Evaluating Spatial Generalization of Stacked Autoencoders in Wind Vector Determination

Richard McAllister and John Sheppard
Gianforte School of Computing
Montana State University
Bozeman, MT

Abstract

Unique areas in the Earth's atmosphere may be influenced by forces that differ subtly from area to area. Using an extensive set of neural networks for assessing atmospheric conditions introduces a situation where many (perhaps millions) networks must be trained and maintained. We explore the extent to which deep learning using Stacked Autoencoders can be generalized across a spatial range of the atmosphere as a method of expanding the applicability of a network trained in a specific area of the atmosphere to the areas that surround it. As a prelude to exploring techniques for transfer learning, we demonstrate that a Stacked Autoencoder is capable of capturing some knowledge universal to this situation and make approximations of the functions that determine wind vector components.

Introduction

From a human perspective, the major factors that influence weather conditions in one area of the Earth's atmosphere and those that influence weather conditions in another seem to be the same. However, considering that the atmosphere is fluid, that conditions vary in time, that we have no access to the initial conditions of this fluid, and that we cannot enumerate the myriad forces that compose these factors, prediction is squarely in the domain of machine learning. If we assume that consistency holds between disparate locations around the Earth, transfer learning can be used to reduce the complexity of whole-Earth models.

In transfer learning, a machine learning system uses some information that has been learned in one setting to improve generalization in another setting (Goodfellow, Bengio, and Courville 2016). Here we are considering separate, discretized areas of the Earth's atmosphere to be these disparate settings. Some consistencies that exist among these settings are that they are all composed of atmospheric gases, they are influenced by their own conditions and by the conditions of the areas that immediately surround them, and that this is not an entirely closed system. The differences are in that they have differing initial conditions and that they are affected by external forces in different ways. For example, one area of the atmosphere may be over mountainous land and another may be over water, thus radically changing the associated weather dynamics. Further, one area may receive more solar radiation than another during the day.

Given such variation, one may be inclined to think that each area of the Earth would need to train its own model, resulting in a whole-Earth system that would be required to maintain millions of models (depending on the spatial resolution of each model). We seek a method whereby a fraction of these models would be required. In this paper we discuss experiments testing the generalization ability of representations learned by Artificial Neural Networks (ANN) over a primary area of the Earth to areas adjacent to this primary area. We demonstrate that the functions learned to approximate wind vector components using radiometric data can be used to make determinations about the general trend of the wind vectors in these peripheral areas.

In the next section we will provide background on the problem domain. Next we will discuss the data that we are using and motivation how this study will support research into transfer learning. We will then discuss our approach. Our experimental design, execution, and results will then be explained. We will then discuss the broader implications of our results. Finally, we discuss our conclusions and the next steps for this research.

Background

Related Work

Stacked Autoencoders: In this work we use Stacked Autoencoders (Ackley, Hinton, and Sejnowski 1985). Autoencoders are used to create compact, reconstructible representations of data at a lower dimensionality. These representations can be stacked, using the outputs of hidden layers of one Autoencoder to encode deeper levels in a process called Unsupervised Pre-Training (UPT). In this way, we can capture hierarchical internal representations of (theoretically) the most important features of the data (Erhan et al. 2010).

Deep Learning in Weather Parameter Prediction: Research using deep neural networks for weather forecasting has been increasing over the past few years. This is partially in response to the advancement in training efficiency that was achieved by Hinton, *et al.* (2006). This is motivated by how important weather prediction is becoming to many industries. As a domain-specific example, Singh (2016) used

deep networks to determine how much wind energy was expected given the wind speed, humidity, and generation time.

Earlier, Dalto, Matusko, and Vasak (2015) used deep networks for ultra-short-term wind forecasting. Their study demonstrated that deep neural networks benefit from an intelligent reduction in the number of input variables, allowing network training to complete in a reasonable time. They used Partial Mutual Information-based Input Variable Selection, which is a technique for determining how much redundancy is captured in the input variables, and selected variables that were most separate from each other. Similar to our approach, they used separate networks for orthogonal wind vector components, not considering the vertical dimension at all. They used pre-trained Stacked Denoising Autoencoders as their network model.

Narejo and Pasero (2017) performed "meteonowcasting," (i.e., predicting conditions of the present or very near future) using multilayer pre-trained Restricted Boltzmann Machines. They used a method based on Mutual Information to determine the relevancy of atmospheric parameters at the prior time step to the current or future prediction of these parameters. The previous temperature and time of day were found to be quite useful for determining the temperature in the current time or near future. Each of these atmospheric parameters was treated separately, and the collection of parameters and topology of the networks used differed with each parameter they were trying to predict.

Most recently, the usefulness of Stacked Autoencoders with UPT for wind vector determination from radiometric data was demonstrated by McAllister and Sheppard (2017). The focus of this work was on demonstrating that UPT could be deployed efficiently in this problem space, thereby filling a gap in the capabilities of Numerical Weather Prediction (NWP) systems that work off of radiometric data. Their results showed that Stacked Autoencoders could be used to estimate wind vectors based on this data to a high level of accuracy and much more efficiently than standard NWP systems. The results in this paper extend this work, demonstrating the extent to which trained networks can generalize to nearby regions in space.

Transfer Learning: Transfer learning is the act of enabling a learner to use information from a model that was trained in one domain as a way to bootstrap training in another, related domain (Pan and Yang 2010; Weiss, Khoshgoftaar, and Wang 2016). One of the motivations for transfer learning is that data may not be available for a domain for which one wishes to make predictions. So the idea is to train on a domain in which there is plentiful data that is somehow related to the domain of interest. We assert that another motivation for using transfer learning is when there exists an extremely large number of domains that differ in subtle but important ways. Such a set of domains could be manifest as many geographic points on the globe that each have different forces influencing the conditions in their respective areas.

Hu, Zhang, and Zhou (2016) used transfer learning for short-term wind speed prediction. They were addressing the problem where data for wind speeds over new wind farms was unavailable. The authors trained deep Autoencoders in areas that varied with respect to terrain, weather, and topography and still were able to produce models that learned abstractions that predicted wind speed in new farm areas effectively.

Problem Strategy

In our experiments, we attempt to determine current wind vector values using only radiometric data. Radiometers are unable to measure wind directly, but we assert that radiometric readings still provide information about wind vectors that can be used to train associated predictive models. The wind vector data we use comes from a separate wind-oriented model over the same region at the same spatial and temporal resolution as the radiometric data we use for training. This wind vector data is used as ground truth for our models.

In predicting wind vectors, we want to know the extent to which the knowledge encoded in a neural network trained in one geographical area can be leveraged in making accurate predictions about conditions in another area. In each location around the Earth, weather conditions are all described by the same number and type of dimensions. This means that we are ensured of the ability to describe different locations using the same parameters. This is why we hypothesize the ability to transfer knowledge geographically; the consistency in parameters for each area allows us to feed the trained model with data from different regions.

Data

As was done in McAllister and Sheppard (2017) we used a dataset created by Zhang and Gasiewski (2016) using an NWP simulator called the Weather Research and Forecasting Model (WRF). This dataset was created to support developing models from higher spatial and temporal resolutions than are currently available. The simulation used measurements of the East Coast of the United States during 24 hours of Hurricane Sandy in 2012. Specifically, the data is within an area bounded in the southwest at 26.4902°N, 81.6064°E and in the northeast at 41.2117°N, 60.3809°E (Figure 1). Actual radiometric and wind measurements were taken at a far lower spatial and temporal resolution during this storm. WRF was used to interpolate measurements between these basis measurements so the simulated dataset has a spatial resolution of 5km and a temporal resolution of 15 minutes.

For our study, we focused on the four points shown in Figure 1. Location 1 is an area that was in the eye of the storm for the duration of the simulation. Location 2 was in an area that was relatively unaffected. Strong rain bands were occurring in Location 3, and Location 4 had already been hit by the storm. Table 1 shows the latitudes and longitudes of these locations.

The data was provided in the form of *point clouds*, which are collections of points that exist in some spatial coordinate system. To make deep learning possible, we needed to bin these points spatially into discrete cells so that each data point in one cell would be representative of the conditions in that cell at that time. We used a Discrete Global Gridding System (DGGS) as the spatial binning mechanism (Sahr, White, and Kimerling 2003). Each data point in the

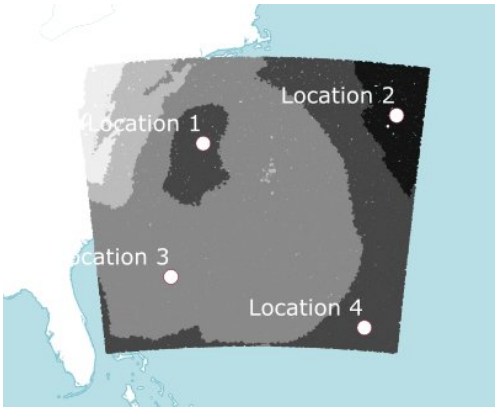


Figure 1: The barometric pressure variation across the dataspace.

Name	Latitude	Longitude
Location 1	37.07	-73.79
Location 2	38.34	-62.63
Location 3	30.69	-75.65
Location 4	28.14	-64.49

Table 1: Latitude and longitude of the locations of interest

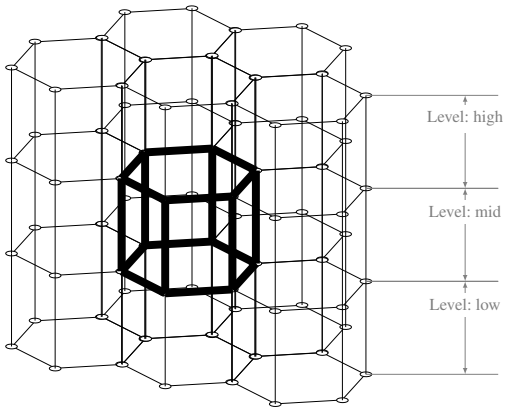


Figure 2: The 21 3D DGGS Cell Lattice

point clouds was given a GIS Point designation on the globe, which allowed the data to be discretized into the DGGS using spatial queries. Each DGGS cell has 15km resolution in our model.

Radiometers are capable of obtaining vertical soundings of the atmospheric parameters. Therefore, our simulation included 60 vertical levels of readings, which we discretized into 10 levels.

Approach

Cell Lattices

Figure 2 depicts our view of the space surrounding a given center cell for which we want to predict the wind vector components. This 21 cell lattice includes the 7 cells above the center cell, the 6 cells surrounding the center cell on the same level, and the 7 cells below the center cell. When training, we used measurements for representative points from all of these cells in the previous time slice (including wind vector information), as well as the wind vector components from the current time slice for the center cell as ground truth. In a second experiment, we just use radiometric data (no wind vectors) from the previous time slice.

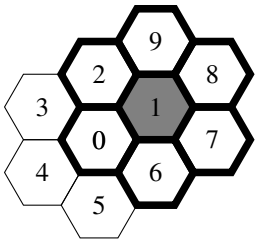


Figure 3: 7 Cells Representing One DGGS Level

The focus of this study is on determine whether we can transfer the knowledge from a model trained on one DGGS cell to a neighboring cell. This situation is depicted in Figure 3, which limits the process in one layer[1]. The idea is that we have trained a network for cell 0 using the information captured from the cell 0 and its surrounding cells. We then shift to a neighboring cell, such as cell 1 (in gray). For this cell, we use the network trained for cell 0 together with the measurements from the previous time slice in cells surrounding cell 1 (outlined in bold). We repeat this process, determining wind vector components in cells 2 through 6 as well.

We call the collection of all of these cells, along with their data, the *super lattice* for cell 0. Thus, the super lattice includes all of the cells surrounding cell 0 which we call the *center lattice*, as well as all of the cells surrounding cells 1 through 6 as center cells. Working in the space of the super lattice provides a sense of the ability of the network trained for cell 0 to generalize to the neighboring cells, thus indicating a degree to which the trained network can be used in for transfer learning.

Our deep learning architecture consists of a four-layer Stacked Autoencoder, trained using greedy unsupervised pre-training. We used the same architecture and parameters discussed in McAllister and Sheppard (2017) in these experiments.

Experiments

Overview

Our experiments proceeded using the following steps:

1. For a location of interest, instantiate the super lattice.
2. Pre-train the Stacked Autoencoder layers using the data from the center lattice.
3. Fine-tune the Stacked Autoencoder using the data from the center lattice.

[1]Even though we have limited the presentation to one layer, the actual experiments considered the measurements over all three layers surrounding the target cell.

Reading Source	Reading Name
Radiometry Measurements	Temperature
	Pressure
	Cloud Density
	Rain Density
	Ice Density
	Snow Density
	Graupel Density
Wind Speed	Wind u (East/West)
	Wind v (North/South)
	Wind w (Up/Down)

Table 2: Features for Each Data Point

4. Use the trained Stacked Autoencoder to predict the wind vector components for each of the peripheral cells.

Super Lattice: Table 2 shows the data available for each cell in the super lattice. The top section of the table shows what measurements are available through passive microwave radiometry and the bottom portion shows what is available using other instruments (dropsondes, direct observation, etc.). We conducted experiments for data that included the wind vector components for the previous time step for all of the cells in the super lattice, as well as data that did not include these wind vector components. This approach allowed us to examine the ability of networks trained using only radiometric data to predict wind vectors and to assess the need for the wind vector data when moving to neighboring cells. The data that included the wind vectors from the previous time step had 231 dimensions and for the data without wind vectors had 168 dimensions.

Pre-Training and Fine Tuning: Our Stacked Autoencoders had 150, 140, 130, and 120 neurons per hidden layer when proceeding from input to output. As mentioned, the input layer consisted of 231 or 168 neurons depending on whether or not prior wind vectors were included. These layers were pre-trained using the training data from each cross validation fold. Following pre-training, the Stacked Autoencoder layers were fine-tuned using backpropagation.

Each wind vector component (u, v, and w) was trained with a separate network. We did this because, according to Gasiewski (2017), these three components should be treated as if they are independent. Preliminary experiments where we combined all components into a single network supported this recommendation.

Generalization: Once we trained our networks on the center lattice, we used each network to predict the current values of the wind vectors using data from the 20 neighboring lattices, using the 20 peripheral cells as the center cells.

To the best of our knowledge, we are the only ones using this data in this manner. Therefore, we are unable to compare our results directly to similar methods reported in the literature. Therefore, our comparison focused more on the potential dependence on prior wind vector information in generalizing predictive power in different geographic regions.

Results

We used the root mean squared error (RMSE) as our performance measure, which was the same performance measure used in (McAllister and Sheppard 2017). Table 3 shows the average RMSE of predictions for all three vector components for all cells surrounding the center cell (six surrounding azimuths above and below) for Location 3. This location is interesting because it lies directly in the rain band area south of the eye of the hurricane. Table 4 shows the coefficient of determination values (R^2) for the six cells surrounding the center cell at the same elevation, where we calculated R^2 as follows:

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2}$$

where f_i is the ith predicted value from the neural net, y_i is the ith target (observed) value, and $\bar{y}$ is the mean target (observed) value over the range of prediction.

The results for the peripheral cell at azimuth 1 at the medium elevation for Location 3 in Figure 1 are represented in Figure 4. The first column of charts shows the results of the predictions made for the data that included wind vectors from the previous time slice, and those on the right did not include wind vectors from the previous time slice. The first row depicts predictions for the u component, the second for the v component, and the third for the w component. This location, azimuth, and relative elevation was chosen because it was representative of the results obtained over the entire dataspace of peripheral cells.

Discussion

For the locations studied, we found that our Stacked Autoencoders produced approximations of wind vectors that generally mimic what was happening in reality. We intentionally chose cells that were adjacent to the training cell to allow us to deviate spatially from the training space in a gradual manner. This is based on an assumption of locality that we made with regard to atmospheric dynamics—the forces that cause the atmosphere to behave the way it does—differ more as one moves further away from the initial training point (Collins et al. 2013). If this is true then the networks so trained will be unable to capture a general model of the atmosphere when trained on any single area; however, we may be able to identify conditions under which generalization and transfer would be feasible.

The network trained on the center cell performed best at the middle elevation. The conditions in the cells above and below the center cell were different, but we were generally able to approximate the shape of the functions of the wind vectors in those areas. The average RMSE for the cells in the layers above and below the level depicted in Figure 4 were significantly higher than for the cells at the same level as the cell upon which we trained the network. We may infer from this that there is an altitude-based sensitivity to the predictive capabilities to networks trained in this manner, though this conclusion will have to be tested using much larger datasets.

	u Component		v Component		w Component	
Level	With Wind	Without Wind	With Wind	Without Wind	With Wind	Without Wind
High	97.345	65.290	16.704	7.917	0.003	0.002
Mid	0.492	0.345	0.205	0.162	0.001	0.001
Low	55.841	53.823	4.600	1.369	0.005	0.003

Table 3: Average RMSE by Level for the Cells Around Location 3

Level	Azimuth	u Component		v Component		w Component	
		With Wind	Without Wind	With Wind	Without Wind	With Wind	Without Wind
Mid	Azimuth 1	0.989	0.993	0.984	0.985	0.817	0.658
	Azimuth 2	0.993	0.995	0.988	0.992	0.866	0.769
	Azimuth 3	0.989	0.992	0.984	0.988	0.823	0.736
	Azimuth 4	0.992	0.993	0.986	0.991	0.853	0.797
	Azimuth 5	0.993	0.995	0.987	0.991	0.726	0.627
	Azimuth 6	0.989	0.993	0.982	0.983	0.810	0.729

Table 4: Coefficient of Determination (R^2) of Deep Networks for the Mid Level Around Location 3

Conclusions and Future Work

Based on the results of these experiments, we believe that we are able to apply deep learning with Stacked Autoencoders to capture important generalizations about the state of the atmosphere with respect to radiometric data and wind vectors. We have confirmed that these generalizations are capable of producing approximations for wind vector values in closely adjacent areas, thus suggesting the potential to be able to apply deep learning in NWP without the need for massive numbers of spatially distinct models. How versatile and transferable these generalizations are will be a subject of future research.

We intend to focus the next steps of our research on determining what useful generalizations are captured when the networks are trained on one area and applied in a variety of novel situations. Among these situations will be locations that are further apart and the same locations in a different dataset depicting a different storm system (or no storm system). This investigation will also include an analysis of the effectiveness of pre-training on a primary region of interest and fine-tuning on the target region of interest of interest.

References

Ackley, D. H.; Hinton, G. E.; and Sejnowski, T. J. 1985. A Learning Algorithm for Boltzmann Machines. *Cognitive Science* 9:147–169.

Collins, S. N.; James, R. S.; Ray, P.; Chen, K.; Lassman, A.; and Brownlee, J. 2013. Grids in Numerical Weather and Climate Models. *Climate Change and Regional/Local Responses* 256.

Dalto, M.; Matusko, J.; and Vasak, M. 2015. Deep Neural Networks for Ultra-Short-Term Wind Forecasting. In *Proceedings of the IEEE International Conference on Industrial Technology*, 1657–1663.

Erhan, D.; Bengio, Y.; Courville, A.; Vincent, P.; and Bengio, S. 2010. Why Does Unsupervised Pre-training Help Deep Learning? *Journal of Machine Learning Research* 11:625–660.

Gasiewski, A. 2017. Personal Communication with Director of the Center for Environmental Technology, University of Colorado.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. MIT Press.

Hinton, G. E.; Osindero, S.; and Teh, Y.-W. 2006. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* 18(7):1527–1554.

Hu, Q.; Zhang, R.; and Zhou, Y. 2016. Transfer learning for short-term wind speed prediction with deep neural networks. *Renewable Energy* 85:83–95.

McAllister, R. A., and Sheppard, J. W. 2017. Deep Learning for Wind Vector Determination. In *Proceedings of the IEEE Swarm Intelligence Symposium (to Appear)*.

Narejo, S., and Pasero, E. 2017. Meteonowcasting Using Deep Learning Architecture. *International Journal of Advanced Computer Science and Applications* 8(8):16–23.

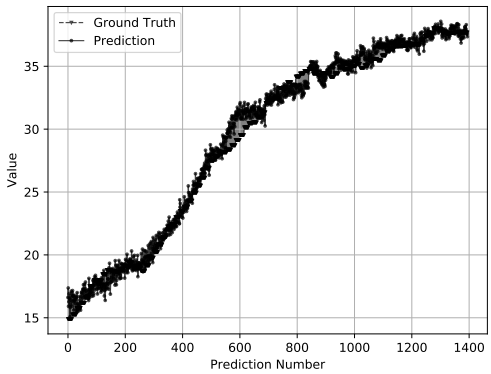
Pan, S. J., and Yang, Q. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 1(10):1345–1359.

Sahr, K.; White, D.; and Kimerling, A. J. 2003. Geodesic Discrete Global Grid Systems. *Cartography and Geographic Information Science* 30(2):121–134.

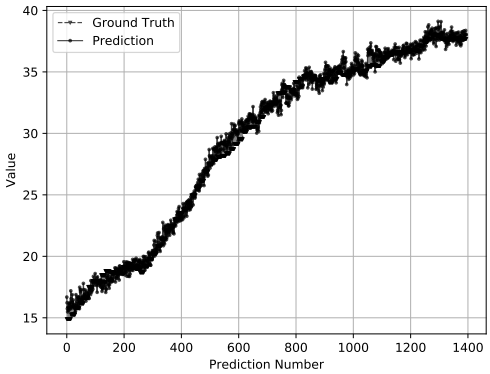
Singh, V. 2016. Application of Artificial Neural Networks for Predicting Generated Wind Power. *International Journal of Advanced Computer Science and Applications* 7(3):250–253.

Weiss, K.; Khoshgoftaar, T. M.; and Wang, D. 2016. A Survey of Transfer Learning. *Journal of Big Data* 3(1).

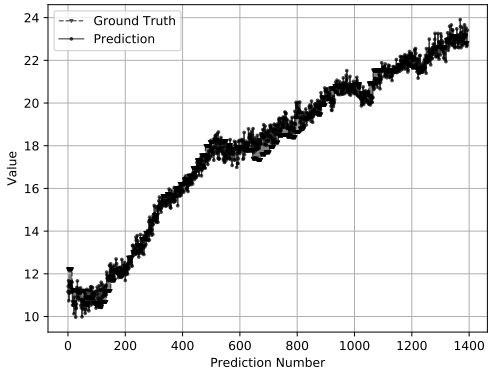
Zhang, K., and Gasiewski, A. J. 2016. Microwave CubeSat Fleet Simulation for Hydrometric Tracking in Severe Weather. In *IEEE International Geoscience and Remote Sensing Symposium*, 5569–5572.



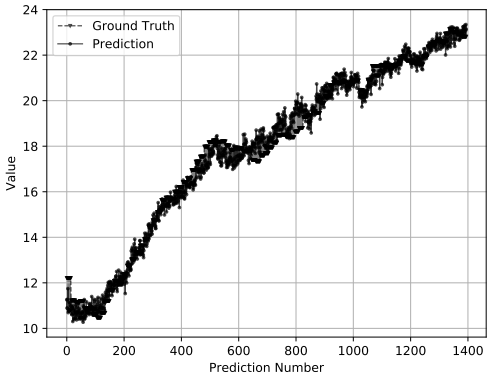
(a) u Component With Wind



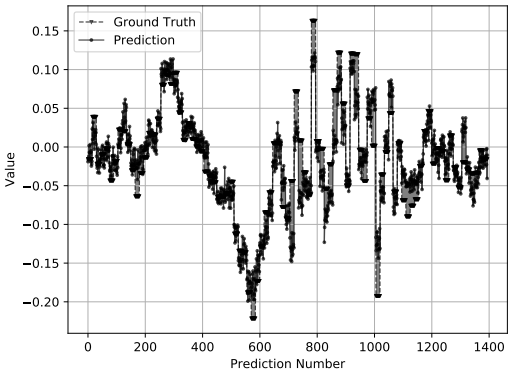
(b) u Component Without Wind



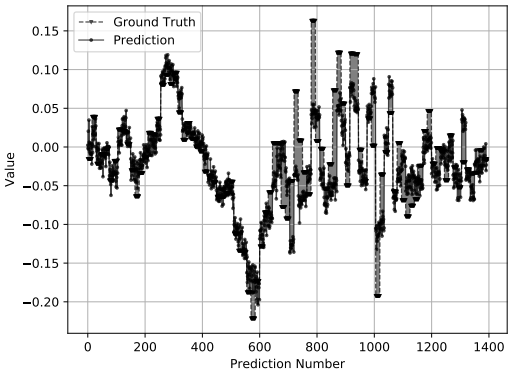
(c) v Component With Wind



(d) v Component Without Wind



(e) w Component With Wind



(f) w Component Without Wind

Figure 4: Predicted vs Actual plots for the peripheral cell at azimuth 1 at the middle elevation for Location 3. The magnitude of error over the range of the predictions is shown by the shaded region between the plots.