

# Data Filling of Wind Fields in Complex Terrain for high-resolution nowcasting

Nipun Gunawardena, Eric R. Pardyjak, Pierre Durand, Thierry Hedde

---

## Abstract

In this paper, two different methods for nowcasting and data filling spatially varying meteorological variables (wind direction/speed, specific humidity, and virtual potential temperature) at the microscale in regions marked by complex terrain are compared. Multivariable linear regression and artificial neural networks are implemented using data gathered from a system of eleven automated weather stations deployed in the Cadarache Valley of southeastern France from December 2016 to June 2017. It is found that the linear regression method can be used to nowcast almost as well as artificial neural networks. These methods can be used to estimate gaps in incomplete datasets or to perform in-depth field experiments using limited equipment.

## 1 Introduction

To study the meteorology of complex terrain such as urban or mountainous regions, researchers often conduct field experiments. During these field experiments, a large amount of data is typically collected with, but not limited to: tethered balloons, radiosondes, remote sensing instruments (LIDAR, SODAR, RASS, etc.), meteorological towers, and small, distributed weather stations [18]. Field experiments typically last from a few weeks [48], to a few months [16], to even a few years [56]. Regardless of the instrumentation used, it is often removed after the field experiment is completed, obviously eliminating the capability to measure weather in that area. However, in many cases, the field experiment is conducted in an area that has permanent weather stations installed. For example, the MATERHORN experiment [20] at Dugway Proving Grounds, Utah, the BLLAST experiment in southern France [40], and the KASCADE experiment described herein, were all field experiments where the scientific equipment used supplemented permanent weather stations. In such an instance, the end of the field experiment doesn't necessitate a cessation of measurements, just a substantial reduction of measurements. In other words, the field experiment offers the opportunity to make dense measurements, while the operational instrumentation deployment offers sparse measurements.

In a situation such as this, it is often desirable to continue to get dense measurements after the field experiment is over. While it is obviously impossible to get true meteorological

measurements from an area where a sensor is no longer deployed, the process of *nowcasting* can allow for approximations of what the measurements would be. In the meteorological sense, nowcasting is a process that describes the current and near-future (hourly scale) state of the atmosphere [45]. The process of nowcasting is also used in the field of economics, where it is defined “as the prediction of the present, the very near future, and the very recent past” [7]. In this paper, we present and compare two methods of nowcasting dense atmospheric measurements in complex terrain using only a sparse set of deployed sensors. The two methods are multiple variable linear regression and artificial neural networks, and we specifically are predicting present values.

## 2 Background

There are two main reasons to study the meteorology of complex terrain. The first is to increase our understanding of the flow physics associated with complex terrain that inherently violate many of the meteorological assumptions usually invoked in models and analysis. However, the second main reason to study the meteorology of complex terrain is to improve our weather prediction capabilities [11]. For example, since over half of the world’s population lives in cities [19], and air pollution is considered a serious human health risk [31], it is often important to predict air pollution concentrations in cities [11]. This can involve studying phenomena such as cold air pools [36] and valley flows [43], both of which affect pollution levels and fog formation [27]. Another reason to improve weather prediction in complex terrain is to better predict snow and ice storms [11]. Adams et al. [2] estimate that improved snow prediction can potentially produce 1.3 billion US dollars of benefit annually, in addition to the number of lives saved due to prevented accidents. It is important to note that the atmosphere in complex terrain often exhibits high spatial and temporal variability [1, 38], which can make weather prediction difficult at the microscale. These microscale difficulties contribute to weather prediction difficulties at larger scales, which is why boundary layer meteorology is necessary to improve large-scale weather models [65].

While these are very broad reasons to study the meteorology of complex terrain, there are some focused applications as well. The agricultural community is interested in airborne pollen and pathogen transmission, evapotranspiration, as well as small-scale microclimate and its effects on crops [6, 42, 39, 49]. The nuclear energy community is often required to consider the side-effects of a breach, and how the contaminants spread [16, 8, 64]. The US military is interested in dust transport and its effects on military equipment [68], while the EPA is interested in dust transport because of air pollution [54, 55]. Finally, urban planners are interested in reducing pollution [37], increasing pedestrian comfort [61], and installing clean energy sources [32]. A good high-level overview of the benefits of complex terrain meteorology studies can be found in Chow et al.’s review [11].

A method that is also important to meteorology (and other fields) is nowcasting. As

mentioned previously, nowcasting is a process that describes the current and near-future (hourly scale) state of the atmosphere [45]. Nowcasting has been used in many different areas of meteorology. Both Xu et al. [71] and Novak [50] have presented systems to nowcast storms using radar, and Wilson et al. [69] published a review of storm nowcasting methods prior to 1998. Rasmussen et al. [59] created a nowcasting system to aid in de-icing decision making, and Gultepe et al. [24] attempted to use nowcasting to predict ice fog, though they were not successful. Yates et al. [72] used a precipitation nowcasting system to improve flash flood prediction. Demetriades et al. [14] use a lightning strike detection network to nowcast information about tropical cyclones. Han et al. [26] used support vector machines to nowcast radar reflectivity data, showing that machine learning has a place in meteorological nowcasting.

The differences between the nowcasting systems presented in this paper, and those presented above are significant. The nowcasting systems above are applied to large scales, sometimes over hundreds of kilometers. The nowcasting systems presented here are designed to be used at small scales, of the order of a few kilometers maximum. Another large difference is the time scale. All the aforementioned studies nowcast in the sense of predicting the near future, whereas our systems predict present values. In addition, many of the previous studies use radar as inputs to their nowcasting systems, whereas we use averaged data from distributed sensor stations. Therefore, it is important to review nowcasting systems in the literature which use distributed sensor stations or wireless sensor networks.

Distributed sensor stations or wireless sensor networks (WSNs) are systems of small meteorological stations that are easily deployed en masse, and provide excellent spatial coverage of a given area. They have been used in wildlife habitat monitoring [44], forest fire detection [73], structural health monitoring [33], and more [3]. Distributed sensor systems and WSNs have been used for nowcasting. Furquim et al. [23] used wireless sensor networks and machine learning techniques to forecast flash floods. They too consider nowcasting to be predicting the near future, instead of the present as done here. De Haan et al. [13] use meteorological data derived from deployed aircraft to improve measured wind fields, and therefore improve nowcasting techniques. While aircraft are not traditional distributed sensor systems, they are deployed in great numbers and offer very high spatial coverage. Furquim et al. [23] used a WSN and several different machine learning techniques, including artificial neural networks, to nowcast flash floods in São Paulo, Brazil.

As outlined in Shalev-Shwartz and Ben-David's book [60], artificial neural networks (ANNs) are a class of biologically-inspired algorithms that can be used for regression or classification tasks. While there are many types of ANNs, the simplest is the multilayer perceptron, also known as a standard feedforward neural network. A feedforward neural network is a directed acyclic graph where the nodes are called *neurons*, and the edges are called *connections*. Each neuron has an associated *bias* and *activation function*. Each connection has an associated *weight*. The graph is organized into layers, where each layer contains a certain amount of neurons and is connected to the layer above and below it.

The bottommost layer is called the *input layer*, and any gathered data are passed into the neurons of the input layer. The input layer has as many neurons as there are *input variables*. The topmost layer is called the output layer, and is the output of the entire feedforward network. There can be multiple outputs, and the number of neurons in the output layer is equal to the number of *target variables*. The layers in-between the input layer and the output layer are called the *hidden layers*. Therefore, data are fed into the input layer; its output is then fed into the first hidden layer, whose output goes into the second hidden layer, and so on, until the output from the last is sent to the output layer. The output of any given neuron is the weighted sum of all the neurons in the layer before it, passed through the activation function. The values of the weights and biases are found by minimizing the network’s mean squared error, in a process known as *backpropagation*. The process of finding the mean squared error using known data is called *training*, while using the ANN to predict unknown values is called *testing*. More information can be found the previously cited book by Shalev-Shwartz and Ben-David.

While ANNs can be used for classification or regression purposes (e.g., Furquim et al. [23] used an ANN to whether certain conditions would lead to a flash flood or not), in this paper and in many other nowcasting contexts, ANNs are used for the purpose of regression. For example, French et al. [21] used an ANN to spatially nowcast rainfall up to one hour ahead. Oztopal et al. [53] used an ANN to predict the wind speed at one station, given the wind speeds at nine different stations. While this is similar to the work presented in herein, Oztopal et al. did it on a much larger spatial scale. Their stations were distributed on a scale of hundreds of kilometers, and our work focuses on scales of a few kilometers.

Philippopoulos et al. [56] also used ANNs to predict wind speeds, though this time on the spatial scale of tens of kilometers. They compared the ANN’s performance to several spatial interpolation methods. In this paper, we compare our ANN performance to a multiple linear regression performance, which they do not do. In addition, we show that an ANN can predict variables other than wind speed, on a much shorter time scale.

Benvenuto et al. [10] use an air quality monitoring network in Mestre, Italy, and ANNs to nowcast pollution concentrations in the near future and to interpolate missing data. While Benvenuto’s work is similar to ours, there are some key differences. Benvenuto nowcasts for the near future (one hour ahead and three hours ahead), while we predict the present values. Benvenuto et al. also interpolates missing data using an ANN that predicts forward in time. Here we show that the ANN doesn’t have to predict forward in time to interpolate missing data. Finally, Benvenuto et al. used data gathered in an urban area, whereas we used data gathered in a hilly area. Videnova et al. [67] also used an ANN to predict air pollution, and they also predicted forward in time.

In this paper, we use data gathered from the Cadarache Valley, a valley with complex topography located in southeast France, which has been the subject of several studies designed to better forecast local-scale winds. For example, Duine et al. [16] developed a simple method, based on potential temperature differences routinely observed from a tall tower, to nowcast the existence of down-valley winds. More recently, Dupuy et al. [17] used

an ANN to downscale Weather Research and Forecasting (WRF,[62]) forecasts. Instead of using observational data as neural network inputs, Dupuy et al. used a low-resolution WRF output as inputs to an ANN. Their work effectively demonstrates that ANNs can be used to downscale physics-based weather models.

One novel aspect of the work presented herein is the comparison of results from ANNs and multiple linear regressions (MLRs). A multiple linear regression is a statistical tool where a set of *explanatory variables* linearly models a single *target variable* [60]. Linear regressions take the form of  $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$ , where  $y$  is the target variable,  $x_p$  are the explanatory variables, and  $\beta_p$  are the *regression coefficients*. Since multiple linear regression is considered a basic statistical tool, the general topic will not be explained here any further. A more in-depth discussion of the method can be found in Shalev-Shwartz and Ben-David's book [60]. There has been some previous work done in the atmospheric science domain comparing ANN and MLR performance. For example, Sousa et al. [63] used MLRs and ANNs to predict next day hourly ozone concentrations. Sousa et al. also analyzed the performance of ANNs with principal components as the inputs, which is something we do not do.

We are predicting the measurements at any given sensor station using the measurements from other stations in the area. While we (and many others) call this nowcasting, the underlying principle is the same as spatial interpolation. As such, there are other methods that are capable of this task that are worth mentioning. One of the most common methods, called kriging, has been used by both Asa [5] and Friedland et al. [22] to spatially interpolate wind data. The name "kriging" is in fact somewhat unique to geospatial statistics applications, and is more generally known as a gaussian process [58]. Osborne et al. [52] used gaussian processes to interpolate sensor readings. Hart et al. [28] used sensor stations along with satellite imagery to spatially interpolate evapotranspiration data. Finally, both Apaydin et al. [4] and Luo et al. [41] have written spatial interpolation comparison papers, where they compare methods such as kriging, inverse distance weighting, polynomial interpolation, splines, and more.

Ultimately, we believe this is novel work. We are nowcasting current values, while many others nowcast in the near future. We also nowcast several different environmental variables, on small spatial scales in complex terrain, and over short time scales. Finally, we compare ANN performance to MLR performance, while many others only use ANNs.

## 3 Methods

### 3.1 Experiment Overview

The data used for this publication were gathered during the Katabatic winds and Stability over Cadarache for Dispersion of Effluents (KASCADE) experiment of 2017. KASCADE 2017 is a follow-on experiment to the KASCADE experiment conducted in 2013 [16] that was focused on understanding the vertical structure of the atmosphere in the Cadarache

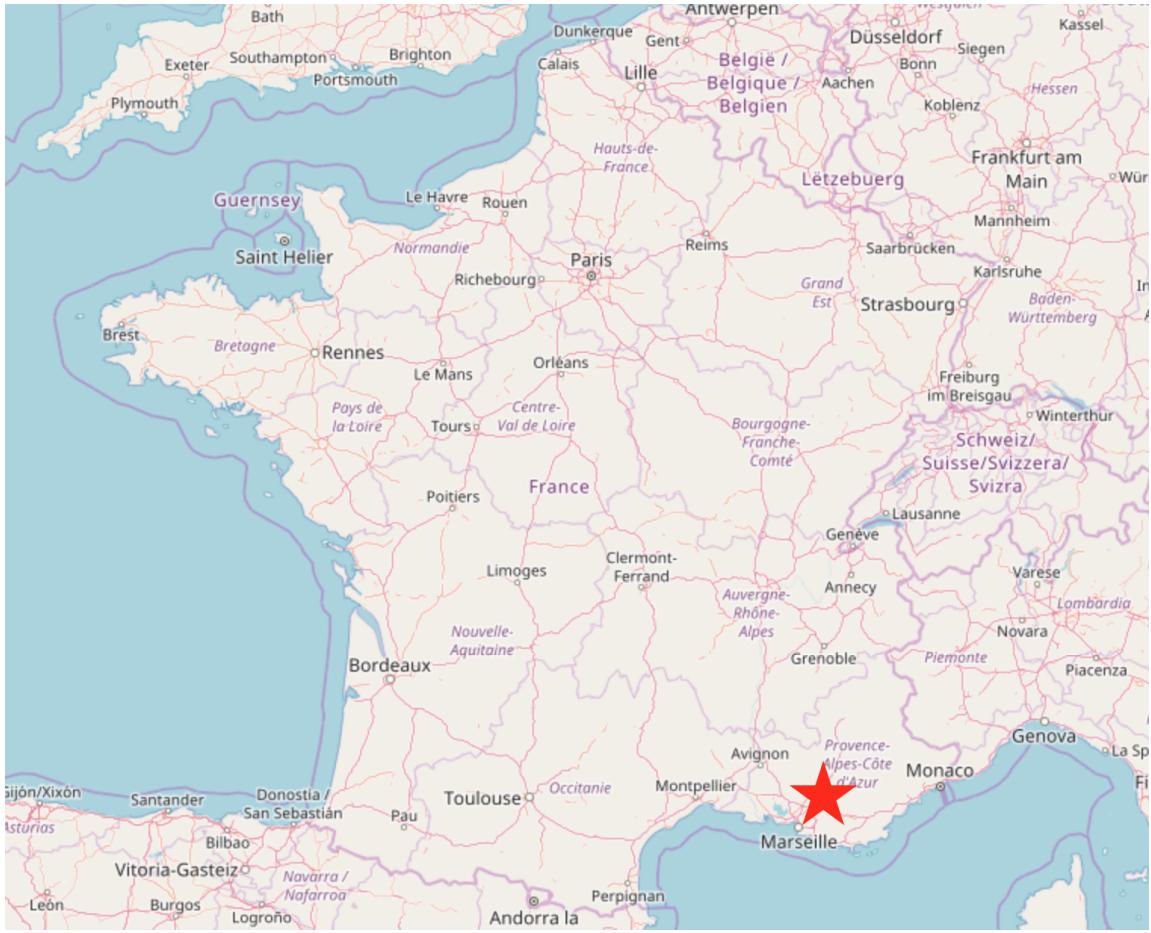


Figure 1: Map of Cadarache valley location in South-Eastern France. ©OpenStreetMap contributors.

Valley at night during stable atmospheric conditions.

KASCADE 2017 was conducted in the Cadarache Valley of the Bouches-du-Rhône department in southeastern France from December 2016 through June 2017 (See Figure 1). The Cadarache Valley contains the French Alternative energies and Atomic Energy Commission (CEA) research center and the International Thermonuclear Experimental Reactor (ITER) is located in the adjacent Durance Valley. The CEA performs various types of nuclear research including the study of contaminant dispersion in the event of an accident. To better understand and predict dispersion events, it is critical to have detailed knowledge of small-scale winds and other atmospheric variables. Therefore, increasing our understanding of these phenomena was the main objective of the experiment.

As illustrated in Figure 2, the Cadarache valley is a small valley about 6 km long by

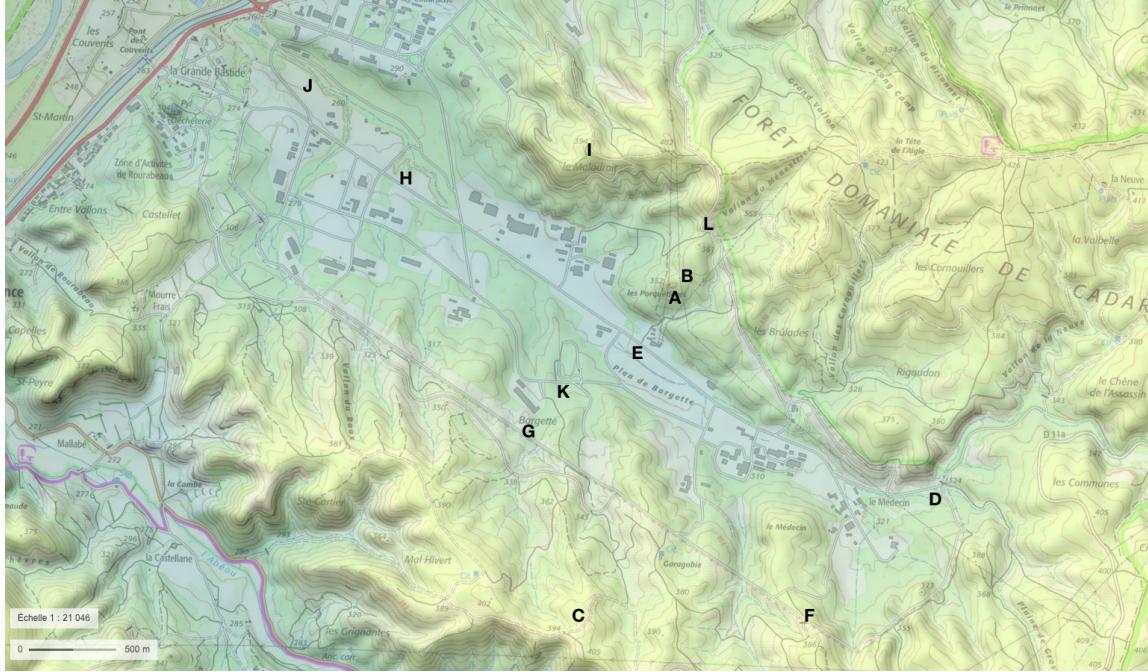


Figure 2: Map of Cadarache valley. The instrument locations are marked by the bold letters. [Geoportail.gouv.fr](http://Geoportail.gouv.fr) [institut national de l'information géographique et forestière (IGN)].



1 km wide. The elevation difference between the floor of the valley and the peaks are about 100 m. The mouth of the valley is connected to the Durance valley, which runs approximately perpendicular to the Cadarache Valley [17]. The land cover and land use within the valley is heterogeneous, with a combination of buildings, roads, grassy areas, and light forests.

For KASCADE 2017, the Cadarache Valley and the surrounding areas were heavily instrumented. Included in the deployment were: four sonic anemometer stations, one surface flux station, two sodar stations, wind and temperature measurements from a 110 m tower, two general meteorological stations, and 12 Local Energy-budget Measurement Stations (LEMS). In addition to these continuous observations, during ten Intensive Observation Periods (IOPs), radiosondes were released every three hours. In this paper, we use a subset of all the data collected. Namely, we use the data collected by the LEMS from January through March 2017.

LEMS are small, low-cost meteorological stations that are capable of taking surface and subsurface measurements. The LEMS used for this experiment are the second version of the instrument. The first version was designed, built, and characterized in 2013 [25]. The second version of the LEMS has a better radiation shield (the Socrima Multiplate radiation shield outlined in van der Meulen et al. [66]), a better processor, and updated

Name	Latitude	Longitude	Elevation (m)	
LEMS A	43.68483056	5.768038889	332	
LEMS B	43.68568056	5.76885	347	
LEMS C	43.66839167	5.761425	397	
LEMS D	43.67518889	5.786719444	328	
LEMS E	43.68263889	5.765680556	293	
LEMS F	43.66871111	5.777919444	383	
LEMS G	43.67848889	5.757638889	325	
LEMS H	43.69141111	5.749188889	276	
LEMS I	43.693	5.762538889	385	
LEMS J	43.69548056	5.743236111	262	
LEMS K	43.68038889	5.760038889	317	
LEMS L	43.68879722	5.770719444	368	

Table 1: Table of LEMS locations

sensors. The LEMS are open source, and information and build instructions can be found at <https://github.com/madvoid/LEMSv2>.

Each LEMS at Cadarache measures the following variables at approximately 2 m: wind speed and direction, incoming shortwave solar radiation, air temperature, and air relative humidity. Barometric pressure is measured at approximately 1 m. In addition, LEMS measure surface temperature, as well as soil moisture content and temperature at two different heights ( $\approx 5$  and  $\approx 25$  cm) below the surface. The heights of the sensors relative to the ground for each LEMS is approximately the same, and each LEMS has the same kind of sensor for each measurement. The wind speed and direction measurement for each LEMS were made using a cup and vane anemometer. Therefore, the data can be inaccurate at low wind speeds, and may also demonstrate overspeed problems as observed in the literature [34, 70]. The LEMS were deployed at 12 different locations around the Cadarache Valley. Each location can be viewed in Figure 2, and information about each LEMS location can be seen in Table 1.

Each LEMS station gathered data at 0.1 Hz. The data were quality controlled and averaged, with an averaging period of five minutes. These 5-minute averages are used for all methods described in this paper.

### 3.2 ANN Details

The ANNs used in this paper were implemented using MATLAB’s Neural Network Toolbox [9]. Each ANN was a standard feedforward neural network with one hidden layer. The number of hidden nodes in the hidden layer, as well as the number of inputs and outputs, changed with each experiment, so those values will be presented in each of the results sections. The initial values for the weights and biases of the neural networks were

randomly generated, and were dependent on the random seed. Since the random seed was varied across some experiments, they will be presented alongside the results. While many ANNs use stochastic gradient descent (SGD) as their training algorithm [60], we use the Levenberg-Marquardt algorithm. The Levenberg-Marquardt algorithm is used for training since it is recommended by MATLAB as being the fastest training algorithm [46]. The transfer function for the hidden layer is the hyperbolic tangent sigmoid function, while the transfer function for the output layer is the linear transfer function. The ANN performance function is the mean squared error (MSE), and normalization and regularization happens internally in MATLAB. MATLAB can do preprocessing within the Neural Network Toolbox. For this application, MATLAB's preprocessing consisted of removing constant inputs, and mapping the minimum and maximum values of all inputs to -1 and +1 respectively. The inverse of these preprocessing steps were taken for the output of the network. While we created our own training and testing data for the experiments, it is important to note that MATLAB uses the dataset to create its own internal training, testing, and validation data partitions. For all the experiments conducted, MATLAB split the given training data into 75% internal training data, 20% internal validation data, and 5% internal testing data. The training and testing splits we created will be presented alongside the results for each experiment. Since the inputs and outputs for each experiment are different, they will also be presented alongside the results for each experiment. Finally, ensemble averaging was frequently implemented. When ensemble averaging is utilized, multiple models with different initial weights are trained, and their outputs averaged [35]. This ensemble model typically has greater performance than a single model, and is less likely to show outlier performance. If an ensemble average is used, it will be specified alongside the results.



### 3.3 MLR Details

As with the ANNs, the MLRs were implemented using MATLAB. In particular, the Statistics and Machine Learning Toolbox [47] was used. Since the inputs, outputs, and training/testing splits are experiment dependent, they will be presented alongside the results. No extra preprocessing was done for any of the MLRs. Since the “linear” part of linear regression refers to the regression coefficients, it is possible to have a linear regression model that is not linear with respect to the explanatory variables, e.g.  $y = \beta_0 1 + \beta_1 x_1 + \beta_2 x_1 x_2 + \beta_3 x_3^2$ . However, that was not done for any of the MLRs in this paper. That is, all MLR models are of the form  $y = \beta_0 1 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$ . While ANNs have many hyperparameters such as batch size, learning rate, loss functions, weight initialization, etc., MLRs do not. Therefore, the MLRs can be run without specifying many hyperparameters beforehand.

## 4 Results and Discussion

This section is broken into four subsections. Subsections 4.1 and 4.3 include results from the ANN and the MLR model tests respectively. These two sections show the results from tests that were performed identically except for the nowcasting methods used. As a result, the plots in these two sections have identical formatting, are directly comparable, and should be viewed with the objective of comparing the two different models. Subsections 4.2 and 4.4 discuss the results from both models. Specific conclusions about the hyperparameters of each models can be found there.

### 4.1 ANN Results

To test the effectiveness of an ANN in nowcasting microclimate parameters, a specific test was run. In this test, the wind components, surface temperature, barometric pressure, and virtual potential temperature from LEMS I, J, and K were used to predict a given parameter (i.e., virtual potential temperature, specific humidity, and wind speed components) from each of the other LEMS, excluding LEMS C (i.e., the parameters from LEMS A, B, D, E, F, G, H, and L were predicted). LEMS I, J, and K were used because they captured the different kinds of flows present in the Cadarache Valley: slope flows, valley flows, and ridge flows. Since they represented the different flows present in the valley, we hypothesized that they would be the best predictor variables. They were also at different elevations, so the temperature differences due to the adiabatic lapse rate could be captured. LEMS C *wasn't* used because a complete dataset for that station *wasn't* available. Since there were eight targets, eight different ANNs were trained, each with 15 inputs (5 parameters each from 3 LEMS), and one output. Each ANN had fourteen hidden nodes, and were randomly initialized using the same seed. Each figure represents a different target variable. Each figure shows a different target variable:

- Figures 3 and 4 show the ANN prediction of the U component of the wind velocity
- Figures 5 and 6 show the ANN prediction of the V component of the wind velocity
- Figures 7 and 8 show the ANN prediction of the virtual potential temperature
- Figures 9 and 10 show the ANN prediction of the specific humidity

There are two test periods for each variable. These two test periods were chosen because they exhibit very different flow characteristics, giving a poorly trained nowcasting method more opportunity to fail. The test period starting 15 January 2017 displayed a very stable atmosphere with cold air pooling (**synoptic driven flows**). The second test period, starting 27 January 2017, displayed an atmosphere with stronger winds and broken stratification (**thermally driven flows**). Since **thermally driven flows break up synoptic flows**, these two test periods represent two extremes.

In each figure, there are 3 subfigures. The training data is identical for all runs: 5-minute averages of the data from 16 December 2016 to 15 March 2017, excluding the given run's test data. This training period was chosen because it is the period where there was a full deployment of sensors. The data were averaged every five minutes to smooth out noise. The following list gives a description of the subfigures:

- Subfigure (a) shows the ANN prediction time series, along with the measured time series, for three different LEMS (A, B, and D). The dots are the actual data, and the solid lines are the ANN predictions.
- Subfigure (b) shows the 1-to-1 plots of the ANN predictions vs. the measured data. The solid line is a linear fit of the data, and the  $R^2$  value is displayed on each plot. The title for each subplot is the target LEMS.
- Subfigure (c) shows statistics for the ANN prediction time series, along with statistics for the measured time series. Two statistics are shown. The first is the standard deviation of all **eight** target variables, for that time step. The second is the absolute difference between the minimum value target variable and the maximum value target variable, for that time step.



These tests will be shown again in section 4.3, except with an MLR as the prediction model instead of an ANN.

## 4.2 ANN Discussion

All neural networks have a number of hyperparameters (parameters which are not part of the training process) [12] that can be tuned to change the model performance. In this section, we present the results of tuning some, but not all, of the hyperparameters available in the model. Since the MATLAB Neural Network Toolbox was used for the implementation, many of the default parameters of the toolbox are used, and not expanded upon here. Some example hyperparameters that aren't discussed are the hidden node activation function, the training algorithm, and the training batch size. Exploration of the variation of these hyperparameters is presented in various sources in the literature (e.g., [51]). We also do not explore the number of hidden layers used in the ANNs, as there is often no practical reason to have more than one hidden layer [29].

One of the major hyperparameters that can be tuned is the number of hidden nodes in the hidden layers. To determine the optimal number of hidden nodes for our application, we ran a simple test where we varied the number of nodes in the ANN, but kept all else constant. In this test, we used the U component of the wind velocity data from all the LEMS (except LEMS E) to predict the U component of the wind velocity data at LEMS E. Note that the inputs for this test only consist of the wind velocity U component, whereas the inputs for the tests described in 4.1 consist of the wind components, virtual potential temperature, surface temperature, and barometric pressure. Testing data were taken from



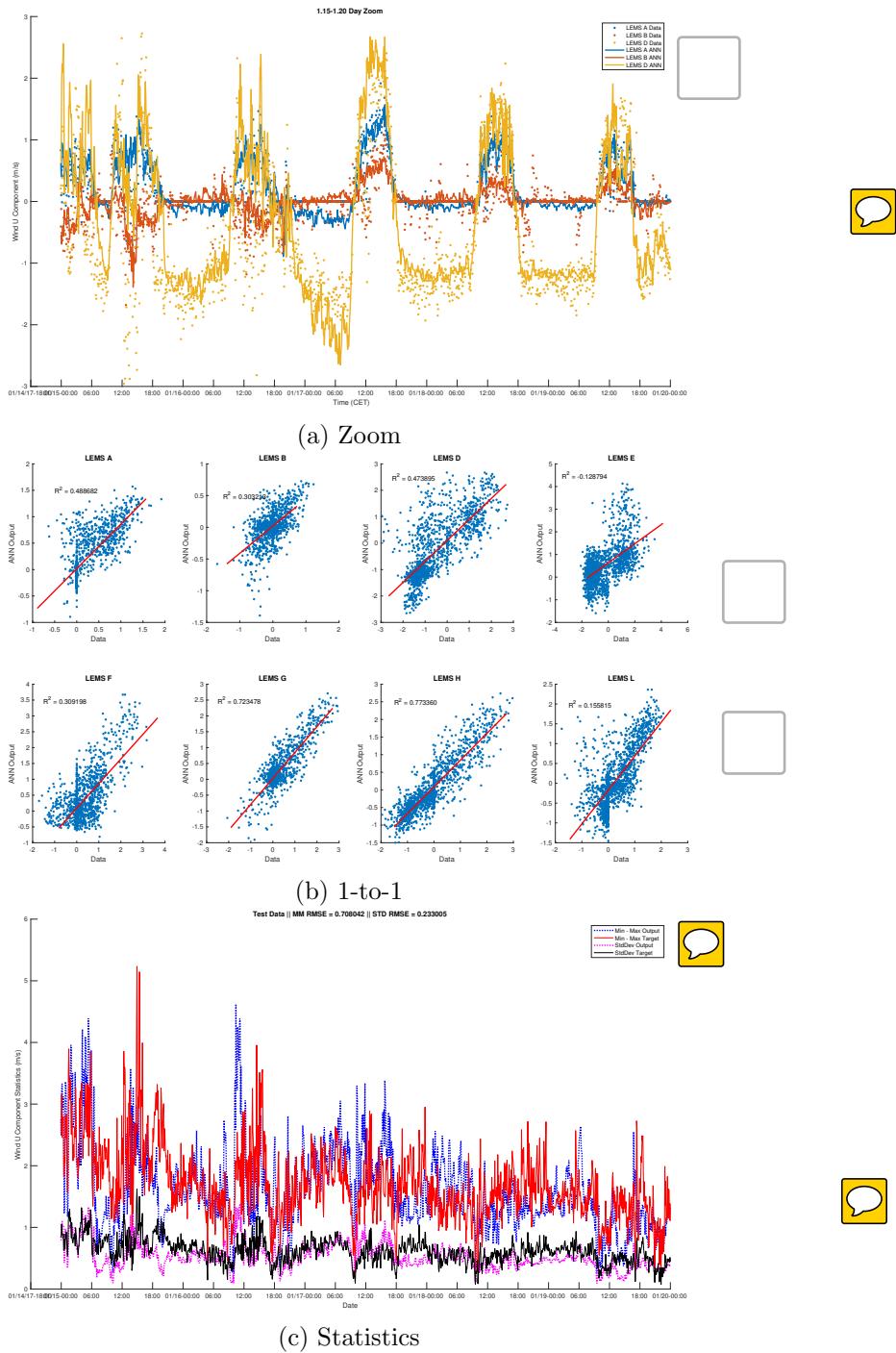
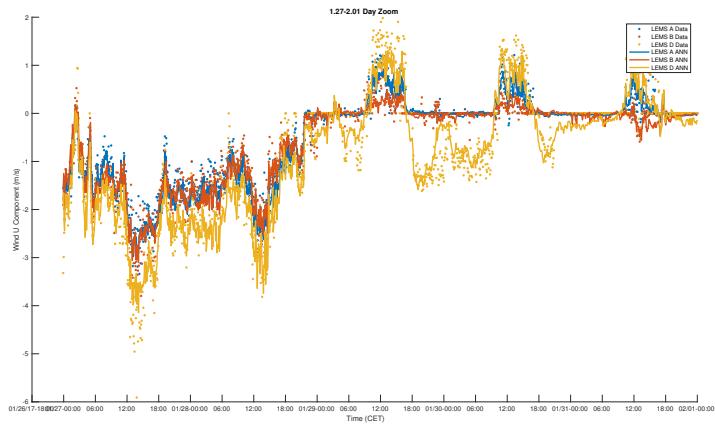
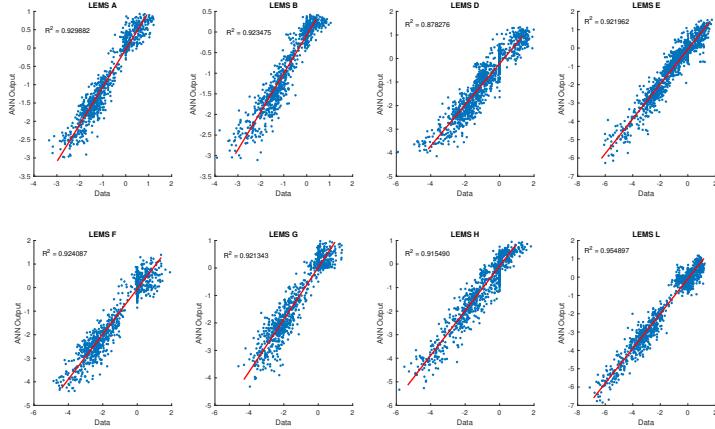


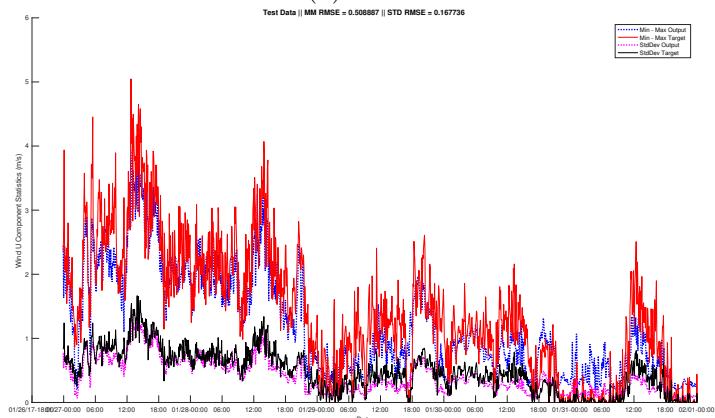
Figure 3: ANN prediction of U component of wind velocity 2017/1/15 - 2017/1/20



(a) Zoom

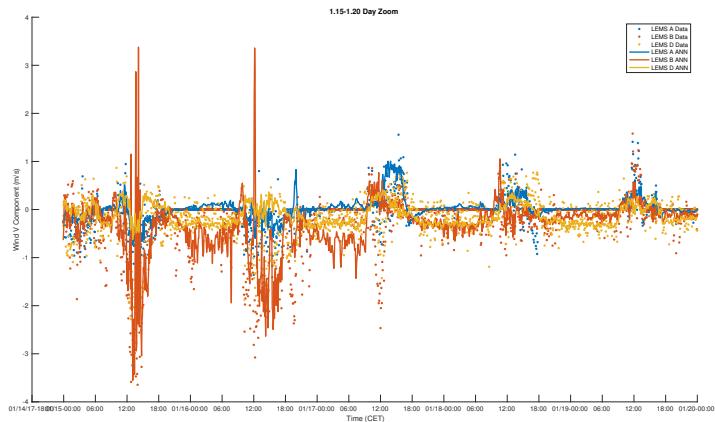


(b) 1-to-1

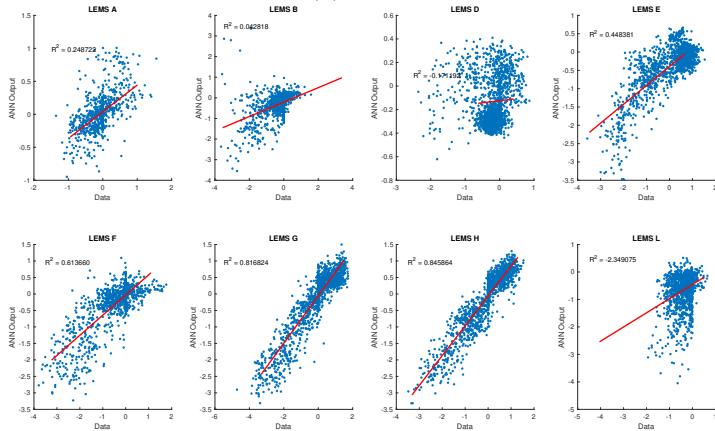


(c) Statistics

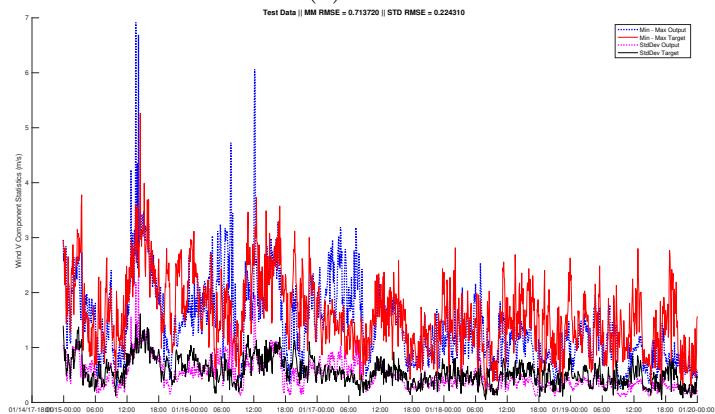
Figure 4: ANN prediction of U component of wind velocity 2017/1/27 - 2017/2/01



(a) Zoom

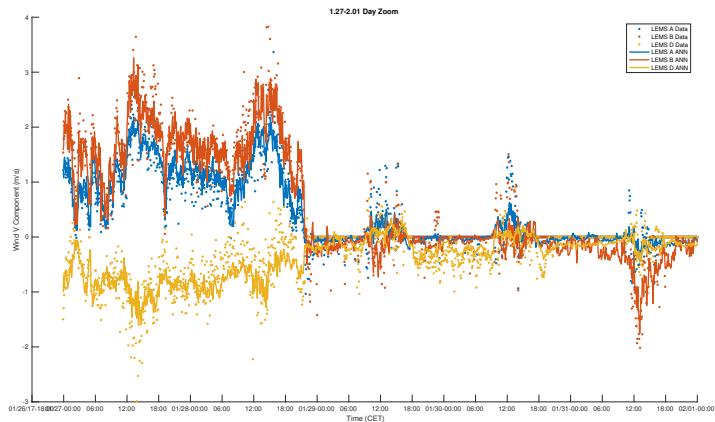


(b) 1-to-1

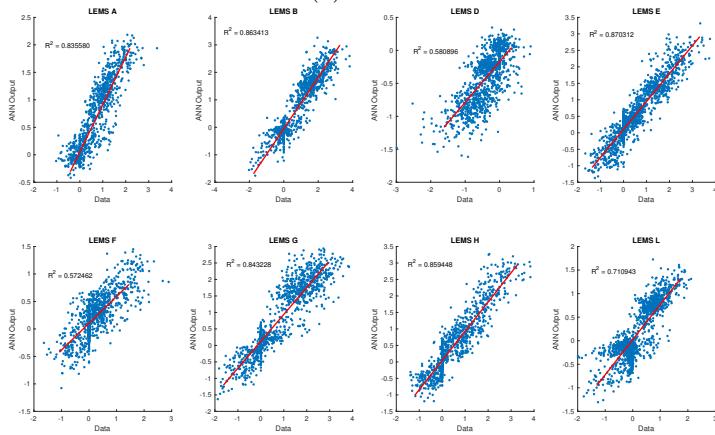


(c) Statistics

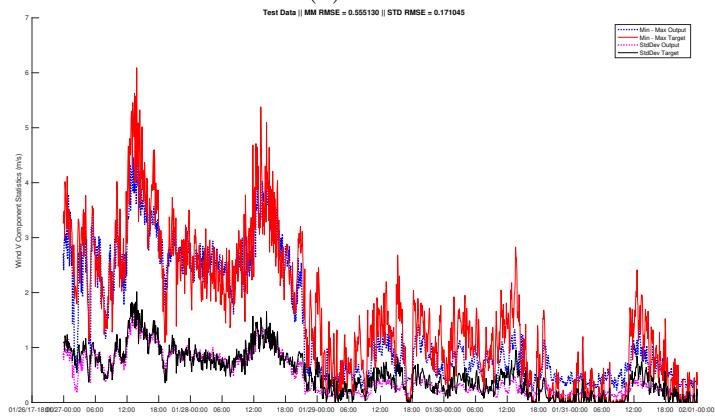
Figure 5: ANN prediction of V component of wind velocity 2017/1/15 - 2017/1/20



(a) Zoom



(b) 1-to-1



(c) Statistics

Figure 6: ANN prediction of V component of wind velocity 2017/1/27 - 2017/2/01

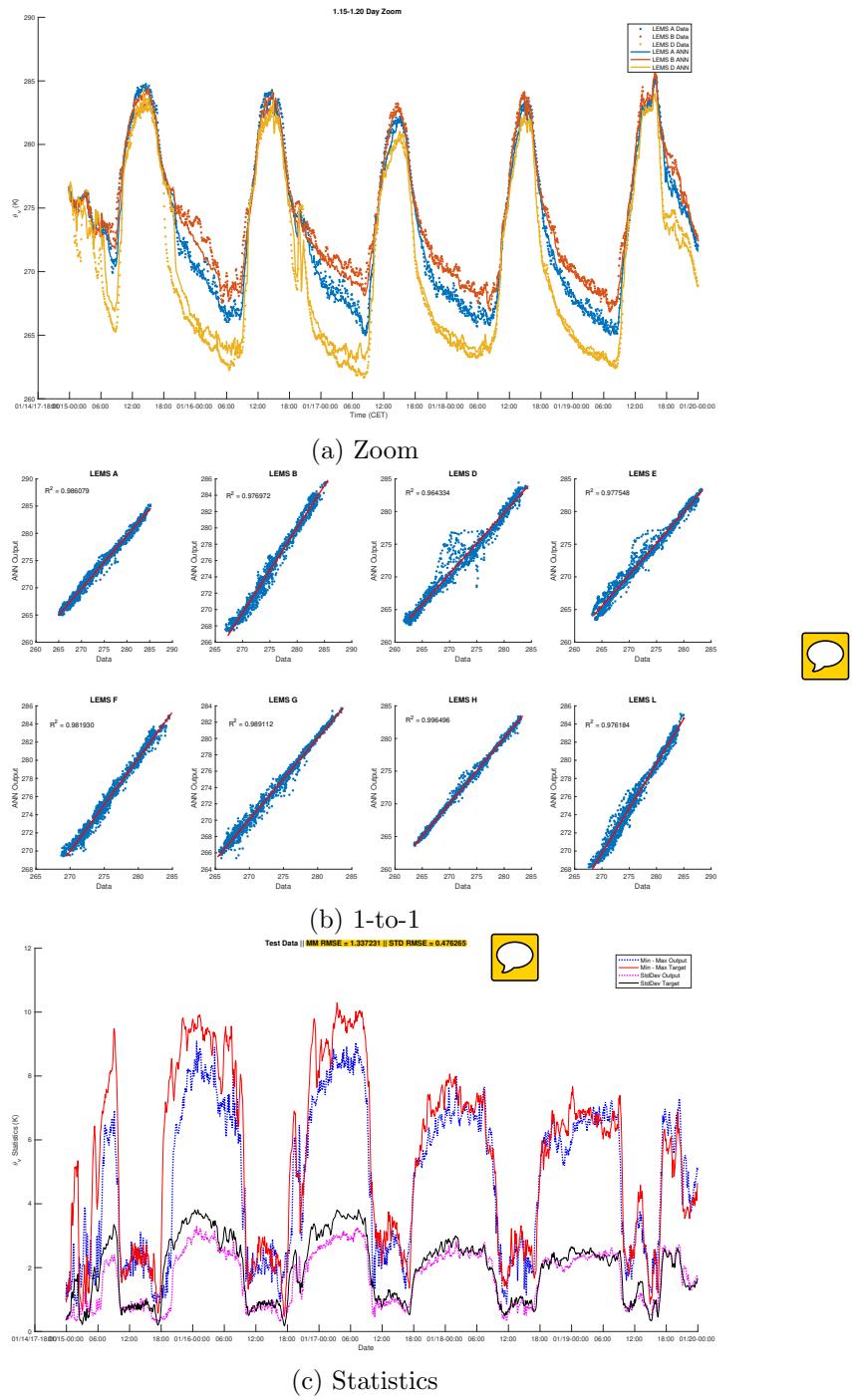


Figure 7: ANN prediction of virtual potential temperature 2017/1/15 - 2017/1/20

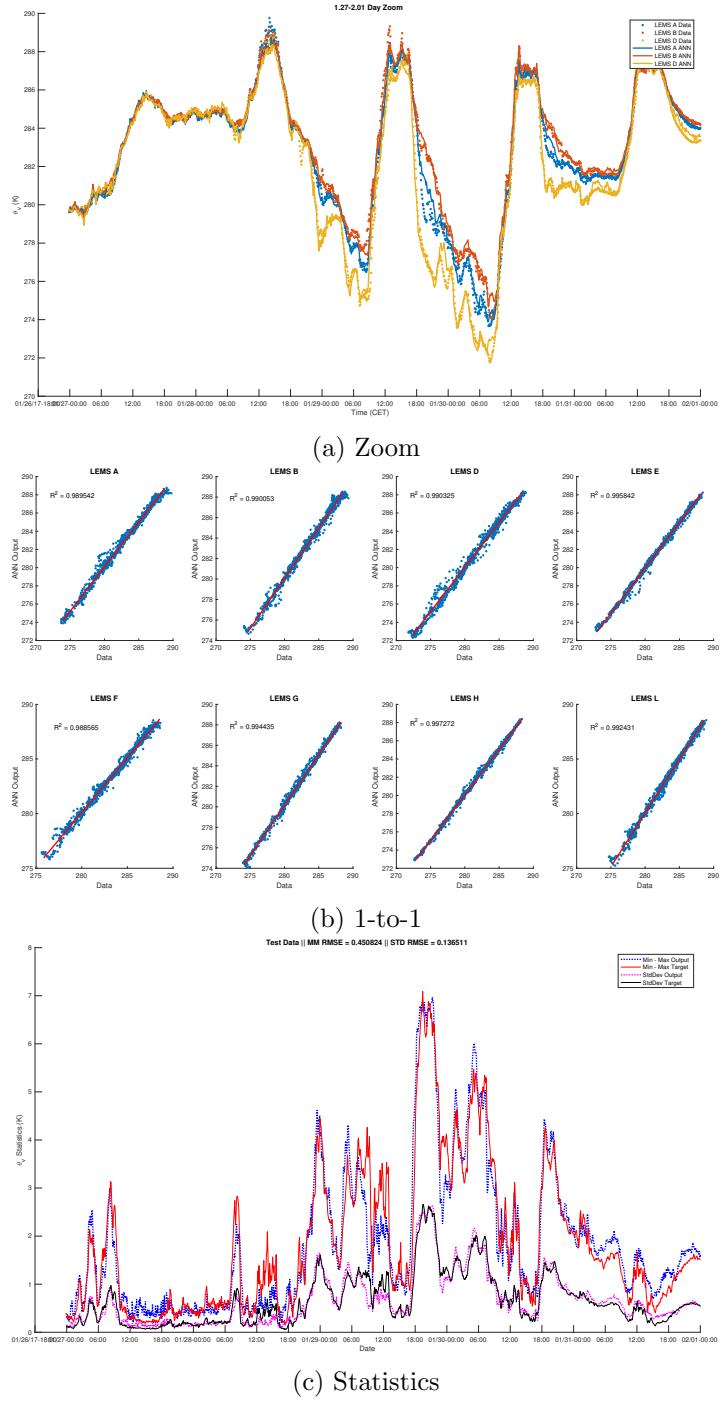
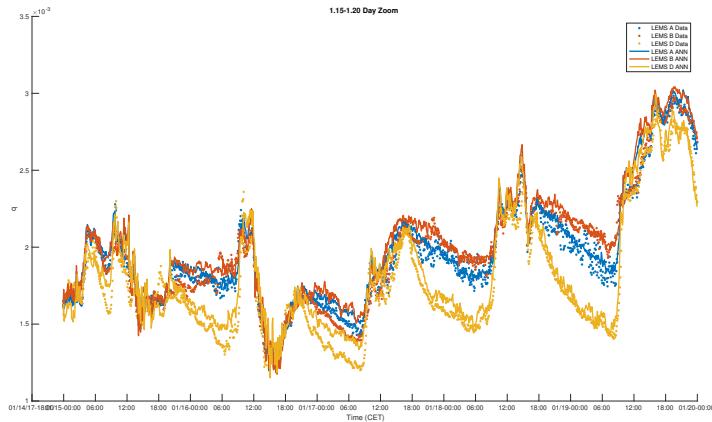
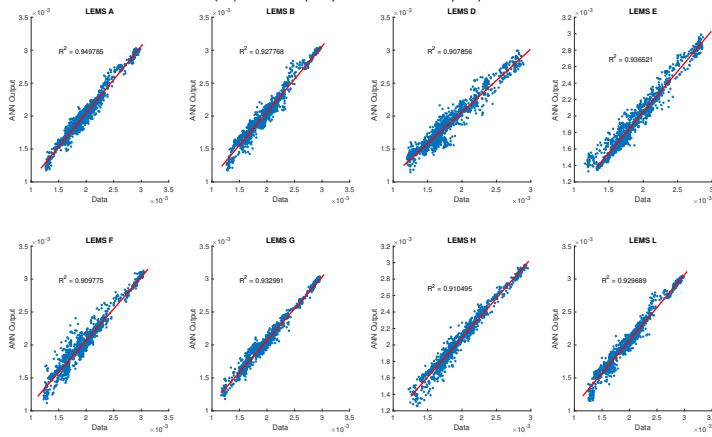


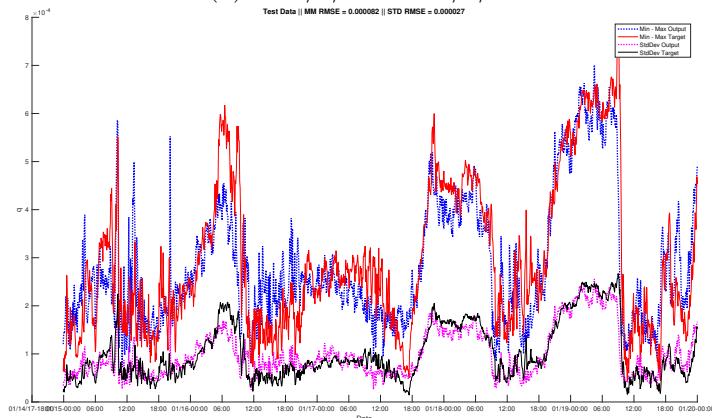
Figure 8: ANN prediction of virtual potential temperature 2017/1/27 - 2017/2/01



(a) 2017/1/15 - 2017/1/20



(b) 2017/1/15 - 2017/1/20



(c) 2017/1/15 - 2017/1/20

Figure 9: ANN prediction of specific humidity 2017/1/15 - 2017/1/20

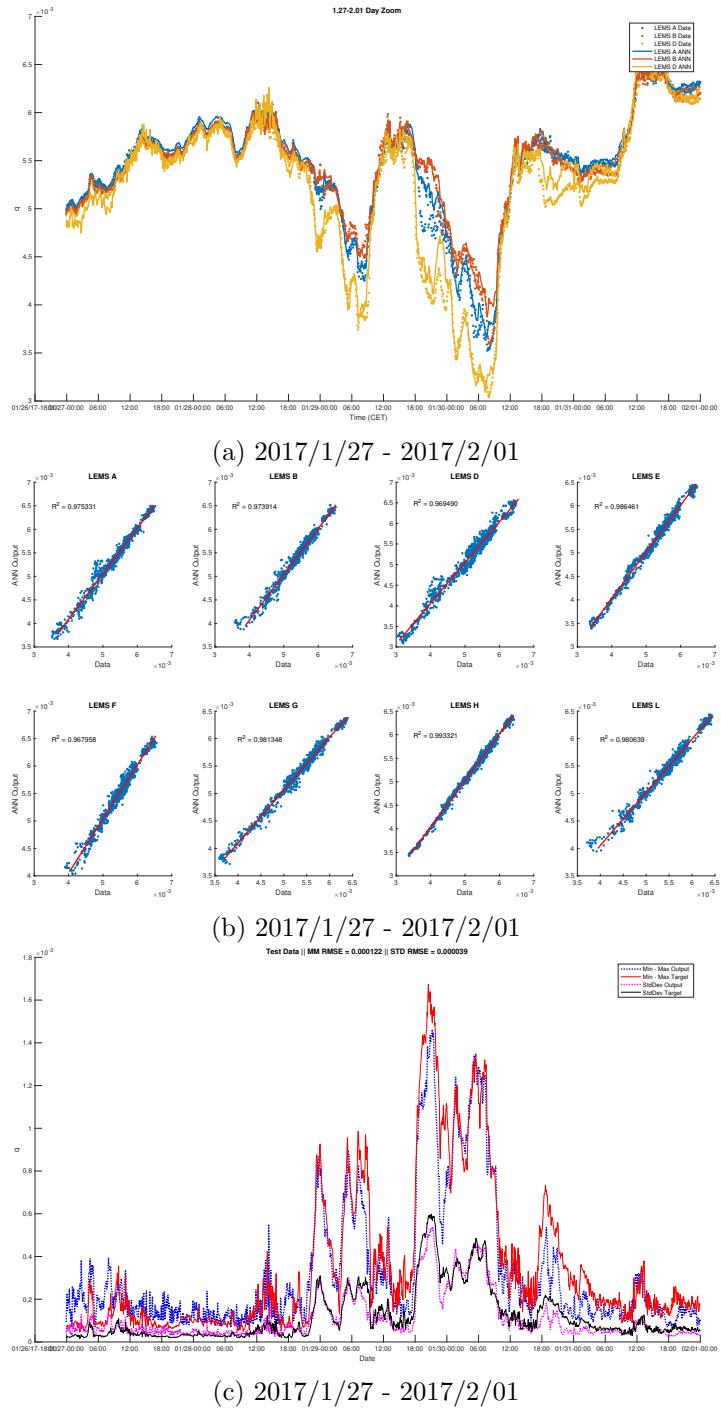


Figure 10: ANN prediction of specific humidity 2017/1/27 - 2017/2/01

15 January 2017 to 20 January 2017. The training data were from 16 December 2016 to 15 March 2017, excluding the testing data period. Five different neural networks, each with identical architecture but different random weights, were run and the outputs of the five averaged (A.K.A an ensemble average). The number of hidden nodes were varied from one to thirty. The root mean square error was taken between the network output and the actual data. The results are plotted in figure 11. This plot shows that increasing the number of hidden nodes beyond about 11 will likely not yield improved performance. It also shows that while a 1-node ANN performs worse than an 11-node ANN, it doesn't perform that much worse. In fact, this result is what motivated us to try using a linear regression to predict the wind field (more information in section 4.4). This test was repeated using the virtual potential temperature instead of the U component of the wind velocity as the output. The results can be seen in figure 12. This was done because the ANNs typically performed better on virtual potential temperature predictions than wind velocity predictions. As seen in figure 12, the performance difference in the number of nodes is still not very pronounced. In fact, for the virtual potential temperature, a single-node neural network performs better than a 21-node neural network.

Another hyperparameter that is relevant to this specific usage of ANNs is the number of output nodes. ANNs are not limited to one output; they can have several outputs for any number of inputs. To produce the results of section 4.1, eight different ANNs were trained with one output each. We repeated the tests from section 4.1 identically, except we trained a single ANN with eight outputs. We found no discernible or systematic difference between the results of the two approaches. Hence, we have not presented any plots from the multiple-output test. However, we did notice that the time to train one multiple-output neural network was longer than training eight separate single-output networks.



On a more practical note, a hyperparameter that is important when using MATLAB's Neural Network Toolbox is the train/test/validate data split percentage. When training the ANN, MATLAB internally splits the data into a training subset, a testing subset, and a validation subset. By default, MATLAB uses 70% of the original data as training data, 15% as validation data, and 15% as test data. The test data are only there so the user can view the performance of the neural network. The ANN performance on validation data, however, is often used by MATLAB as a training stopping condition. Therefore, in this instance, having a small test split won't affect us very much, as we have our own separate test data that the ANN has never "seen" before. But, having more training data will increase neural network performance, and having more validation data will improve generalization. This is why all ANNs presented in this paper have a train/test/validate split of 75%/20%/5%.

The last test we performed was not related to hyperparameters, but to the type of inputs. The results of 4.1 always used data from the same three LEMS as inputs: LEMS I, J, and K. These LEMS were chosen to be the inputs because their locations were spread across the measurement area (both horizontally and vertically) to captured many phenomena associated with thermal circulation in complex terrain (e.g., cold pools, slope/valley flows).

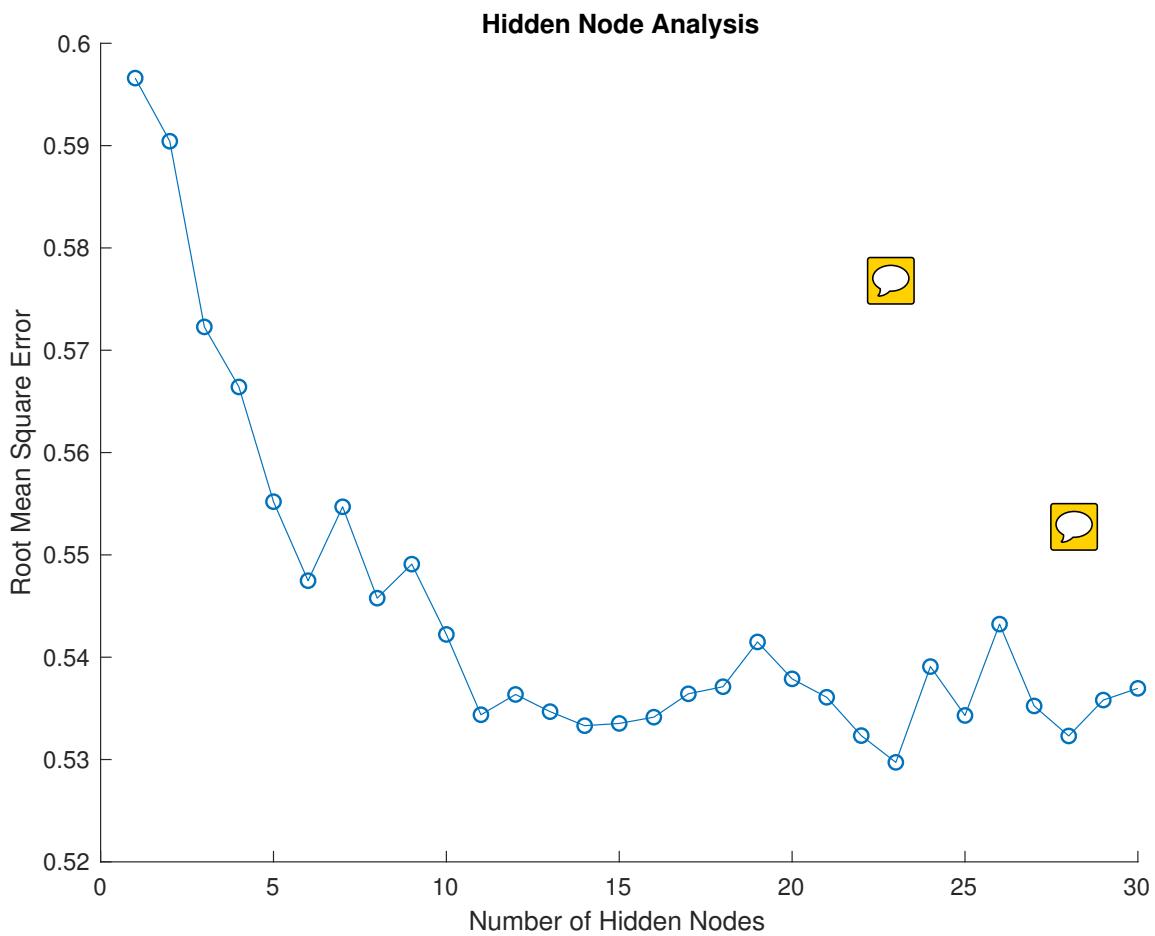


Figure 11: Hidden node analysis for U component of wind velocity

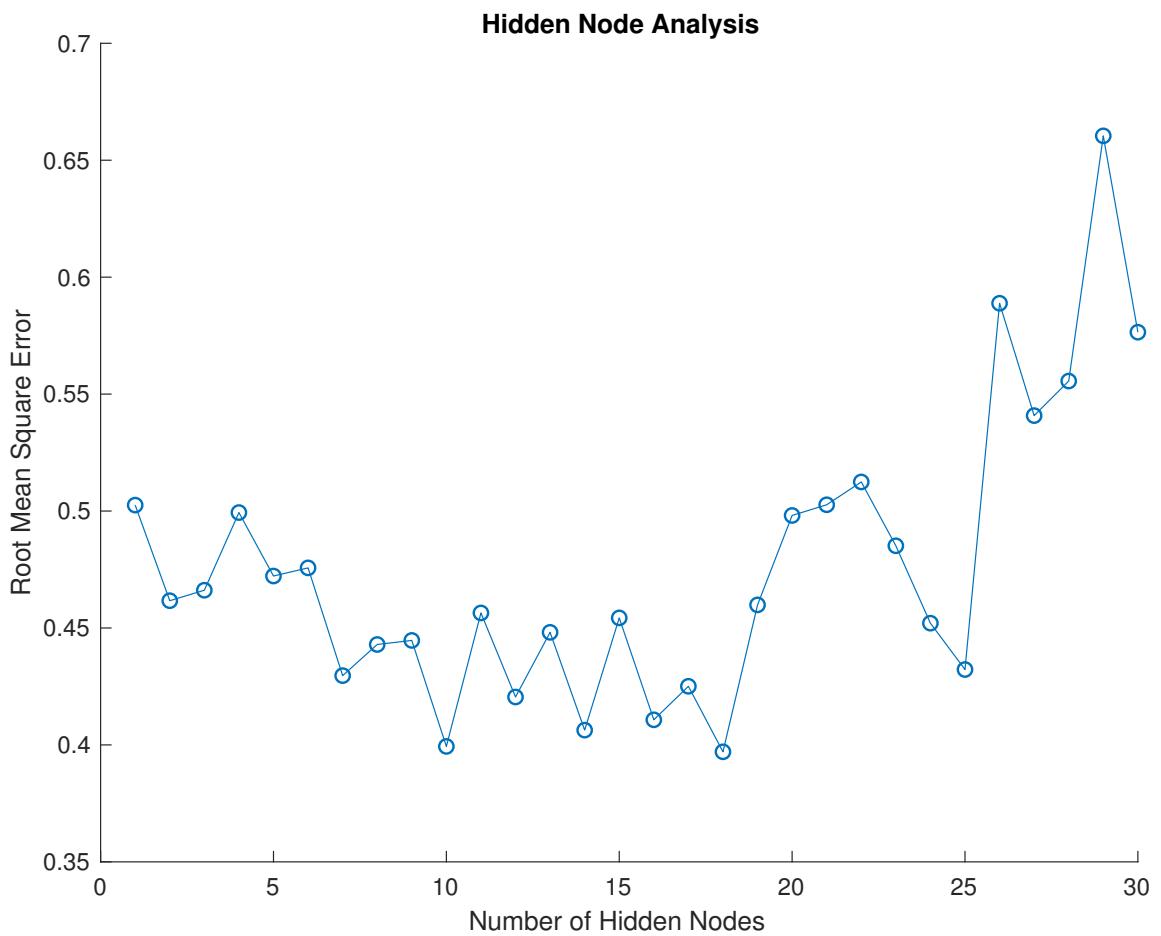


Figure 12: Hidden node analysis for virtual potential temperature

A priori, one might assume that the good performance exhibited by the ANNs is due to the locations of the input LEMS, and not because of the inherent power of the ANN. To test this, a combination analysis was performed, where every possible combination of three LEMS was used to predict values at the other nine LEMS. The inputs were identical to those from the tests from section 4.1. Specifically, the inputs were: wind velocity components, surface temperature, barometric pressure, and virtual potential temperature from three LEMS. The output was the virtual potential temperature of the other nine LEMS. Nine single-output ANNs were trained instead of a single nine-output ANN since the training time was shorter that way.

The testing data were taken from 15 January 2017 to 20 January 2017. The training data were from 12 January 2017 to 15 March 2017, excluding the testing data period. Ensemble averages were not used and the number of hidden nodes was 14. There are  $\frac{12!}{3!(12-3)!} = 220$  different combinations of input LEMS. Throughout this document, when we refer to the “combination number”, we mean a specific combination out of the 220 combinations. For example, combination “1” would have LEMS A, B, and C as input LEMS, and the rest as output LEMS. Since 9 different ANNs were trained for each combination, there are 9  $R^2$  values associated with each combination; each  $R^2$  value is computed with the difference between the ANN model prediction and the experimental data. We can also calculate the same statistics calculated in section 4.1, which are, the standard deviation of all the target variables, and the absolute difference between the minimum value target variable and the maximum value target variable, for a given time step. Using these statistics, we can calculate the RMSE between the statistics for the ANNs and the actual data, giving us a single statistic that quantifies the performance of any given combination of input LEMS.

Figure 13 shows the results of the combination analysis for prediction of the virtual potential temperature. As seen in subplot (a), most combinations have very low RMSE values between both statistics, but there are some combinations of input LEMS that have very high RMSE values between both statistics. Subplot (b) shows the mean  $R^2$  value for each combination, which correlates with the top subplot. Subplot (c) shows the median  $R^2$  value, which shows that all combinations perform quite well, and that there are outlier  $R^2$  values. We verified this manually. In every combination where the mean  $R^2$  value was below 0, we looked at the ANN response to the input test data. For every one, there was one run (out of nine) where the ANN exhibited a highly nonlinear and incorrect response, and the prediction did not match the measured data at all.

To ensure that this was not systematic, we re-ran the entire combination analysis multiple times using a different random seed. **Each time, the combination numbers that exhibited poor performance changed.** To show this, we performed the combination analysis three times with three different ANN seeds: MATLAB’s default seed, the shuffled seed (where the seed is based on the current time), and a constant (the number 14). Then, the combinations from each of these tests were sorted in order of smallest to greatest  $\text{abs}(\text{max-min})$  RMSE. This ranking is insightful because well-performing combinations will

	Default	14	Shuffle	
Default	1.000	0.026	-0.099	
14		1.000	0.026	
Shuffle			1.000	

Table 2: Table of Spearman’s rank correlation coefficient for ANN combination analysis.

consistently be at the top of the rankings. For example, if combination “3” (LEMS A, B, and E as inputs) consistently performed better than other combinations, it would be near the top of the rankings for each of the three different random seeds. To quantify the similarity between between the three different rankings, Spearman’s rank correlation coefficient was calculated for the rankings. Spearman’s rank correlation coefficient is the Pearson correlation of two different rankings, and is an efficient way to compare rankings [15]. Spearman’s rank correlation coefficient will be 1 when the rankings agree completely, -1 when the rankings are perfectly inverted, and 0 when there is no relationship between the rankings. Table 2 shows Spearman’s rank correlation coefficient for the three different seeds. It is evident that there is no relation between the rankings, implying that there are no combinations that consistently work better than others.

We conclude that for this experiment and data, an ANN can perform nowcasting well, regardless of the input LEMS. We also think that hyperparameter tuning is important, but not necessarily required for good performance. We hypothesize that this is because multiple linear regressions also work for this case, which is discussed in the next sections.

### 4.3 MLR Results

The tests run with ANNs (section 4.1) were repeated identically for MLRs. In the tests, the wind components, surface temperature, barometric pressure, and virtual potential temperature from LEMS I, J, and K were used to predict a given parameter from each of the other LEMS, excluding LEMS C (i.e., the parameters from LEMS A, B, D, E, F, G, H, and L were predicted). Since there were eight targets, eight different MLRs were trained, each with 15 explanatory variables (5 parameters each from 3 LEMS), and one target variable. Each figure represents a different target variable. Each figure shows a different target variable:

- Figures 14 and 15 show the MLR prediction of the U component of the wind velocity
- Figures 16 and 17 show the MLR prediction of the V component of the wind velocity
- Figures 18 and 19 show the MLR prediction of the virtual potential temperature
- Figures 20 and 21 show the MLR prediction of the specific humidity

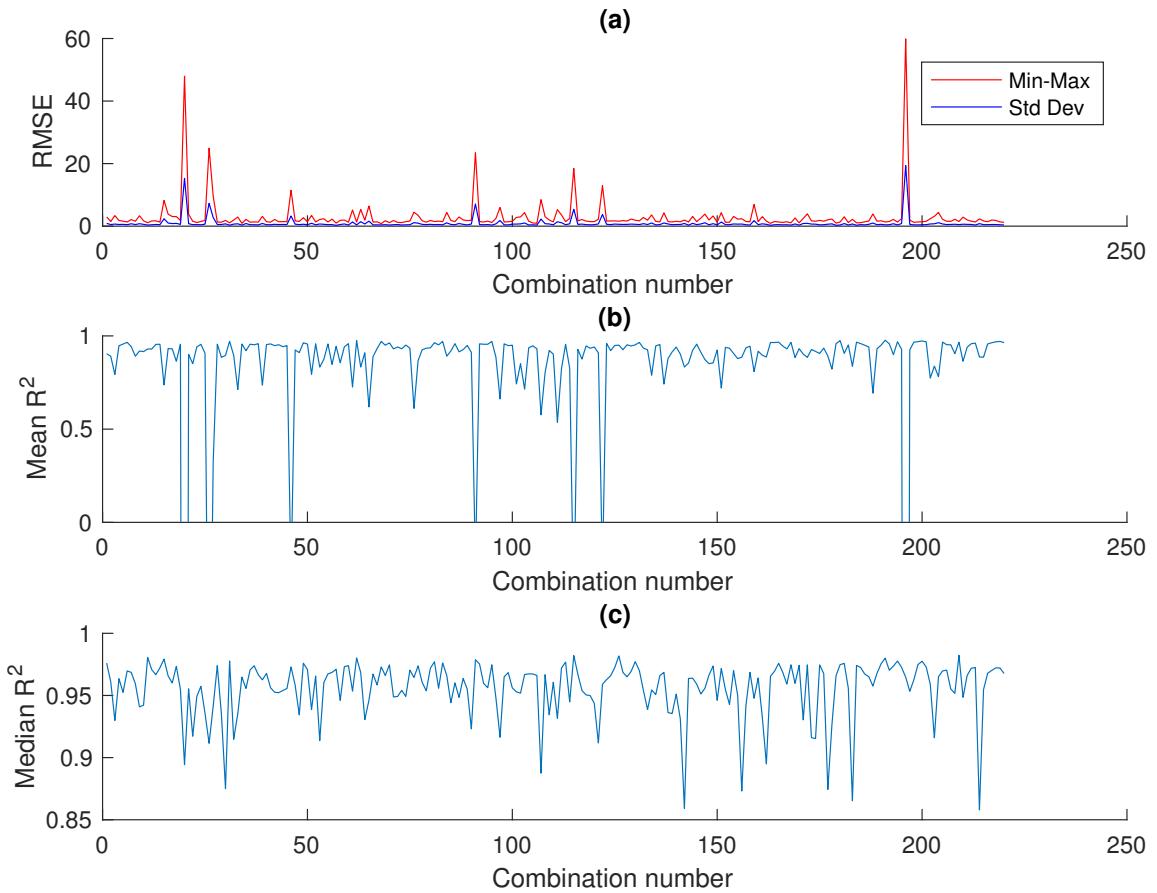


Figure 13: Example of combination analysis for the virtual potential temperature. For every combination of three input LEMS and nine output LEMS, three statistics were determined, which can be seen in the plot above. Some input combinations produce bad results, but that's only because one output (out of the nine) performed poorly. For this figure, the ANN weights were randomly initialized.

In each figure, there are 3 subfigures. The training data are identical for all runs: 5-minute averages of the data from 2016/12/16 to 2017/03/15, excluding the given run's test data. The following list gives a description of the subfigures:

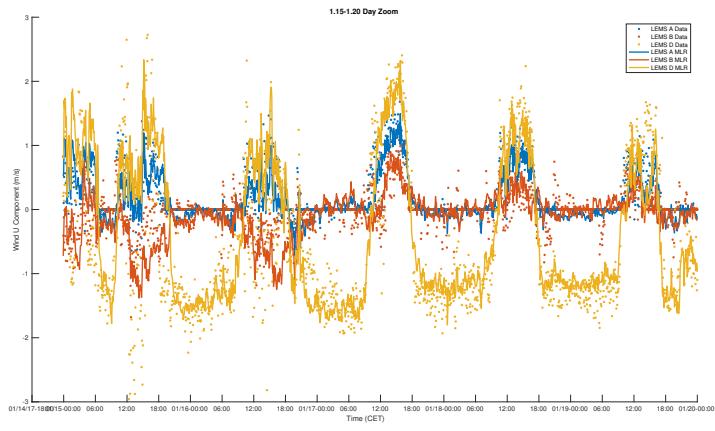
- Subfigure (a) shows the MLR prediction time series, along with the measured time series, for three different LEMS (A, B, and D). The dots are the actual data, and the solid lines are the MLR predictions.
- Subfigure (b) shows the 1-to-1 plots of the MLR predictions vs. the measured data. The solid line is a linear fit of the data, and the  $R^2$  value is displayed on each plot.
- Subfigure (c) shows statistics for the MLR prediction time series, along with statistics for the measured time series. Two statistics are shown. The first is the standard deviation of all eight target variables, for that time step. The second is the absolute difference between the minimum value target variable and the maximum value target variable, for that time step.

#### 4.4 MLR Discussion

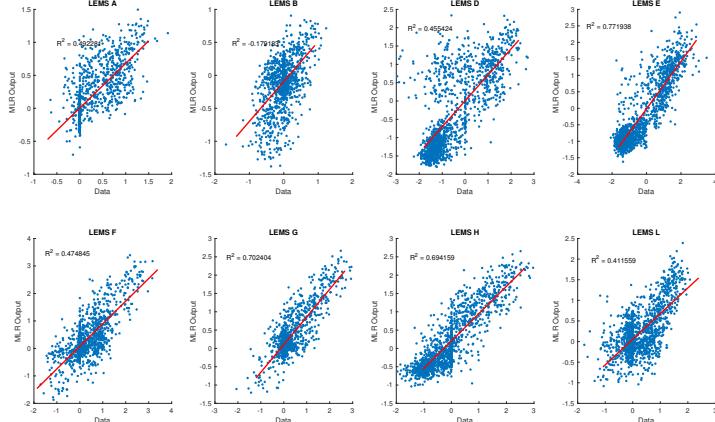
Since MLRs are much simpler than ANNs, there are no hyperparameters to tune. In addition, since MLRs do not have random initialization and iterations, they produce the same result every time, assuming the training data is identical. We originally didn't even consider even trying an MLR, because we assumed that the relationship between the measurements of the different LEMS was highly non-linear. However, when conducting the ANN hidden node analysis (see section 4.2), we noticed that a single-node ANN performed nearly as well as a multi-node ANN. Since a single-node ANN is essentially a multiple linear regression whose output is passed through a sigmoidal function, we decided to try an MLR as the nowcasting method. Surprisingly, as seen in 4.3, it worked well. When comparing figures 3 - 9 to figures 14 - 20, one can see that the ANN usually performs better than the MLR, but not always. In addition, the two models don't consistently perform better on one set of dates compared to others.

There are two main reasons to use an MLR instead of an ANN if possible. The first is computational runtime. Even on powerful computers, ANNs take much longer to train when compared to MLRs. The second is interpretability. ANNs tend to be black boxes that are difficult to interpret, whereas MLRs are transparent and easy to interpret. However, the biggest concern about using MLRs is the assumptions that need to be met. According to Poole et al. [57], there are six critical assumptions being made when using an MLR successfully. One of these assumptions is that the independent variables are linearly independent of each other. When they are not, collinearity exists, and the precision of the regression coefficient decreases [30]. During our analysis, we noticed that there were high Variance Inflation Factors (VIFs) for many of the coefficients, implying multicollinearity. This makes intuitive sense. For example, when the sun sets, all stations will measure

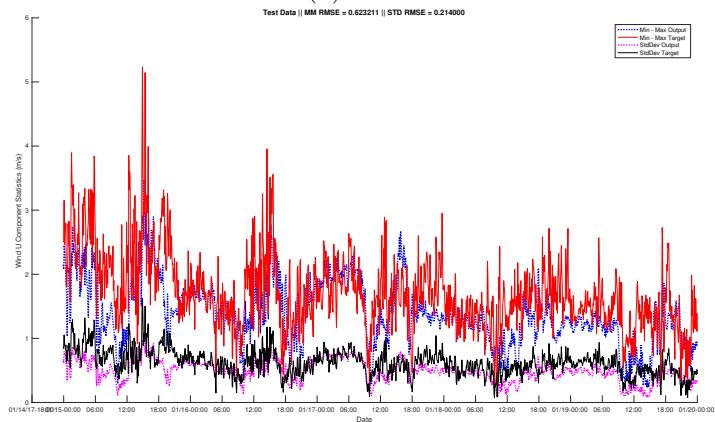




(a) Zoom

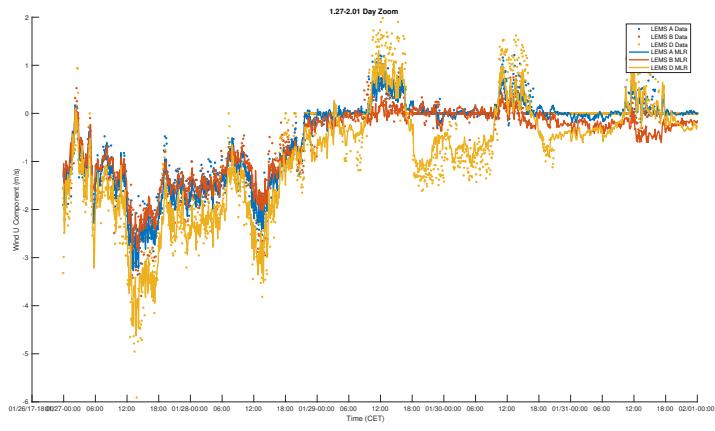


(b) 1-to-1

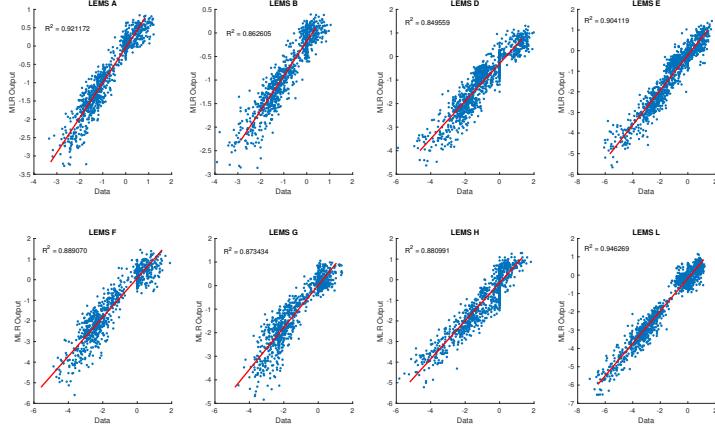


(c) Statistics

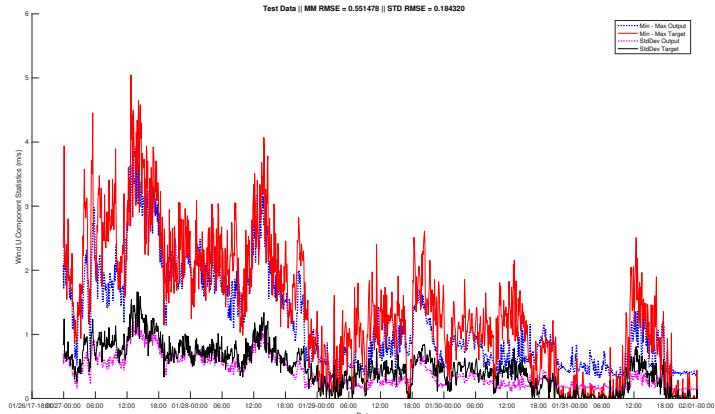
Figure 14: MLR prediction of U component of wind velocity 2017/1/15 - 2017/1/20



(a) Zoom

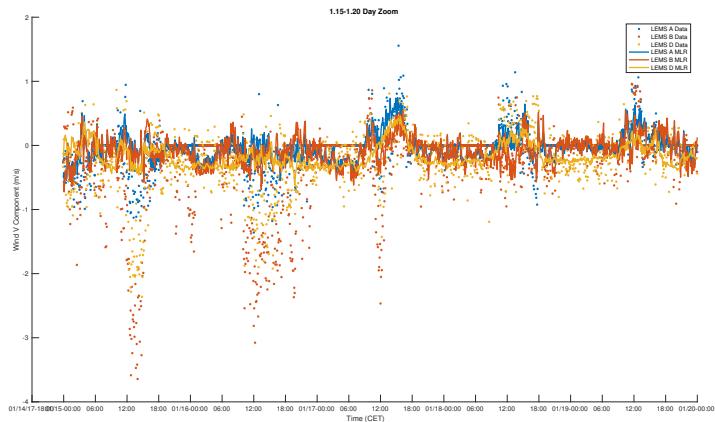


(b) 1-to-1

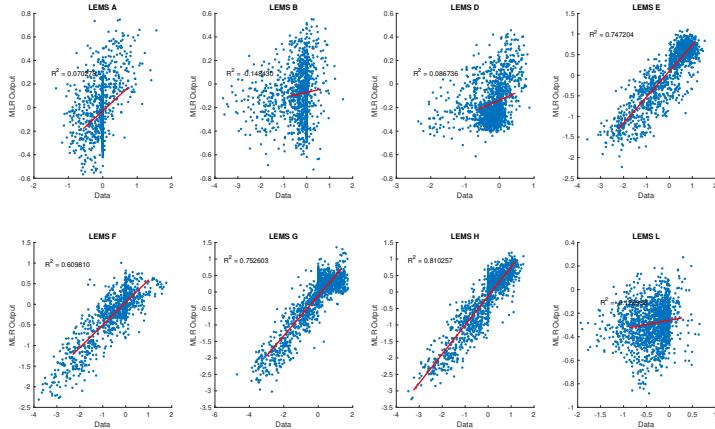


(c) Statistics

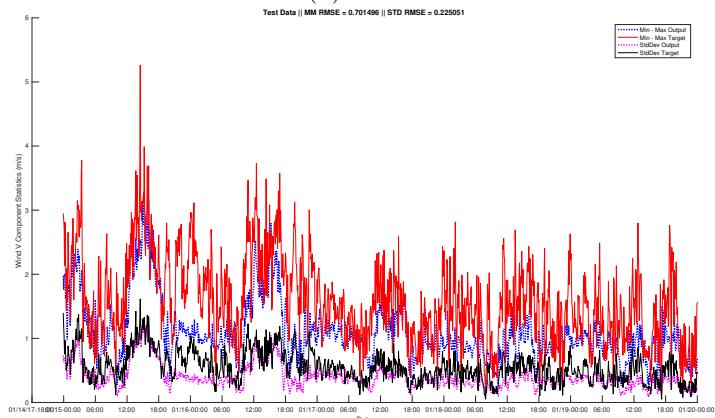
Figure 15: MLR prediction of U component of wind velocity 2017/1/27 - 2017/2/01



(a) Zoom

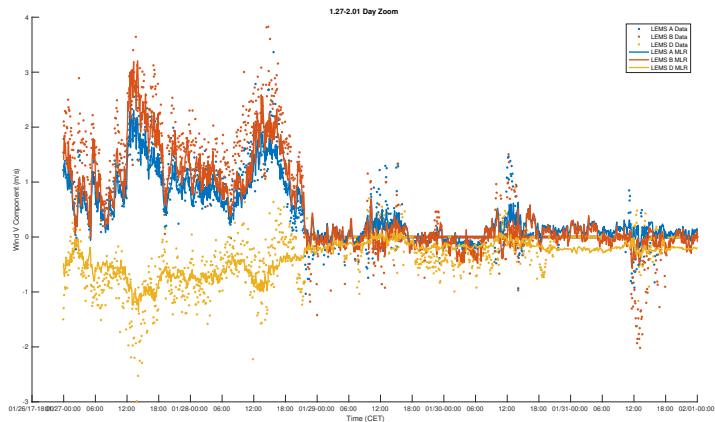


(b) 1-to-1

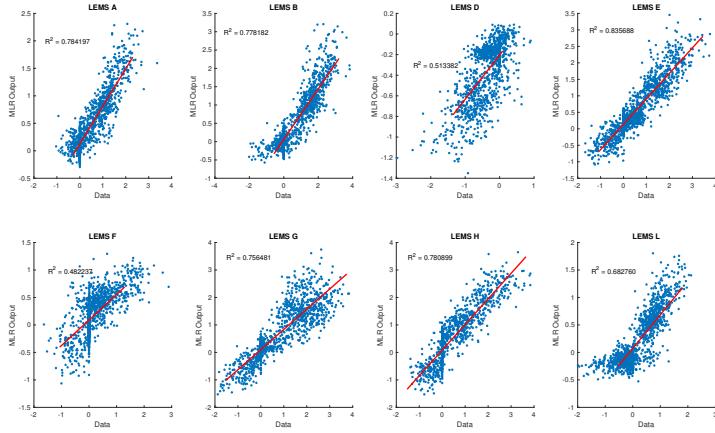


(c) Statistics

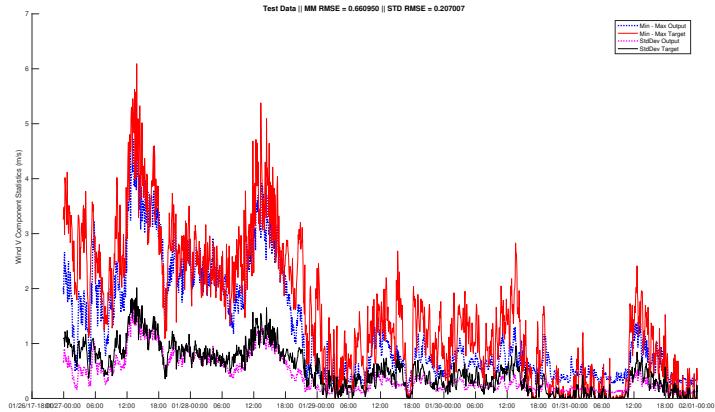
Figure 16: MLR prediction of V component of wind velocity 2017/1/15 - 2017/1/20



(a) Zoom



(b) 1-to-1



(c) Statistics

Figure 17: MLR prediction of V component of wind velocity 2017/1/27 - 2017/2/01

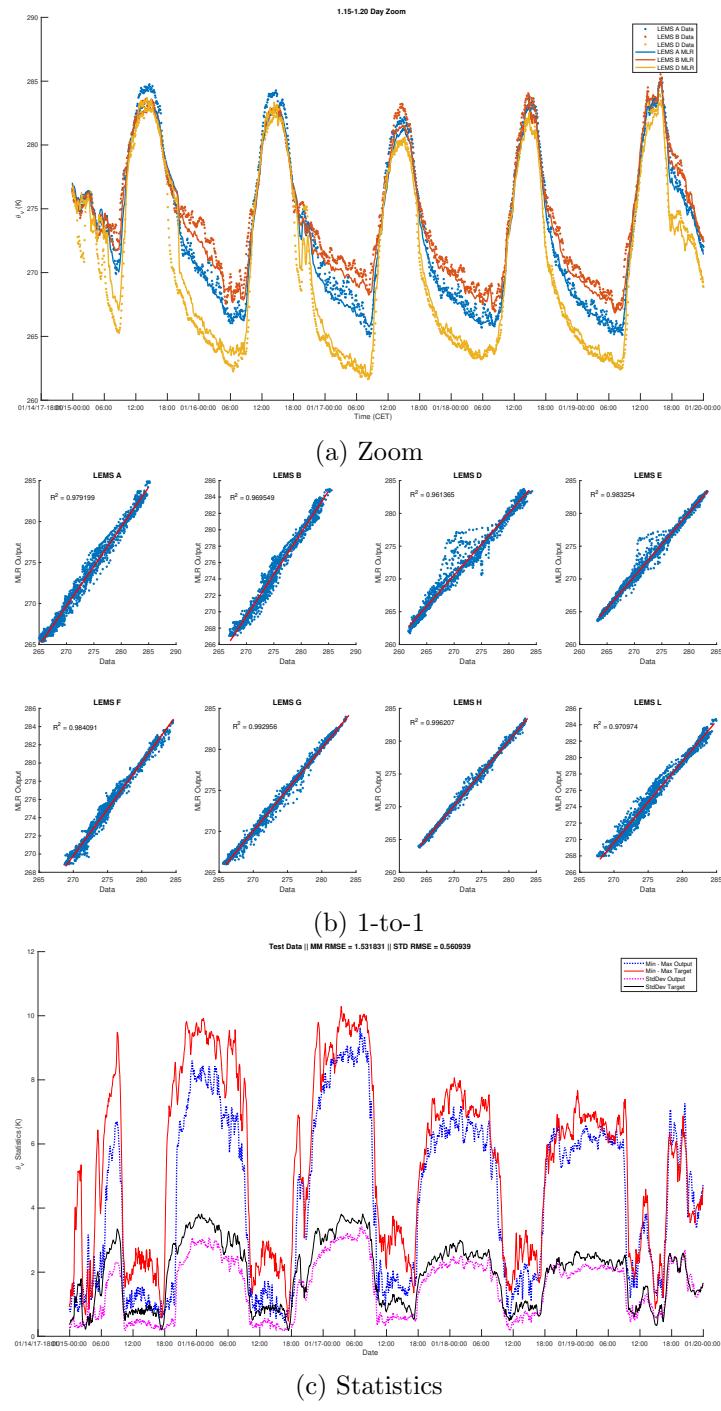


Figure 18: MLR prediction of virtual potential temperature 2017/1/15 - 2017/1/20

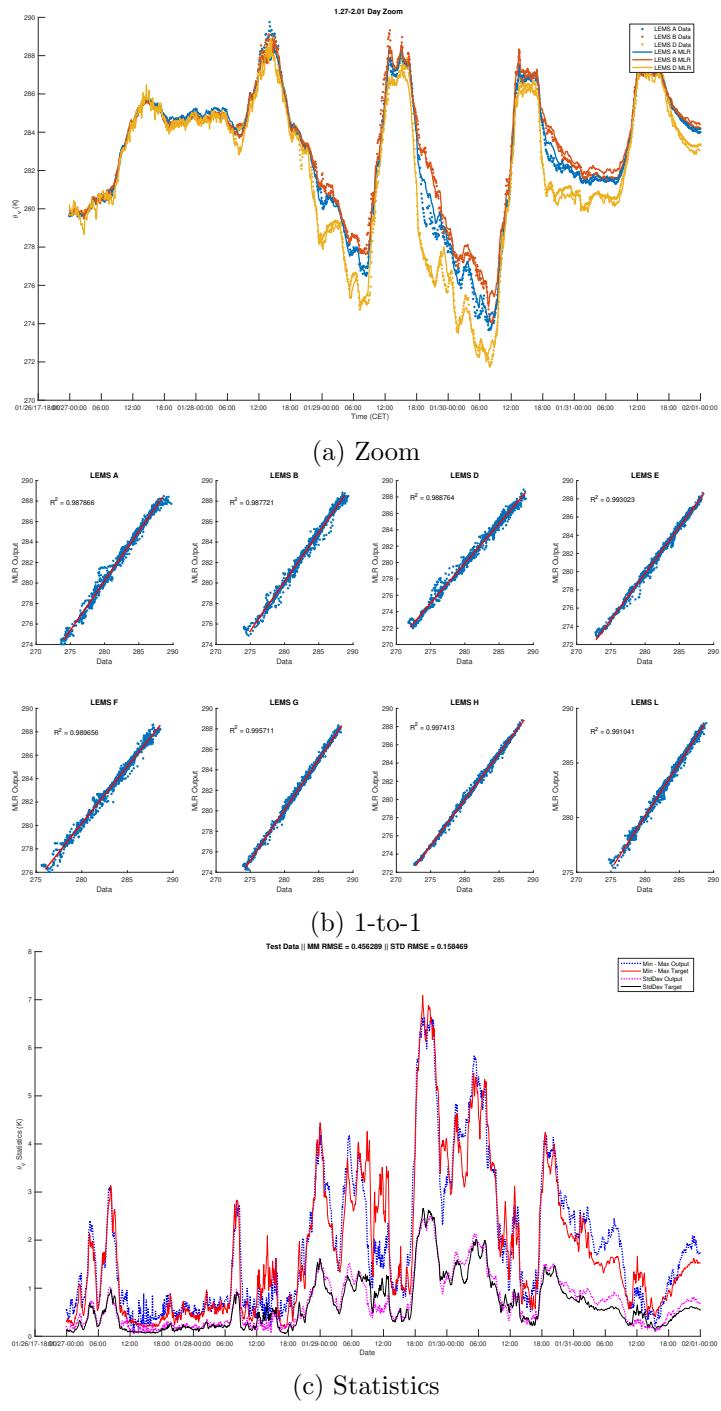
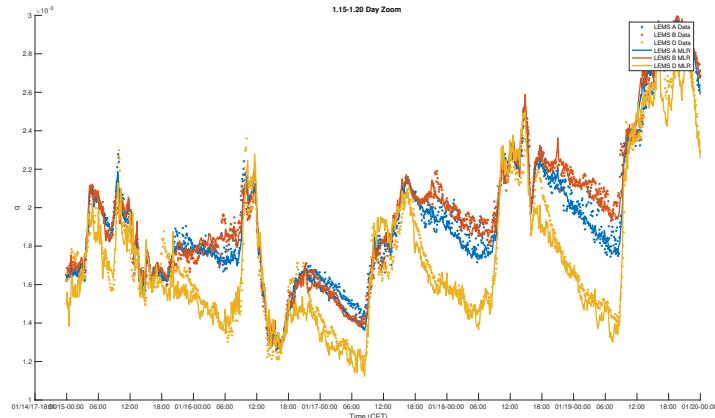
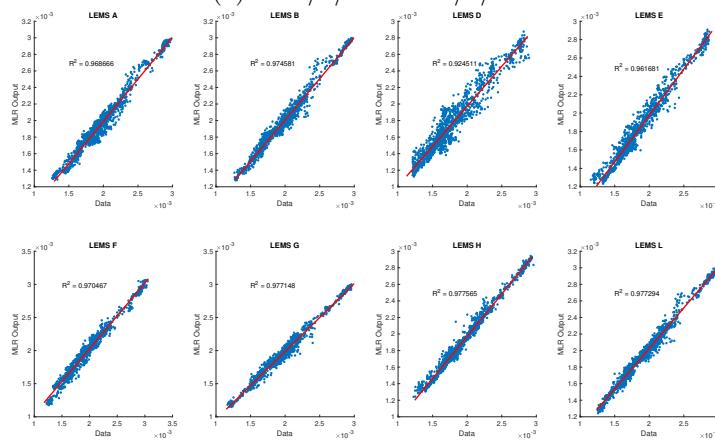


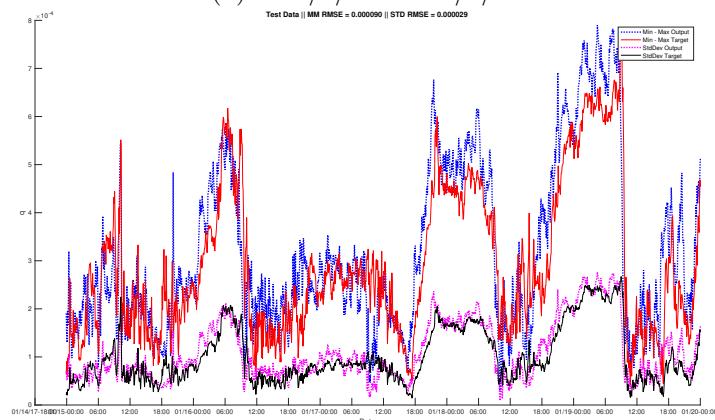
Figure 19: MLR prediction of virtual potential temperature 2017/1/27 - 2017/2/01



(a) 2017/1/15 - 2017/1/20

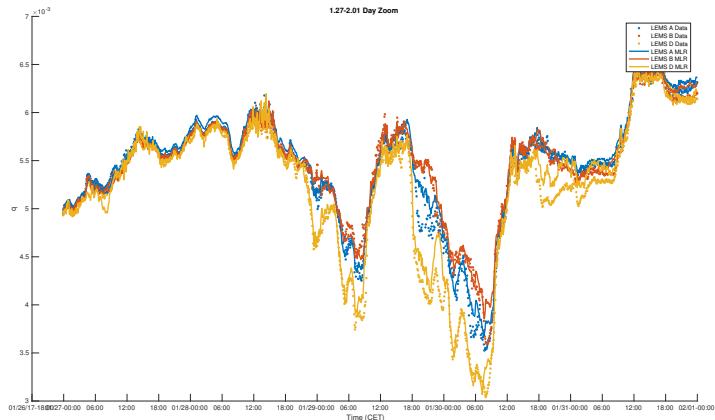


(b) 2017/1/15 - 2017/1/20

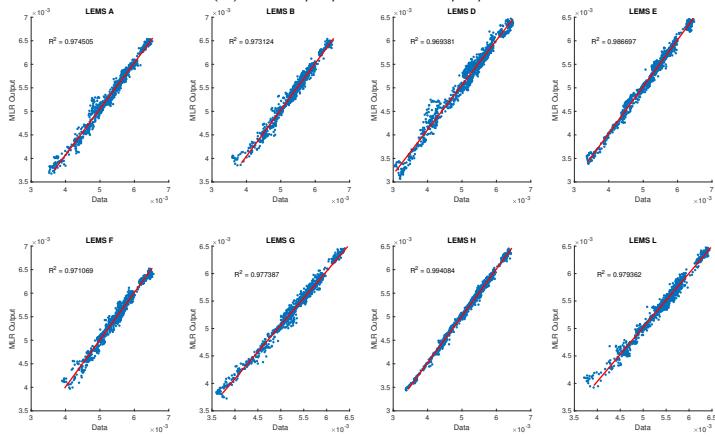


(c) 2017/1/15 - 2017/1/20

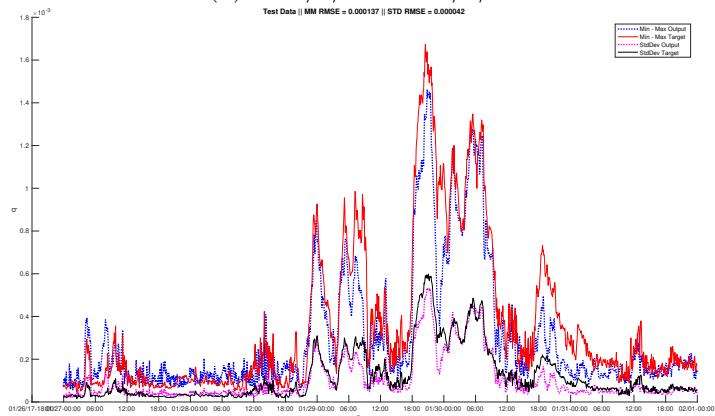
Figure 20: MLR prediction of specific humidity 2017/1/15 - 2017/1/20



(a) 2017/1/27 - 2017/2/01



(b) 2017/1/27 - 2017/2/01



(c) 2017/1/27 - 2017/2/01

Figure 21: MLR prediction of specific humidity 2017/1/27 - 2017/2/01

a temperature drop and the radiation measurements and the air temperature measurements will correlate. While we've shown that the MLR can nowcast successfully in this instance, the reduced accuracy of the regression coefficients due to multicollinearity means that interpreting the models must be done with care. To successfully interpret the models, one must reduce the number of explanatory variables until multicollinearity is minimized. Manually, this can be done by calculating correlation coefficients between all explanatory variables and removing highly correlated ones. Automatically, this can be done by running algorithms such as lasso regression, which automatically remove unneeded variables. While some preliminary work has been done with regards to this, it wasn't included in this paper as it felt out of the scope.

## 5 Dugway Proving Grounds Test



While the above results are promising, it is possible that these methods work in this location and these variables, but not in other situations. To test this, we used an MLR and an ANN to predict environmental variables collected from Dugway Proving Grounds, Utah.

In 2013, the Mountainous Terrain Atmospheric Modeling and Observations (MATER-HORN) experiment [20] was conducted at Dugway Proving Grounds, which is located in the Utah west desert. Dozens of instruments were deployed for the experiment, including meteorological towers, LEMS stations, LIDARs, SODARs, Ceilometers, tethered and untethered balloons, and various meteorological stations (SAMS, mini-SAMS, hobos, PWIDs). Out of the meteorological stations, the PWIDS were highly concentrated on an area called Granite Mountain, specifically on the east slope of Granite Mountain (see Figure 22). The data from these stations were used to test the ANNs and MLRs to ensure that the methods work in locations and equipment other than those in Cadarache, France.

Two tests were run, one with an ANN and one with an MLR. The input PWIDs were PWIDs 79, 90, and 93. The input variables were the relative humidity, air temperature, and wind vectors from these three PWIDS (twelve inputs total for each model). The output variable was the air temperature from PWIDs 33, 37, 72, 75, 78, 92, 96, and 108. While figure 22 shows more PWIDs than those listed here, they weren't used due to data availability issues. Similar to the tests conducted in Section 4.1 and 4.3, eight different ANNs and MLRs were trained with one output. The total dataset consists of the data starting 15 September 2012 and ending at 30 October 2012. The testing data consists of the data from 18 October 2012 to 21 October 2012, and the training data is the rest of the dataset. Note that the performance of each model is still good, even though there is less training data.

As seen in figure 23, both the ANN and MLR models work well. This is despite the different location, different variables and stations, and with different amounts of training data.



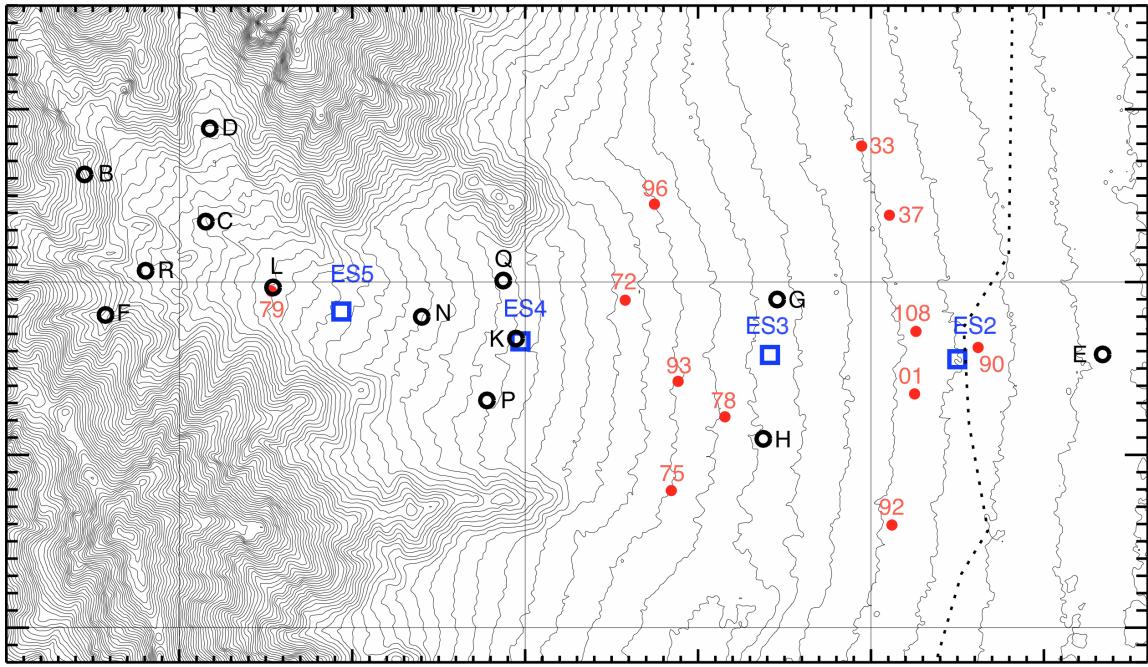
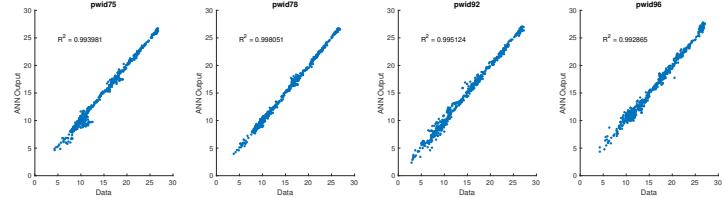
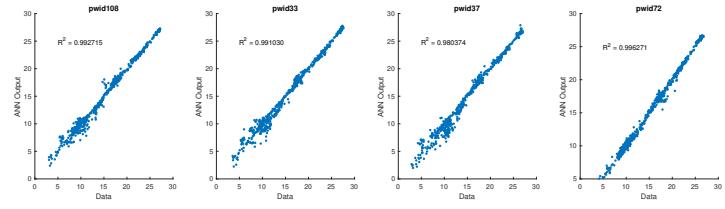
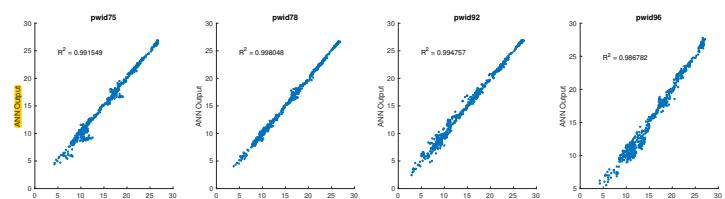
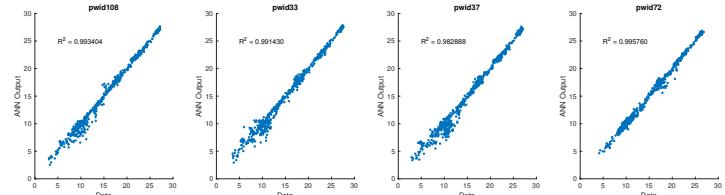


Figure 22: Topographical map of the east slope of Granite Mountain. The PWIDs are marked by red dots. The PWIDS used for the ANN and MLR tests were 33, 37, 72, 75, 78, 79, 90, 92, 93, 96, and 108.



(a) ANN 1-to-1



(b) MLR 1-to-1

Figure 23: ANN and MLR prediction of air temperature at Dugway Proving Grounds, Utah. This demonstrates that this method works in different locations, with different variables, and with different amounts of training data.

## 6 Summary



It is shown that artificial neural networks and multiple linear regression can be used to nowcast environmental measurements in small-scale complex terrain. This is demonstrated using measurements taken from open-source weather stations deployed in Cadarache, France, and from research-grade weather stations deployed at Dugway Proving Grounds, Utah. Out of all the variables predicted (wind velocity components, virtual potential temperature, specific humidity, air temperature), the most difficult variables to predict were the wind velocity components. The ANNs perform about as well as the MLR does for all variables, though the ANNs will sometimes randomly perform very poorly. While the MLR is easier to interpret than the ANN, multicollinearity means one should be cautious when doing so. This work has the potential to perform data-filling for incomplete data in the future.

## References

- [1] Otávio C. Acevedo and David R. Fitzjarrald. The Early Evening Surface-Layer Transition: Temporal and Spatial Variability. *Journal of the Atmospheric Sciences*, 58(17):2650–2667, 2001. URL: [http://journals.ametsoc.org/doi/abs/10.1175/1520-0469\(2001\)058<2650:TEESLT>2.0.CO;2](http://journals.ametsoc.org/doi/abs/10.1175/1520-0469(2001)058<2650:TEESLT>2.0.CO;2). doi:10.1175/1520-0469(2001)058<2650:TEESLT>2.0.CO;2.
- [2] R Adams, L Houston, and R Weiher. The value of snow and snow information services. *Report prepared for NOAA's National Operational Hydrological Remote Sensing Center, Chanhassen, MN, under contract DG1330-03-SE-1097*, 2004.
- [3] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.
- [4] Halit Apaydin, F. Kemal Sonmez, and Y. Ersoy Yildirim. Spatial interpolation techniques for climate data in the GAP region in Turkey. *Climate Research*, 28(1):31–40, 2004. doi:10.3354/cr028031.
- [5] Eric Asa. Nonlinear Spatial Characterization and Interpolation of Wind Data. *Wind Engineering*, 36(3):251–272, 2012. doi:10.1260/0309-524X.36.3.251.
- [6] Brian N Bailey, Rob Stoll, Eric R Pardyjak, and Walter F Mahaffee. Effect of vegetative canopy architecture on vertical transport of massless particles. *Atmospheric environment*, 95:480–489, 2014.
- [7] Marta Banbura, Domenico Giannone, Michele Modugno, and Lucrezia Reichlin. Nowcasting and the real-time data flow. 2013.
- [8] Ronald L Baskett, John S Nasstrom, and Rolf Lange. Emergency Response Model Evaluation Using Diablo Canyon Nuclear Power Plant Tracer Experiments. In *Air Pollution Modeling and Its Application VIII*, pages 603–604. Springer, 1991.
- [9] Mark Hudson Beale, Martin T. Hagan, and Howard B. Demuth. MATLAB Neural Network Toolbox User's Guide. 2017.
- [10] F Benvenuto and A Marani. Neural Networks for Environmental Problems: Data Quality Control and Air Pollution Nowcasting. 2(3):281–292, 2000.
- [11] Fotini Katopodes Chow. *Mountain Weather Research and Forecasting: Recent Progress and Current Challenges*. 2013. doi:10.1007/978-94-007-4098-3.
- [12] Marc Claesen and Bart De Moor. Hyperparameter Search in Machine Learning. *CoRR*, abs/1502.0, 2015. URL: <http://arxiv.org/abs/1502.02127>, arXiv:1502.02127.

- [13] Siebren De Haan and Ad Stoffelen. Assimilation of high-resolution Mode-S wind and temperature observations in a regional NWP model for nowcasting applications. *Weather and Forecasting*, 27(4):918–937, 2012.
- [14] Nicholas W S Demetriades and R L Holle. Long range lightning nowcasting applications for tropical cyclones. In *Preprints, Conf. Meteorology Application of Lightning Data, Atlanta*, pages 353–365, 2006.
- [15] Allen B. Downey. *Think Stats*. Green Tea Press, 2011.
- [16] Gert-Jan Duine, Thierry Hedde, Pierre Roubin, and Pierre Durand. A Simple Method Based on Routine Observations to Nowcast Down-Valley Flows in Shallow, Narrow Valleys. *Journal of Applied Meteorology and Climatology*, 55(7):1497–1511, 2016. doi: 10.1175/JAMC-D-15-0274.1.
- [17] Florian Dupuy, Gert-Jan Duine, Pierre Durand, Thierry Hedde, Pierre Roubin, and Eric R. Pardyjak. Valley-winds at the local scale: A downscaling method based on an artificial neural network applied to routine weather forecasting. *Journal of Applied Meteorology and Climatology*, 2017.
- [18] Stefan Emeis. *Measurement methods in atmospheric sciences: in situ and remote*. Gebr. Borntraeger Science Publishers, 2010.
- [19] H J S Fernando. Fluid dynamics of urban atmospheres in complex terrain. *Annual review of fluid mechanics*, 42:365–389, 2010.
- [20] H J S Fernando, E R Pardyjak, S Di Sabatino, F K Chow, S F J De Wekker, S W Hoch, J Hacker, J C Pace, T Pratt, Z Pu, and Others. The MATERHORN-unraveling the intricacies of mountain weather. *Bulletin of the American Meteorological Society*, (2015), 2015.
- [21] Mark N. French, Witold F. Krajewski, and Robert R. Cuykendall. Rainfall forecasting in space and time using a neural network. *Journal of Hydrology*, 137(1-4):1–31, 1992. doi:10.1016/0022-1694(92)90046-X.
- [22] Carol J. Friedland, T. Andrew Joyner, Carol Massarra, Robert V. Rohli, Anna M. Treviño, Shubharoop Ghosh, Charles Huyck, and Mark Weatherhead. Isotropic and anisotropic kriging approaches for interpolating surface-level wind speeds across large, geographically diverse regions. *Geomatics, Natural Hazards and Risk*, 5705(August 2017):1–18, 2016. URL: <http://dx.doi.org/10.1080/19475705.2016.1185749>, doi:10.1080/19475705.2016.1185749.
- [23] Gustavo Furquim, Filipe Neto, Gustavo Pessin, Jo Ueyama, Joao P. De Albuquerque, Maria Clara, Eduardo M. Mendiondo, Vladimir C.B. De Souza, Paulo De Souza, De-

- sislava Dimitrova, and Torsten Braun. Combining wireless sensor networks and machine learning for flash flood nowcasting. *Proceedings - 2014 IEEE 28th International Conference on Advanced Information Networking and Applications Workshops, IEEE WAINA 2014*, (Section V):67–72, 2014. doi:10.1109/WAINA.2014.21.
- [24] Ismail Gultepe, Thomas Kuhn, M Pavolonis, C Calvert, J Gurka, Andrew J Heymsfield, P S K Liu, B Zhou, R Ware, B Ferrier, and Others. Ice fog in Arctic during FRAM–Ice Fog Project: Aviation and nowcasting applications. *Bulletin of the American Meteorological Society*, 95(2):211–226, 2014.
  - [25] Nipun Gunawardena, Eric Pardyjak, Rob Stoll, and Anup Khadka. Development and evaluation of an open-source low-cost distributed sensor network for environmental monitoring applications. *Measurement Science and Technology*, 2017.
  - [26] Lei Han, Juanzhen Sun, Wei Zhang, Yuanyuan Xiu, Hailei Feng, and Yinjing Lin. A machine learning nowcasting method based on real-time reanalysis data. *Journal of Geophysical Research: Atmospheres*, 122(7):4038–4051, 2017.
  - [27] C Hang, D F Nadeau, I Gultepe, S W Hoch, C Román-Cascón, K Pryor, H J S Fernando, E D Creegan, L S Leo, Z Silver, and E R Pardyjak. A Case Study of the Mechanisms Modulating the Evolution of Valley Fog. *Pure and Applied Geophysics*, 173(9):3011–3030, 2016. doi:10.1007/s00024-016-1370-4.
  - [28] Q J Hart, M Brugnach, B Temesgen, C Rueda, S L Ustin, and K Frame. Daily reference evapotranspiration for California using satellite imagery and weather station measurement interpolation. *Civil Engineering and Environmental Systems*, 26(1):19–33, 2009.
  - [29] Jeff Heaton. *Introduction to neural networks with Java*. Heaton Research, Inc., 2nd editio edition, 2008.
  - [30] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.
  - [31] Marilena Kampa and Elias Castanas. Human health effects of air pollution. *Environmental pollution*, 151(2):362–367, 2008.
  - [32] B R Karthikeya, Prabal S Negi, and N Srikanth. Wind resource assessment for urban renewable energy application in Singapore. *Renewable Energy*, 87:403–414, 2016.
  - [33] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steven Glaser, and Martin Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *Information processing in sensor networks, 2007. IPSN 2007. 6th international symposium on*, pages 254–263. IEEE, 2007.

- [34] Leif Kristensen. Cup anemometer behavior in turbulent environments. *Journal of Atmospheric and Oceanic Technology*, 15(1):5–17, 1998.
- [35] Anders Krogh, Anders Krogh, Jesper Vedelsby, and Jesper Vedelsby. Neural Network Ensembles, Cross Validation, and Active Learning. *Nips*, pages 231–238, 1995.
- [36] Neil P. Lareau, Erik Crosman, C. David Whiteman, John D. Horel, Sebastian W. Hoch, William O. J. Brown, Thomas W. Horst, Neil P. Lareau, Erik Crosman, C. David Whiteman, John D. Horel, Sebastian W. Hoch, William O. J. Brown, and Thomas W. Horst. The Persistent Cold-Air Pool Study. *Bulletin of the American Meteorological Society*, 94(1):51–63, jan 2013. URL: <http://journals.ametsoc.org/doi/abs/10.1175/BAMS-D-11-00255.1>, doi:10.1175/BAMS-D-11-00255.1.
- [37] M Lateb, R N Meroney, M Yataghene, H Fellouah, F Saleh, and M C Boufadel. On the use of numerical modelling for near-field pollutant dispersion in urban environments- A review. *Environmental Pollution*, 208:271–283, 2016.
- [38] Margaret a. LeMone, Kyoko Ikeda, Robert L Grossman, and Mathias W Rotach. Horizontal Variability of 2-m Temperature at Night during CASES-97. *Journal of the Atmospheric Sciences*, 60(20):2431–2449, 2003. URL: [http://cobserv.cpt.ac.jp/~rotach/cases97/jas60\\_2431.pdf](http://cobserv.cpt.ac.jp/~rotach/cases97/jas60_2431.pdf), doi:10.1175/1520-0469(2003)060<2431:HVOMTA>2.0.CO;2.
- [39] Brenda B Lin. Agroforestry management as an adaptive strategy against potential microclimate extremes in coffee agriculture. *Agricultural and Forest Meteorology*, 144(1-2):85–94, 2007.
- [40] Marie Lothon, Fabienne Lohou, David Pino, Fleur Couvreux, E R Pardyjak, Joachim Reuder, Jordi de Arellano, Pierre Durand, O Hartogensis, Dominique Legain, and Others. The BLLAST field experiment: Boundary-layer late afternoon and sunset turbulence. *Atmospheric chemistry and physics*, 14(20):10931–10960, 2014.
- [41] W Luo, M C Taylor, and S R Parker. A comparison of spatial interpolation methods to estimate continuous wind speed surfaces using irregularly distributed data from England and Wales. *International journal of climatology*, 28(7):947–959, 2008.
- [42] Walter F Mahaffee and Rob Stoll. The ebb and flow of airborne pathogens: Monitoring and use in disease management decisions. *Phytopathology*, 106(5):420–431, 2016.
- [43] L. Mahrt. Stably Stratified Flow in a Shallow Valley. *Boundary-Layer Meteorology*, 162(1):1–20, 2017. doi:10.1007/s10546-016-0191-4.
- [44] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM*

*international workshop on Wireless sensor networks and applications*, pages 88–97. ACM, 2002.

- [45] Clifford Mass. Nowcasting: The promise of new technologies of communication, modeling, and observation. *Bulletin of the American Meteorological Society*, 93(6):797–809, 2012.
- [46] MathWorks. Levenberg-Marquardt backpropagation - MATLAB trainlm, 2006. URL: <https://www.mathworks.com/help/nnet/ref/trainlm.html>.
- [47] MathWorks. MATLAB Statistics and Machine Learning Toolbox User’s Guide. 2016.
- [48] Nathan E Miller, Rob Stoll, Walter Mahafee, Tara Neill, and Eric R Pardyjak. Field-scale particle transport in a trellised agricultural canopy during periods of row-aligned winds. (June):2016, 2016.
- [49] J L Monteith, C K Ong, and J E Corlett. Microclimatic interactions in agroforestry systems. *Forest Ecology and management*, 45(1-4):31–44, 1991.
- [50] Petr Novak. The Czech Hydrometeorological Institute’s severe storm nowcasting system. *Atmospheric research*, 83(2):450–457, 2007.
- [51] Genevieve B Orr and Klaus-Robert Müller. *Neural networks: tricks of the trade*. Springer, 2003.
- [52] Michael a. Osborne, Stephen J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings. Towards Real-Time Information Processing of Sensor Network Data Using Computationally Efficient Multi-output Gaussian Processes. *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pages 109–120, 2008. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4505467> <http://ieeexplore.ieee.org/document/4505467/>, doi:10.1109/IPSN.2008.25.
- [53] Ahmet Öztopal. Artificial neural network approach to spatial estimation of wind velocity data. *Energy Conversion and Management*, 47(4):395–406, 2006. doi:10.1016/j.enconman.2005.05.009.
- [54] E. R. Pardyjak, S. O. Speckart, F. Yin, and J. M. Veranth. Near source deposition of vehicle generated fugitive dust on vegetation and buildings: Model development and theory. *Atmospheric Environment*, 42(26):6442–6452, 2008. doi:10.1016/j.atmosenv.2008.04.024.
- [55] Alexandre Petroff, Alain Mailliat, Muriel Amielh, and Fabien Anselmet. Aerosol dry deposition on vegetative canopies. Part I: review of present knowledge. *Atmospheric Environment*, 42(16):3625–3653, 2008.

- [56] Kostas Philippopoulos and Despina Deligiorgi. Application of artificial neural networks for the spatial estimation of wind speed in a coastal region with complex topography. *Renewable Energy*, 38(1):75–82, 2012. URL: <http://dx.doi.org/10.1016/j.renene.2011.07.007>. doi:10.1016/j.renene.2011.07.007.
- [57] Michael A Poole and Patrick N O’Farrell. The assumptions of the linear regression model. *Transactions of the Institute of British Geographers*, pages 145–158, 1971.
- [58] Carl Edward Rasmussen and Christopher K I Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.
- [59] Roy Rasmussen, Mike Dixon, Frank Hage, Jeff Cole, Chuck Wade, John Tuttle, Starr McGettigan, Thomas Carty, Lloyd Stevenson, Warren Fellner, and Others. Weather Support to Deicing Decision Making (WSDDM): A winter weather nowcasting system. *Bulletin of the American Meteorological Society*, 82(4):579–595, 2001.
- [60] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.
- [61] Xing Shi, Yiyu Zhu, Jin Duan, Runqing Shao, and Jianguo Wang. Assessment of pedestrian wind environment in urban planning design. *Landscape and Urban Planning*, 140:17–28, 2015.
- [62] William C Skamarock, Joseph B Klemp, Jimy Dudhia, David O Gill, Dale M Barker, Wei Wang, and Jordan G Powers. A description of the advanced research WRF version 2. Technical report, National Center For Atmospheric Research Boulder Co Mesoscale and Microscale Meteorology Div, 2005.
- [63] S I V Sousa, F G Martins, M C M Alvim-Ferraz, and M C Pereira. Multiple linear regression and artificial neural networks based on principal components to predict ozone concentrations. *Environmental Modelling & Software*, 22(1):97–103, 2007. URL: <http://linkinghub.elsevier.com/retrieve/pii/S1364815205002240>. doi:10.1016/j.envsoft.2005.12.002.
- [64] Andreas Stohl, P Seibert, G Wotawa, D Arnold, John Faulkner Burkhart, Sabine Eckhardt, C Tapia, A Vargas, and T J Yasunari. Xenon-133 and caesium-137 releases into the atmosphere from the Fukushima Dai-ichi nuclear power plant: determination of the source term, atmospheric dispersion, and deposition. *Atmospheric Chemistry and Physics*, 12(5):2313–2343, 2012.
- [65] Roland B Stull. *An introduction to boundary layer meteorology*, volume 13. Springer Science & Business Media, 2012.

- [66] J P van der Meulen and T Brandsma. Thermometer screen intercomparison in De Bilt (The Netherlands), Part I: Understanding the weather-dependent temperature differences. *International Journal of Climatology*, 28(3):371–387, 2008.
- [67] Ivanka Videnova, Dimitar Nedialkov, Maya Dimitrova, and Silvia Popova. Neural networks for air pollution nowcasting. *Applied Artificial Intelligence*, 20(6):493–506, 2006. doi:10.1080/08839510600753741.
- [68] Walter D Wilkerson. Dust and sand forecasting in Iraq and adjoining countries. Technical report, AIR WEATHER SERVICE SCOTT AFB IL, 1991.
- [69] James W Wilson, N Andrew Crook, Cynthia K Mueller, Juanzhen Sun, and Michael Dixon. Nowcasting thunderstorms: A status report. *Bulletin of the American Meteorological Society*, 79(10):2079–2099, 1998.
- [70] J C Wyngaard. Cup, propeller, vane, and sonic anemometers in turbulence research. *Annual Review of Fluid Mechanics*, 13(1):399–423, 1981.
- [71] Ke Xu, Christopher K Wikle, and Neil I Fox. A kernel-based spatio-temporal dynamical model for nowcasting weather radar reflectivities. *Journal of the American statistical Association*, 100(472):1133–1144, 2005.
- [72] David N Yates, Thomas T Warner, and George H Leavesley. Prediction of a flash flood in complex terrain. Part II: A comparison of flood discharge simulations using rainfall input from radar, a dynamic model, and an automated algorithmic system. *Journal of Applied Meteorology*, 39(6):815–825, 2000.
- [73] Liyang Yu, Neng Wang, and Xiaoqiao Meng. Real-time forest fire detection with wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, volume 2, pages 1214–1217. IEEE, 2005.