Approximating Missing Sensor Readings Using An Artificial Neural Network

Tyler Kostuch, Trent Thomas
David Block, Tim Julius, Josette Staples, Jayson Walberg
François Neville
Department of Mathematics and Computer Science
Bemidji State University
Bemidji, MN 56601
Tyler.Kostuch@gmail.com, Trent.Thomas@live.bemidjistate.edu

Abstract

Professional mariners rely on accurate knowledge of ocean conditions to perform their jobs and prevent accidents. Events occurring in coastal waters can potentially impact not only the local economies of coastal communities but have consequences to the national economy as well. Ocean currents are monitored with coastal radar stations and buoys that have wireless communication links with each other. These may occasionally malfunction during harsh weather conditions however, or even due to normally-occurring atmospheric conditions. As continuous monitoring of ocean conditions allows naval workers to plan efficiently and reduce opportunities for any negative impact, a suggested approach to maintaining continuous ocean current readings under uncertain conditions is the artificial neural network computational model.

Neural network models allow for missing data to be approximated by internalizing past patterns that have been observed in the data record. The model is inspired by biological networks of neurons and consists of interconnected computational units, forwarding intermediate results from one unit to the next. Weights associated with the connections between these computational units can dynamically adapt as new patterns and data are observed by the system.

Internally, the model initializes all weights randomly between the neurons. The model is fitted, or trained, by introducing a set of data with known output to the model and each weight adjusts accordingly in small increments. After evaluating training data, the model is validated by executing over a separate set of data with known outputs to which the model has not been exposed. The average error of these results is compared to a desired level of tolerance. Alternation between the training and validation processes continues until the model's error reaches an acceptable range.

One difficulty that this model has to overcome in the proposed application is due to the region from which the data was collected and the tendencies of the current in that area. Naturally, currents in the Gulf of Maine display sinusoidal behavior; however, in some sub-regions, currents can rapidly change and present extremely high velocities. To overcome this difficulty, a model initially trained on overall data was subsequently retrained on smaller sub-regions, allowing for improved results in those areas.

Approximations resulting from the neural model appear to reliably outperform common predictive techniques such as auto-regressive functions, areal averaging and the persistence model. As most linear extrapolation models perform poorly for this application due to the sinusoidal patterns exhibited in tidal signals as well as the difficulties presented by wireless communication links, the Neural Network appears to be a promising approach for modeling and approximating missing readings for these natural phenomena.

1. Introduction

With the continuing development of Wireless Sensor Networks (WSN) to enable us to monitor our environment, increasingly innovative techniques are required to process the volumes of data collected, as well as approximate missing data readings should the sensors in question fail. Natural forces have ever presented a challenge to man-made constructions with floods, tornadoes and other natural disasters periodically causing problems of varying severity world-wide. As technology and our understanding of the world grows, we are better able to prepare and predict such events to help prevent losses. But setting catastrophes aside, simply monitoring the real-time data of the prosaic events taking place within nature on a daily basis can be intriguing for the researchers that spend much of their time studying and working in the field.

Like a weather report for the general public, information gleaned from monitoring ocean currents has benefits for all mariners in a monitored coastal region. Tidal forces change on a daily basis with varying degrees of magnitude and particular areas may have potentially hazardous currents. Providing real time readings of the current allows mariners to plan their route efficiently, adjust their course should problems occur and avoid endangerment. Fishermen may also be able to use the results to follow a particularly abundant source of food, while freighters may plan their routes around routine current changes to maximize their efficiency. The Coast Guard could warn those on the seas of pending dangers or localize lost cargo that has drifted out to sea. Marine Biologists, oceanographers and other marine scientists may benefit as well by being better able to perform their studies by relying on the results produced by ocean current monitoring [2, 8]. Regardless of the task at hand, accurate real-time measurements of currents can be a crucial aspect to many professions.

Nor are the benefits of monitoring coastal waters localized only to those most directly involved. Coastal communities are economically impacted by the fishing, transportation and leisure activities that operate in their region due to their close proximity to the sea and its resources. Beyond that, a significant portion of the United States' economy is reliant upon ship-borne imports and exports, whose follow-on economic effects are then felt throughout the country. For all of these reasons, the coastal ocean is an important environment in which to have reliable real-time monitoring systems embedded.

Limitations exist, however, on what our present sensor technology is capable of. Currently, coastal radar stations may obtain accurate surface ocean current readings only when their signal coverage overlaps. Another method of data collection are buoys that transmit the data via radio frequencies. But either method has various benefits and downsides depending on the particular circumstances.

The coastal radar stations' reliance on overlapping signals presents potential difficulties as geographical distance in combination with inclement weather circumstances may cause interference with the radar signal. Similarly, buoys are expensive to produce and maintain simply due to their often-remote locations, as well as being embedded in a corrosive salt water environment that is not conducive to the long-term survival of the electronic components they carry. It is for these reasons that data retrieved from these systems may often be incomplete or unsatisfactory for display.

Different mathematical models can be used to approximate the missing data as it occurs in order to accurately display the data. Linear models may be used to forward-extrapolate for missing data in a time-series at a buoy-location, for example; however natural signals are infrequently linear, nor are they stationary, complicating the modeling effort. Having a long series of past data for a location can often enable an accurate fit of a sinusoidal signal to observed readings, so long as power, memory and computational resources are not limited. However, sensor nodes embedded *in situ* in remote, natural environments do not usually benefit from a reliable, constant power source, but must instead make do with stored power in the form of batteries. The Artificial Neural Network (ANN) is particularly promising in this context, as it is a highly adaptable mathematical model used for interpolation and extrapolation and has proven to be successful in the face of both of these difficulties -- nonlinear approximation and limited input data sets [1, 5]. It has further been found to be resistant to the nonstationarity common to natural data readings [9] and is thus an intriguing approach to investigate regarding its capacities in providing accurate approximations of surface tidal currents. Since such models can be pretrained in the laboratory and electronically delivered to the sensor station *in situ*, the expense of regular service-visits to each marine sensor platform can be avoided. As the models are pretrained, no extensive computations need to be made on the fly in order to fit data to a nonlinear function whenever the model needs to provide a result, thus reducing some battery usage. And finally, since an ANN is adaptive, models once delivered to their disparate platforms may each continue to self-modify and evolve into accurate approximators for their particular regions, as opposed to a static model whose performance may degrade more quickly with the changing of the seasons, for example.

2. METHODOLOGY

2.1 Ocean Surface Current Measurements

The Gulf of Maine, on the northeastern coast of the United States, extends from the northern tip of Cape Cod, MA in the southeast to the southern tip of Cape Sable, Nova Scotia in the northeast. It encompasses nearly 58000 km^2 of ocean, and is known to have some of the highest tidal variations on earth. The data used in this study were drawn from the Gulf of Maine Ocean Observation System (GoMOOS) repository currently maintained by the School of Marine Sciences at the University of Maine. The collection system consists of an array of buoy-mounted sensors, as well as coastal radar (CODAR) stations located to provide maximal coverage of the sea-surface of the Gulf of Maine [8]. Data was collected in June 2005 from the CODAR system of 4.3-5.4 MHz SeaSonde HF radar stations deployed along the perimeter of the Gulf of Maine. Each radar station periodically transmits radar signals in a 360º pattern, directed out towards the surface of the ocean. Reflected signals undergo a Doppler frequency-shift, from which one radial component of the surface current velocity vector may be determined. Combining the radial surface readings of all CODAR stations from a single point in time enables the synthesis of a field of 2-D surface current velocity vectors, each assigned a location in a regular square grid with cells measuring roughly 20 x 20 km, running parallel to the coast of Maine. We refer to the two components of these velocity vectors as u and v. Fields of surface currents are thus determined once per hour, and are used as a proxy for a notional network of wireless sensor nodes embedded upon the ocean surface. Further information regarding CODAR array operation can be found in [8].

2.2 The Artificial Neural Network Model

The Artificial Neural Network is a computational model which recognizes patterns inductively by mapping relationships between particular inputs and the corresponding outputs. The general model is based on an abstract representation of biological neurons (computational nodes) and synapses (links) which interconnect the nodes with a weighted value. Many variations exist on the base model, generally categorized as either regression- or classification-type models.

The functionality of the nodes and links is almost trivial and provides the basis for the function of the entire model. Every node has a set of input links that provide a value. The receiving node then sums all incoming values and evaluates the sum through an activation function. This value is then fed forward via outgoing weighted links to subsequent nodes where the process is repeated.

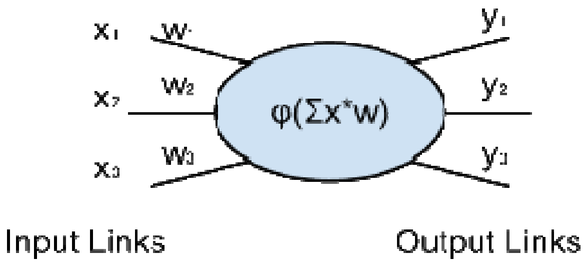

Figure 1 : Computational Node within an ANN

Figure 1 illustrates a particular node within the neural network. The incoming *x* values generated by other nodes are passed through the left links which scale the *x* value by the weight w_i associated with the link. The node then sums all inputs and applies the activation function Φ. The result then gets sent out as the *y* values (i.e. the output of the node multiplied by the link's weight w_i) which become the *x* values for subsequent nodes.

The model organizes the nodes into three or more ordered layers which determines where synapses are formed. In the standard feed-forward ANN model, there are no recurrent connections (i.e. links between nodes within the same layer) thus any one node in a particular layer receives its inputs from nodes in the preceding layer, and outputs its results to nodes in the subsequent one.

The first layer to a typical neural network model is termed the *input layer*, which contains only output synapses. The input layer links to the first of *n* hidden layers. The hidden layers are one or more layers containing an arbitrary number of neurons. The last of these hidden layers then connects to the output layer, which contains nodes that only have input synapses and the values contained within the nodes is the model’s outputs.

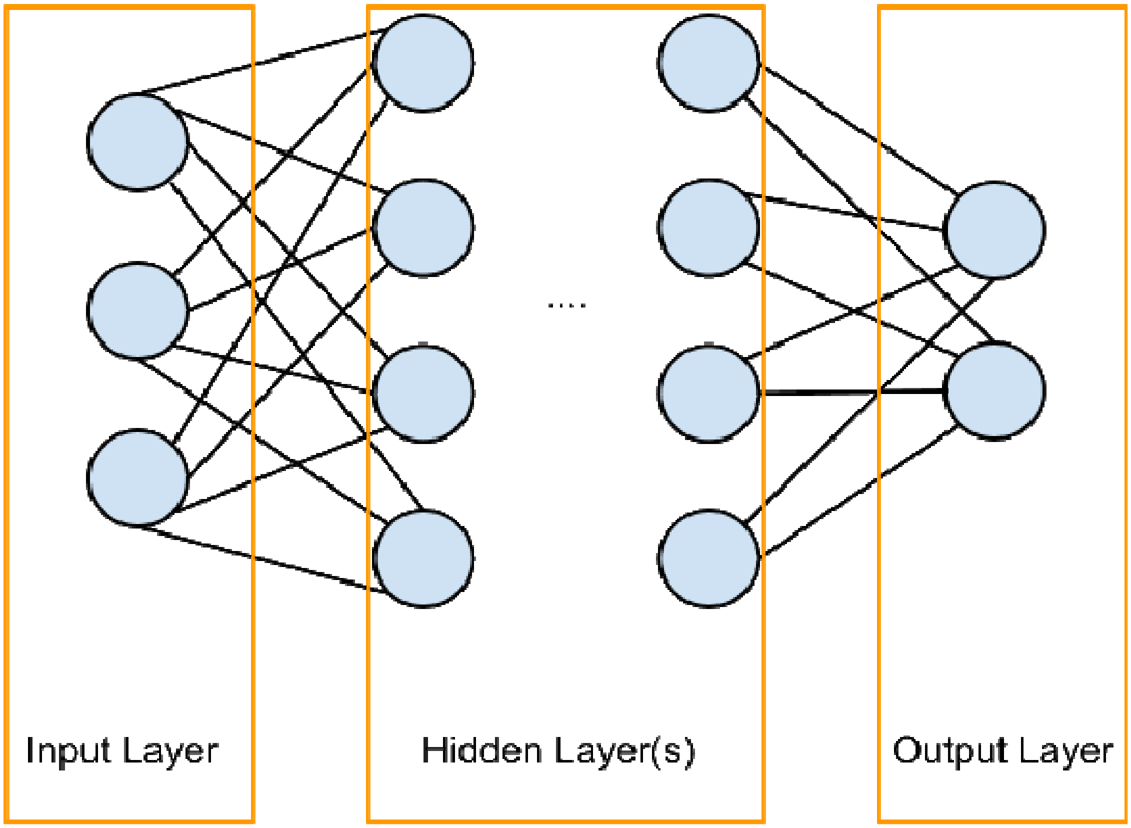

Figure 2 : An illustration of the ANN model.

In Figure 2, each circle represents a node and the links represent the synapses that connect the neurons. The arbitrary model depicted here has three input nodes, two output nodes and four nodes in each hidden layer. Note that the amount of neurons in any given layer is wholly design-dependent, based on the particular circumstances of the models application.

The ANN model initializes all of its weights to small random values which will change as the model is *trained*. Training is the act of passing known inputs values through the model, comparing the model's output results to desired values, and then adjusting the weights of the synapses through an algorithm known as *backpropagation* [1, 7]. The backpropagation algorithm iteratively passes through all the links, working backwards, adjusting weights so that the results of output nodes are within an arbitrary acceptable error of desired model output values.

$$\delta_o = (t_o - r_o) * f'_o(act_o) \quad (1)$$

$$\delta_h = f'_h(act_h) * \Sigma w_{ho}\delta_o \quad (2)$$

$$\Delta w = -\eta * \delta_o * act_x \quad (3)$$

$$w_o = w_o + \Delta w_o \quad (4)$$

Backpropagation works by finding the difference between model outputs and the desired target values (Equation 1), and propagating these backwards to obtain deltas for the weights of each link (Eqs. 1, 2). This delta value is multiplied to the node's activation function to obtain the gradient of the error contributed by each weight . Once each gradient is obtained, an arbitrary ratio known as the *learning rate*, often denoted η, is used to modify the weight in the opposite direction of the gradient (Eq. 3). Each time this process is performed, model weights are adjusted (Eq. 4) to produce a more accurate approximation of the desired output.

The choice of a *learning rate* that properly adjusts the weights is crucial for back propagation to work correctly. If the learning rate η is too large then the change in weight for the synapses will be too large as well, potentially leading to model divergence and failure. Therefore a relatively small η is necessary, however; choosing a too-small η will result in unnecessarily slow convergence to a solution. In our implementation, η was initialized to 0.1

After the entire set of training data has been trained with the back-propagation algorithm, a different data set with known outputs is processed by the model to generate validation results. This validation data is used to get a general idea of the error-level of the model on data it has never previously seen, and thus the accuracy we can expect of it once deployed in the field. Should this error value meet the application's accuracy requirements, it is ready for deployment. Otherwise the backpropagation process is continued.

2.3 The Temporal Persistence Model

The persistence model is among the most simplest approaches for populating missing data. A temporal version of the more well-known spatial nearest-neighbor approach, this technique uses the last known value at that location as the value of the missing data. While this method might seem too primitive to give any reliable results, it does perform surprisingly well considering that tides typically do not shift magnitude or direction too rapidly under normal circumstances, and assuming the substituted value is not too old.

Due to the simplicity of this model, it was used in the preliminary stages when making comparisons with artificial neural network. While the persistence model was quickly beat once the ANN. took shape, initially the model offered a time-efficient comparison to judge the effectiveness of the ANN.

To conceptualize this, consider the set of known readings at a particular point, $\{x_0, x_1, ..., x_n\}$. The persistence model, $p(t)$, is simply defined as:

$$p(t) = x_{t-1} \qquad (5)$$

2.4 The Linear Approximation Model

The linear model generates a best fit line from the three prior readings then uses that line to predict (or extrapolate) the missing datum. Though relatively easy to implement and straight-forward, the model’s linear nature makes it considerably inaccurate whenever the (sinusoidal) tides begin to change, and provides no ways to compensate for this which ultimately accounts for the model's high error.

To generate the linear model with a given set of readings at a particular point which will be denoted as $\{(t_0,x_0),(t_1,x_1),...,(t_n,x_n)\}$ where the t_0 is the first reading in the set, first the slope, m, was computed with:

$$m = \frac{n(\Sigma tx) \quad (\Sigma t)(\Sigma x)}{n(\Sigma t^2) \quad (\Sigma t)^2} \tag{6a}$$

and then the intercept, denoted b, was found with:

$$b = \frac{\Sigma x \quad m(\Sigma t)}{n} \tag{6b}$$

The resulting linear equation is $y(t) = mt + b$. Evaluating $y(t_n + 1)$ will generate the predicted value for the next hour.

2.5 The Areal Averaging Model

The areal averaging model uses the surround area to a missing reading and averages those values to obtain an approximation to what should be in the center of those readings. The downside of this model is that it is reliant upon the immediate surrounding data and thus if that data is also missing then the model is inapplicable or less accurate.

Given a missing point at the center of a 3x3 matrix, the available surrounding values are averaged, weighted equally, to generate the missing center value.

3. Experimental Results and Discussion

3.1 ANN Implementation

Our ANN model's primary purpose is extrapolation and interpolation by means of function approximation, and is thus a regression-type model. A separate model was dedicated to each vector component (i.e. *u* or *v*) and therefore two separate models were trained to obtain final current-vector approximations. After numerous configurations of the model underwent trials, the present configuration was determined to offer optimal accuracy.

To allow verification of the model's accuracy, all available input vectors with known outputs were place into one of two sets. The first, denoted the training set, was used by the model's backpropagation algorithm to train the weights of the synapses to provide accurate results. The second set, known as the validation set, was initially withheld from the model during the training process but subsequently used to obtain all reported results. The data was collected from a week long period of hourly readings measured by coastal radar stations and sensory buoys covering the Gulf of Maine [8]. Only five-thousand valid input vectors, in our case three consecutive readings, were obtained to constitute the two data sets. Roughly two-thirds of the data were used for training while the remaining third were dedicated to validation.

The model consisted of three layers, namely: an input layer, one layer of hidden nodes, and an output layer. The input layer consists of four nodes which use the three prior values at the particular point which is being extrapolated and the forth node is a bias node with a constant input of one. Through experimentation, it was observed that eight hidden nodes produce the lowest average error in the models. The sigmoid activation function used within the nodes was the hyperbolic tangent which is a standard choice and works well for most applications of the ANN model.

The learning rate utilized was determined through experimentation and is dynamic as the backpropagation algorithm proceeds. Initially, the learning rate is relatively large and decreases as the training progresses. This allows the model to quickly become accurate at the beginning and adjust in greater precision further into training. The final learning rate function consists of a six-step step-function with each step comprising of nine iterations of the backpropagation algorithm.

During initialization the weights of every synapses are randomly generated in the interval of [-1, 1]. After fifty iterations of the back propagation algorithm were performed on the training data set, the weights reverted to produce the lowest average error for the model.

3.2 Model Comparisons

The linear model is intuitive and easy to implement. Although the linear model does not necessarily provide an accurate representation of a nonlinear function, it provides a good baseline for comparison against more sophisticated models.

The persistence model is another model that is easily implemented and provides relatively accurate approximations. In most cases the persistence model is capable of outperforming the linear model assuming a suitably small temporal resolution. Consider Figure **x** which clearly shows the persistence approximation is closer than the linear one in a particular instance.

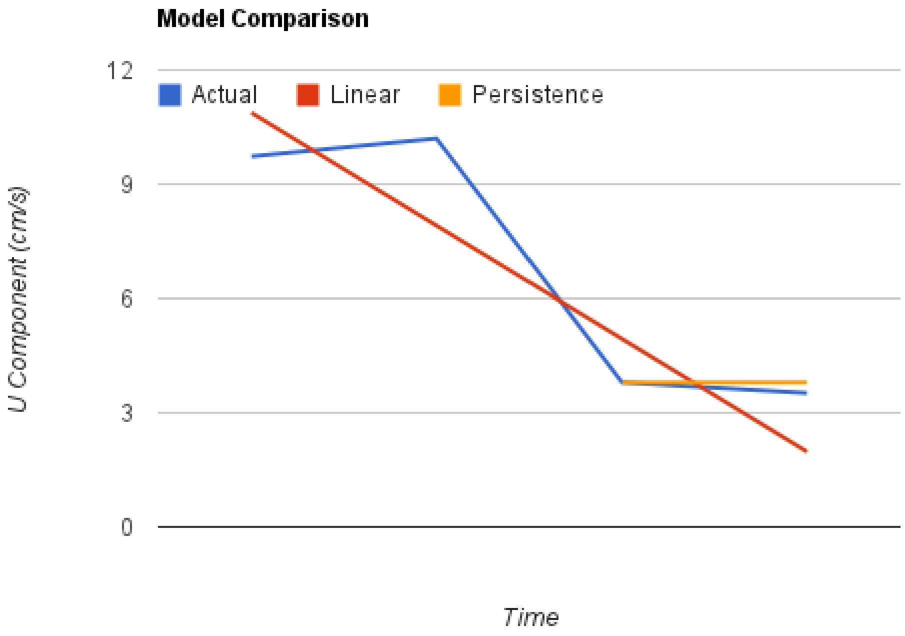

Figure 3 : Linear & Persistence Model Extrapolation

Each surface ocean current reading consists of two component vectors, *u* and *v*. These component vectors are summed to obtain the actual current-vector. Approximations are similarly generated as component vectors that produce our predicted current-vector. To obtain the error vector, the vector difference between the actual current-vector and the predicted current-vector is taken. The magnitude of this error vector is taken. The root mean square of the magnitudes of each model's error vectors is represented in the following figure.

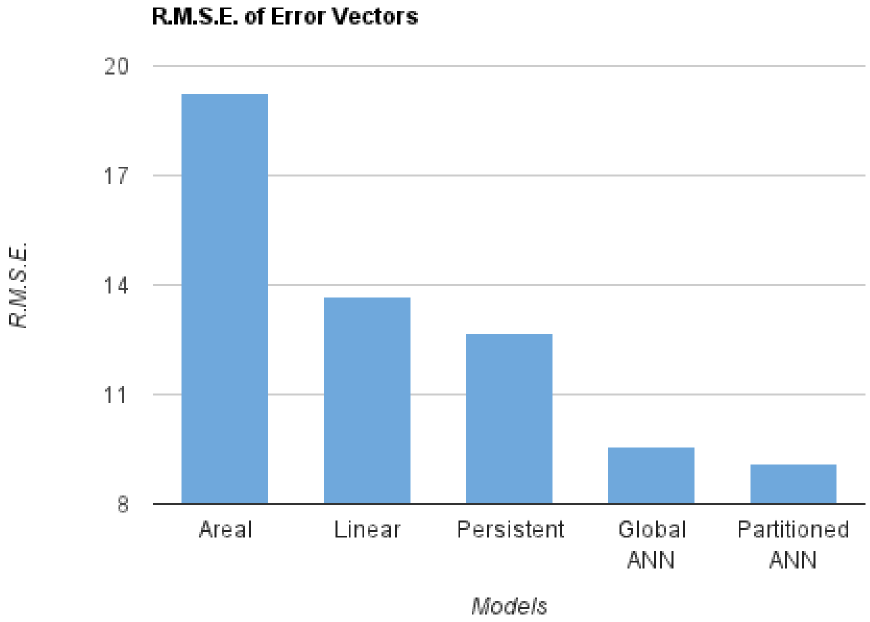

Figure 4 : R.M.S.E. Model Comparison

As the figure 4 demonstrates, the areal model clearly exhibits the worst performance among the chosen approaches. Despite the magnitude of the error, it is important to keep in mind that this result would often be considered an acceptable approximation for the missing data point. In comparison, the linear model can be seen to exceed the areal performance, to the point of approaching the effectiveness of the persistence model.

It is interesting to note that partitioned ANN model does not seem to have any additional improvement over the global ANN model. It does stand to reason that a nonlinear model should have surpassed linear models performance; its reassuring however to observe that the ANN has also outperformed the persistence model, as the latter's relatively simple approach tends to deliver impressive performance (on paper, at least), however ultimately unsatisfactory from a practical standpoint.

4. Conclusion

There are many ways to populate missing observations from a set of data. Some methods use intuitive approaches (such as averaging the surrounding area's data or using a prior value) but for nonlinear phenomena these methods tend to have relatively high error due to an insufficient number of degrees of freedom, or simply not being a good fit for the function being modeled. The Artificial Neural Network allows for accurate data extrapolation when compared to other models and is low cost once the model has been trained. The adaptive pattern recognition that the neural network performs allows the model to account for the sinusoidal motion that tidal currents exhibit over the course of time. Many models cannot account for this behavior due to their linear nature, and nonlinear models are often too resource intensive for devices to perform remotely.

There are significant drawbacks to using an Artificial Neural Network. A major issue \with the model is that it is considered a *black box* [4] as it is difficult to determine the actual inner operations of a trained model, and thus prove its validity compared to alternate approaches. The ANN relies on an extensive and representative initial data set to train the model, and results cannot be generated until there are enough readings. Additionally, the number of nodes and resulting links results in a large amount of multiplication operations as input nodes and particularly as hidden nodes are added. Though these add additional degrees of freedom to the ANN’s operation, too many lead to over-fitting, yet little research exists to suggest how many nodes are enough or too many [5].

Overall, the results produced by the Artificial Neural Network were capable of outperforming other initial approaches to the problem. While other extrapolation models exist and may outperform the Artificial Neural Network, the nature of the Wireless Sensor Network problem requires that the approximations be accomplished with minimal energy expenditure, be it via transmission or computational. A well trained model may be sent to such devices and reduce the computational effort required, whereas alternate similarly-complex models would require the device to perform either extensive computation or receive/transmit data from neighboring sensor platforms, either of which would place an unacceptable drain on limited power reserves. Additionally, the devices may further refine the model by continuing training with data collected in situ, which allows computational model to self-modify, whereas static models would need to be periodically redistributed to sensor platforms.

5. Future Work

There are several interesting research avenues that remain unpursued due to time constraints. The model used was designed and built from the ground up and because of this a good portion of development time was dedicated to ensure that our model was correctly implemented and performing as expected. Having dedicated more time or resources might possibly have offered more accurate models or more conclusive results.

During the development phase, time was taken to modularize the model and allow for varying levels of complexity. An aspect that was left unimplemented was support for having multiple hidden layers and implementing such a model could possibly increase the accuracy of the model. Additional hidden layers can allow the model to recognize more complex patterns by having more weighted links to adjust during the back propagation.

An aspect of the model that was implemented allowed support for varying amounts of input nodes however all models created had only three input nodes due to the interface with the database. Increasing the number of inputs would allow for the model to recognize patterns over larger spans of time and adds the possibility of incorporating a greater variety of inputs such as geographical coordinates or derivative values. Correlations may be recognized by the model between the coordinates, time of day or other inputs that could potentially enhance the overall accuracy of the model [6]. A further reason an ANN model may be a particularly adept choice for multiple-predictor problems is that whereas a normal polynomial function's parameters grow exponentially with the number of predictors *p*, an ANN's parameters grow merely linearly with *p* [4]. Again, the time constraints of our research did not allow us to implement and form conclusive results with other inputs.

In addition to increasing the input nodes, increasing the number of output nodes could reduce the number of models by combining the u-components and v-components into a single model. Besides reducing the number of models, this may also enhance accuracy by modeling a relationship between the u and v components which is likely to exist.

The data set utilized for this project was collected over the span of a single week with hourly readings. This produced a data set with about twenty thousand entries but considering the magnitude of the region and time frame the data was collected in, the usable data was spread relatively thin. Testing our model on alternate data sets from regions other than the Gulf of Maine [2] for performance comparisons would provide more conclusive results in demonstrating the power and versatility of the model.

6 References

[1] Haykin S. (1998) Neural Networks: A Comprehensive Foundation. 2nd ed: Prentice Hall

[2] Kim S. Y., Cornuelle B. D., and Terrill E. J. . (2008) "Mapping Surface Currents from HF Radar Radial Velocity Measurements Using Optimal Interpolation." *American Geophysical Union*. Journal of Geophysical Research Oceans, 25 Oct. 2008. Web. 14 Jan. 2012. <http://www.agu.org/pubs/crossref/2008/2007JC004244.shtml>.

[3] Lee T.-L. (2004) Back-propagation neural network for long-term tidal predictions. Ocean Engineering 31, 225-238

[4] Marzban C. (2009) Basic Statistics and Basic AI: Neural Networks. In: Haupt S.E., Pasini A., Marzban C., editors. Artificial Intelligence Methods in the Environmental Sciences: Springer; 2009, p. 15-47

[5] Reed R.D., Marks R.J. (1998) Neural smithing: supervised learning in feedforward artificial neural networks. Cambridge, MA: MIT Press;

[6] Rigol J.P., Jarvis C.H., et al. (2001) Artificial neural networks as a tool for spatial interpolation. International Journal of Geographical Information Science 15(4), 323-343

[7] Rumelhart, D.E., Hinton G.E., et al. (1986) Learning Representations By Back-propagating Errors. Science 323(9), p. 533-536

[8] Pettigrew N.R., Roesler C.S., et al. (2008) An Operational Real-Time Ocean Sensor Network in the Gulf of Maine. In: Nittel S., Labrinidis A., Stefanidis A., editors. LNCS: Springer Berlin / Heidelberg; 2008, p. 213-238

[9] Zealand, C. M., D. H. Burn, et al. (1999). "Short term streamflow forecasting using artificial neural networks." Journal of Hydrology 214: p. 32-48.

Acknowledgements

The authors gratefully acknowledge Professor François Neville for all his advice and assistance in the development of this paper.