniform station density. On the other hand, none of these studies have explored the use of LSTMs, which are designed to handle sequence dependency. To the best of our knowledge, no other study applied ELMs nor LSTMs to the downscaling of climate variables in regions with complex topography.

Our method learns a mapping between a low-resolution reanalysis dataset and the climate at specific locations, using ANN, ELM and LSTM models. The method has two variants: (a) mapping to a location for which there is a past observational record (i.e. a station location) and (b) mapping to a location without a past observational record, in which case other stations in the region are used for training data. The method implicitly incorporates local climate characteristics and topographic features, as they are captured in the observational data.

III. Model development

A. Theoretical background

In this section, we develop a downscaling method for climate data, combining two sources of inputs: observational station records and reanalysis gridded datasets. We begin with a single neural network model, for example an LSTM (Figure 1). LSTMs are part of a family of neural networks called recurrent neural networks (RNNs). As opposed to ordinary ANNs, RNNs allow forward and backward connections between neurons, which makes them well fit for processing sequential data. Consequently, LSTMs, which use purpose-built memory cells

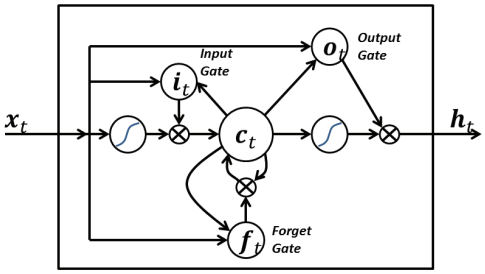

Fig. 1: An LSTM block with input, forget, and output gates, which control respectively, the extent to which a new value flows into memory, remains in memory and is used to compute the output activation of the block. [30].

to store information, have a better ability to find and exploit long range dependencies in the data. For a more detailed and formal coverage of the topic, the reader is directed to [28]. We start by defining the LSTM update equations in the commonly-used version from [29]:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \odot c_{t-1} + b_i) \quad (1)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \odot c_{t-1} + b_f) \quad (2)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \odot c_t + b_o) \quad (4)$$
$$h_t = o_t \odot tanh(c_t) \quad (5)$$

where $\odot$ denotes the Hadamard product and σ denotes the logistic sigmoid function. The gating functions are represented by i_t, f_t, o_t which denote the input, forget and output gates at time t respectively. c_t, x_t and h_t denote the cell activation vector, the input feature vector and the hidden output vector respectively.

The weight parameters are W_{xi}, W_{xf}, W_{xc}, W_{xo}, W_{hi}, W_{hf}, W_{hc} and W_{ho}. These weight parameters connect the different inputs and gates with the memory cells and outputs. The bias terms are represented by b_i , b_f , b_c and b_o. Optional connection weights W_{ci}, W_{cf} and W_{co} further influence the gates' operation.

In addition to LSTM, we also investigate the use of MLR, ANN and ELM models. The theoretical background for MLR and ANN is provided in [31] and [32] respectively. For ELM, the model is based on a single layer feed-forward neural network architecture (SLFN), with a single nonlinear hidden layer. This architecture is different from the traditional feedforward back propagation (FFBP) ANN, as it starts by fixing the hidden layer weights and biases, drawn from a continuous probability distribution function, and then provides a closed-form least squares solution to the output weights achieved through a Moore-Penrose inverse function (rather than using the iterative solution used in FFBP-ANNs) [33]. ELM has proved to be efficient in online and real-time time-series prediction, with faster learning and running speed than traditional neural networks [34, 35, 36]. In our method, an ELM model was used to train the (x_i, y_i) data pairs. For a set of N training samples, the SLFN with L hidden neurons is expressed as [33]:

$$f_L(x) = \sum_{i=1}^{i=L} h_i(x).\beta_i = h(x)\beta \quad (6)$$

where $h_i(x)$ is the i_{th} hidden neuron, $h(x)$ is the hidden neuron outputs representing the randomized hidden features of the predictor x_i, and β is the output weight matrix. The output functions of the hidden neurons $h_i(x)$ can be represented as $h_i(x) = G(a_i, b_i, x)$ with $a_i \in R^d$ and $b_i \in R$, and $G(a_i, b_i, x)$ is defined using the hidden neuron parameters (a, b). We compared five commonly used activation functions: (1) the tangent Sigmoid $G(a, b, x) = \frac{2}{1+e^{-2(-ax+b)}} - 1$, (2) the logarithmic sigmoid $G(a, b, x) = \frac{2}{1+e^{(-ax+b)}}$, (3) the hard limit $G(a, x, b) = 1$ if $ax + b > 0$ or 0 otherwise, (4) the triangular basis $G(a, x, b) = 1 - |ax + b|$ if $1 \leq ax + b \leq 1$ or 0 otherwise, and (5) the radial basis $G(a, b, c) = e^{[-(-ax+b)^2]}$. The next step is to minimize the approximation error by solving for the weights connecting the hidden and the output layer (β) using least square fitting: $\min_{\beta \in R^{L*m}} \|H\beta - T^2\|$ where $\|\cdot\|$ is the Frobenius norm and H is the randomized hidden layer output matrix in the form:

$$H = \begin{bmatrix} g(x_1) \\ \vdots \\ g(x_N) \end{bmatrix} = \begin{bmatrix} g_1(a_1x_1 + b_1) & \cdots & g_L(a_Lx_1 + b_L) \\ \vdots & \cdots & \vdots \\ g_1(a_Nx_N + b_1) & \cdots & g_L(a_Lx_N + b_L) \end{bmatrix} \quad (7)$$

and the target matrix T is expressed as:

$$T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix} = \begin{bmatrix} t_{11} & \cdots & t_{1m} \\ \vdots & \cdots & \vdots \\ t_{N1} & \cdots & T_{nM} \end{bmatrix} \quad (8)$$

An optimal solution is obtained by solving a system of linear equation $\beta^* = H^+T$ where H^+ is the Moore-Penrose generalized inverse function $(+)$. The optimal solution is then plugged in Equation (6) to predict the downscaled climate variable.

B. Features

The predictand of the models is the expected value of a given climate variable at a specific location and time. For predictors, many explanatory variables could be included in the analysis. Traditional downscaling techniques draw on data from digital elevation models (DEMs), orographic and oceanographic effects, the orographic slope, aspect ratio curvature, radiation balance or the proximity to lakes of different sizes and depths [37]. However, our study explores the minimal amount of information needed to accurately downscale a climate variable. For the first task in this study, the models' predictors consisted of the reanalysis values at the 16 gridded nodes surrounding the location of interest. For the second task, the predictors included the location's latitude, longitude and elevation, in addition to

the reanalysis values. The coordinates were included to enable a choice mechanism, so that the models can downscale to any specific location (latitude, longitude, elevation) within the bounds of the inputs domain. The inclusion of elevation is advantageous, as it directly correlates with major gradients in climate on a small scale; this is also true for latitude on a larger scale. However, the inclusion of lateral the coordinates (longitude and latitude) is also necessary to capture horizontal gradients, which are often captured in the gridded reanalysis data, but do not necessarily follow lines of elevation or latitude. For example, in British Columbia a strong gradient from the South-west to the North-east exists in both, temperature and precipitation, due to proximity to the coast and rain-shadowing by mountain ranges.

It should be noted here, that as a preliminary step, this approach was applied to the raw observational data, without the inclusion of the reanalysis data as predictors. However, the models (not reported here due to space constraints) performed poorly under this scenario. Hence, the inclusion of the reanalysis data as part of the models'inputs.

C. Task 1: Downscaling to locations with observational record

For the first task, we downscale gridded reanalysis data to a location for which past observations are available. In this scenario, the historical values recorded at the station were used as the predictand, and the reanalysis data at 16 grid points around the station were used as model predictors, selected such that the location of the station of interest is at the center grid cell of a 4 x 4 sub-grid or square. We refer to these 16 grid points as the station's neighborhood. The studies in [38, 5] showed that the sixteen grid points around a station of interest all supply relevant information to the model.

D. Task 2: Downscaling to locations without observational record

For the second task, the goal was to explore how a gridded dataset can be downscaled to locations where no past observational record is available. The methodology used here is similar to [38], where the focus was on predicting solar energy over a spatial grid by developing a support vector machine model for each individual cell of a gridded dataset.

We develop a model for a location of interest, using the information available from that location's neighborhood. Again, we use the square formed by the nine grid cells (i.e., 16 grid points) around the location of interest as the location's neighborhood. As there is no data for the location of interest, we use other stations within the given neighborhood. For the training set, the input variables are the reanalysis values from the sixteen grid points surrounding these stations along with each stations' coordinates (i.e., latitude, longitude and elevation), and the output variables are the observations recorded at the stations that fall within the neighborhood. In case the station of interest had no neighboring station within the 4×4 sub-grid , the sub-grid was incremented by one cell, moving from using 16 grid points (i.e., 4×4) to using 25 grid points (i.e., 5×5), until at least one neighboring station was included.

To test the method, we select one station as the location of interest, and exclude its data from the training set. The output variable in our tests corresponds to the observational data recorded at this location, and the input variables are the reanalysis values from the sixteen grid nodes around said location, and the location coordinates. During the training phase, the model has not been fed any value related to the location of interest, and during the testing phase, the model's only input is the information from the reanalysis dataset, and the location's coordinates.

Figure 2 illustrates the construction of a test set for a model used to downscale to a location of interest (s1) with three neighboring stations (s9, s11 and s12) used for training. The location of interest (s1) is located in the center cell of a sub-grid composed of nine grid cells. Each grid cell is defined by four grid point. Following this methodology, the models can be used to downscale to any location (any latitude, longitude, elevation), whether or not it is in the testing set. In particular, this means that it would be possible to generate two-dimensional maps of downscaled climate variables by using this method in conjunction with a digital elevation model of suitable resolution.

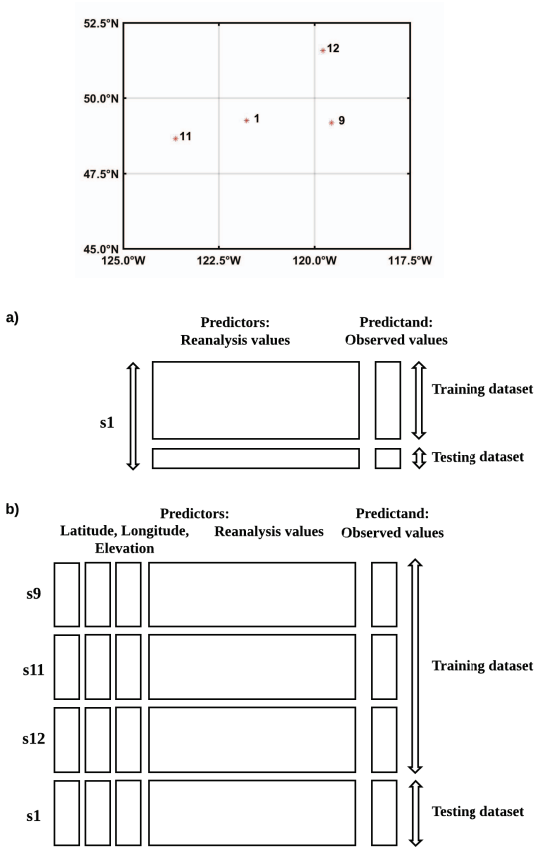

Fig. 2: Development of downscaling models' datasets for a) task 1 and b) task 2 at location s1.

To summarize, our method downscales to a given location by training a model from the gridded data and the stations within that locations' neighborhood, and we test our approach by excluding one station form the training set, to be treated as the location of interest. In a way, this methodology is similar to leave-one-out cross-validation, applied to stations instead of training examples.

IV. Application and Experiments

In this section we present experimental results when applying our method on a daily mean air temperature dataset for British Columbia. The province of British Columbia in Canada is located at high latitudes, extending across the Canadian Pacific coast, and the coastal and interior mountain ranges including the Canadian Rockies. The province lies within the bounds 49°00′N to 60°00′N and 114°04.1′W to 139°03′W, and covers an area of 944,735 km^2. The province is generally characterized by a milder climate compared to the rest of the country, but varies significantly from one region to another, due to the influence of the Pacific Ocean and the mountain range.

A. Data

The reanalysis data used as the models' predictors (inputs) are from the NCEP/NCAR (National Centers for Environmental Prediction/National Center for Atmospheric Research) reanalysis dataset, with 55 years spanning 1960–2015 for air temperature [39]. NCEP/NCAR dataset is a combination of physical process and model forecast gridded data at the 2.5°x 2.5°spatial resolution. This resolution corresponds to approximately a 278 km x 162 km spatial scale: over British Columbia, 2.5°in latitude is 278 km, and 2.5°in longitude varies from 128 km (at 45°N latitude) to 196 km (at 62.5°N latitude). Details regarding this dataset's development can be found in [39].

The station data used in our study consists of the observed values of daily mean air temperatures. These were obtained for twelve stations that are part of the Environment and Climate Change Canada network [40]. This dataset is homogenized, meaning it has been processed to identify and correct non-climatic shifts due to the relocation of the stations and changes in observing practices and automation [41].

There are currently 54 meteorological stations unevenly distributed across the province [40]. Of these stations, we chose 12 from a short list of stations initially selected for completeness of record then further refined based on topographic complexity. In the first step, we selected stations with the longest uninterrupted climate record. The length of the reanalysis dataset (55 years) determined the length of the stations that could be selected. Then, from the short-listed stations, the 12 sites used in this study were selected because they all fall in interesting topographies and are likely to pose challenges to downscaling. For instance, stations were selected if they were located close to large bodies of water: the Agassiz station (s1) located close to the Fraser river and the Fort St. James station (s5) located close to the Stuart lake. Other stations were selected due to their location in the dry climate region of the east coast (Sandspit station, s7), or the wet climate region of the west coast (Estevan Point station, s4). We excluded stations for which the observational records were obtained by joining multiple neighboring stations. Figure 3 shows the geographical location of British Columbia with the distribution of the twelve selected stations and the reanalysis grid.

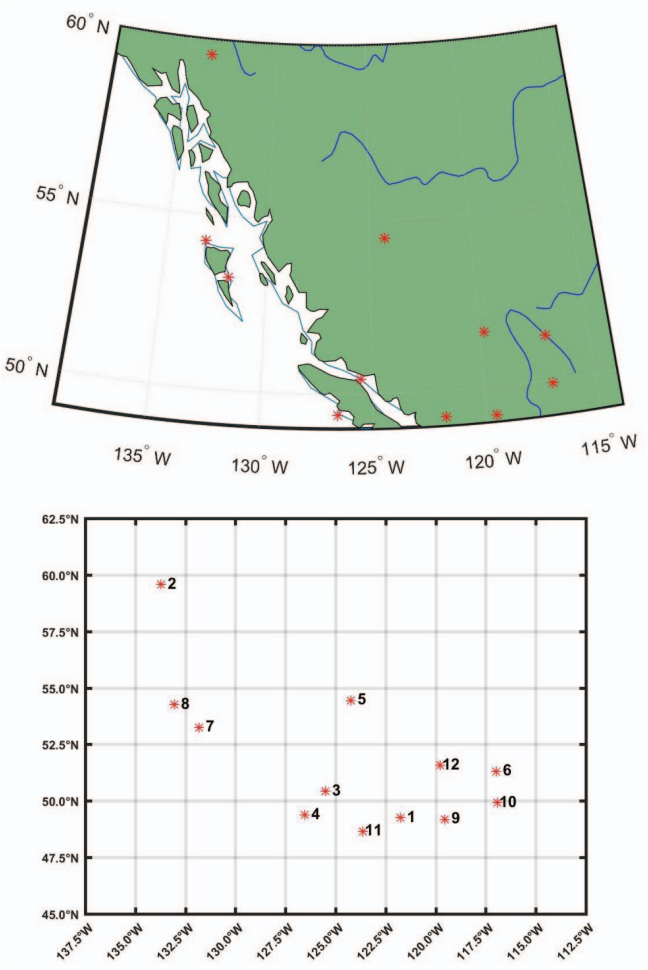

Fig. 3: A map of the study area (British Columbia, Canada) on the top, and the grid nodes used as neighbor cells of the stations under study on the bottom.

The predictand and predictor data were standardized to fall within a range of [0, 1]. By standardizing the variables and recasting them into dimensionless units, the arbitrary effect of similarity between objects is removed. The data was partitioned into a training and testing set. In all cases, 10% of the available data was used to test the models. Parameter tuning was achieved through cross-validation.

B. Model structure

All of the ANN, ELM and LSTM models were based on a three-layer architecture. The number of neurons in the first layer corresponded to the number of input variables used and was equal to 16 for the first task and 19 for the second task.

The number of neurons in the third layer was fixed to 1 and represented the climate variable to be downscaled.

When it comes to the number of hidden neurons used for the ANN models, [42] and [43] presented two different approaches to determining the optimal number of neurons in the hidden layer. In [42], the optimal number of hidden neurons was determined to be equal to log N (N being the training sample size), whereas [43] showed that number to be equal to 2n+1 (where n is the number of input neurons). Therefore, the ANN models were developed first using a minimum number of hidden neurons equal to log N (i.e., ≈ 4) and this number was incremented by 1 until it reached a maximum of 2n+1 (i.e., 39). The ANN model structure that lead to the lowest error during validation was selected as the optimal architecture. The developed ANN models were all based on the feed forward multi-layer perceptron (MLP) architecture and trained using the Levenberg-Marquardt algorithm. All the ANN models were developed in Matlab R2017a using the neural network toolbox.

The number of hidden neurons for the ELM models was determined through a trial and error approach starting by a number of 5 hidden neurons and incrementing this number by 5 to a maximum of 150 hidden neurons. The complexity (as controlled by the density of the hidden layer) that lead to the highest R^2 values during validation was used as the optimal number of hidden neurons for the ELM models. Furthermore, five activation functions were tested, including the sigmoid 'Sig', the sine 'Sin', the hard-limit 'Hardlim', the triangular basis 'Tribas' and the radial basis function 'Radbas'. The model architecture that lead to the best generalization skill was reported in the results table. All the ELM models were developed in Matlab R2017a. The code available in [44] was modified and used for the ELM models' development.

Similarly, the number of neurons in the hidden layer of the LSTM models was progressively varied from a minimum of 2 to a maximum of 100, with increments of 2. The optimal number of hidden neurons, leading to the lowest generalization error was found to be equal to 20. The size of the lookback window was determined via a grid search on a range of [1, 30] and a size of 6 was selected as it lead to the highest downscaling accuracy as measured by R^2. Finally, the number of epochs and batches were also selected based on a grid search and the optimal parameters were found to vary between 150 and 200 and 20 and 30 respectively. Dropout was not used here. All LSTM models were developed in Python 3.5 using the Tensorflow library.

In order to compare the developed models' performance, the following measures of goodness of fit were used: the root mean square error ($RMSE$), the mean absolute error (MAE), the mean absolute deviation (MAD) as well as the coefficient of determination (R^2). The latter is expressed as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(O_i - P_i)^2}{\sum_{i=1}^{n}(O_i - \bar{O})^2} \tag{9}$$

where O_i is the observed value, P_i the predicted value, $\bar{O}$ the mean of the observed values, and n the number of data points. The coefficient of determination measures the degree of association among the observed and predicted values. Values for R^2 range from 0 to 1, with a value of 1 showing perfect predictive skill.

TABLE I: Results of the best models at test time for each station for task 1.

Station	Task 1				
	Method	Model structure	$RMSE$	R^2	R^2% inc.
1	ANN	(16-32-1)	1.92	0.91	2%
2	ANN	(16-25-1)	2.59	0.94	1%
3	LSTM	(16-20-1),lb=6	1.30	0.94	11%
4	ANN	(16-27-1)	1.03	0.94	3%
5	ANN	(16-33-1)	**4.00**	**0.84**	**3**%
6	LSTM	(16-20-1),lb=6	2.13	0.96	4%
7	ANN	(16-29-1)	1.30	0.92	2%
8	ANN	(16-19-1)	**0.85**	**0.96**	**1**%
9	LSTM	(16-20-1),lb=6	2.18	0.95	3%
10	LSTM	(16-20-1),lb=6	1.98	0.95	4%
11	ANN	(16-25-1)	1.34	0.95	1%
12	LSTM	(16-20-1),lb=6	2.28	0.94	5%

TABLE II: Results of the best models at test time for each station for task 2.

Station	Task 2				
	Method	Model structure	$RMSE$	R^2	R^2% inc.
1	ANN	(19-35-1)	1.83	0.92	18%
2	ANN	(19-35-1)	2.68	0.93	15%
3	LSTM	(19-20-1),lb=6	1.48	0.92	39%
4	ANN	(19-39-1)	1.10	0.92	3%
5	LSTM	(19-20-1),lb=6	**3.26**	**0.90**	**5**%
6	LSTM	(19-20-1),lb=6	2.33	0.95	45%
7	ANN	(19-27-1)	1.40	0.91	6%
8	ANN	(19-19-1)	**0.90**	**0.95**	**13**%
9	LSTM	(19-20-1),lb=6	2.30	0.94	59%
10	LSTM	(19-20-1),lb=6	1.93	0.95	5%
11	ANN	(19-28-1)	1.38	0.95	8%
12	LSTM	(19-20-1),lb=6	2.40	0.94	8%

V. Results and discussion

Tables I and II present the $RMSE$ and R^2 values for the best performing models for each station, along with the relative improvement (in %) in R^2 over MLR. In each case, the model structure for the best performing model is given as (number of inputs-number of hidden nodes-number of outputs) for each model. The size of the lookback window used for the LSTM models is also reported as 'lb'. The performance of all the models for each station is presented in the Appendices A and B, where the model structure for the MLR models is reported as (number of inputs-number of outputs).

The R^2% *inc.* columns in Tables I and II show the improvement over a standard MLR model. The machine learning-based techniques consistently outperform MLR for all stations in both tasks, and the improvement is especially large for the second task. The results in Table III show that overall, the daily mean air temperature was predicted with high accuracy as the RMSE values, at test time, range between 0.85 and 4.00 for the first task, and between 0.90 and 3.26 for the second task. For the

first task, downscaling to locations where past observations are available, LSTM proved to be the best performing model for 5 of the 12 stations examined. For the second task, where the objective is to downscale to locations with no past observational records, LSTM led to the best performance for 6 out of the 12 stations. For the other half of the stations, ANN was the best performing model.

The results in Appendices A and B further confirm that the ANN, ELM and LSTM models performed better than the multi-linear regression models for all of the 12 stations. The better performance observed for the LSTM models could be explained by their ability to learn long-term dependencies in sequential datasets. In the case of ordinary neural networks (ANNs), the inputs are assumed to be independent of each other, whereas LSTMs perform the same task for every training example in the sequence, with the output being depended on the previous computations. In other words, LSTMs have a "memory" which enables them to look back few steps (in this study, 6 steps back), capture information about what has been calculated so far and determine whether or not it is useful to pass it along to the next iteration. In general, the results show that using non-parametric and non-linear methods, such as those based on ANNs and LSTMs, for spatial downscaling leads to better performance. However, non-parametric methods are based on more complicated theory, compared to MLR, and the results can be difficult to interpret.

For both tasks, the station where the predictive accuracy was highest is station 8, which is located in Langara Island. The results show that the testing R^2 value exceeds 0.9 for both tasks. This result seems unusually high, considering that this station has only one station (s7) in the neighborhood.[1] There are two likely reasons for this result: 1.) station 7 and station 8 are both located on islands in the Canadian Cordillera in relatively close proximity, so that they likely have very similar micro-climates, making them excellent predictors for each other (and in fact both show very good results); 2.) by virtue of being located on islands in front of the coastal mountain ranges, the weather at the two stations will be dominated by large storm systems moving in from the Pacific, and remain relatively unaltered by topography, so that the predictability on the basis of low resolution predictors is unsurprisingly very good much better (the predictability of station 8 is likely higher, because it is less shielded by the islands, which are not resolved in the reanalysis fields).

On the other hand, the station where the performance was the worst is station 5, which corresponds to Fort St. James station. The poorer performance for this station, although not bad at $R^2 = 0.84$ for task 1, and $R^2 = 0.90$ for task 2, can be explained by the fact that this station has only one neighbor (s3), which is located in a region with a very different climate: station 5 is located in the northern Interior Plateau, which has a cold and fairly arid climate, due to the rain shadow of the Coast Mountains, while station 3 is located on the windward flanks of the Coast Mountains, which is a region dominated by coastal rainforest, where temperatures are moderated by the Pacific ocean and precipitation totals are very high.

[1]In principle this could indicate duplicated data records between station 8 and its neighbor; the datasets for both stations were checked and no duplicate records were found.

TABLE III: Results of the best models at training and testing for each station for part I.

Station	Worst - Station 5		Best - Station 8	
	Task 1	Task 2	Task 1	Task 2
Method	ANN	LSTM	ANN	ANN
Model Structure	(16-33-1)	(19-20-1),lb=6	(16-19-1)	(19-19-1)
$RMSE$	**4.00**	**3.26**	**0.85**	**0.90**
R^2	0.84	0.90	0.96	0.95
$R^2\%$ increase	3	5	1	13

These results show that the predictive power of the downscaling algorithm does not only depend on the structure of the network, but also on the representativeness of the stations used to train the network. Furthermore, they also suggest that the potential predictability depends on the specific region and is limited by the extend to which key characteristics are represented in the predictor fields.

When it comes to the impact of the models' structure on the performance of the machine learning techniques, we noticed that the performance only slightly changes as the number of hidden neurons is varied. In general, networks with a smaller number of hidden neurons gave poorer performance, and so did networks with a high number of hidden neurons, as they resulted in underfitting and overfitting respectively. Overall, the best performance was obtained when the number of hidden neurons varied between a minimum of 19 and a maximum of 39.

Finally, the results show that when it comes to the second task, the key factor impacting the results is the number and representativeness of stations in the neighborhood or square surrounding the station of interest. Apart from station 8 and 7, the smaller the number of neighboring stations, the lower the models's skill at downscaling the reanalysis data. Downscaling at stations 2 and 5, with 0 and 1 neighbor respectively, has resulted in the lowest accuracy, as measured by $RMSE$ and R^2 for both tasks.

Key difference between downscaling techniques

what structural features of LSTM confer its capacity to outperform ANN

VI. Conclusions and future work

This study presented a new downscaling method for two specific tasks: downscaling at locations where past observations are available to train the models, and downscaling for locations where there is no past record, using neighboring stations to train the models. We illustrated our proposed method in the challenging region of British Columbia, due to its complex topography. We compared the performance of three non-parametric and non-linear machine learning techniques and multi-linear regression applied to downscaling 55-year daily mean temperatures for selected stations in British Columbia.

The comparison showed that non-parametric and non-linear machine learning techniques, specifically ANN and LSTM proved superior to traditional MLR, and lead to more accurate results. The results indicate that our method offers a novel and portable downscaling framework that can be used for complex topographical regions (such as mountains) where the station coverage is sparse.

In this paper, we selected a very limited set of input data for prediction of daily mean temperatures. While the results were good, we anticipate they can be improved further by including additional information in the training data, including additional meteorological variables (e.g. wind speed and direction) and basic topographical data (e.g. orographic slope). However, there is a trade-off between predictive accuracy and generality. A key benefit of our approach over existing downscaling methods is that it does not depend on detailed topographical analysis of the region of interest, and so can be readily applied to any region.

In further work, we plan to test the application of these methods for downscaling additional climate variables, including precipitation and climate extremes (e.g. daily high and low temperatures), as these are important for assessing climate change impacts, and for planning adaptation strategies for future climate change.

REFERENCES

[1] Candela L., Tamoh K., Olivares G, and Gomez M. "Modelling impacts of climate change on water resources in ungauged and data-scarce watersheds. Application to the Siurana catchment (NE Spain)." In: *Science of the Total Environment* 440 (2012), pp. 253–260.

[2] Vandal T., Kodra E., Ganguly S., Michaelis A., Nemani R., and Ganguly A.R. "DeepSD: Generating High Resolution Climate Change Projections through Single Image Super-Resolution". In: *arXiv preprint arXiv:1703.03126* (2017).

[3] Jeffrey S.J., Carter J.O., Moodie K.B., and Beswick A.R. "Using spatial interpolation to construct a comprehensive archive of Australian climate data." In: *Environmental Modelling and Software* 16.4 (2001), pp. 309–330.

[4] Price D. T., McKenney D. W., Nalder I. A., Hutchinson M. F., and Kesteven J. L. "A comparison of two statistical methods for spatial interpolation of Canadian monthly mean climate data." In: *Agricultural and Forest Meteorology* 101 (2000), pp. 81–94.

[5] Mahfouf J.F., Brasnett B., and Gagnon S. "A Canadian precipitation analysis (CaPA) project: Description and preliminary results." In: *Atmosphere-ocean,* 45.1 (2007), pp. 1–17.

[6] Wilby R.L. and Dawson C.W. "The statistical downscaling model: insights from one decade of application". In: *International Journal of Climatology* 33.7 (2013), pp. 1707–1719.

[7] Woodward F.I. *Climate and Plant Distribution. 1. Vegetation and Climate.* Cambridge, U.K.: Cambridge University Press, 1987.

[8] Mujumdar P.P. and Kumar D.N. *Floods in a Changing Climate - Hydrologic Modeling*. Cambridge, U.K.: Cambridge University Press, 2013.

[9] Monteleoni C., Schmidt G., Saroha S., and Asplund E. "Tracking climate models". In: *Statistical Analysis and Data Mining* 4.4 (2011), pp. 372–392.

[10] Schmidt G.A., Ruedy R., Hansen J.E., Aleinov I., Bell N., Bauer M., Bauer S., Cairns B., Canuto V., Cheng Y., et al. "Present-day atmospheric simulations using GISS ModelE: Comparison to in situ, satellite, and reanalysis data". In: *Journal of Climate* 19.2 (2006), pp. 153–192.

[11] Maraun D., Wetterhall F., Ireson A.M., Chandler R.E., Kendon E.J., Widmann M., Brienen S., Rust H.W., Sauter T., Themebl M., Venema V.K.C., Chun K.P., Goodess C.M., Jones R.G., Onof C., Vrac M., and Thiele-Eich I. "Precipitation downscaling under climate change: Recent developments to bridge the gap between dynamical models and the end user." In: *Reviews of Geophysics* 48.3 (2010).

[12] Teutschbein C. and Seibert J. "Bias correction of regional climate model simulations for hydrological climate-change impact studies: Review and evaluation of different methods." In: *Journal of Hydrology* 3 (2012), pp. 12–29.

[13] Tveito O., Wegehenkel M., VanDerWel F., and Dobesch H. "Spatialisation of climatological and meteorological information with the support of GIS (Working Group 2)". In: *The Use of Geographic Information Systems in Climatology and Meteorology, Final Report* (2006), pp. 37–172.

[14] Stahl K., Moore R., Floyer J., Asplin M., and McKendry I. "Comparison of approaches for spatial interpolation of daily air temperature in a large region with complex topography and highly variable station density". In: *Agricultural and Forest Meteorology* 193.3 (2006), pp. 224–236.

[15] Hofstra N., Haylock M., New M., Jones P., and Frei C. "Comparison of six methods for the interpolation of daily European climate data". In: *Journal of Geophysical Research* 113 (2008).

[16] Rizzo D. and Dougherty D. "Characterization of aquifer properties using artificial neural networks: neural kriging." In: *Water Resources Research* 30.2 (1994), pp. 483–497.

[17] Tripathi S., Srinivas V.V., and Nanjundiah R.S. "Downscaling of precipitation for climate change scenarios: A support vector machine approach." In: *Journal of Hydrology* 330 (2006), pp. 621–640.

[18] Ho H.C., Knudby A., Sirovyak P., Xu Y., Hodul M., and Henderson S.B. "Mapping maximum urban air temperature on hot summer days." In: *Remote Sensing of Environment* 154 (2014), pp. 38–45.

[19] Pardo-Igúzquiza E., Chica-Olmo M., and Atkinson P.M. "Downscaling cokriging for image sharpening". In: *Remote Sensing of Environment* 102.1 (2006), pp. 86–98.

[20] Rodriguez-Galiano V., Pardo-Igúzquiza E., Sanchez-Castillo M., Chica-Olmo M., and Chica-Rivas M. "Downscaling Landsat 7 ETM+ thermal imagery using land surface temperature and NDVI images". In: *International Journal of Applied Earth Observation and Geoinformation* 18 (2012), pp. 515–527.

[21] Atkinson P.M. "Downscaling in remote sensing". In: *International Journal of Applied Earth Observation and Geoinformation* 22 (2013), pp. 106–114.

[22] Ducournau A. and Fablet R. "Deep learning for ocean remote sensing: an application of convolutional neural networks for super-resolution on satellite-derived SST data". In: *Pattern Recogniton in Remote Sensing (PRRS), 2016 9th IAPR Workshop on*. IEEE. 2016, pp. 1–6.

[23] Moitra S. "Single-image super-resolution techniques: a review". In: *International Journal for Science and Advance Research in Technology* 3.4 (2017), pp. 271–283.

[24] Couralt D. and Monestiez P. "Spatial interpolation of air temperature according to atmospheric circulation patterns in southeast France." In: *International Journal of Climatology* 19 (1999), pp. 365–378.

[25] Xia Y., Fabian P., Winterhalter M., and Zhao M. "Forest climatology: estimation and use of daily climatological data for Bavaria, Germany." In: *Agricultural and Forest Meteorology* 106.2 (2001), pp. 87–103.

[26] Garen D.C. and Marks D. "Spatially distributed energy balance snowmelt modelling in a mountainous river basin: estimation of meteorological inputs and verification of model results." In: *Journal of Hydrology* 315 (2005), pp. 126–153.

[27] Appelhans T., Mwangomo E., Hardy D.R., Hemp A., and Nauss T. "Evaluating machine learning approaches for the interpolation of monthly air temperature at Mt. Kilimanjaro, Tanzania." In: *Spatial Statistics: Part A* 14 (2015), pp. 91–113.

[28] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[29] Graves A. "Generating sequences with recurrent neural networks". In: *arXiv preprint arXiv:1308.0850* (2013).

[30] Greff K., Srivastava R.K., Koutnık J., Steunebrink B.R., and Schmidhuber J. "LSTM: A search space odyssey". In: *IEEE transactions on neural networks and learning systems* (2016).

[31] Glantz S.A. and Slinker B.K. *Primer of applied regression and analysis of variance. 2nd ed.* New York, NY: McGraw-Hill, 2001.

[32] Bishop C.M. *Neural networks for pattern recognition.* Oxford: Clarendon Press, 1995.

[33] Wan C., Xu Z., Xu Z., Pinson P., Dong Z.Y., and Wong K.P. "Probabilistic forecasting of wind power generation using extreme learning machine." In: *IEEE Transactions on Power Systems* 29.3 (2014), pp. 1033–1044.

[34] Huang G.B., Zhu Q.Y., and Siew C.K. "Extreme learning machine: A new learning scheme of feedforward neural networks." In: *IEEE International Joint Conference on Neural Networks* (2004), pp. 985–990.

[35] Butcher J.B., Verstraeten D., Schrauwen B., Day C.R., and Haycock P.W. "Reservoir computing and extreme learning machines for non-linear time-series data analysis." In: *Neural Networks* 38.5 (2013), pp. 76–89.

[36] Guo W., Xu T., and Lu Z. "An integrated chaotic time series prediction model based on efficient extreme learning machine and differential evolution." In: *Neural Computing and Applications* 27.4 (2016), pp. 883–898.

[37] Aalto J., Pirinen P., Heikkinen J., and Venäläinen A. "Spatial interpolation of monthly climate data for Finland: comparing the performance of kriging and generalized additive models". In: *Theoretical and Applied Climatology* 112.1 (2013), pp. 99–111. ISSN: 1434-4483.

[38] Martin R., Aler R., Valls J. M., and Galvan I. M. "Machine learning techniques for daily solar energy prediction and interpolation using numerical weather models". In: *Concurrency and Computation: Practice and Experience* 28.4 (2016). cpe.3631, pp. 1261–1274. ISSN: 1532-0634.

[39] Kalnay E., Kanamitsu M., Kistler R., Collins W., Deaven D., Gandin L., Iredell M., and Joseph D. "The NCEP/NCAR 40-Year Reanalysis Project." In: *Bulletin of the American Meteorological Society* 77.3 (1996), pp. 437–471.

[40] Environment and Climate Change Canada. *Adjusted and Homogenized Canadian Climate Data – Daily Temperature and Precipitation AHCCD – daily T&P.* 2017. URL: http://ccds-dscc.ec.gc.ca/index.php?page=homogenized-data.

[41] Vincent L. A., Wang X. L., Milewska E. J., Wan H., Yang F., and Swail V. "A second generation of homogenized Canadian monthly surface air temperature for climate trend analysis". In: *Journal of Geophysical Research* 117 (2012).

[42] Wanas N., Auda G., Kamel M.S., and Karray F. "On the optimal number of hidden nodes in a neural network". In: *Electrical and Computer Engineering, 1998. IEEE Canadian Conference on*. Vol. 2. IEEE. 1998, pp. 918–921.

[43] Mishra A.K. and Desai V.R. "Drought forecasting using feed-forward recursive neural network". In: *Ecological Modelling* 198.1 (2006), pp. 127–138.

[44] Huang G.B. "What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle". In: *Cognitive Computation* 7.3 (2015), pp. 263–278.

Appendix A: Results of the MLR, ANN, ELM and LSTM models at training and testing for each station for task 1.

Station	Method	Model structure	Training				Testing			
			$RMSE$	MAE	MAD	R^2	$RMSE$	MAE	MAD	R^2
1	MLR	(16-1)	1.979	1.532	5.842	0.906	2.104	1.550	5.219	0.895
	ANN	(16-32-1)	1.750	1.354	5.207	0.927	1.918	1.422	5.170	0.913
	ELM	(16-80-1), 'sin'	1.846	1.426	5.141	0.918	2.118	1.518	5.091	0.894
	LSTM	(16-20-1), lb=6	1.868	1.442	5.181	0.916	1.939	1.475	5.351	0.911
2	MLR	(16-1)	4.183	2.827	10.126	0.857	2.794	2.097	8.650	0.928
	ANN	(16-25-1)	3.891	2.555	8.594	0.877	2.590	1.863	8.383	0.938
	ELM	(16-135-1), 'radbas'	3.896	2.565	8.572	0.876	2.705	1.976	8.412	0.932
	LSTM	(16-20-1), lb=6	2.928	2.191	8.441	0.930	2.866	2.172	8.155	0.924
3	MLR	(16-1)	1.410	1.107	4.696	0.924	2.115	1.513	4.196	0.828
	ANN	(16-27-1)	1.233	0.969	4.219	0.942	1.873	1.324	4.243	0.865
	ELM	(16-135-1), 'sig'	1.293	1.015	4.189	0.936	2.055	1.440	4.219	0.837
	LSTM	(16-20-1), lb=6	1.292	1.004	4.285	0.936	1.297	1.017	4.344	0.935
4	MLR	(16-1)	1.204	0.944	3.661	0.906	1.216	0.943	3.110	0.908
	ANN	(16-27-1)	1.027	0.808	3.275	0.932	1.027	0.795	3.179	0.935
	ELM	(16-145-1), 'sin'	1.090	0.853	3.212	0.923	1.112	0.860	3.107	0.923
	LSTM	(16-20-1), lb=6	1.224	0.957	3.268	0.903	1.204	0.934	3.407	0.910
5	MLR	(16-1)	3.023	2.289	9.040	0.916	4.300	2.899	8.400	0.814
	ANN	(16-33-1)	2.585	1.941	8.246	0.939	3.998	2.634	8.070	0.839
	ELM	(16-140-1), 'radbas'	2.728	2.041	8.163	0.932	4.291	2.792	8.171	0.814
	LSTM	(16-20-1), lb=6	2.822	2.166	8.115	0.927	4.482	3.073	8.014	0.797
6	MLR	(16-1)	3.115	2.348	9.197	0.908	2.888	2.237	8.467	0.917
	ANN	(16-17-1)	2.781	2.072	8.279	0.928	2.460	1.880	8.470	0.940
	ELM	(16-145-1), 'sig'	2.842	2.115	8.239	0.924	2.596	1.980	8.431	0.933
	LSTM	(16-20-1), lb=6	2.148	1.649	8.362	0.956	2.132	1.655	8.374	0.955
7	MLR	(16-1)	1.329	1.027	4.357	0.923	1.430	1.109	3.796	0.906
	ANN	(16-29-1)	1.199	0.933	3.908	0.937	1.295	0.999	3.852	0.923
	ELM	(16-120-1), 'sin'	1.275	0.984	3.886	0.929	1.366	1.057	3.847	0.915
	LSTM	(16-20-1), lb=6	1.477	1.152	3.738	0.905	1.517	1.179	3.742	0.895
8	MLR	(16-1)	1.018	0.774	3.668	0.940	0.924	0.716	3.266	0.948
	ANN	(16-19-1)	0.930	0.718	3.390	0.950	0.850	0.658	3.303	0.956
	ELM	(16-120-1), 'sig'	0.957	0.732	3.357	0.947	0.903	0.695	3.277	0.950
	LSTM	(16-20-1), lb=6	1.136	0.862	3.296	0.925	1.088	0.820	3.301	0.928
9	MLR	(16-1)	2.645	2.040	8.466	0.917	2.643	2.068	7.711	0.922
	ANN	(16-27-1)	2.350	1.807	7.569	0.934	2.306	1.790	7.798	0.941
	ELM	(16-135-1), 'sig'	2.479	1.905	7.486	0.927	2.482	1.921	7.727	0.931
	LSTM	(16-20-1), lb=6	2.105	1.629	7.589	0.947	2.176	1.661	7.909	0.947
10	MLR	(16-1)	2.103	1.633	7.413	0.934	2.605	1.950	6.842	0.908
	ANN	(16-24-1)	1.881	1.451	6.796	0.947	2.370	1.742	7.033	0.923
	ELM	(16-150-1), 'sin'	1.978	1.535	6.715	0.942	2.534	1.879	6.913	0.913
	LSTM	(16-20-1), lb=6	1.741	1.339	6.802	0.955	1.975	1.488	7.204	0.947
11	MLR	(16-1)	1.446	1.133	5.444	0.942	1.495	1.176	4.911	0.943
	ANN	(16-25-1)	1.300	1.015	5.011	0.953	1.343	1.053	5.041	0.954
	ELM	(16-140-1), 'radbas'	1.349	1.056	4.949	0.949	1.447	1.140	4.915	0.946
	LSTM	(16-20-1), lb=6	1.566	1.232	5.022	0.932	1.529	1.219	5.270	0.940
12	MLR	(16-1)	3.027	2.280	8.648	0.900	3.041	2.299	7.919	0.895
	ANN	(16-16-1)	2.630	1.993	7.612	0.925	2.664	1.978	7.810	0.919
	ELM	(16-120-1), 'sig'	2.770	2.071	7.564	0.916	2.823	2.118	7.854	0.910
	LSTM	(16-20-1), lb=6	2.322	1.764	7.685	0.941	2.279	1.712	7.759	0.941

Appendix B: Results of the MLR, ANN, ELM and LSTM models at training and testing for each station for task 2.

Station	Method	Model structure	Training				Testing			
			$RMSE$	MAE	MAD	R^2	$RMSE$	MAE	MAD	R^2
1	MLR	(19-1)	2.566	1.915	7.672	0.911	3.243	2.592	6.583	0.748
	ANN	(19-35-1)	2.185	1.613	6.710	0.933	1.825	1.384	5.123	0.917
	ELM	(19-126-1), 'hardlim'	3.125	2.396	6.651	0.867	3.086	2.432	4.245	0.772
	LSTM	(19-20-1), lb = 6	2.197	1.716	6.687	0.934	2.248	1.773	5.314	0.879
2	MLR	(19-1)	1.286	0.949	4.059	0.920	5.049	3.425	6.089	0.790
	ANN	(19-35-1)	2.219	1.241	4.687	0.898	2.684	1.933	8.245	0.931
	ELM	(19-16-1), 'sig'	1.315	0.975	3.612	0.916	4.574	2.956	7.139	0.828
	LSTM	(19-20-1), lb = 6	1.355	1.028	3.603	0.911	5.710	3.283	7.692	0.731
3	MLR	(19-1)	2.211	1.542	6.295	0.922	3.388	2.986	4.480	0.563
	ANN	(19-37-1)	1.848	1.284	5.616	0.944	1.445	1.067	4.137	0.916
	ELM	(19-16-1), 'sig'	2.195	1.537	5.630	0.923	2.217	1.789	3.709	0.813
	LSTM	(19-20-1), lb = 6	2.145	1.550	5.898	0.926	1.475	1.135	4.489	0.917
4	MLR	(19-1)	1.527	1.186	5.136	0.925	1.307	1.020	3.382	0.891
	ANN	(19-39-1)	1.278	0.989	4.399	0.943	1.095	0.842	3.127	0.921
	ELM	(19-51-1), 'radbas'	1.471	1.139	4.529	0.931	1.294	0.988	3.285	0.893
	LSTM	(19-20-1), lb = 6	1.469	1.149	4.668	0.931	1.353	1.060	3.496	0.883
5	MLR	(19-1)	1.493	1.149	4.741	0.915	3.913	2.936	6.329	0.859
	ANN	(19-28-1)	1.944	1.394	6.003	0.945	3.457	2.316	7.957	0.880
	ELM	(19-81-1), 'tribas'	1.448	1.110	4.163	0.920	4.563	3.367	7.170	0.808
	LSTM	(19-20-1), lb = 6	1.325	1.032	4.316	0.933	3.256	2.345	7.782	0.902
6	MLR	(19-1)	2.733	2.082	8.331	0.911	7.084	6.209	7.388	0.524
	ANN	(19-35-1)	2.406	1.816	7.477	0.933	2.553	1.962	8.205	0.934
	ELM	(19-121-1), 'hardlim'	3.935	3.088	6.418	0.815	4.715	3.733	7.140	0.789
	LSTM	(19-20-1), lb = 6	2.218	1.737	7.352	0.941	2.330	1.794	8.250	0.948
7	MLR	(19-1)	1.009	0.768	3.664	0.941	1.807	1.455	3.622	0.857
	ANN	(19-27-1)	1.048	0.802	3.587	0.943	1.396	1.057	3.861	0.913
	ELM	(19-111-1), 'hardlim'	1.325	1.037	3.258	0.898	2.946	2.312	2.189	0.619
	LSTM	(19-20-1), lb = 6	1.159	0.876	3.364	0.922	1.568	1.215	3.816	0.892
8	MLR	(19-1)	1.339	1.035	4.366	0.921	1.709	1.392	3.362	0.829
	ANN	(19-19-1)	1.161	0.886	3.711	0.935	0.898	0.687	3.283	0.952
	ELM	(19-131-1), 'hardlim'	1.633	1.282	3.806	0.883	2.864	2.247	1.675	0.521
	LSTM	(19-20-1), lb = 6	1.464	1.138	3.840	0.906	1.220	0.914	3.366	0.913
9	MLR	(19-1)	2.741	2.055	7.954	0.907	7.256	6.701	6.795	0.380
	ANN	(19-32-1)	2.354	1.752	7.100	0.931	2.304	1.795	7.582	0.937
	ELM	(19-32-1), 'sin'	2.704	2.039	7.013	0.909	2.905	2.281	6.941	0.901
	LSTM	(19-20-1), lb = 6	2.076	1.577	7.223	0.947	2.300	1.807	7.198	0.938
10	MLR	(19-1)	3.033	2.312	8.935	0.907	2.603	2.035	7.632	0.900
	ANN	(19-39-1)	2.569	1.930	7.869	0.932	2.139	1.625	6.782	0.933
	ELM	(19-16-1), 'sig'	3.096	2.356	7.734	0.903	3.716	3.038	8.304	0.797
	LSTM	(19-20-1), lb = 6	2.266	1.755	7.867	0.948	1.930	1.506	7.012	0.945
11	MLR	(19-1)	1.775	1.350	4.874	0.888	2.158	1.751	3.914	0.872
	ANN	(19-28-1)	1.481	1.123	4.285	0.923	1.382	1.089	4.882	0.948
	ELM	(19-21-1), 'sig'	1.815	1.380	4.129	0.883	1.896	1.507	4.301	0.901
	LSTM	(19-20-1), lb = 6	1.643	1.271	4.103	0.904	1.808	1.437	4.608	0.910
12	MLR	(19-1)	2.729	2.084	7.942	0.907	3.546	2.858	7.297	0.862
	ANN	(19-31-1)	2.355	1.775	7.093	0.931	2.766	2.060	7.588	0.913
	ELM	(19-22-1), 'radbas'	2.802	2.154	6.953	0.902	3.139	2.431	7.410	0.892
	LSTM	(19-20-1), lb = 6	2.042	1.573	7.098	0.948	2.397	1.820	7.725	0.937