Deep Learning for Wind Vector Determination

Richard McAllister, John Sheppard
Numerical Intelligent Systems Laboratory
Montana State University
Bozeman, MT
richard.mcallister@msu.montana.edu, john.sheppard@montana.edu

***Abstract*—Numerical Weather Prediction (NWP) is a process of using numerical simulation to track and predict weather patterns over time. NWP is presented with a number of challenges in that it aims for precision in a situation that is by its nature uncertain and influenced by factors that are very difficult to enumerate. As an alternative, Empirical Weather Prediction (EWP) attempts to use observational data to construct models of weather phenomena. We evaluate a deep learning method applied to EWP to make wind vector determinations from radiometric data using data collected from Hurricane Sandy. Our approach uses unsupervised pre-training of stacked autoencoders to construct multilayer perceptrons. We then discuss the role of our approach as an important step in positioning EWP as a viable alternative to NWP.**

I. Introduction

The use of artificial neural networks in weather prediction systems has been growing due to the effectiveness that they have seen in a variety of other areas. Some of these domains, such as meteorology, have been dominated by disciplines that favor precise mathematical inference over the imprecision inherent in the use of machine learning. Today, weather prediction is done mostly using mathematical tools from atmospheric science to extrapolate from the current conditions of the atmosphere into the future.

Mathematical precision in weather prediction can be computationally expensive. Very large supercomputers have to be dedicated to the problem of weather prediction[1], since the operations that are necessary to make these predictions often involve performing numerical integration and differentiation at a high precision[2]. Imprecision creeps into the system at every level, since there are many causes of uncertainty in a system as complex as the weather. In addition to the uncertainty surrounding weather prediction, sometimes the state of the features of the atmosphere that we want to determine (in this case, the wind) must be inferred using other data that do not include the previous states of that features (in this case, radiometric data).

Radiometric measurements of the atmosphere, which can be used directly to determine temperature, moisture content, and precipitation type as point observations, do not include wind data. If the main interest is in determining wind vectors, and radiometric data is all that is available, inferences must be made. We hypothesize that wind vector determinations can be made using deep multilayer perceptrons that have been conditioned using unsupervised pre-training on radiometric data.

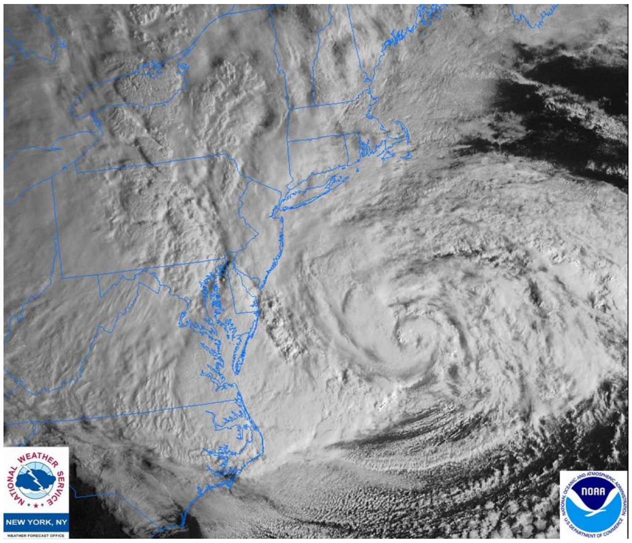


Figure 1. Hurricane Sandy on October 29, 2012 [3]

Figure 1 shows a satellite image of Hurricane Sandy, which affected the East Coast of the United States in the Fall of 2012. Considering that the economic and social impact of this storm was significant, and because a variety of conditions exist within a hurricane that might be representative of the larger wind vector determination problem, we considered it a suitable data set to test our hypothesis. Our goal is for techniques like ours to play a role in enhancing the situational awareness of those affected by such storms.

II. Background

A. Related Work

Shi, *et al.* [4] performed “nowcasting” of precipitation by formulating it as a spatiotemporal sequence forecasting problem that can be solved using a sequence-to-sequence learning framework. The goal of their nowcasting task is “to perform precise and timely predictions in a local region over a short period of time.” They used multiple Convolutional LSTM layers (i.e., FC-LSTM models that have convolutional structures in both the input-to-state and state-to-state transitions) to form an encoding-forecasting structure for capturing spatiotemporal correlations from the data.

After acknowledging that weather data is nonlinear and follows irregular trends, Abhishek, *et al.* developed nonlinear models using ANNs to create weather data simulations

978-1-5386-2726-6/17/$31.00 ©2017 IEEE

[2]. In this paper they asserted "... massive computational power required to solve the equations that describe the atmosphere, error involved in measuring the initial conditions, and an incomplete understanding of atmospheric processes..." were significant factors motivating the ANN approach. Their approach specifically addressed temperature forecasting and used an assemblage of temperature data, year over year, to predict a recent daily maximum temperature. In contrast to the deterministic (NWP) approach, they observed high accuracy in the predictions made by their ANNs.

Using deep methods for rainfall prediction has seen an increasing trend in recent years. Hernández, *et al.* [5] used deep learning to predict accumulated daily precipitation a day in advance. They used a "multilayer autoencoder" (*not* a stacked autoencoder), where the hidden layers of the autoencoders are connected directly to single nodes in the single hidden layer of a multilayer perceptron. They used a variety of input variables related to rainfall, such as relative humidity, temperature, dew point, previous rainfall, sun brightness, and barometric pressure, collected over ranges varying from the previous three to the previous five days. Their technique outperformed using a multilayer perceptron alone and significantly outperformed the work of Abhishek *et al.* [6] who used a multilayer perceptron three layers deep with 10 and 20 nodes per layer. Our approach differs from Hernández, *et al.* in that we are using unsupervised pre-training to train a deep network rather than simply training the nodes of a multilayer perceptron.

Salman *et al.* compared the prediction performances of recurrent neural networks, conditional restricted Boltzmann machines, and convolutional neural networks to investigate hierarchical weather representations that result from training these networks [7]. Specifically, they wanted to predict the behavior of El Niño / Southern Oscillation (ENSO) parameters. Given their study was preliminary, their methodology was largely exploratory, and they did not claim that they had a superior technique.

Grover *et al.* studied combining discriminatively trained predictive models with a deep neural network that models the joint statistics of weather related variables [8]. They used the spatial characteristics of the data, learning long-range spatial dependencies in the process. They applied their technique to predict wind speeds, dew points, geopotential heights, and temperature for a period of time ranging from 6 to 24 hours, and their technique performed well against some state of the art methods that are currently being used at NOAA.

B. Related Technologies

1) Discrete Global Grid Systems: NWP uses the Discrete Global Gridding Systems (DGGS) as a method of geographical binning. A DGGS is a system of adjacent polygons that cover the entire planet [10]. Such systems allow the collection of point observations that occur in the same vicinity so that each point collected in a polygon is seen as representative of that polygon. The size of these polygons determines the resolution of the weather information system. In our experiments we did not average the data points for each polygon in each time slice because this allowed us to provide more individual data points to the neural networks.

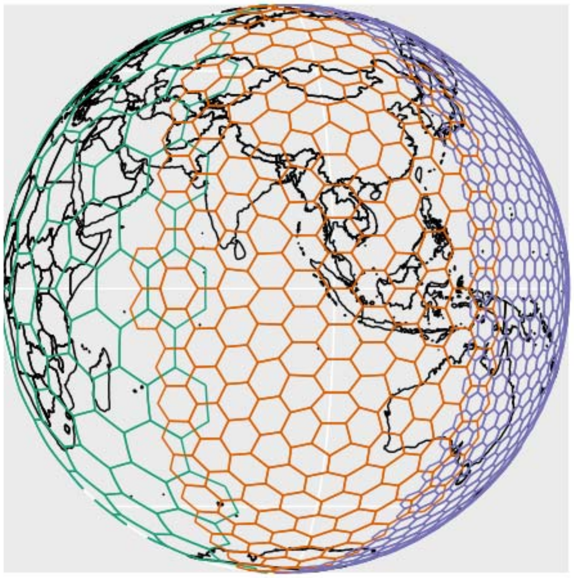


Figure 2. Intuitive Depiction of the Hexagonal Discrete Global Grid System [9]

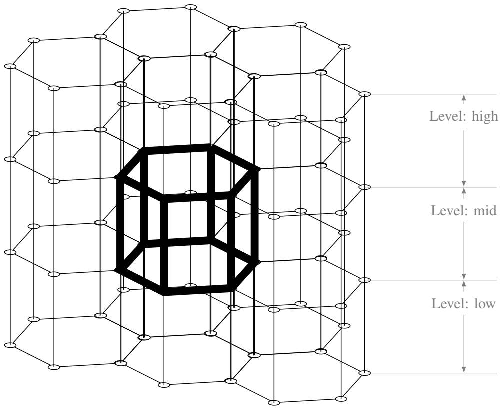


Figure 3. The 21 3D DGGS Cell Lattice

We use 15 kilometer grid cell sizes, so our system can be described as having a resolution of 15 kilometers at sea level. In addition, we use a geodesic DGGS grid that is based on hexagonal grid shapes. This provides superior resistance to shape distortion and size consistency [11]. To obtain a representation of a DGGS suitable for insertion into a GIS database we used dggridR [9], which is a system that creates base polygons for this purpose. As an example, Figure 2 shows a hexagonal DGGS superimposed over the entire planet at three different resolutions.

Each cell in an area of study is represented by $c_{h,a,t}$. The symbol h represents the height relative to the center cell, or cell of interest. The heights will be labeled "high," representing the cells above the center cell level, "middle" (mid) for those at the same level, and “low” for cells at below the center cell. The symbol a is the azimuth of the cell. Since these are hexagonal cells there will be six cells surrounding the center cell. These six cells are represented by azimuths 1 through 6 with the center cell represented by azimuth 0. The symbol t represents the time slice relative to the time about which we are attempting to make predictions, with a value of -1 representing the conditions from the previous time slice. In our study, the cell about which we are attempting to make predictions is designated $c_{\text{mid},0,0}$. We assume that the current weather conditions cell $c_{\text{mid},0,0}$ are influenced directly by the condition of this cell in the previous time slice $c_{\text{mid},0,-1}$ and the conditions in the previous time slice of all of the cells in the 3D lattice that are adjacent to this cell. In other words, we incorporate conditions from $c_{\{\text{high},\text{mid},\text{low}\},\{0-6\},-1}$.

2) Autoencoders and Unsupervised Pre-Training: Autoencoders, first introduced by Ackley *et al.* [12] are simple neural networks that encode a vector of inputs into a lower-dimensional representation, with the aim of decoding them into their original feature space with minimal distortion [13]. They form the basis of our neural networks used in our experiments. In Unsupervised Pre-Training (UPT), autoencoders are trained and stacked iteratively, where an autoencoder is trained to capture the input’s main variations [14]. Figure 4 depicts the operation of UPT. In the experimental analysis performed by Erhan *et al.*, results are offered that indicate that “unsupervised pre-training guides the learning towards basins of attraction of minima that support better generalization from the training data set.” Consequently, in this study, we used UPT to learn what is hypothesized to be an abstract hierarchy of features to characterize the input data.

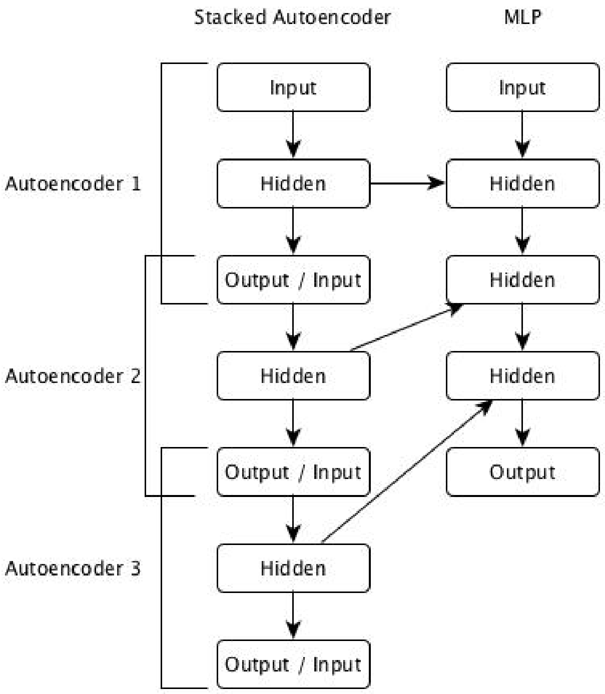


Figure 4. Unsupervised Pre-Training

III. Dataset

We used the outputs of a Weather Research and Forecasting (WRF) model simulation of Hurricane Sandy created by Zhang and Gasiewski [15]. This storm system devastated the East Coast of the United States in 2012 and was the second costliest hurricane in United States history [16]. The simulation made use of actual measurements from the storm but then enabled generation of data with higher spatial and temporal resolution. This is a common practice in NWP so that mesoscale weather models can be improved to keep pace with instruments being developed that provide data at these resolutions. Specifically, WRF provides simulated data at a 5 km grid spacing and generates the entire field at an interval of 15 minutes. Figure 5 shows the area of the Eastern seaboard covered by the simulation during a 24 hour period. The area is bound in the southwest at 26.4902°N, 81.6064°E and in the northeast at 41.2117°N, 60.3809°E.

Figure 5a shows the standard deviation of the barometric pressure for various points of the storm over the 24 hour time period. Here, lighter shades show lower standard deviation. The area with the highest variation in barometric pressure was in the northeast of the dataspace. Likewise, in Figure 5b the lighter areas represent areas where the standard deviation of the temperature over the entire 24 hour dataset was lower and the darker areas show where it was higher. As can be seen, the area of greatest variability is in the extreme northwest of the data space. This is the area where the storm is making landfall and is affected by different atmospheric conditions than it is over the ocean.

IV. Approach

A. Overview

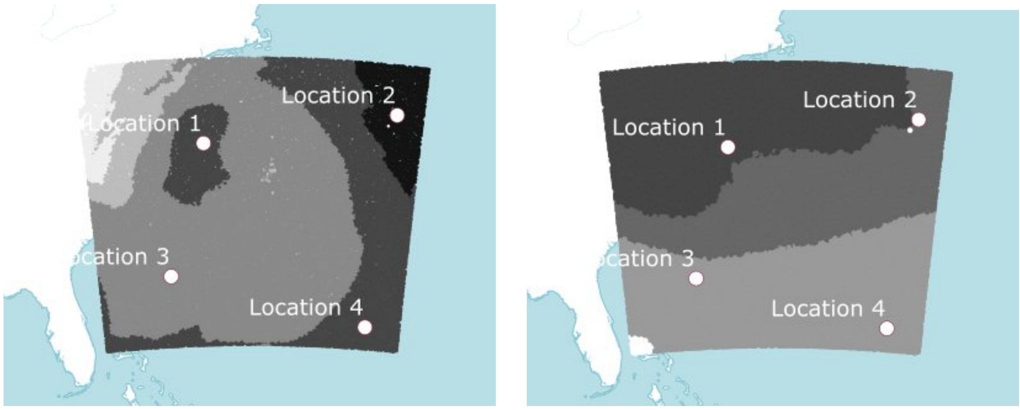
We assume for our experiments that the current weather conditions of any location depend on the following:

1) The previous weather conditions at that location.
2) The previous weather conditions at the locations immediately surrounding that location.

Essentially, what this does is define a spatiotemporal “Markov blanket” around the cell over which we are predicting the wind vectors.

Autoregressive models like ARMAX [17] could be considered to make these predictions. But our ultimate goal was in predicting wind vector components without using wind as explicit inputs. The ARMAX more naturally applies the knowledge of the prior wind vectors, thus defeating our purpose. We wanted to ignore the wind vectors from the previous time slice from consideration altogether.

For each experiment, we retrieved all data from each cell for the entire duration of the simulation. These readings were sorted by time. To simplify, each reading, including the wind vector reading, was regarded as representative of the entire cell from which it came. Since our DGGS cell resolution is 15 km and the simulation resolution was 5 km, there were



(a)

(b)

Figure 5. Left (a) shows the barometric pressure variation, right (b) shows the temperature variation. Example locations for analysis were chosen at four different areas with different standard deviations for barometric pressure and temperature.

several readings for each time slice for each cell. Therefore, in most cases, the radiometric data do not correspond to the exact locations of the wind vector data, but the two do always occur in the same cell.

B. Data Storage

Our data was stored in a star-schema of point-cloud rasters using the PostGIS extension of the PostgreSQL database management system. This differs from extant meteorological resources, which use database management systems to store metadata and flat files to store the actual point data. The reason we used PostGIS is because of the ease with which queries for points internal to a 3D polygon (a "surface" in GIS parlance) can be performed. This functionality comes with a cost in storage space, however: The non-indexed system occupied approximately 450 GB of storage while the indexed data system for our 24-hours of 15-minute data required 1.2 TB of storage space. The data structures containing the DGGS exist alongside the point data in the PostGIS database, which facilitates queries for the data inside the polygons. The extra space occupied by the DGGS grid itself was negligible.

C. Data Preparation

We analyzed each of 100 different locations distributed evenly across the dataspace. Our experiments cover a variety of pressure and temperature conditions, as can be seen in Figures 5a and 5b; however, we highlight the results associated with the four locations shown in this figure.

V. Experiments

A. Overview

Our experimental procedure consisted of the following steps:

1) For a location of interest, determine DGGS cell containing that location.
2) Query the database for the surrounding grid cells.
3) Query the database for all data points from all of the relevant cells.

Table I
Raw Features for Each Point in the Simulation

Reading Source	**Reading Name**
Radiometry Measurements	Temperature
	Pressure
	Cloud Density
	Rain Density
	Ice Density
	Snow Density
	Graupel Density
Wind Speed	Wind u
	Wind v
	Wind w

4) Scale the data.
5) Time shift the wind data.
6) Obtain cross validation splits for model evaluation.
7) For each cross validation trial, do the following:
   a) Perform unsupervised pre-training over the stacks of the autoencoder on the training set.
   b) Train on training set.
   c) Test on the test set.

The results of the cross-validation trials were then aggregated to obtain a summary of the performance of our approach.

1) Locate DGGS Cells: For each of our points of interest on the map, a PostGIS query is made to the database to determine the DGGS cell that contains this point and another to determine all of the cells surrounding this cell. For each of these cells, their azimuths are recorded so they can be used later to identify the location of individual readings.

2) Retrieve Cell Data Points: For each of the 21 cells retrieved for each point of interest from the database [1], all of the points contained in those cells are retrieved. Table I shows the dimensions of a single data point. Wind u, Wind v, and Wind w are the u, v, and w wind speed components of the wind vector at that point. These are the points of data that we are interested in for training our model. Note that u is the east

[1] Fourteen if the altitude of interest is at sea level.

/ west vector component, v the north / south component, and w is the vertical component. Each of the readings are tagged with the elevation and azimuth information from the cell in which the reading was taken. For example, the temperature for the highlevel cell at azimuth 260 would be tagged as “azimuth_260_high_temperature.”

3) Scale the Data: The model resulting from training our neural networks performed far better when we scaled all of the data to a range of -1 to $+1$ and then de-scaled when evaluating the predictions. To do this, all data retrieved from the datastore, including the test data, was normalized to this range.

4) Time Shift Wind Data: For each of the time slices the target dimensions (in this case the wind vector component values) from the next time slice are appended to the set of dimension representing the current time slice. This is in keeping with our assumption that the current wind vector conditions are dependent on the values from the previous time slice. We restrict this experimentation to only single-order temporal consequences, meaning that we are only trying to predict the wind vector values of the current time slice using information from the time slice immediately preceding the current one.

5) Obtain Cross Validation Splits: Arranging the data as we have done with the above time shifting procedure allows us to use k-fold cross-validation in our experiments. This would not be possible if we were to assume that there were order-k temporal consequences to each configuration of the training data, and that these consequences were also reflected in the test data. Fortunately, by defining the spatiotemporal Markov blanket around each cell, this assumption becomes fairly safe. If this were not the case, then we would only be able to test wind vector determination on the latter part of the day, training the networks sequentially up until then. Though other configurations are reasonable, in this investigation we use 10-fold cross-validation.

6) Unsupervised Pre-Training: For unsupervised pre-training we stacked four layers of autoencoders, with layer sizes from input to output of 150, 140, 130, and 120 respectively. By using this configuration we include a “bottleneck” that is important for autoencoders to perform feature abstraction. We also allow the network to learn an internal hierarchical representation inherent in this data. Each layer was conditioned using unsupervised pre-training. We chose this network topology after experimenting with a variety of network widths and numbers of layers. We found that this topology offered the best performance.

7) Neural Network Training: After pretraining and assembling the network, training proceeded using the normal backpropagation training process. Each wind vector component, u, v, and w was treated separately by training three different networks using the pre-trained stacked autoencoder. This follows from the assumption that the forces that influence latitudinal, longitudinal, and vertical motion are independent from each other as is the norm for the numerical weather prediction community [18].

Table II
LATITUDE AND LONGITUDE OF CLOSELY INSPECTED LOCATIONS

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

8) Testing the Networks: Unfortunately, since we are the only ones using this data in this manner, and it is a new dataset with the unique characteristics that enable this type of analysis and prediction, we are unable to directly compare our results to the results of anyone else. However, we do provide a novel investigation into the use of this data using machine learning techniques and we plan to build upon this knowledge.

B. Experimental Design

1) Distribution of Locations For Analysis: We chose four points of interest to demonstrate the effectiveness of this methodology. Note, however, that we ran our analysis on all 100 data points described above. The points of interest described here were chosen because they represent different pressure/temperature profiles and are generally representative of all of the results obtained. Table II shows the latitudes and longitudes of these locations, and Figure 5 shows where the points are located on the map.

To predict each of the three vector components (u, v, and w) we used three separate networks, both pre-trained and trained separately. We did this because these three components should be treated as if they are independent. The assemblage of three networks produced three different vector components. We claim that it is not reasonable to recombine the three components from all observations since unique cross validation splits were created for each of the three vector components at each point of interest.

2) Evaluating the Use of Wind Vector Measurements: For comparison, we wanted to get a sense for how having knowledge of all of the previous time slice’s wind vector measurements influenced the predictions that were made. In one set of experiments we removed all wind vector measurements, relying solely on radiometric data for wind vector prediction. For the other thread, we included all wind vector measurements for each cell except for the cell about which we were making predictions.

C. Experimental Results

Table III shows the root mean squared error (RMSE) of predictions for all three vector components for the locations studied, concentrating on the four closely inspected locations with an average over the k folds. It also shows an average RMSE over all 100 locations studied. Table IV shows the coefficient of determination values (R^2) for these same locations with an average R^2 value over all 100 locations studied. Here

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

Table III
Deep Network RMSE for the Locations Studied

	u component		v component		w component	
	With Wind	Without Wind	With Wind	Without Wind	With Wind	Without Wind
Location 1	0.0794	0.1042	0.1218	0.1894	0.1590	0.1022
Location 2	0.1688	0.1505	0.0631	0.1121	0.1415	0.1523
Location 3	0.0115	0.0596	0.1031	0.1284	0.1103	0.1324
Location 4	0.1048	0.1245	0.1736	0.2107	0.1373	0.1241
Avg Over All 100	0.0850	0.1235	0.0958	0.1326	0.1203	0.1359

Table IV
Coefficient of Determination (R^2) of Deep Networks for Locations Studied

	u component		v component		w component	
	With Wind	Without Wind	With Wind	Without Wind	With Wind	Without Wind
Location 1	0.4630	0.3300	0.4558	0.1737	-2.1930	-0.5132
Location 2	0.0769	0.0627	0.7621	0.5473	-0.0673	-0.1679
Location 3	0.9606	0.7963	0.4401	0.3246	0.0818	-0.0407
Location 4	0.5161	0.4617	0.1710	0.0286	-0.1488	-0.1368
Avg Over All 100	0.5956	0.4265	0.5462	0.3888	-0.1931	-0.3395

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 of one representative fold from Location 1 in Figure 5 are represented in Figure 6a–f. This location corresponds to the eye of the storm, which is an area of both greater barometric pressure and temperature variability. The ground truth values ("Actual" values) differ between charts of the same vector component because they were randomly drawn from the data in the k-fold cross-validation process. We chose to show the results from Location 1 because it is representative of the types of results that we achieved without being either the best or the worst result.

VI. Discussion

In general, we found that the stacked autoencoders performed well across the 100 points examined. We begin by examining Location 1, which corresponds to the eye of the storm. Given this is the eye which has high variability among pressure and temperature, we were not surprised when we found wind vector functions that varied more dramatically. Even with the highly variable network output, it appears in Figure 6 that the predicted values generally simulate the underlying distribution of the actual values. It is interesting that the difference between actual and predicted values is greater when the previous time slice's wind vector information is used for prediction. At this time, we cannot explain the increase in error, and this will be a topic of future research.

When considering the w vectors specifically, we found that the data was highly noisy and difficult to predict. Based on conversations with research meteorologists at the University of Colorado, we were advised that the w vector component should be regarded as Gaussian noise [18]. When examining the results in Table IV, we found that the R^2 values are mostly negative. This happens when the predictions do not explain the variability in the input data very well. The positive results for RMSE (Table III), on the other hand, may be showing that our stacked autoencoder was still successful learning the Gaussian distribution.

Location 2 was on the leading edge of the storm. The predictions generated by the stacked autoencoders were more in keeping with the ground truth for the v (east–west) component of the wind vector. The RMSE values were correspondingly lower here as well. The u component, however, presented more of a challenge due to much higher variability. In our future experiments using this dataset, we intend to find a meteorological basis for this.

Location 3 proved to be the most predictable of the four locations discussed here. This is a location that is well inside the rain bands of the Hurricane, which is an area where storms exhibit the greatest organization in terms of wind direction and speed. This is especially true when the values of the wind vector components were known beforehand.

Location 4 was in a situation that is somewhat opposite from what is seen at Location 2. This location was at the trailing edge of the storm over open water. In this case it was the v vector component that exhibited a greater degree of predictability. As with Location 2, further experimentation using this dataset will necessitate an understanding of why this occurs, as well as clarification on whether or not there are predictable differences in the conditions at the leading and trailing edges of hurricanes in general.

From our experiments we see that general simulation of wind vector functions is possible using only radiometric data. We believe that the reasons for these positive results are that physical influences transfer spatially in a way that is predictable by the recognition of higher-order patterns in the data describing the adjacent spaces and their histories. Surprisingly we do not see consistent improvement when information about the wind vectors from the previous time slice is included. Although this may be a question for the meteorological community, this work may be a contribution to that community by identifying scientific questions to ask. Even so, our results are interesting and encouraging, since it is a goal of this research to enable wind vector predictions from radiometric data alone. Thus our long-term intent is to use wind only to provide the ground truth during training.

We observe that training a deep stacked autoencoder is able to approximate the distribution of wind vector components using nothing but radiometric data. We also observe that the predictions tend to have a smoothing effect on the outliers in the data, though we do not know exactly how to characterize the outlying observations in the first place. For example, it is possible that these outliers are an calibration artifacts of the simulation or the original instrumentation. It is also possible that noise was added into the simulation to bring it closer to reality or to define a desirable test scenario.

VII. Conclusions and Future Work

Remote sensing data is different from what is normally termed as "big data," since this data is highly structured. However, data collection in this area results in enormous volumes that must be spatially and temporally correlated. Because of this, to use remote sensing data effectively in machine learning (i.e., create systems that are relatable and queryable such that timely predictions can be made) different data deployment regimes than those traditionally used in the meteorology community must be exploited. We intend to experiment with large scale distributed storage systems to facilitate an environment more favorable for this type of analysis. Such a system would be necessary to make any of this analysis usable for real-world data.

A successful outcome of research into appropriate data systems will facilitate an expansion of the spatial extent of the inputs to our networks. This will allow us to explore more complex hierarchical relationships among phenomena in the weather space, such as n-order spatial and temporal influences.

An avenue of research that will closely follow this paper is testing the longer-term (on the scale of hours) prediction capabilities of this procedure. This will involve generating models for all involved dimensions and so will require considerably more time for processing. The process will be to train the networks on the first part of the day, extrapolate for the next several hours, and test against the actual (WRF-simulated) data at the end of the day.

The fact that we are inferring wind vector begs the question: can we create an accurate simulation environment using wind vector values that we predict (essentially using newly predicted wind vector values as inputs)? How far into the future can we run these predictions and still expect reasonable accuracies from the models? In doing this we would have to compare this deep-learning prediction framework with current states of the art in NWP (such as WRF) and see which system ends up with values that are closer to reality.

Finally, a major intent of this research is to identify encodings in deep neural networks for use in transfer learning. We hypothesize that neural networks exhibit organizational characteristics that allow them to be partitioned based on the features learned, and that these partitions can be used to construct new networks that combine relevant partitions from the networks. We intend to use test this hypothesis in the weather domain as well as other compatible domains such as precision agriculture (e.g., yield prediction).

Acknowledgments

We wish to thank Dr. Albin Gasiewski and Kun Zhang of the University of Colorado for providing data and expert guidance in this research. We also thank members of the Numerical Intelligent Systems Laboratory at Montana State University for their advice and support through this project.

References

[1] R. J. Firth, "A Novel Recurrent Convolutional Neural Network for Ocean and Weather Forecasting," Ph.D. dissertation, Louisiana State University, 2016.

[2] K. Abhishek, M. Singh, S. Ghosh, and A. Anand, "Weather Forecasting Model Using Artificial Neural Network," *Procedia Technology*, vol. 4, pp. 311–318, 2012.

[3] N. W. S. US Department of Commerce, NOAA, "Hurricane Sandy," 2013. https://www.weather.gov/okx/HurricaneSandy http://www.weather.gov/okx/HurricaneSandy

[4] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," *Advances in Neural Information Processing Systems*, pp. 802–810, 2015.

[5] E. Hernández, V. Sanchez-Anguix, V. Julian, J. Palanca, and N. Duque, "Rainfall prediction: A deep learning approach," in *Hybrid Artificial Intelligent Systems (HAIS)*, 2016, pp. 151–162.

[6] K. Abhishek, A. Kumar, R. Ranjan, and S. Kumar, "A Rainfall Prediction Model Using Artificial Neural Network," *IEEE Control and System Graduate Research Colloquium*, vol. 7, pp. 82–87, 2012.

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

[8] A. Grover and E. Horvitz, "A Deep Hybrid Model for Weather Forecasting," *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 379–386, 2015.

[9] R. Barnes, "dggridR: Discrete Global Grids for R," 2016. https://cran.r-project.org/web/packages/dggridR/vignettes/dggridR.html

[10] G. Percivall, "Geodata fusion study by the Open Geospatial Consortium," in *SPIE 8747, Geospatial InfoFusion III*, M. F. Pellechia, R. J. Sorensen, and K. Palaniappan, Eds., 2013.

[11] R. Gibb, A. Raichev, and M. Speth, "the Rhealpix Discrete Global Grid System." http://www.discreteglobalgrids.org/wp-content/uploads/sites/33/2016/10/PlanetRiskDGGS.pdf

[12] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A Learning Algorithm for Boltzmann Machines*," *Cognitive Science*, vol. 9, pp. 147–169, 1985.

[13] P. Baldi, "Autoencoders, Unsupervised Learning, and Deep Architectures," *JMLR Workshop on Unsupervised and Transfer Learning*, vol. 27, pp. 37–50, 2012.

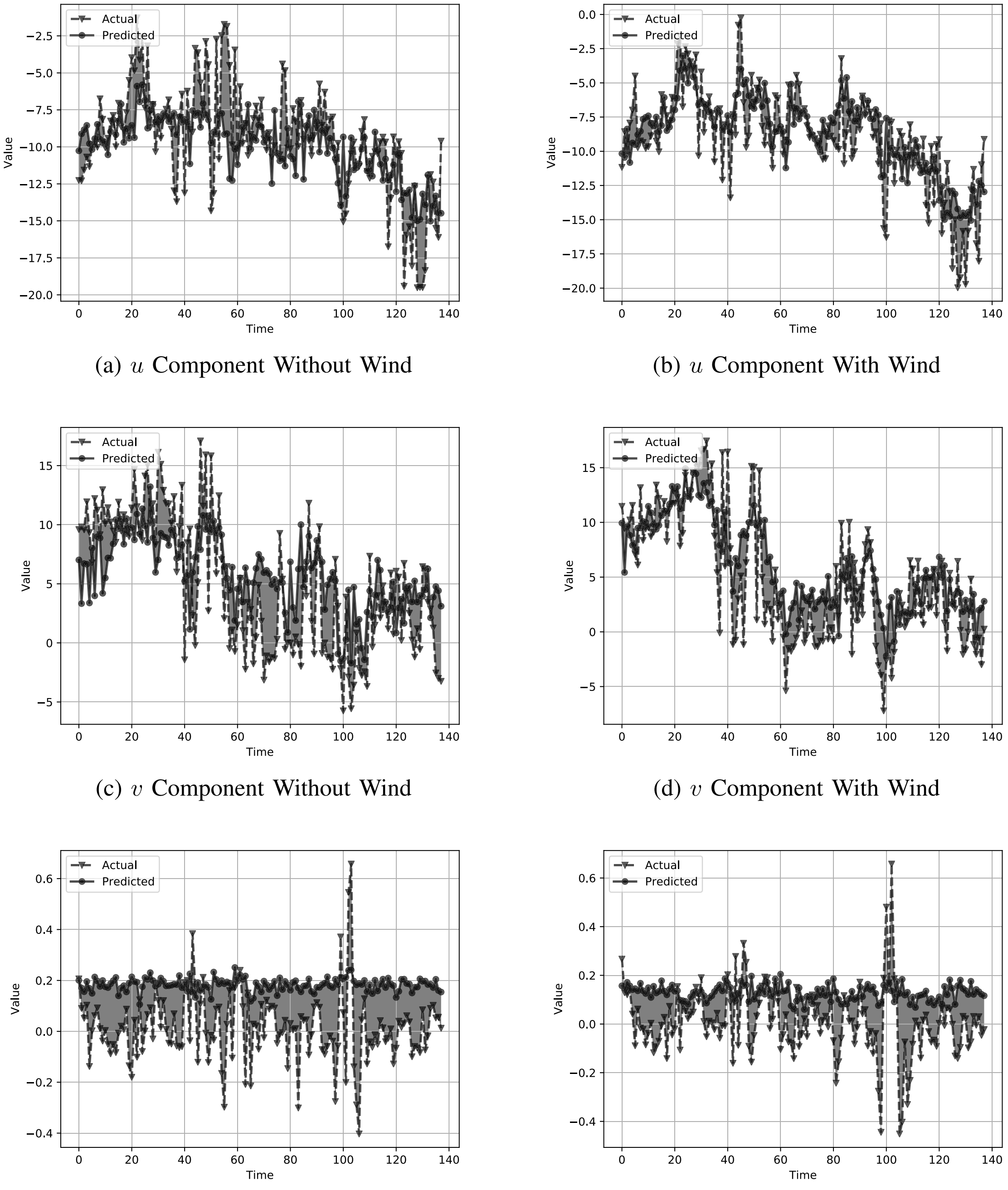
[14] D. Erhan, Y. Bengio, A. Courville, P. Vincent, and S. Bengio, "Why Does Unsupervised Pre-training Help Deep Learning?" *Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

[15] K. Zhang and A. J. Gasiewski, "Microwave CubeSat fleet simulation for hydrometric tracking in severe weather," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016, pp. 5569–5572.

[16] A. Fallis, "Tropical Cyclone Report Hurricane Sandy (AL182012) 22 – 29 October 2012 Eric," *Journal of Chemical Information and Modeling*, vol. 53, no. 9, pp. 1689–1699, 2013.

[17] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis*. Hoboken, NJ: John Wiley & Sons, Inc., 6 2008. http://doi.wiley.com/10.1002/9781118619193

[18] A. Gasiewski, "Personal Communication," 2017.



(a) u Component Without Wind

(b) u Component With Wind

(c) v Component Without Wind

(d) v Component With Wind

(e) w Component Without Wind

(f) w Component With Wind

Figure 6. Predicted vs Actual plots for one representative cross validation fold at Location 1. The magnitude of error over the range of the predictions is shown by the shaded region between the plots.