Summer 8-30-2015

Spatiotemporal Wireless Sensor Network Field Approximation with Multilayer Perceptron Artificial Neural Network Models

François Neville
University of Maine - Main, fneville@bemidjistate.edu

SPATIOTEMPORAL WIRELESS SENSOR NETWORK FIELD APPROXIMATION WITH MULTILAYER PERCEPTRON ARTIFICIAL NEURAL NETWORK MODELS

By

François Neville

B.S. Computer Science, Nebraska Wesleyan University, 1992

M.S. Computer Science, University of Nebraska-Lincoln, 1995

A DISSERTATION

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

(in Spatial Information Science and Engineering)

The Graduate School

The University of Maine

August 2015

Advisory Committee:

Kate Beard-Tisdale, Professor of Spatial Informatics, Advisor

Neal Pettigrew, Professor of Oceanography

Max Egenhofer, Professor of Spatial Informatics

Reinhard Moratz, Associate Professor of Spatial Informatics

Phillippe Tissot, Associate Professor of Physics

DISSERTATION ACCEPTANCE STATEMENT

On behalf of the Graduate Committee for Francois Neville I affirm that this manuscript is the final and accepted dissertation. Signatures of all committee members are on file with the Graduate School at the University of Maine, 42 Stodder Hall, Orono, Maine.

Dr. Kate Beard-Tisdale, Professor of Spatial Informatics Date

LIBRARY RIGHTS STATEMENT

In presenting this dissertation in partial fulfillment of the requirements for an advanced degree at The University of Maine, I agree that the Library shall make it freely available for inspection. I further agree that permission for "fair use" copying of this thesis for scholarly purposes may be granted by the Librarian. It is understood that any copying or publication of this dissertation for financial gain shall not be allowed without my written permission.

Signature:

Date:

SPATIOTEMPORAL WIRELESS SENSOR NETWORK FIELD APPROXIMATION WITH MULTILAYER PERCEPTRON ARTIFICIAL NEURAL NETWORK MODELS

By François Neville

Dissertation advisor: Dr. Kate Beard-Tisdale

An Abstract of the Dissertation Presented
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy
(in Spatial Information Science and Engineering)
August 2015

As sensors become increasingly compact and dependable in natural environments, spatially-distributed heterogeneous sensor network systems steadily become more pervasive. However, any environmental monitoring system must account for potential data loss due to a variety of natural and technological causes. Modeling a natural spatial region can be problematic due to spatial nonstationarities in environmental variables, and as particular regions may be subject to specific influences at different spatial scales. Relationships between processes within these regions are often ephemeral, so models designed to represent them cannot remain static. Integrating temporal factors into this model engenders further complexity.

This dissertation evaluates the use of multilayer perceptron neural network models in the context of sensor networks as a possible solution to many of these problems given their data-driven nature, their representational flexibility and straightforward fitting process. The relative importance of parameters is determined via an adaptive backpropagation training process, which converges to a best-fit model for

sensing platforms to validate collected data or approximate missing readings. As conditions evolve over time such that the model can no longer adapt to changes, new models are trained to replace the old.

We demonstrate accuracy results for the MLP generally on par with those of spatial kriging, but able to integrate additional physical and temporal parameters, enabling its application to any region with a collection of available data streams. Potential uses of this model might be not only to approximate missing data in the sensor field, but also to flag potentially incorrect, unusual or atypical data returned by the sensor network. Given the potential for spatial heterogeneity in a monitored phenomenon, this dissertation further explores the benefits of partitioning a space and applying individual MLP models to these partitions. A system of neural models using both spatial and temporal parameters can be envisioned such that a spatiotemporal space partitioned by k-means is modeled by *k* neural models with internal weightings varying individually according to the dominant processes within the assigned region of each. Evaluated on simulated and real data on surface currents of the Gulf of Maine, partitioned models show significant improved results over single global models.

ACKNOWLEDGEMENTS

This manuscript has taken (too) many years to come to fruition. But of course no creative work of any consequence is ever truly accomplished alone. I have received assistance and encouragement from many sources over the years, and am grateful for this opportunity to thank some of them.

Before all else I would like to extend my heartfelt gratitude to my thesis advisor Dr. Kate Beard-Tisdale for her years of generous financial support, and who mitigated my worst writing throughout this dissertation, enhanced my best, and proved a font of constant support and wise counsel throughout it all. Truly her influence can be felt through every chapter of not only this work, but my graduate career entire.

My sincere thanks go also to Dr. Neal R. Pettigrew, who provided me not only my initial financial support, but also with the opportunity to study applications of Artificial Neural Networks in real-world physical test environments, as well as the data with which to do so. It was through his original project to apply ANNs to buoy and CODAR data to fill spatial and temporal holes in acquired data that this thesis was first conceived.

Dr. Max Egenhofer, for teaching me to be a good graduate student and researcher; who encouraged me to read widely, and write often -- a wise prescription followed imperfectly, but ultimately (I hope!) successfully.

Dr. Reinhard Moratz, for the sharing of his time and expertise, and for being available when it mattered most.

And Dr. Phillippe Tissot, for generously befriending me at my first national conference, and for providing guidance and encouragement from afar ever since.

To the members of Dr Pettigrew's research Physical Oceanography Group (PhOG) past and present: Karl Schlenker, who took me out to see my first CODAR installation in person; Carol Janzen inspired me to think spatially as well as temporally, ultimately leading me to my home department on campus; John Wallinga, who kindly proofed my early simulations of buoy data; Neil Fischer, for providing me with reams of CODAR data whenever needed; Bob Fleming first turned me onto the Python language, which I use regularly to this day (thanks again Bob!); and Linda Mangum, who always knew where to direct me for any information I needed; and failing that, made available plentiful golden furry bundles of stress-relief as required.

I received the support of several crucial federal and state research grants, including the following:

NSF Grant# DGE-0504494 (SSEI IGERT)

NGIA SmartMaps Grant #NMA201-01-2003

Maine Economic Improvement Fellowship (MEIF), University of Maine Graduate School, 2010

To my colleagues at Bemidji State University, who encouraged me relentlessly to finally finish my dissertation: Dr. Marty Wolf, Christopher Brown, Dr. Derek Webb,

Dr. Randy Westhoff, Dr. Eric Lund, Dr. Glen Richgels, Dr. Colleen Livingston, Dr. Todd Frauenholz, Dr. Heidi Hansen – and Dr. Colleen Greer.

To my mentor, Dr. Dan Kaiser, who encouraged me to consider teaching as a career, and introduced me to the first, best, and most-fulfilling job I had ever had up to that point.

To my parents, Dan and Micheline Neville, who have long awaited this moment. *C'est vrai, c'est bien fini maintenant!*

To my amazing wife, Melinda Otto Neville, without whom I would be lost, and none of this would have been possible.

And to anybody I may have forgotten, my sincere abject apologies.

François Neville

August, 2015

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

LIST OF TABLES xii

LIST OF FIGURES xiii

CHAPTER 1: SPATIOTEMPORAL FIELD INTERPOLATION 1

1.1 Introduction 1

1.1.1 Properties of Spatial Fields 2

1.1.2 Sensor System Data as Spatial Fields 3

1.2 Formal Definitions 6

1.3 Problem Definition: Sensor Platform-based Interpolation in Time and Space 8

1.4 Proposed Model for the Spatiotemporal Interpolation Task 11

1.4.1 Proposed Approach 13

1.4.2 Discussion 15

1.4.3 Qualifying Assumptions 17

1.5 Hypothesis 20

1.6 Key Research Questions 21

1.7 Scope of Thesis 22

1.8 Research Approach 22

1.9 Expected Outcomes 22

1.10 Intended Audience 23

1.11 Thesis Organization 23

CHAPTER 2: LITERATURE REVIEW 24

2.1 Introduction 24

2.2 Traditional Approaches to the Spatial Interpolation Problem 25
2.2.1 Brief Overview of Spatiotemporal Extensions to the Kriging Model 27
2.2.2 Summary of Advantages and Disadvantages of Explicit Spatiotemporal Trend Modeling 30
2.3 The Multilayer Perceptron Approach 31
2.3.1 Model Structure 32
2.3.2 Backpropagation Training 33
2.3.3 MLP Model Benefits 33
2.3.4 MLP Weaknesses 37
2.3.5 MLP Models Used for Spatial Interpolation 40
2.4 Addressing Nonstationarity 41
2.4.1 Partitioning Techniques 42
2.4.1.1 The k-means Algorithm 44
2.4.1.2 Monte Carlo BestK-Approximation (MCBestK) 47
2.4.2 Assessing Model Inputs 48
2.4.3 Classification and Regression Trees 49
2.5 Uncertainty Estimates for Spatiotemporal Models 52
CHAPTER 3: MULTILAYER PERCEPTRON FIELD INTERPOLATION OF SPATIOTEMPORAL DATA 54
3.1 Introduction 54
3.2 Model Data – Ocean Surface Current Measurements 57
3.3 Methodology 58
3.3.1 Radial Component Approximation 58

3.3.2 Delineating the Spatial Interpolation Space by Convex Hull 60
3.3.3 Velocity Vector Approximation 60
3.3.4 Quantitative Measures 61
3.4 Model Configuration 62
3.4.1 The Multilayer Perceptron Model 62
3.4.2 MLP Model Specification 63
3.4.3 The Ordinary Kriging Model 66
3.4.4 The Temporal Persistence Model 67
3.5 Results and Discussion 68
3.6 Conclusions 73
CHAPTER 4: EVALUATING MLP MODELS ON PARTITIONED SYNTHETIC DATA 74
4.1 Introduction 74
4.2 Methodology 75
4.2.1 Application Context 75
4.2.2 The K-Problem: How Many Partitions? 78
4.3 Synthetic Datasets 81
4.3.1 Synthetic Orthogonal Testbed 82
4.3.1.1 Design of Phenomenological Regimes 82
4.3.1.2 Stochastic Signal Perturbation 83
4.4 Synthetic Orthogonal Results 90
4.5 Orthogonal Dataset Conclusions 105
4.6 Synthetic “Natural” Dataset 106

4.6.1 Creating the Synthetic "Natural" Testbed 106
4.7 MLP Design and Feature Relevance 120
4.8 Conclusions 122
CHAPTER 5: MLP PERFORMANCE ON NATURAL DATA SET FROM THE GULF OF MAINE 124
5.1 Introduction 124
5.2 Experimental Setup 126
5.2.1 Gulf of Maine Dataset 126
5.2.2 Proposed WSN Operation 126
5.2.3 Determining K 127
5.2.4 Partitioned MLP Simulation Results 139
5.2.5 Gulf of Maine Dataset Feature Relevance 142
5.3 Conclusion and Subsequent Work 145
5.3.1 Conclusions 145
5.3.2 Subsequent Work 146
CHAPTER 6: EVALUATION OF AUTOCORRELATED RESIDUALS, AND REFRESHING THE SYSTEM 147
6.1 Introduction 147
6.2 Spatial Autocorrelation of the Residual Fields 149
6.2.1 Rationale for Excluding Edges of the I-Field 150
6.2.2 Comparison of Moran's I in the U/V fields between Global and Partitioned Models 151
6.2.3 Determining the Change in Mean Autocorrelation Between Models 154

6.2.4 Excluding the Possibility of a ST-Leviathan Volume in Residual Fields 161
6.2.5 Maximum Significant I Persistence 162
6.3 Determining Refresh-Timing 166
6.4 Comparing Populations of Readings for Automated Refresh-Timing 167
CHAPTER 7: CONCLUSIONS AND FUTURE WORK 171
7.1 Conclusions 171
7.1.1 Model of Choice: MLP 171
7.1.2 Research Questions 172
7.2 Future Work 173
REFERENCES 177
APPENDIX A: MONTE CARLO BESTK MATLAB CODE 186
APPENDIX B: SYNTHETIC DATA GENERATIVE CODE 189
BIOGRAPHY OF THE AUTHOR 192

LIST OF TABLES

Table 2.1: General Backpropagation algorithm 33

Table 3.1: Overall performance comparison against the TPM model for a 36-hour test period in June 2005 68

Table 4.1: Results of Single model error vs. Partitioned model error on Orthogonal Dataset 91

Table 4.2: Sample readout of Hypothesis test results 92

Table 4.3: Normalizing the axes to approximate BestK in Figure 4.12 96

Table 4.4: Cluster centers resulting from kmeans processing of second synthetic dataset for K=5 111

Table 4.5: Results of Single model vs. Partitioned model 118

Table 5.1: Cluster centers resulting from non-Z-scored k-means processing 138

Table 5.2: Non-Zscored Partitioning Scheme Simulation Results 140

Table 6.1: Paired t-test results for April 2014 Moran's I fields as produced by Global and Partitioned models, aggregated both monthly and daily by location 156

LIST OF FIGURES

Figure 1.1: A conceptual partitioning of the Gulf of Maine testbed including local datasources 16

Figure 2.1: Perceptron model, with illustration of processing sequence in output nodes 32

Figure 2.2: Notional contour map of an error function over a spatial region 53

Figure 3.1: Map of Gulf of Maine CODAR testbed region with active sites labeled..... 58

Figure 3.2: Data preprocessing steps involved in this MLP interpolation..... 65

Figure 3.3: a) Gulf-wide Root Mean Square Error of TPM, MLP and OK models compared to b) Gulf-wide surface current variance magnitude (and component variances) 70

Figure 3.4: Apparent regionalization in variance maps of *u* and *v* component velocity data of monitored region *M*..... 72

Figure 4.1: Proposed process-chart for proposed CatSTANN framework for partitioned-region MLP model applications 76

Figure 4.2: Illustration of phenomenological regimes induced into synthetic datasets 83

Figure 4.3: Illustration of intracluster sum of squared error vs. K as generated by (Peeples 2011) 84

Figure 4.4: Intracluster sum of squared error vs. K as generated by (Peeples 2011) on a logarithmic scale 85

Figure 4.5: Result of a short-term k-means partitioning of the dataset with K=4 86

Figure 4.6: Synthetic Orthogonal dataset clustering at time-steps 1, 21, 41, 61 87

Figure 4.7: Time-series of cluster populations over time 88

Figure 4.8: TreeBagging results of dominant terms in orthogonal dataset's K=4 clustering solution 90

Figure 4.9: Spatial distribution of mean error-reduction resulting from use of partitioned model 93

Figure 4.10: Non-normal histogram of partitioned-model's mean, per-location error reduction 93

Figure 4.11: Non-normal distribution of partitioned-model's mean, per-location error-reduction 94

Figure 4.12: Spatial-temporal WCSS vs. Number of Clusters 95

Figure 4.13: Inequality plot of: $f(x) < (d + 0.5)$ 99

Figure 4.14: Root Mean Square Error and Effort-ratio as K increases 101

Figure 4.15: Example of change in K 102

Figure 4.16: a) Time-series of instantaneous Best-K determinations, and
b) the $Mode_{20}$-smoothed series 103

Figure 4.17: *Mean*- and *Mode-Smoothed* Best-K error-rates..... 104

Figure 4.18: Snapshot of regimes in region *M* at time t=0 106

Figure 4.19: Snapshots of the progression of the green-circle partition over time..... 108

Figure 4.20: Feature-weights of the K=5 clustering solution as determined by
random-forest tree-bagging..... 109

Figure 4.21: 2D and 3D representations of kmean cluster centers, and feature-
contributions to instance classification 110

Figure 4.22: View of 5-part membership before and after critical point t=400..... 113

Figure 4.23: Result of MCBestK heuristic, with normalized mean slope-
approximation techniques 114

Figure 4.24: WCSS solutions for Z-scored data for varying number of clusters K..... 116

Figure 4.25: Actual cluster self-identifications over a selected time-period of the
NP simulation..... 116

Figure 4.26: Histogram of differences between results of Single model, and
Partitioned (K=5) model 119

Figure 4.27: Normality plot for paired-differences histogram in Figure 4.26 119

Figure 4.28: A typical boxplot of Z-scored Partitioned model comparison for naturalistic synthetic dataset 120

Figure 4.29: Relative feature-weight valuations of Orthogonal synthetic dataset for a) Global model, and b) Partitioned model 121

Figure 4.30: Relative feature-weight valuations of "Natural" synthetic dataset for a) Global model, and b) Partitioned model 122

Figure 5.1: Monte Carlo results of raw Codar data compared to 100 randomly permuted versions of dataset 127

Figure 5.2: Parameter relevance results of two clustering schemes of one month of CODAR data 130

Figure 5.3: Random partitioning of data instance vectors in the Gulf of Maine taken over three consecutive hours (3/06/14 17h00 – 19h00 UTC) 130

Figure 5.4: Total vectors remotely sensed in GoM on 03/06/2014 at 17h00 and 21h00 UTC (PhOG, 2014) 132

Figure 5.5: Z-scored partitioning scheme over three consecutive hours (3/06/14 17h00 - 19h00 UTC) 133

Figure 5.6: Population of Z-scored clusters over five days' time 134

Figure 5.7: Non-Z-scored partitioning scheme over same three consecutive hours (3/06/14 17h00 – 19h00) 135

Figure 5.8: Cluster populations of non-Z-scored clusters over five days' time 135

Figure 5.9: Snapshot of reversing tide in Gulf of Maine at 19h00 UTC 137

Figure 5.10 : Cluster-centers and attribute influence displayed in 2D and 3D digraphs 138

Figure 5.11: Boxplot of (non-Zscored) Partitioned model error comparison for CODAR testbed data 141

Figure 5.12: Histogram of paired differences of results from the Single and 5-Partitioned (non-Zscored) model, and the normality plot of said histogram 141

Figure 5.13: Relative feature-weight valuations of 2013-2014 Gulf of Maine dataset for a) Global model, and b) Partitioned model 143

Figure 5.14: Relative feature-weight valuations of 2013-2014 Gulf of Maine dataset for a) Global model, and b) Partitioned model, including spatial attributes latitude and longitude 144

Figure 6.1: Three views of averaged V-component fields on March 20, 2014 – residual V-component values, significantly correlated V-component Z-scores, and I-field of V-comp autocorrelations 149

Figure 6.2: Comparison of the U-component's Z-fields and I-fields resulting from the averaged results of the Global mode and Partitioned model on March 20, 2014 152

Figure 6.3: Comparison of the V-component's Z-fields and I-fields resulting from the averaged results of the Global model and Partitioned model on March 20, 2014 153

Figure 6.4: Histograms for the differences of paired Moran's I values (PART – GLOBAL) produced by U-component, V-component and total residual magnitude for the month of April 2014 155

Figure 6.5: Relative differences in I-values of U-component residuals between Global and Partitioned models 158

Figure 6.6: Relative differences in I-values of V-component residuals between Global and Partitioned models 159

Figure 6.7: Relative differences in I-values of Total Magnitude residuals between Global and Partitioned models 160

Figure 6.8: Global model MSIP images of the U-component and V-component at 0.01 level of significance for April 2014 163

Figure 6.9: MSIP values for U-component error, V-component error, and residual magnitude from Partitioned model results for April 2014 164

Figure 6.10: Seasonality in error is exhibited by an MLP model fitted to data collected by GoMOOS buoy I in the Gulf of Maine over first half of 2004, 2005 166

Figure 6.11: Series of p-values resulting from paired-t-tests over 7-day populations of collected readings 169

Figure 6.12: Series of p-values resulting from paired t-tests over mean-aggregated populations of collected readings..... 170

CHAPTER 1

SPATIOTEMPORAL FIELD INTERPOLATION

1.1 Introduction

One of the most well-known and long-standing dichotomies in the discipline of spatial science is one of model representation: whether geographic phenomena are best interpreted as objects or fields (Couclelis 1992; Goodchild 1992; Camara *et al.* 1994). The choice between these two alternative approaches comes down to balancing the various needs of conceptualization and perspective with context and scale. A mountain range may be conceptualized in a computer model as an object whose height at any point along its length creates a rain-shadow upon adjacent space. The same mountain range may also be rendered as a field of elevation values within a digital elevation model. As most if not all geographic phenomena can be expressed through either conceptual perspective (Peuquet et al. 1999), the context for which the representation is needed will generally determine which approach is employed. Scale of the phenomena may also play a role in the modeler's choice of computer implementation. The concepts of cognitive typology (Zubin 1989) relate scale and perspective interactions to define the human experience of cognitive geography. Entities which are perceived to occupy spaces smaller than the human body inhabit A-space; while B-space entities appear to human cognition to be scaled-up versions of A-spaces. Entities in C-space can only be apprehended partially, as from a vantage point; and an entity in D-space (e.g., the solar system) is likely outside direct human experience, and only deduced from a collection of

clues, readings, and other pieces of knowledge we are able to collate. This categorization of spaces defines a continuum which seemingly links the two extremes of human conceptual perspective: entities in A- and B-space appear to be more amenable to a discrete object-modeling approach, while C- and D-space entities tend towards a more field-like representation.

1.1.1 Properties of Spatial Fields

A spatial field is often defined as a functional mapping for a given instant in time from a collection of locations in space to a domain of attribute values (Worboys and Duckham 2004). For example a gridded topographic map is a discretized representation of a continuous theoretical field of elevation values for a given terrain. The field is represented by a continuous function relating each pair of grid coordinates *(x,y)* to some elevation attribute *z* at that location. Fields may similarly be defined for any other attribute that can be measured over the monitored space. Examples of fields include air temperature, land cover type, change in yearly average rainfall, or wind magnitude and direction. From these examples we can observe that fields will take on the characteristics of their component measurements. They may be composed of single-valued readings (temperature), or vector-valued (wind direction, magnitude); they may be continuous (elevation) or discrete (cover type). They may exist at any of the Stevens scales of measure: categorical, ordinal, interval, and ratio (Stevens 1946).

From the examples above, it is clear that the measurements composing the fields can vary at different temporal scales. A field of elevation values, for example, might be considered for most practical purposes to be time-invariant; we would not expect any

substantive change in the field in shorter than geologic timescales (with the possible exception of active volcanic regions). Whereas measurements of temperature or wind-speed are in constant flux, and can be expected to change significantly over short time periods. Spatiotemporal fields must incorporate temporal changes as well as spatial location into their functional mapping.

Finally, although fields of measurements taken of continuous phenomena are in theory unlimited, they must of necessity frequently be bounded in practice. For example, the field of elevation values on a topographic map is limited by the edges of the printed sheet. Similarly, a field of values collected by a remote sensor array or an in-situ network of sensor nodes will be limited by the placement and effective range of the individual contributing sensing stations. The continuous representation of a field is realized by a function encompassing all of its discretely-sampled readings. Since such a function cannot be appropriately constrained outside the effective range of its sensing source(s), the natural spatial bound of the field might thus be set to the *convex hull*, or perimeter of the range of the contributing sensor nodes.

1.1.2 Sensor System Data as Spatial Fields

As systems of sensors are increasingly utilized for the task of regular, long-term geographic observation, geographic entities and physical processes are increasingly being interpreted from the sensor's point of view in discrete bits and pieces. From this discretized view of entities and processes, there is frequently the need to generate a more complete representation of the underlying spatiotemporal field, for example, the

representation of the 15-minute temperature, precipitation, or wind field from a set of widely spaced weather stations measuring these variables on the hour.

Remote-sensing systems typically create spatial fields of readings in the course of capturing their data. Systems such as satellite, radar, and wireless sensor networks are the source of a growing collection of spatiotemporal data sets. A data set generally only represents discrete readings, however. As a sensor system's resolution is finite, the continuous structure of a phenomenon is necessarily discretized as it is sampled, geolocated, time-stamped, and finally stored. The density of the observations, as well as their spatial and temporal regularity, has implications regarding the accuracy of the function which is finally determined to define the field. A sparsely-sampled field of values creates greater uncertainty between sampled locations, especially in highly-variable regions of the field. Missing data in the field has much the same effect, as the local density of readings is reduced. The occurrence of regionalized clusters of missing data exacerbates this effect, as the lack of nearby readings to constrain the field's associated function at a location will cause higher degrees of uncertainty.

Gaps in the data record are problematic for accurate scientific data analysis and can take on additional importance in the face of accidents or other serious events having a potential to lead to economic or ecological destruction. In the event of an oil spill or a hazardous red tide algal bloom at sea, accurate readings of surface currents must be known to track the progress of contaminated waters in order to protect public and/or environmental health.

Frequently the results of disparate readings can be combined in order to indirectly arrive at a result that cannot be sensed directly. A coastal weather radar system may not

be primarily designed to detect fishing vessels, for instance, but may be able nonetheless to deduce their locations and track them over time from the perturbations the vessels create in the generated radar field. Fields of discrete readings are central to the manner in which our electronic sensing tools experience and communicate back to us their perception of the world, thus the matter of missing readings can become significant. For example, a contributing factor to the 2004 Indian Ocean tsunami death toll is attributed to the lack of effectively functioning tsunami early-warning system sensors along a vulnerable and densely populated coastline. This was the motivation for the subsequent development of the Indian Ocean Tsunami Warning System (UNDG 2009).

Natural environments are replete with phenomena that interfere with the gathering of sensor readings: clouds regularly obscure satellite photogrammetry, ionospheric interference in the earth's atmosphere disturbs radio and radar signals; and the growth of plant matter over time or other gradual environmental change may come to obscure or block a sensor's view of a desired phenomenon. Beyond the list of potential natural causes of data loss, a host of potential technical failures must also be considered. Natural environments are notoriously corrosive to the electronic components on which most sensors are based, requiring environmental hardening of a sensor node without compromising its ability to gather high-quality readings. Sensors embedded in the natural environment have limited access to power. They often draw power from batteries, which may incur a high cost in the form of regular maintenance, and impose limitations on sensor operation in terms of frequency of readings taken, extent of spatial coverage, deployment length, and costs of communicating collected readings back to the data store. An array of sensors is not likely to have access to either a directly-connected

communication network or a reliable, unlimited on-demand power source. These challenges create the need for an efficient, tractable and lightweight means of field representation on the sensing platform itself. For if the field and its dynamics could be simply encoded to the sensor node collecting the readings, then approximations of missing readings might be reconstituted as they occur. Implementing field approximations presumes some storage and computational capacity on the part of the sensor node; but given the continual upgrades to computational capacity and performance of electronic sensor platforms, the assumption is justified (Akyildiz, Melodia *et al*. 2007) (Akyildiz, Su *et al*. 2007).

This dissertation will endeavor to describe a framework within the wireless sensor network context that will allow the individual network node to provide approximations for missing readings in the field under observation as they occur. As it suggests the use of one or more artificial neural network models to do so, and categorized by a clustering method, its proposed name is CatSTANN, for Categorized Spatial/Temporal Artificial Neural Network.

1.2 Formal Definitions

Clarification of some of the basic concepts underlying this thesis with some formal definitions follows. Throughout this thesis when reference is made to *space* and *time*, these should respectively be taken unless otherwise noted as *geographic space*, and *experiential time* as measured in instants or standard intervals such as seconds or minutes.

A sensor-monitored region *M* is a subregion of geographic space *S*. Although geographic space is three-dimensional, we shall assume throughout that only one sensor reading of a given attribute will ever be taken at any (globally-available) time *t* for a given location $(x, y) \in M$, and shall thus constrict both *M* and *S* to two-dimensional Euclidean space, or more formally:

$$M \subset S \subset R^2 \tag{1.1}$$

Given a geographic space *S* (consisting of all individual geolocations *s*), and a class of scalar values *V*, a spatial field is a function *f* whose domain is *S* and codomain is *V* (Worboys and Duckham, 2006).

Definition 1 A *spatial* field is defined as follows:

$$f : R^2 \rightarrow V \tag{1.2}$$

A spatiotemporal field incorporates the concept of time to separate subsequent states of *f* along the temporal domain *T*.

Definition 2 A *spatiotemporal* field *F* is defined as follows:

$$F : M \times T \rightarrow V \tag{1.3}$$

In general, references to fields will refer specifically to spatiotemporal fields. Although as mentioned earlier both *f* or *F* can be either continuous or discrete and exist within any of the Stevens scale of measurement categories, we restrict *F* in this thesis to spatiotemporally continuous functions, yielding either interval or ratio values. Depending on the attribute field described, these values may be scalar or vector. For simplicity, for each value sensed at a space-time point (*s*, *t*) in *M* we will define the field

F's value as *F*(*s*, *t*). For generic reference to a sensor node location *s* in *S* from which a reading may be taken, reference shall be made to *F*(*s*).

One focus of this thesis will involve the partitioning of *F* into *n* partitions P_i. Each partition P_i. is thus a subregion of *F*, and *F* is wholly represented by the collection of its subcomponent regions P_i. Formally:

$$P_i \subset F \tag{1.4}$$

and

$$\bigcup_{i=1}^{n} P_i = F \tag{1.5}$$

Individual models representing fields over a single region P_i will generally be referred to as *local* models, while a single model representing *F* as a whole will be termed *global*. The composite of local models representing *F* will be termed the *partitioned* model.

A sensor node, or simply *node*, will refer to any stationary sensor platform containing sensors capable of taking reading(s) at a location *s* of one or more field attributes. We assume no more than one reading per attribute will be generated per location *s* at any time *t*. The collection of nodes as a whole will comprise a sensor network, which will be presumed a wireless sensor network (WSN) unless stated otherwise.

1.3 Problem Definition: Sensor Platform-based Interpolation in Time and Space

As sensor networks and/or arrays are increasingly deployed in internal or external, natural or constructed environments, the vision of an instrumented world draws ever closer to reality (Estrin, 2002; Akyildiz, 2007; Nittel, 2009). However, natural

environments especially are often corrosive and generally inhospitable to the electronic components of sensor monitoring systems. When combined with potential technical issues pertinent to monitoring systems, environmental conditions will often lead to intermittent data loss. Although such data are often visualized as is, gaps in the data record make analysis difficult, thus an interpolation technique must be chosen to complete the record with synthetic values.

Viewing an environment through the sensor's point of view is much like navigating D-space (Zubin, 1989). A spatial field can be decomposed into two parts: a mean function representing large scale spatial variation – or first order effects – and small scale spatial variation representing autocorrelation in deviations from the mean – or second order effects. Discretized readings reveal hints of confounded first- and second-order effects in spatial phenomena occurring within the sensor's immediate neighborhood, but with little other indication of the global (or first-order) picture. Attributes or features in an observed field *F* may not be detectable by a single sensor type, but may instead require the combination of readings from several different sensors (Abadi, 2005). Detecting the occurrence of the live birth of mice in a research laboratory breeding cage might, for example, require both temperature and humidity sensor readings at a minimum in order to isolate the birth event. Should naturally-occurring environmental interference or some other intermittent technical issue materialize that prevents data readings from being taken, uncertainty levels in that region of the field may increase, possibly affecting spatially adjacent or temporally subsequent readings if data are shared within the sensor node or between adjacent network nodes.

When interpolating for missing values in a spatiotemporal field, time must be taken into account in conjunction with whatever local spatial predictors are available. However, the inclusion of time into the spatial interpolation problem can introduce several complexities in the construction of spatiotemporal models. First among these is that standard units of time are not directly comparable with standard units of space. For problems where field variability is mostly concentrated on the spatial, rather than temporal domain (such as for digital elevation models), accurate approximations for missing values can be made at any point in time using a strictly-spatial interpolation method such as kriging, or inverse-distance weighting. However any field phenomena that derives a significant portion of its variability from the time domain will require some customization or extension to traditional spatial models (Kyriakidis and Journel 1999; Gao and Revesz 2006). As spatiotemporal phenomena occur on varying spatial and temporal scales, such customizations may result in models that are difficult to compare. Some extensions require the generation of multiple covariance matrices to capture interactions between different locations in space and time (Kyriakidis and Journel 2001), while other approaches incorporate such relations with shape functions (Li and Revesz 2004). However, interactions and dependencies within field phenomena may evolve over time, requiring all matrices and shape functions to be periodically reanalyzed and recalibrated. The difficulties of spatiotemporal field interpolation are exacerbated when implemented on a distributed network of sensors. The periodic reanalysis process described above may require wide-ranging data access involving significant communication costs, extensive computation, and intensive draw on a strictly-limited

local power source, all primary limiting factors in independent *in situ* sensor node deployments. (Duckham *et al*. 2005) (De La Piedra *et al*. 2013).

Simplifying Assumptions

The task of spatiotemporal interpolation can be visualized in several ways, depending on the temporal point of reference. For this thesis, the following simplifying assumptions apply:

- spatiotemporal interpolation takes place at the most recently logged time-instant t_i, before any values $f(t_{i+1}, p)$ are known. Thus all available data for the interpolation of $f(t, p)$have taken place at or prior to t.
- we assume the interval Δt between readings to be regular and constant.
- although in practical real-world situations, sensor readings logged as taken at a time t are actually taken within the range $t \pm \Delta tt$ ($\Delta tt << \Delta t$), we shall consider all such readings to be taken at the closest integer-value t.
- we restrict our focus solely to spatial field-based models to the exclusion of the alternate spatial object-based modeling approach.

1.4 Proposed Model for the Spatiotemporal Interpolation Task

Given our continuous function-based modeling approach to fields, we require a method to approximate smooth, continuous functions that take into consideration inputs from previous points in time as well as nearby locations in space. Previous approaches to this problem are more fully described in Chapter 2. They include techniques such as the generation of covariance matrices at each measurement location in S (Kyriakidis and

Journel 2001), and the combination of distinct random functions for each relevant variable to create an effective spatiotemporal kriging variogram function (Kyriakidis and Journel 1999). These approaches for effective spatiotemporal interpolation can become cumbersome as more locations are added, or as more relevant variables become available. Further, methods requiring kriging are generally not well-suited to implementation in a wireless sensor network context (Jin 2009). With regard to convenience of implementation, one might imagine an optimal approach to have the following properties:

- capable of representing smooth, continuous functions
- capable of accepting both temporal and spatial input parameters
- capable of computing scalar or vector-valued output parameters
- capable of reproducing functional surfaces of any of the Stevens scales of measure
- adaptive and self-modifying, to account for incremental change over time
- representable as a finite collection of scalar parameters, for easy transmission from node to node in a sensor network
- implementable by a finite number of addition and multiplication operations

Our chosen model that suits this list of requirements is the *Multilayer Perceptron* (MLP), a dynamic nonlinear regression technique from the family of artificial neural network (ANN) models. Chapter 2 provides some background information on MLP model structure and operation. Chapters 3 and 5 describe an application of MLP field interpolation in a conceptual network of stationary (moored) sensors deployed in the Gulf of Maine. The results are not limited to a particular testbed, but could be realized in

nearly any distributed, sensor-monitored environment. The core original contribution of this thesis is the methodology for localized spatial and spatiotemporal field interpolations within the context of distributed sensored computational platforms, such as those used to make up wireless sensor network systems.

1.4.1 Proposed Approach

When a sensor platform requires the combination of the local sensor reading(s) with those taken by nearby sensor nodes, it can be handicapped when necessary readings become unavailable. One approach is to estimate missing data by querying the value of the variable from other geolocated sources, and use of a kernel function to spatially interpolate the needed estimate (Jin 2009). Although this technique has the advantage of simplicity, even a customized kernel function can do little more than provide a weighted-average estimate. Although a weighted-average estimate may be sufficient in situations of isotropic data variation, such behavior is seldom the case for many stochastic natural phenomena at work in regions monitored by sensor networks. For these situations therefore, a compact spatiotemporal model that captures the temporal *dynamics* of the monitored field (e.g., some representative function F_{Pi}), may yield more accurate results than a simple spatial weighted average.

A node in a wireless sensor network has access to more information than just the outputs of its own sensors, or even the information directly addressed to it from its neighbors. Wireless communication travels outward from the source in all directions, and is accessible to all neighboring nodes within range just as it is to the particular node to which the communication is officially addressed. A node's "listening in" on

broadcasts within its range but not addressed to it is called *peeking* (Stemick *et al.* 2008). Through peeking, a node can maintain a short history of the last *n* readings reported by any of its neighboring nodes. These historical readings can be used by an autoregressive function to make a temporal prediction of a particular neighbor's next likely reported reading. Combining these temporal inputs with local spatial guiding parameters (for example the spatial trend of readings currently reported by neighboring nodes) could further enhance a node's interpolation accuracy were it required to estimate a non-responsive neighbor's expected sensor reading – or even to provide an approximation to its own currently-nonresponsive sensor.

Given a collection of recent readings, the system dynamics underlying the phenomenon producing these readings can be induced into a finite set of weights comprising a MLP model. The induction process, called *training*, is computationally intensive, however, and is not likely to take place on any nodes within the sensor network, but rather on a server connected to the network, with more reliable access to power and computational resources. We assume therefore that all necessary data collected by the network of nodes is transmitted to the server so that this training may occur. Once the MLP model is appropriately trained, its component weights may be transmitted to the network nodes that require them for in-system estimations. This transmission may impose a temporary communications-induced drain on the network, but such updates should be infrequent under normal circumstances. It is assumed that any long-term wireless sensor network deployment would be equipped with some power recharging capacity, and that MLP updates would not occur so often as to deplete power stores in the interim. Offloading the intensive computation to an out of network server

and updating network nodes with new MLP models only when necessary allows the network of sensor platforms to benefit from increased estimation accuracy while mitigating some of the limiting factors of the sensor network platform.

1.4.2 Discussion

Through their training process, MLP models converge to a maximum-likelihood estimation function of the process they are meant to approximate, given the input parameters presented. A sensor node on its own or in combination with its neighbors may only be able to perceive a local view of the process, in which first-order effects (i.e., general trend) and second-order effects (i.e., smaller-scale autocorrelation in deviations from the mean) are confounded. Modeling this process as a collection of weights allows us to provide a first-order picture of its dynamics relatively inexpensively for a sensor node with only a local view of it. Nodes containing a copy of the MLP model (i.e., a finite collection of weights representing the trained neural network) can use it to point-estimate missing readings either from neighboring nodes, or at other nearby locations of interest. A regular series of local point-estimates may then be combined to approximate the underlying surface of observed phenomena, in response to network *spatial window queries* (Jin and Nittel, 2008), for example. While such estimates might also occasionally be able to be implemented via a spatial Gaussian kernel function, the MLP is capable of integrating additional supports (such as temporal trend, or readings generated by other, nearby sensors) in order to potentially shape a more accurate view of the underlying field surface.

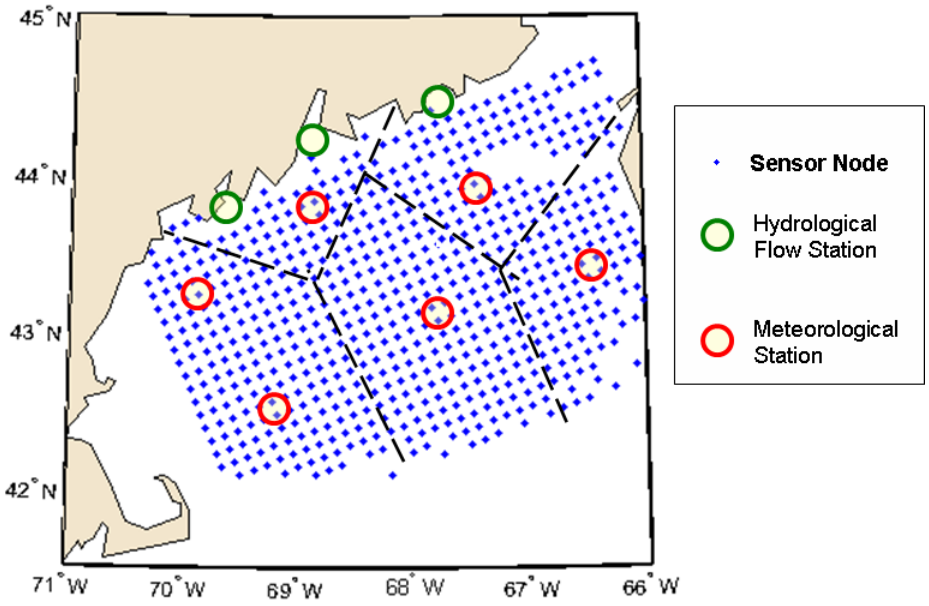


Figure 1.1: A conceptual partitioning of the Gulf of Maine testbed including local datasources

A frequent enhancement applied to spatial studies is to partition the study space to avoid issues of data nonstationarity, and reveal improvements that may be gained (Brunsdon *et al.* 1998). Such studies allow the consideration of variables with impacts limited only to local subregions of *M*, whose significance could be lost in the global model. This approach appears particularly fitting to a partitioned MLP-based model of a region, given the model's architectural flexibility. Different partitions P_i of the monitored region *M* would be assigned MLP models with potentially differing structures and input sets, depending on the local influences and data sources available to each partition (Figure 1.1). Furthermore, communication strain on the entire network can be avoided if only a subset of nodes requires an update of an onboard model.

Finally, MLP models are adaptive and have therefore some capability to respond to gradual change. Although MLP models emerge from training with a marked bias

toward the dynamics of the data on which they were trained, subsequent data readings can be used to continue to incrementally modify the set of trained weights they were initially provided. If the dynamics of the observed process remain similar to those of the training data, or change only slowly, the MLP model can remain accurate and functional longer than static, non-adaptive models. Should the process dynamics change too rapidly for the adaptation process to keep up, the adaptive capacity still out-performs static models. This could be a useful property when a model no longer meets performance standards, but cannot be immediately replaced. This might occur due to temporary loss of connectivity to the rest of the network, or in cases when transmission of a new model might place too great a strain on a portion of the network. If an expected model update is delayed, readings should not be too far from reality as network operation continues in the interim.

1.4.3 Qualifying Assumptions

The potential application of MLP models to the domain of wireless sensor networks will obviously suffer the same list of standard limiting factors as any WSN-based application:

- limited energy sources
- constrained communication
- limited processing, storage and memory capacity
- volatile nature of individual nodes.

In many ways an MLP model seems well-suited to such an environment. It can potentially represent complex system dynamics in a relatively compact form, meeting

memory space requirements. It can be processed efficiently as a sequence of multiplications and additions, interspersed with occasional simple functional transforms, in line with limited processing requirements. It allows nodes to make up for the temporary disappearance of neighbors from whom a reading is required. Clearly the major logistical hurdle to this implementation in WSNs is the communication cost of distributing a new MLP representation to all the nodes requiring it. The choice to use an MLP model instead of one with a more compact representation (but also presumably less representative power) is one of the many tradeoffs that is balanced in the design of a monitoring network. If a kernel method cannot provide the required accuracy, the more expensive MLP (both in terms of computation and transmission-cost) may be the next best choice. As broadcast data transmissions are the greatest power draw for individual WSN nodes, one would like to avoid unduly draining the system's power reserves when an update is distributed.

Data compression can help somewhat in this regard. Network partitioning may help further, as the whole network need not bear the cost of distributing a model to a limited region. Some assumptions can be made of a system that would require an MLP modeling system. The main underlying assumption is that it is a long-term, rather than temporary, environmental deployment. Short-term deployments may not require the same level of adaptability over time as would platforms spending several seasons in the field, for example. A long-term network of sensing platforms might feature some capacity for power source regeneration, for example via solar panels. Although several prototype WSN systems exist (Johnson *et al*. 2009), such as the Crossbow Mica family of motes that power themselves either from a direct power source or standard AA cell batteries, the

constant maintenance these battery-changes would require generally make such systems unworkable for long-term environmental deployments, where direct access to a constant power source is rare. Other assumptions are that:

- models will be trained/fitted outside the network, and communicated to the individual sensing platforms comprising the network. These platforms will also regularly transmit their data-record of readings to the training platform, to enable subsequent refittings.
- platform memory capacity is sufficient for model storage (that is, to store the finite collection of floating-point coefficients necessary to transmit or represent the model).
- platform computational capability is sufficient for smooth logistic (or other nonlinear) function transforms.
- the refractory period between model transmissions is sufficient for partially-depleted nodes within the network to recover energy stores.

Many of these capabilities are already adequately covered by most existing sensor systems. If we expect that these requirements will continue to increase, we can also expect Kurzweil's *Law of Accelerating Returns* – an extension of Moore's Law of Circuit Capacity to technology as a whole – to continue to hold as well: that is, as current technologies approach a new technical barrier to their function, new technologies are invented to allow us to cross that barrier. Given the increasing importance and scientific value placed on the development of effective environmental monitoring systems, it is unlikely that MLP solutions will prove too expensive for use in WSN systems for long, assuming they are currently out of reach at all.

1.5 Hypothesis

A Multilayer Perceptron model is a universal approximator, adept at shaping itself to generally conform to presented data in order to approximate its underlying structure. Thus the MLP is a likely candidate for spatiotemporal field representation, especially when interpolation of missing field data is required. Spatiotemporal fields can exhibit complex behaviors, however, and might, for example, display temporal as well as spatial nonstationarity. We expect therefore that if a single MLP model is not capable of satisfactorily rendering a spatiotemporal field on its own, field representation may be more accurately accomplished through some means of partitioning the field and representing each partition with a separately designed and trained MLP model. Since a spatiotemporal field may change over time, we anticipate that its change can be represented by a state-model, which will signal change events requiring, for example, an increase or reduction in the number of component MLPs required to adequately maintain the changing field's representation. Therefore the hypothesis for this thesis is: a partitioned system of MLP models will show performance gains and improved interpolation results over that of a non-partitioned, or single-model system in the presence of detectable spatial and temporal non-stationarities.

1.6 Key Research Questions

The research questions this work addresses are the following:

Is the MLP a reasonable model to apply to spatiotemporal applications? MLP models are flexible and powerful, but their capabilities come at a price. Expanding the set of a model's input parameters makes the model very flexible, but it also expands the dimensions of the search space within which a solution must be found, with accompanying explosive growth in the number of local minima (i.e., suboptimal solutions) in that space. Chapter 3 evaluates this question.

Can the partitioning of space be a benefit? Regionalization of the MLP model may serve to counteract the "curse of dimensionality" (Bellman 1957), as input parameters may be constrained to only the most relevant local readings. Local conditions may also suggest particular partitioning schemes that are more effective than others. Chapter 4 addresses the question of partitioning benefits on synthetic datasets, while Chapter 5 explores this question on a real data set.

When and how should partitioning schemes be reevaluated? As conditions change with the passage of time, the number of partitions needed to represent the system may change as well. Chapter 6 investigates this question.

1.7 Scope of Thesis

The focus of this thesis is on modeling system dynamics of monitored environmental phenomena using available local datastreams to approximate missing spatiotemporal field readings as they occur (or as they are needed). As natural dynamics are not typically stationary in space or time, spatiotemporal partitions are investigated to improve model specification to regional dynamic influences driving observed phenomena. Since natural processes are not stationary in time, models may be periodically refreshed to keep up with evolution in the dominating influences within a region as well.

1.8 Research Approach

This dissertation intends to develop an effective and generally applicable adaptive modeling system to deal with spatiotemporal nonstationarity as it emerges in individual phenomena evolving over space and time. It may be seen as an application-outgrowth of current work in the spatial computing and reasoning community, with theoretical roots spanning (Dube & Egenhofer, 2014; Worboys & Duckham 2006; Egenhofer & Al-Taha, 1992).

1.9 Expected Outcomes

A first expected outcome of this work is the confirmation of the utility of MLP models for application in the spatiotemporal domain, especially as applied to spatiotemporal field interpolation in the context of WSN deployments. It is further expected that in the presence of non-stationarities, partitioning the global domain into regions for which MLPs may be specialized will lead to significant improvements over a single global MLP implementation.

1.10 Intended Audience

The audience for this work includes scientists and practitioners of spatial information and computer science, geography, sensor science, artificial intelligence, and the geographic and scientific database communities. This dissertation provides approaches for the effective interpolation of spatiotemporal information in a volatile and lossy WSN context – that is, a setting in which readings may occasionally be lost due to equipment glitches, adverse environmental conditions or transmission error. This dissertation may be useful to designers of next-generation WSN platforms who plan to implement the capacities required for robust real-time observations of the physical world in an uncertain and continuously changing environment.

1.11 Thesis Organization

Chapter 2 of this dissertation provides some background and an overview of the state of the art in spatiotemporal interpolation and prediction models. Chapter 3 presents basic global MLP model experiments and performance measures. Similar experiments and performance measures are presented in Chapter 4 for partitioned synthetic datasets processed by groups of specialized MLPs. Chapter 5 applies the techniques developed in chapter 4 to a nonstationary natural dataset collected from the Gulf of Maine. Chapter 6 explores spatial and temporal dimensions for indications of second-order effects to determine the potential utility of using further models to process the residuals of the initial model. It also explores a comparison of error distributions of trained models over time to determine an appropriate time to initiate a model-refitting process. Finally conclusions and proposed future work are discussed in Chapter 7.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Sensors that provide the discrete readings comprising spatial data fields are typically electro-mechanical, and embedded into the natural environments that they monitor. Natural environments, however, are notoriously corrosive to electronic sensor components. As such, any number of natural sources of interference and/or technical issues can combine to create gaps in the readings gathered to define spatial and spatiotemporal data fields. Given that spatiotemporal fields cannot be sampled in their entirety, and that data may periodically be lost, spatiotemporal interpolation methods are needed to fill gaps in a sensed field, as well as to make finer-grained representations of an existing field.

Various numerical techniques have been developed to interpolate missing readings in fields of values, such as linear, Fourier or polynomial modeling for data series, and kernel methods or kriging for spatial or spatiotemporal data sets. But most such techniques assume stationarity in data values; and in the case of kriging, might require access to the entire field of values to generate accurate models of spatial data variability (Webster and Oliver 1993). Values gathered from natural environments often display stochastic, nonmonotonic and nonstationary behavior. Global access to all known data is often impractical or impossible in wireless sensor networks, where local

calculations may nonetheless be required to be made relatively accurately, and in near real time. A first major focus of this work is to investigate whether a data-driven Multilayer Perceptron (MLP) model using primarily local information will be sufficiently resistant to temporal and spatial non-stationarity in a natural spatiotemporal dataset to give accurate interpolation performance for simulated gaps in the dataset.

2.2 Traditional Approaches to the Spatial Interpolation Problem

Traditional statistical interpolation schemes tend to follow one of two general approaches: exploiting either temporal or spatial autocorrelation of the data to achieve their goal. Univariate linear or polynomial regression models attempt to take advantage of the temporal autocorrelation in a time series. Cyclical processes are commonly modeled with sinusoidal basis functions through Fourier, or "harmonic," analysis. Multivariate versions of these approaches combine the mathematical transfer functions associated with each parameter to relate the multiple predictors to observed results on the spatial plane. Some assumptions underlying these approaches can make them unfit solutions as a practical matter, however. Linear models are clearly unfit for problems that display significant nonlinearity, or non-monotonicity, for example. Multivariate models generally suffer from the "curse of dimensionality" (Bellman 1957), wherein the problem-space expands exponentially with each additional predictor integrated into the model.

Spatial kriging is a traditional cornerstone technique of geostatistics that exploits the spatial autocorrelation between a collection of spatial readings of some attribute *z* to generate an interpolated surface providing values at unobserved locations (Matheron 1963; Journel and Huijbregts 1978). Knowledge of the entire spatial field of values is required to estimate the covariance between values of *z* at different spatial lags. If covariance is not known or difficult to quantify, the average semivariance between data pairs can be estimated instead. Semivariance, covariance and correlation are intrinsically related, so if one is known, the other can be derived (Bailey and Gatrell 1995). A continuous model is fit to these average semivariances. The semivariogram model constructed can then provide semivariance values at any given spatial lag to drive the kriging interpolation process.

These models do not typically take time into account, however, and integrating additional parameters into the kriging model is cumbersome. Further, some level of data stationarity is assumed, as for nearly all standard statistical models, yet natural spatial surfaces and natural spatiotemporal data sets frequently do not conform to this requirement.

Although hybrid approaches do exist that take both spatial and temporal data into account, they tend to be fewer in number, more complex and/or computationally expensive (Gething et al. 2007), and somewhat less generalizable to situations beyond the particular one they were designed to solve (Kyriakidis and Journel 1999). A possible reason for the generalization problem may be due to the different scales of space and time implicit in most spatiotemporal datasets. Besides being impossible to compare in a physical sense, different processes operate on different scales (Lam and Quattrochi

1992), thus it is expected that customizations made to a model in order to isolate the dynamics of a particular phenomena may be incompatible for datasets incorporating phenomena operating at different space-time scales. One simple hybrid spatial interpolation approach, known as *spatial time series* (Bennett 1979; Kyriakidis and Journel 1999), extrapolates forward from past values of a spatial field surface to approximate missing values in the current one. Spatiotemporal kriging models also exist, but come with a variety of additional requirements, as described in the next section.

2.2.1 Brief Overview of Spatiotemporal Extensions to the Kriging Model

Dimitrakopoulos and Luo (1996) proposed several spatiotemporal trend models for kriging systems that require linearly independent component functions to arrive at a unique solution. The component trend functions must ensure *tensorial invariance* of the spatiotemporal kriging system, so that the kriging estimator and variance are invariant under changes of the origin and/or units of the coordinate system. Three types were proposed: traditional polynomial functions, Fourier expressions, and hybrid combinations of the two – all of which are linearly independent. These hybrid trend models do not necessarily maintain tensorial invariance, however, and must be checked (Kyriakidis and Journel 1999). Polynomial trend forms of order K in space and L in time meet tensorial invariance only if all polynomial orders up to K-1 and L-1 are present; similarly, Fourier models of order K must have all terms up to K as well to satisfy tensorial invariance (Dimitrakopoulos and Luo 1996)

In the general case of a nonstationary spatiotemporal problem, a stationary mean is assumed; however, obtaining the necessary residual covariance values is problematic

and they are therefore estimated. The results of this residual estimator vary depending on the algorithm used to estimate the spatiotemporal stationary mean, thus bias can be introduced

Other proposed models have similar difficulties and complexities. Rouhani and Hall (1989), and Christakos (2013) adapted the intrinsic random functions of order K (IRF-K) model (developed by Matheron (1973) for a purely spatial context) to a spatiotemporal context (IRF-KL) for application in the earth sciences. But this method had difficulties in non-gridded contexts, and the generalized covariance model it generated was not guaranteed to be unique. The family of generalized covariance models is limited; iterative trial-and-error selection and fit of generalized covariance components tends to yield models with large nugget effect because of the sensitivity of the fitting algorithm to outlier data (Journel, 1989). Although some of these problems can be avoided by basing inference of residual covariance on values *not* affected by the spatiotemporal trend, such values can be difficult to determine in a context of complex space-time interactions.

Journel and Rossi (1989) proposed to get around the issue of data nonstationarity over large regions by restricting the kriging to sufficiently small neighborhoods around the point to be approximated. They concluded that the method worked as well as a method incorporating an explicit, nonstationary trend model, such as models proposed by Eynon and Switzer (1983), Haas (1995) and Gething *et al* (2007). However, due to the stochasticity inherent in spatiotemporal phenomena, this property does not necessarily hold for forward extrapolation of values in time.

When a single spatiotemporal random function (RF) model Z(*s*, *t*) is considered, spatiotemporal continuity is modeled by a joint space–time covariance function, whereas under the multiple RF models approach, spatiotemporal continuity is modeled via the Linear Model of Coregionalization (LMC) (Journel and Huijbregts, 1978). A joint space–time covariance function allows estimation (through kriging in a space–time context) at any location *s* in space and any instant *t* in time. The LMC model however can only approximate values at predefined points *s* in space and *t* in time; while the single random function model's joint space–time covariance function allows approximations at any points *s* and *t*. Determination of a joint space–time covariance function requires high data density at the same space location/time instant however, whereas this is not a limitation for LMC. LMC implementation becomes progressively more difficult as measurements in the spatiotemporal domain become more abundant, as t(t + 1)/2 (for time-instants) or n(n + 1)/2 (for spatial locations) auto- and cross-covariance matrices have to be computed. Thus the number of random functions the LMC model implements is generally kept small.

Kyriakidis (2001) proposed a relatively straightforward extension to the spatial time series model that spread time series vectors out to the spatial domain by regionalizing temporal trend parameters in space. This approach avoided the need to generate n(n + 1)/2 direct and cross-variograms/covariances for each geolocated time series (Kyriakidis and Journel 2001).

2.2.2 Summary of Advantages and Disadvantages of Explicit Spatiotemporal Trend Modeling

Conclusions we may glean from the approaches described above are the following. Explicit modeling of spatiotemporal data may allow for the incorporation of local information specific to the dataset (Gething et al. 2007), though often at the cost of making the resulting solution too problem-specific to generalize to different data sets. Integration of additional parameters or monitored spatial locations s_i tends to expand either computation time or the problem solution-space exponentially in explicit models such as LMC, however, and no interpolation between predefined locations s_i may be done. Random function-based models (i.e., non-explicit) may extrapolate for any location s_i so long as sufficient data density exists, and the approach generalizes readily to alternate applications. Additional parameters are not as easily incorporated into this model.

The multilayer perceptron model marries some of the advantages of these two alternate approaches, while avoiding some of their respective limitations. It can incorporate additional parameters while implicitly moderating the dimensional growth of the solution space. Its solution generates a random surface that is not limited to a predefined set of locations s_i, and its approach generalizes well to alternate spatiotemporal domains.

2.3 The Multilayer Perceptron Approach

Multilayer Perceptron (MLP) network models are a subset of a group of artificial intelligence techniques called Artificial Neural Network (ANN) models. The term "Artificial Neural Network" encompasses a wide variety of mathematical models designed for automated information processing and/or machine learning tasks effected through various means. Models falling under this umbrella term have names such as: perceptron, self-organizing map, Hopfield net, bidirectional associative memory, and many others. The tasks they are designed to effect can fall under the general headings of either classification (e.g., clustering, or identification), or function approximation (e.g., prediction, regression, or interpolation). Although the MLP can be used for classification tasks, it is discussed here solely in its typical regression context.

This type of mathematical regression model consists of a weighted collection of simple processing nodes designed to associate a set of input parameters (predictors) to a set of desired output values. Like traditional regression techniques, such as linear or polynomial regression, it combines a set of basis functions to converge to an optimal solution surface. The basis functions most commonly used by MLP are nonlinear, monotonic sigmoidal functions. Nonlinear basis functions allow the MLP to model highly nonlinear surfaces, but its method of converging to that final predictive surface is wholly different. It is a data-centered approach, meaning that there is no intrinsic bias at the outset toward what form the final solution will take. Instead, the data encountered cause the appropriate solution-surface to gradually take shape. Originally conceived in the late 1950s as a simple model called the Perceptron (Rosenblatt 1958), the MLP came into its own with the development of a technique called backpropagation (Rumelhart et al. 1986; Werbos 1994), a distributed regression technique which allowed the model to

reliably converge to an appropriate solution by evolving a set of parameters (via ordinary least squares estimates) that minimize an objective function of model error, typically mean square error (MSE).

2.3.1 Model Structure

The general template for the original Perceptron model consists of two connected layers of nodes: input and output. The input layer consists of *i* input nodes, that simply capture the *i*-element data signal. Data processing occurs in the output nodes. Each input element is propagated forward along weighted connections to each node in the output layer, where all weighted incoming signals are summed, and this sum is transformed by the basis function – called an *activation function* in the literature.

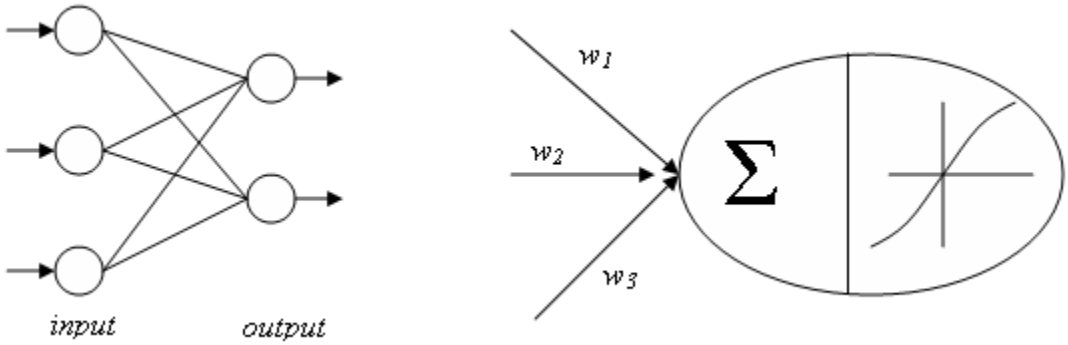


Figure 2.1: Perceptron model, with illustration of processing sequence in output nodes

The MLP model enhances the basic Perceptron model with the insertion of one or more layers of *hidden nodes*. Like the Perceptron output nodes, hidden nodes also sum all incoming signals and transform the sum via their activation function, and propagate their results onward to the output layer via weighted connections. The activation functions incorporated into hidden and output nodes can be seen as analogous to the random functions used by both spatial and spatiotemporal kriging models described earlier.

2.3.2 Backpropagation Training

As the backpropagation algorithm simultaneously modifies all component weights of the MLP model, it performs a distributed implementation of gradient descent that encourages the network as a whole to converge to a state of minimal error. Given a set of target values *t* for the model to replicate, inputs are propagated through an untrained, randomly-initialized model to generate a matching set of output results *r*. Errors are determined at the *o* output nodes (t_o-r_o) and propagated back to the *h* hidden layer nodes proportionately to the activation function results (*act*) which contributed to them. Then modifications to each connection-weight (w_c) are made, often moderated by some learning rate $0 < \alpha < 1$ (see Table 2.1).

Table 2.1: General Backpropagation algorithm

1. Calculate Error-term (δ) for each node (in reverse order)

$$\delta_o = -(t_o - r_o) * f_o'(act_o) \quad \text{(output node(s))}$$

$$\delta_h = f_h'(act_h) * \sum w_{ho}\delta_o \quad \text{(hidden nodes)}$$

2. Compute weight-change for each connection *c* :

$$\Delta w_c = -\alpha\, \delta_d act_s$$

(where d = destination_node; s = source_node)

3. Modify connection weights:

$$w_c = w_c + \Delta w_c$$

Further details on backpropagation and MLP models can be found in Haykin (2008).

2.3.3 MLP Model Benefits

MLP models display several properties that make them interesting, particularly to modelers of natural processes. MLP models can often outperform standard polynomial

regression models at their task of settling to a minimum-error surface as measured by the least-square metric. Part of this ability resides in their use of monotonic sigmoidal basis functions to interpolate between observed values. The combination of these functions results in an Occam's razor effect, encouraging a smoother surface and thus avoiding the artificial high-frequency variation between observations that can easily occur in overfit polynomial regression solutions. Due to the use of these same nonlinear basis functions, MLPs are thought to be superior to standard time-series prediction models when the underlying data is highly nonlinear, or dominated by complicated functions (Weigend 1996).

The dimensional growth in the solution space an MLP must search is restrained in comparison to that of polynomial regression. A polynomial regression with p parameters, for instance, causes exponential growth in the number of coefficients to be estimated, due to the interaction terms between parameters. Such growth requires an exponentially increasing sample size to determine values for each coefficient to an acceptable degree of confidence. The addition of parameters to an MLP model causes only linear growth in the dimensionality of the search-space, thus significantly reducing Bellman's curse of dimensionality for MLP models (Marzban 2009).

MLP models are also often found to be more resistant than standard statistical approaches in the face of incomplete, noisy and non-stationary data (Zealand et al. 1999). This can be ascribed in part to their data-driven nature, which requires a period of training or calibration on a set of representative data. Any set of training data will display a certain amount of variation in its values, but the Central Limit Theorem ensures that any sufficiently large set will approximate a normal distribution in the trained

model's output relative to presented input parameter values. Exposure to this data induces into the MLP a functional surface shaped by maximum probability distribution responses to the training examples provided. In this way a tendency towards mean-moderated responses to inputs is formed, regardless of whether the mean is constant or varying. Thus the influence of noisy data and the outliers it contains is moderated, while at the same time the internal segmentation of its solution-space enables the model to deal more gracefully with non-stationary data sets. Nonetheless, MLP performance can be significantly improved by preprocessing its data to remove trends and render it more stationary. Reducing the dimensionality of the parameter space through such means as principal component analysis is also helpful, as any dimensional reduction in search-space vastly reduces the number of potential local minima to which its gradient-descent-based error-minimization backpropagation procedure might otherwise fall victim (Masters 1995).

Being a data-driven model, no pre-analysis or critical assumptions about the nature of the data, spatial or otherwise, are necessary. This does not mean that the MLP model is wholly assumption-free, however. Reliance on the minimization of overall model error (MSE) in the training phase implies one assumption at least: that errors in the measured data are normally distributed (Marzban 2009). The backpropagation regression procedure ensures that relevant features of the data set will be learned in the course of training (Openshaw and Openshaw 1997). Naturally, this presumes representative data sets in which the desired features are present. One consequence, however, is that even if some specific domain knowledge is known regarding a particular phenomenon or data set, this information cannot be explicitly emphasized or modeled, other than providing it

as an additional input parameter (if possible) to the MLP. Although there is no guarantee that any given parameter will ultimately be utilized, any parameter that provides non-redundant information capable of guiding model outputs to consistently low-error performance is likely to be integrated into the solution process.

The proof that MLP models are universal function approximators – that they are capable in principle of approximating any continuous function – has been made in several studies (Cybenko 1989; Funahashi 1989; Hornik et al. 1989; Castro et al. 2000). Universal approximation itself is not an unusual property in modeling systems, as it only requires that component basis functions maintain linear independence. Polynomial regression and Fourier (or harmonic) decomposition techniques are thus also known to be universal approximators. That MLP models are also found to be so confirms that they have the power to competitively model interesting and significant continuous surfaces, including those formed by the covariance structure interactions relating parameters in the model's input vector to each other. However, none of these proofs are *constructive*; that is, they indicate only the existence of a set of weights yielding the appropriate combination of basis functions for the sought-after result, but give no indication as to how that set of weights may be achieved. Nor do they address the number of basis functions (represented architecturally in the model as the number of *hidden* nodes) needed. For example, a sinusoidal wave can provably be represented by a MLP composed of sigmoidal basis functions, but only so long as the number of hidden nodes approaches infinity (Castro et al. 2000). Although this is an extreme case, and most problems will not in fact require near-infinite numbers of nodes, it illustrates a need for

some pre-analysis to determine an appropriate number of basis functions for the problem at hand.

2.3.4 MLP Weaknesses

As has been suggested above, some properties of the technique do not always necessarily work to its advantage. The architectural flexibility of the model is such that the naïve modeler may be tempted to use all available data streams as predictor parameters, but the greatest difficulty can be in selecting appropriate model inputs. Determining the most relevant inputs is necessary in order to mitigate structural complexity of the model, as well as to limit the dimensions of the solution-space to search. Limiting the MLP to only the most relevant parameters has the additional related effects of decreasing the training time of the model, and increasing the generalization ability of the resulting solution (Zealand et al. 1999). Nor is the MLP ever completely free of the *curse of dimensionality*, as any solution space it must search will feature a plethora of local minima. As noted in the preceding section, choosing an appropriate number of hidden nodes for a model is neither easy nor obvious. The general approach to this issue has traditionally been to implement a battery of bootstrap tests to determine the smallest number of hidden nodes for which model error shows significant decrease (Marzban 2009). Another popular method to determine optimal model structure is to use a *genetic algorithm* evolutionary approach (Ferentinos 2005). Although this latter approach can be more flexible than the former, it can also be much more expensive in terms of time and computational resources. While the solution arrived at by either

approach will typically deliver above average performance, in neither case is the determination of an optimal model structure guaranteed.

Openshaw (1997) lists various critiques leveled at the MLP technique: that the defining processes and patterns of the phenomenon that the model attempts to represent are poorly represented, or hard to extract; and that it is essentially a black box model. Marzban (2009) suggests that the black box reputation is not entirely deserved, however, as the appearance of complexity is not all necessarily due to the model, but can emerge from the problem itself. The apparent necessity in spatiotemporal models for growing numbers of multiple cross-covariance matrices as locations s_i increase would seem to suggest that most spatiotemporal problems too belong in this number. Indeed, *any* complex, highly nonlinear phenomenon with a high level of interaction between parameters will make for a complicated representation, and will be extremely difficult to concisely and meaningfully describe (Marzban 2009). Nonetheless, it is often worthwhile to provide a sensitivity analysis on the final trained model to determine the spectrum of contributions of each input parameter. Olden (2002) proposes one potential technique, and provides a short review of alternate approaches in (Olden and Jackson 2002).

Finally, it is known that the process of finding the existence of an optimum collection of weights for even relatively simple MLP models, such as those comprising a 3-node hidden layer of linear computational nodes, is an *NP-complete* problem (Blum and Rivest 1992). Given the infinite combination of initial weight allocations such a network may be assigned upon initialization, each one defining a different challenging landscape of local minima to navigate, there is little reason to believe that a universally

optimal training algorithm exists to consistently mold the collection of weights to an optimal configuration. As there is a certain level of uncertainty associated with any physical readings taken by in-situ sensors in a natural environment, the optimal weight-set may be impossible to achieve even with a perfect training algorithm. In most cases, it may be sufficient simply to locate the deepest local minima possible. Thus we can expect no single trained MLP to learn the problem-space perfectly (Hansen and Salamon 1990).

Different models trained from different initial weight distributions will inevitably make generalization errors on different subsets of the problem space (Hansen and Salamon 1990). This has led to some active research into *ensemble* MLP systems, wherein multiple MLP models are trained for the same problem, and the final system output is determined by majority support (in the case of classification systems), or by average response for function approximation (Cannon and Whitfield 2002; Baker and Ellison 2008; Watts and Worner 2008). These studies generally support the finding that a collective decision by ensemble is likely to contain less error then the conclusion arrived at by any single member of the ensemble. Ensemble results can therefore be much more trustworthy than reliance on a single model. However, the process of molding a trained model's final weights from the starting-point of a randomly initialized weight distribution is a global optimization problem, for which no optimum approach is known. Any global optimization algorithm will thus experience varying levels of success according to the particulars of the problem set before it (Hansen and Salamon 1990).

Nonetheless, MLP models have experienced a moderate level of adoption by researchers in the natural sciences to model various natural processes. Dawson and Wilby (2001) presented a comprehensive review of MLP methods in the field of

hydrology, showing the applicability of the technique to phenomena with temporally-lagged effects. MLP use has been shown to be simpler and more efficient in environmental modeling scenarios where otherwise mass-consistent or hydrostatic flow models – necessitating the solving of systems of fluid dynamics governing equations – would be used. Models employed for tidal or coastal water-level point forecasts have been seen to offer substantial improvements over harmonic analysis while requiring significantly less data than the more traditional time series prediction methods for that level of performance (Tissot, Michaud et al. 2003; Lee 2004).

2.3.5 MLP Models Used for Spatial Interpolation

Relatively few studies can be found employing MLP models for spatial interpolation. Those that do exist are primarily of two types:

1) studies that approximate aggregated (e.g., maximum, average) physical measurements for a discrete set of geolocated points, inside a network of data stations used as predictor variables (Snell *et al.* 2000; Londhe and Panchang 2007).
2) studies interpolating an entire surface of aggregate values, based on a subset of nearby geolocated data sources (Rigol et al. 2001; Bryan and Adams 2002; Rigol 2005).

The first group of papers above implant MLP models that use readings from a fixed collection of stations *A* to predict cotemporaneous values at a non-overlapping set of static locations *B*. This approach can certainly serve to arrive at solutions by exploiting the covariance relations between the various locations in A with those in B. These

models can say nothing, however, about the regions around and between the distinct points $\{A \cup B\}$. The second group of papers uses MLP models that employ readings from a selection of nearby stations, in addition to the physical parameters at the current location, to interpolate an entire field of values.

Being universal approximators, MLP models should be capable of integrating time series as temporally-correlated lagged predictor variables into a prediction surface (and some studies do employ the time-series technique for point forecasts: (Dawson and Wilby 2001; Tissot *et al*. 2003)). Such an approach is similar in concept to the spatial time series approach for spatial interpolation, but given the MLP advantage in easy extensibility, with the added possibility of incorporating further arbitrary data features available in the region. Rigol (2001) compares the performance of a model interpolating a surface with terrain variables specific to the location in question, for example, against a model also incorporating near-neighbor effects. That the best performing model in Rigol's study incorporated local guiding effects as well as a subset of available terrain variables is further confirmation of the necessity of choosing one's predictors well, but also how crucial the incorporation of local constraints is to a model's ultimate performance level (Bollivier *et al*. 1997).

2.4 Addressing Nonstationarity

A perceptible shift is ongoing in recent approaches taken to spatial problems. Models that have traditionally been designed as global simulations of a process are increasingly being developed as a system of regionalized or partitioned sub-models. One of the more well-known examples of this trend is Brunsdon's concept of *geographically-*

weighted regression (Brunsdon et al. 1998). Certainly the idea is not new: the concept of *divide and conquer* has long been a mainstay in computer science and mathematics, as indeed in every discipline requiring systematic problem-solving methods. Nonetheless partitioning is a very useful technique in spatial applications, where techniques require stationarity for optimal results, yet the data is rarely if ever stationary. As mentioned earlier, restricting kriging to sufficiently small areas can be sufficient to dispense with the problem of data non-stationarity (Journel and Rossi 1989), but can also potentially open the door to greater variance and increased uncertainty in the model's results.

Another potential advantage of partitioning is that different variables may become more relevant as the focus is restricted to smaller regions. Different processes operate at different scales (Lam and Quattrochi 1992). Processes and interrelationships between input parameters that may have been invisible to a global model may be better distinguishable to a model trained on just one part of the whole. In other words, a system of smaller-scale models may allow us to mitigate the Ecological Fallacy (Robinson 1950) that a single global model might be induced to make: that phenomenological behavior in all regions reverts to the mean, global behavior. Given a model such as the MLP which is capable of easily integrating new variables into its approximative function, the spatial partitioning technique may have some potential to impart accuracy improvements over the whole system that a single global system lacks the flexibility to deliver.

2.4.1 Partitioning Techniques

Most partitioning strategies utilize some form of clustering method. Clustering algorithms can be generally classified under two basic approaches: hierarchical agglomerative models that can be represented as dendrograms, such as hierarchical tree

models (Murtagh, 1985); and squared-error based methods (also known as Vector Quantization), such as k-means (Xu & Wunsch, 2005). Many techniques in the latter group can be shown, under specific circumstances, to be identical to the k-means technique, suggesting that this base approach underlies many of the more specialized algorithms targeted to apparently-disparate problems. Furthermore, k-means is a computationally efficient technique at an approximate computational complexity of O(n) (as compared to $O(n^2)$ complexity or more for agglomerative approaches, for example) (Cormen *et al*. 2009).

K-means clustering has general application in any non-spatial context. Openshaw (1977) provides a description of the fundamental spatial partitioning problem referred to as the automatic zoning problem (AZP), as a partitioning of *n* basic spatial units (*bsu*) of measure into distinct collections by the following.

Let $\mathbf{x_1, x_2, \ldots, x_N}$ be $\boldsymbol{N}$ vectors of dimension $\boldsymbol{n}$ representing the *bsu* data:

$$\mathbf{X = [x_1^T, x_2^T, \ldots, x_N^T]^T.}$$

The N vectors of X partition the study area into K zones, denoted 1, 2, …, K; such that $1 \leq K \leq N\text{-}1$. A classification array W is defined as:

$$\mathbf{W = [w_1, w_2, \ldots, w_N]^T}.$$

Finally, it is assumed that there is a model to be used on zone data with $\boldsymbol{m}$ independent variables; $p_1, p_2, \ldots, p_m$;

$$\mathbf{P = [p_1, p_2, \ldots, p_m]^T.}$$

An objective function provides a measure of partition performance in terms of the model and a predefined target value, so that by optimizing this function an

optimum *partition-performance* is obtained. The function F(W, X, P) is a scalar function of the independent variables W and the constant variables X and P. It maps the performance of any partition onto the set of real numbers.

The unconstrained optimal-zone design problem either minimizes or maximizes the function F for the partitioned data, depending on the desired optimal goal. The framework of Fuentes (2005) is quite similar to the AZP problem described by Openshaw, with *N* data items in *X* partitioned into *K* collections in *W*. In the context of this thesis the primary technique used to accomplish partitioning is the well-known k-means algorithm.

2.4.1.1 The k-means Algorithm

Given a set of *d*-dimensional observations ($x_1, x_2, \ldots, x_n$), k-means clustering aims to partition the *n* observations into *k* sets $S = \{S_1, S_2, \ldots, S_k\}$ (where $k \leq n$) so as to minimize the within-cluster sum of squares (WCSS):

$$\min\left(\sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2\right)$$

where μ_i is the mean of the x_j in S_i.

Main Algorithm

Beginning with an initial set of *k* means $m_1^{(0)}, \ldots, m_k^{(0)}$, the algorithm proceeds to alternate between two steps: *instance-assignment*, and *mean-updates*.

a. Instance assignment

Each d-dimensional instance x is assigned to the cluster whose mean happens to be the smallest distance from x. Since the arithmetic mean is a least-squares estimator, this meets the objective of minimizing the WCSS (Note: this process can also be described mathematically as partitioning the observations according to the Voronoi diagram (Voronoi, 1908) generated by the means). Each x_p is assigned to one and only one set S_i, even if it could be assigned to two or more

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - \mu_i^{(t)} \right\|^2 \leq \left\| x_p - \mu_j^{(t)} \right\|^2 \forall j, 1 \leq j \leq k \right\}$$

b. Mean-updates

Once all instances have been assigned to cluster-sets, a new mean for each set is calculated

$$\mu_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j$$

When the μ_i no longer change in the mean-updates step, the algorithm has converged onto a local minima set of μ_i. As both steps optimize the WCSS and the number of partitions is finite, the algorithm must converge to a local minimum. However there is no guarantee that its solution is a global minimum. In practice, this entire algorithm is repeated N times in order to avoid suboptimal solutions and instead isolate a “best” one to a desired level of significance.

Drawbacks

There are two significant drawbacks to using the k-means method: model assumptions and choice of k. In general, k-means assumes a spherical cluster model where clusters are of similar size and density that may not be met in all natural situations. However, a generalization of k-means called the expectation maximization (EM) algorithm provides incremental performance improvement by taking both variance and covariance of cluster-member data instances into account (Dempster *et al*. 1977). The algorithmic simplicity and relative performance efficiency of the basic k-means algorithm however, has advantages for a power-limited WSN context. The most appropriate value of *k* for the current time period is determined by a Monte Carlo simulation, described later in the Monte Carlo BestK-approximation section.

In the context of wireless sensor networks there have been several attempts to implement distributed clustering schemes, since the power-limited quality of such networks is one of the greatest obstacles the network must overcome in its delivery of valid results. Distributed EM schemes are popular, with variants proposed by (Nowak 2003), (Kowalczyk and Vlassis, 2004), (Gu, 2008), (Wolfe et al. 2008) and (Forero et al. 2008). K-means and hybrid distributed methods are proposed in (Chen et al. 2004), (Younis and Fahmy, 2004), (Forero *et al*. 2008) and (Oliva and Setola, 2014). Unfortunately, these works are limited by requirements which may not be supported in WSN, including: particular network topologies and configurations; storage and computing power needed for nodes to solve clustering problems comparable to the overall centralized one; or need to meet assumptions that cannot be easily guaranteed in a

dynamic natural environment. In an effort to maintain generality therefore, we limit ourselves in this work to the basic k-means algorithm to explore the effectiveness of this technique as a first step.

Energy savings is often a prime concern when working with Wireless Sensor Networks (WSN) (Apiletti *et al*. 2011). The basic k-means algorithm may not finally be strictly optimal for all large-scale spatial and temporal data applications (including those exhibiting high data dimensionality) for which separate, related techniques have been developed (e.g., BIRCH, CLARA, DBSCAN, EM etc.) (Jain *et al*. 1999) (Tan *et al*. 2006) (Witten *et al*. 2011). However, techniques have been developed to efficiently apply k-means to large geospatial datastreams (Nittel and Leung, 2004), and we have found it a reasonable, effective and simple clustering technique that is well suited for power-limited processing such as takes place on self-contained wireless sensor nodes embedded within natural environments.

2.4.1.2 Monte Carlo BestK-Approximation (MCBestK)

Choosing the most-appropriate *k* for use in a k-means process is considered a difficult algorithmic problem, made even harder in multidimensional data, even when clusters are well-separated (Hamerly and Elkan, 2004). Consequently, multiple studies and heuristics exist to determine a dataset's most appropriate choice of *k* in order to generate reasonable clusters (Bischof *et al*. 1999) (Pelleg &and Moore, 2000) (Jain *et al*. 1999). This technique is used to determine when a dataset would benefit from partitioning, and also delivers an approximate, discrete best-K value to use when partitioning using a k-means approach. The general algorithm is the following:

```
Collect dataset

for i = 1 to N:
  for k = 1 to maxK:
   Generate sum of within-cluster squared-distances (WCSS) for k into matrix
     mm(i, k)

Determine average solution-series s(k) from N series in mm(i,k)

for i= 1 to statN:
  Generate new version of dataset with randomly-permuted column-values
  for k = 1 to maxK:
    Generate WCSS for each i,k into permuted matrix pm(i, k)


if all series pm(i,_) > s(_):
  normalize x-, y- axes
  for i = 2 to maxK-1:
    determine slope m for K=i
    if m(i)>=-1 and m(i-1)<-1: bestK = i
```

This heuristic is discussed in greater depth in Chapters 4 and 5, where it is applied to datasets in order to find the most appropriate number of partitions to use.

2.4.2 Assessing Model Inputs

Random initialization can be a very powerful design concept in specific situations – for example, it makes a major contribution to security in the creation of a robust encryption scheme – and in this current work to create and fit a model for purposes of dataset exploration, interpretation and missing datum approximation. For example, both MLP and k-means models are randomly initialized, allowing models to investigate and compare different approaches through the multidimensional problem space in order to find an effective solution. Multiple randomly-initialized models are generally created in order to determine a single reasonably well-fit solution, as well as to control for the

tendency of candidate solutions to settle in local minima. Due perhaps in part to the randomness of their origins, a downside of using MLP models (and, for that matter, the k-means model) is that although the best-performing models resulting from these techniques deliver perfectly reasonable results, they are essentially black-box processes (Marzban 2009). That is, it is often difficult to determine *how* their final results were achieved, or even *why* they work as solutions to the problems they are employed to solve. Often these models are developed, discarded and recreated frequently enough that it is not considered worthwhile to analyze their workings, as the next well-performing iterate may well exhibit a completely different structure or approach in delivering its results.

Due to this difficulty in articulating how or why the model delivers the results that it does, we propose to use another randomly-initialized statistical model , a *random forest* (Breiman 2001) – also known as an ensemble of *regression trees* (Breiman *et al*. 1984) – to pry open the black boxes of both our k-means and MLP models in order to gain some insight into the perceived importance of their respective problem inputs. We do this by performing a repeated bootstrap-aggregation process known as Tree-Bagging on a subset of our regression-tree models, and thereby focus on the inputs that the MLP's training process found particularly significant in generating its results.

2.4.3 Classification and Regression Trees

Parametric models specify the form of a relationship between predictors and a response. In many cases, the form of the relationship is unknown, and a parametric model requires assumptions and simplifications. Classification and Regression Trees

(CART) provide a nonparametric alternative (Breiman *et al*. 1984). Binary tree classifiers are constructed by repeated splits of subsets of the set of data instances X into two disjoint, descendant subsets, beginning with X itself. A regression tree is one where all instances of a given class *c* are processed via a regression function f_c. In this thesis regression trees are used to "crack open" the black boxes of other nonparametric models in order to measure the relative significance assigned to parameters by the model.

Small populations of multiple regression-tree models (called *ensembles* or *random forests*) consisting of subsets of the entire available parameter set are generated through *Tree Bagging* to efficiently derive quantitative weightings of model parameters for the model in question (Breiman, 2001). Random forests have a number of characteristics that make them well suited to work in tandem with black-box function-approximation applications such as Artificial Neural Networks. They run efficiently on large data sets, can easily be parallelized, and are relatively robust to outliers and noise. Further, they do not require specification of an underlying data model, can capture non-linear association patterns between predictors and response, and are able to deal with highly correlated predictor variables. Most importantly, they generate an internal unbiased estimate of the generalization error (called OOB, or *out-of-bag*, error), which allows them to determine which variables are important within the regression model.

The concept of random forests combines many binary decision trees built using multiple bootstrap samples of a dataset and randomly choosing at each node a subset of explanatory variables *sv* (Genuer et al. 2010). Kühnlein et al. (2014) provide the following explanation for the general random forests algorithm (Kuhnlein et al. 2014):

i. *s* bootstrap samples are randomly selected from the data set with replacement. For each bootstrap, a different subset of the data set is used to develop a binary decision tree model. A certain number (e.g., 33%) of instances are left out of the sample. This OOB set is used to generate unbiased estimates of the regression error as well as to estimate the importance of predictor variables used to construct the tree.

ii. A regression tree for each of the bootstrap samples is generated (resulting in *s* trees), with one important modification: a subset *sv* of the predictor variables is randomly selected to create the binary rule. In other words, *sv* specifies the number of randomly chosen predictor variables upon which the decision for the best split at each node is made. The variable selected for the split is based upon the lowest residual sum of squares error generated for each of the *sv* variables. *sv* is held constant for each tree in the forest ensemble model.

iii. Each of the *s* tree models is grown out as far as possible; there is no pruning

iv. Approximations are calculated by passing each data instance (whether sample set or OOB) through each tree model and averaging results to produce the final estimate.

Further necessary definitions and heuristics are rendered in detail within (Breiman 1984) and (Breiman, 2001), with various applications in satellite reading enhancement (Kühnlein *et al*. 2014), bioinformatics (Boulesteix *et al*. 2012), and generalized parameter-selection (Genuer *et al*. 2010).

2.5 Uncertainty Estimates for Spatiotemporal Models

Significance differences between the results of single- and multiple-model MLP systems are determined by a 2-sample t-test, where the variance of each population is assumed to be similar, and normality of error-distribution is also assumed (via Central Limit theorem due to large populations of result-instances). The standard equations used are:

$$t = \frac{\overline{X}_1 - \overline{X}_2}{s_{X_1X_2} \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \qquad \text{where: } s_{X_1X_2} = \sqrt{\tfrac{1}{2}\left(s_{x_1}^2 + s_{x_2}^2\right)}$$

Root mean square error is a common performance metric used to evaluate the fitness of trained MLP models to their task. While this may be sufficient for models of a process at some discrete location, such as tide levels at a particular station along a coast, it is a less satisfactory solution in a spatial or spatiotemporal context as performance will typically vary over space as well as time. In a spatial context a more common approach is *contouring*; that is to evaluate the accuracy of a model as a contoured surface of the performance metric, as in (Bailey and Gatrell 1995). This suggests that the performance metric of a spatiotemporal field would be a spatiotemporal field itself, with a formal definition similar to Definition (1.3):

$$E : T \rightarrow M \rightarrow V_{ERR}$$

A visualization of some model's error function E at a given point in time might look like the following:

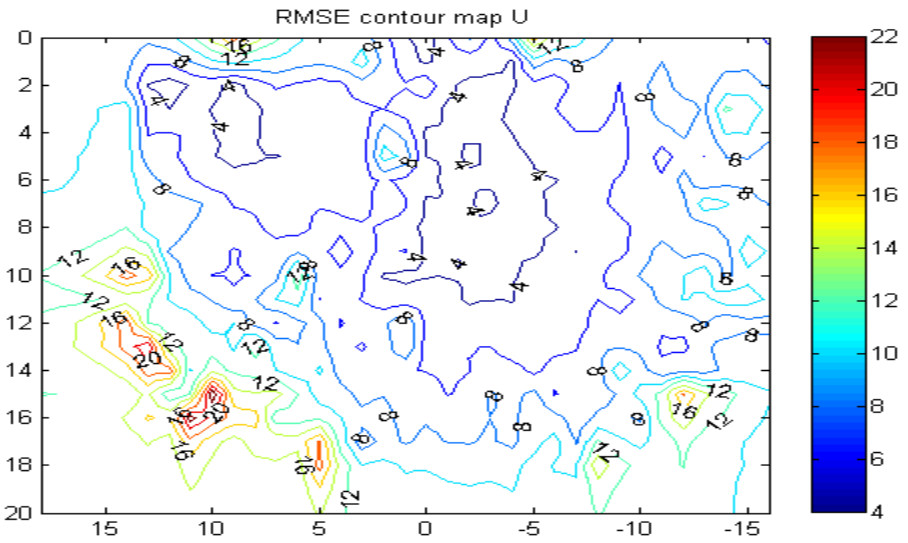


Figure 2.2: Notional contour map of an error function over a spatial region

Visualisations such as this aggregated over time can be useful in determining potential partitionings of the space to better focus on regions of consistently poor performance. These partitioned regions might, for example, have a unique model dedicated to them in order to better deal with influences particular to that region. This is the subject treated in the following chapter, where the chosen model is an MLP.

CHAPTER 3

MULTILAYER PERCEPTRON FIELD INTERPOLATION OF SPATIOTEMPORAL DATA

The goal of this chapter is to evaluate the performance of a global multilayer perceptron (MLP) model at the task of spatiotemporal field interpolation using the local spatial and temporal information available to it. Its performance is gauged in relation to two other common data interpolation techniques: ordinary kriging (OK), and a simple temporal persistence model (TPM). OK is a well-known spatial approximation approach that employs a global view of the available spatial data field to yield high-quality estimations of missing values. TPM simply returns the previous known value at a given location, providing a minimum level of performance the MLP can be expected to improve upon. The OK and TPM interpolation test methods provide for comparison against a more strictly spatial and more strictly temporal performance assessment respectively for the MLP model.

3.1 Introduction

Many techniques exist to approximate missing values in time series as well as spatial datasets. Linear, polynomial and Fourier interpolation are often applied to time series, and spatial kriging is a popular interpolation method for spatial data. Most of these interpolation methods concentrate solely upon either the temporal or the spatial

dimension of the data. Techniques exist that integrate both space and time (Kyriakidis and Journel 1999), but their combination often requires a series of subjective assumptions specific to the dataset (Knotters et al. 1995), and could result in an overly complex model with poorer performance than a single-dimension version (Skøien and Bloschl 2007). Further, nearly all these methods have difficulty representing complex nonstationary relationships within their datasets.

Artificial Neural Network models, and specifically the basic MLP model, have not been widely applied in the GIS domain, although the technique has gained some wider acceptance (Bollivier et al. 1997) despite the model's perceived shortcomings (i.e., its black-box nature). MLP have been applied to the spatial interpolation of daily temperature variables, and have outperformed traditional benchmarks (spatial averaging, near neighbor and inverse distance methods) (Snell et al. 2000). Rigol et al. compared the results of MLP models using various combinations of globally invariant data – such as day of year, latitude and longitude – and additional locally-relevant information such as local terrain aspect and near neighbor readings, to implement the interpolation process (Rigol et al. 2001).

One reason MLP models may be suited for spatial applications is that the sigmoidal basis functions commonly used in their construction enable better performance than that delivered by the more standard linear and log-linear models (Openshaw and Turner 2001). Pariente reported better geographical interpolation performance with stacked Hopfield neural nets (a related artificial neural network model) than standard methods such as kriging (Pariente et al. 1994).

This chapter explores the effectiveness of a trained MLP model to accurately model the spatiotemporal dynamics of surface ocean currents in the Gulf of Maine. Given the harsh climatic conditions that sensors embedded in and around the Gulf of Maine experience, a reliable approximation method for the monitoring system's missing data is necessary. Having a full and accurate picture of conditions in the Gulf is important for sea-based industries, for mitigating effects of accidents (oil spills, for example), and for tracking potentially harmful periodic events such as harmful algal blooms (Townsend et al. 2001). The data for this experiment was obtained from a Coastal Ocean Dynamics Application Radar (CODAR) surface current monitoring system based on the coast of Maine. While not a sensor network in the context of wireless sensor networks (WSN), we view the grid of geolocated surface ocean current readings provided by the CODAR system as though provided by an array of independent sensors, whose communications with the data store are intermittently interrupted due to either natural conditions and/or technical limitations. Our objective is twofold: (1) to test how well MLP models perform as a field interpolator in the spatial domain in comparison to standard geostatistical techniques (such as ordinary kriging); and (2) to demonstrate that situations exist where the use of a MLP modeling system may be preferable to the standard geostatistical solution (primarily due to its performance as a less computationally-intensive spatial-temporal field interpolator than space-time kriging (Guan *et al.*, 2011; Zhong *et al.*, 2015)).

The remainder of this chapter is organized as follows: Sections 3.1 – 3.3 provide relevant information on the source of the data, its composition, preparation and evaluation. Section 3.4 describes the MLP, kriging and TPM baseline models whose

performances are compared in this study. Section 3.5 contains the experimental results, and we conclude with section 3.6.

3.2 Model Data – Ocean Surface Current Measurements

The experimental data used for the spatiotemporal field in this study are 2-D surface ocean current vectors observed over the Gulf of Maine. These data were drawn from the University of Maine Ocean Observing System (UMOOS) that is part of the Northeastern Regional Association of Coastal Ocean Observing Systems (NERACOOS). The data collection system consists of an array of buoy-mounted sensors, as well as CODAR coastal radar stations located to provide maximal coverage of the sea-surface of the Gulf of Maine. Data were 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. Each station periodically transmits radar signals in a radial pattern, directed out towards the surface of the ocean. A physical phenomenon known as Bragg scattering ensures that all signals striking waves traveling directly toward or away from the transmitting station will be reflected back to the station for capture. 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 rectangular square grid with cells measuring roughly 15 x 15 km (Figure 3.1). We refer to the two components of these velocity vectors as *u* and *v*. Fields of surface currents are thus determined once per hour. The three

CODAR stations primarily active during this time-period are Wood Island to the southwest, Greens Island midway up the Maine coast, and Cape St Mary located in Nova Scotia to the east. Further information regarding CODAR array operation can be found in (Pettigrew *et al.* 2008).

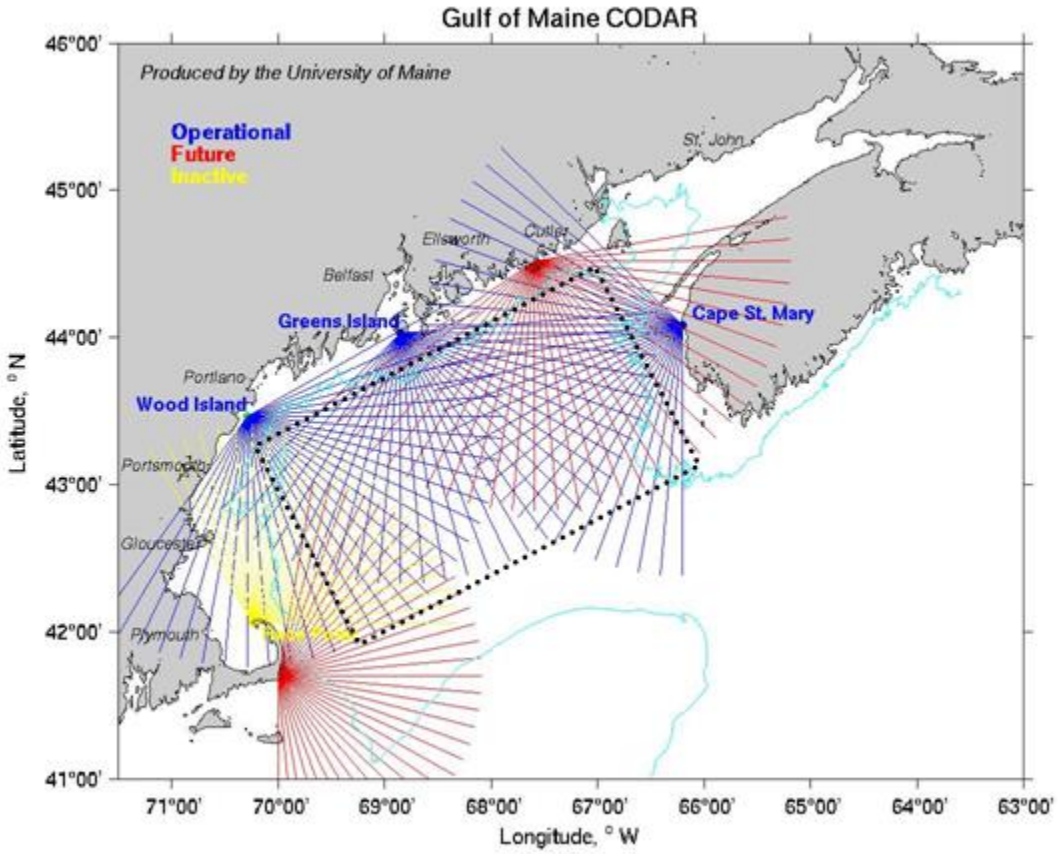


Figure 3.1: Map of Gulf of Maine CODAR testbed region with active sites labeled. The monitored region *M* is indicated by a dashed rectangular outline

3.3 Methodology

3.3.1 Radial Component Approximation

Although CODAR stations will generally attempt to project outgoing radar signals over their entire range of coverage, some surface conditions within that range may

reflect signals back poorly; further, atmospheric or other interference may also contribute to an incomplete field of radial readings being reflected back to source. Since at least two separate radials are required from the radar stations to synthesize a velocity vector in one location, this has the practical effect of causing data drop-outs in the resulting vector field where only one CODAR radial exists (i.e., only one radar station received a reflected signal). The partial information contained in the datum represented by the single radial may hint at surface current behavior in a given location when a history of past values of that single radial (and the total vectors resulting from them) is consulted.

For example, imagine a location *l* exists where the total surface current always flows due east (that is, it never flows west, nor is there ever any north or south component). Assume that a radar station always receives radial signals for *l*, and when these are greatest the magnitude of the surface current at *l* has the value max_l, and when these are smallest, the magnitude at *l* is min_l. Given this serendipitous linear relationship between the radial signal and the surface current vector at *l*, knowing the value of the radial signal will provide a very good idea of the surface current magnitude at *l* , potentially making *radial-value* an effective input for the MLP tasked with approximating surface currents in the region containing *l*. This is just the type of opportunistic data-mining that MLP models can facilitate. On the other hand, should we resolve to use a single-radial attribute as an input parameter to our MLP model, we must ensure that a single-radial value exists wherever (and whenever) the MLP is to be used. Otherwise our model cannot be used anywhere *radial-value* is not available

To ensure that a radial value exists at each location and time-step within our testbed, a synthetic field of radials is generated to fill all empty potential grid locations

with inverse distance-weighted average values from nearby radial readings. In this particular context, the MLP model uses as radials from the Greens Island (GRI) station as one of its input parameters as this station is most centrally-located in the Gulf of Maine, and is therefore most likely to have a value to provide to all monitored locations within the Gulf.

3.3.2 Delineating the Spatial Interpolation Space by Convex Hull

For each time step, a *convex hull* is generated from the known data in the field of surface velocity vectors, bounding the perimeter of the area within which model approximations will be made. This procedure ensures that any approximated value will have the spatial support necessary to make at minimum a linear approximation of its actual value from two independent readings. No approximations are effected outside this bound as the values for large swathes of estimated currents may be based upon the same few actual readings, and are thus insufficiently independent.

3.3.3 Velocity Vector Approximation

The *spatial time series* approach employed by the MLP model requires an uninterrupted time series of prior surface-current readings (in addition to the specified radial-value input described above) to aid in making its approximation. Yet due to the same climatic and technical conditions that affect radial vector collection, there may be gaps in the surface current record. Therefore the historical data record is completed by filling in for any missing values with the results of a spatial kriging model surface for the

requisite time steps. The MLP model results will be compared to Ordinary Kriging (OK) results, thus it is important to note that the MLP model's results never employ the OK model's results at the timepoint for which a value is approximated, although prior OK results may be present in the MLP input set as part of the historical record. MLP model results from previous time steps are *never* employed as inputs, even as historical values in the MLP input set, to prevent the natural bias present in a trained MLP model's results from accumulating in subsequent approximations. All three of these preprocessing steps are illustrated in Figure 3.2 below.

3.3.4 Quantitative Measures

Two general quantitative error measures were employed to compare performance of various models: *simple mean error* (SME) and *root mean square error* (RMSE). The simple mean provides an estimate of model bias in each component of the resulting vector. No bias is expected in the kriging model employed to fill in missing values in the historical data record as required, as it is an unbiased predictor. RMSE gauges overall performance accuracy of the model over its assigned region. These measures are determined by computing model-approximations for missing and known values. Performance measures were determined on the basis of approximation error for all known locations with measured values, and were assumed to be roughly the same for the remaining locations with missing values inside the convex hull. Since the kriging model is an exact interpolator, its approximation errors are obtained from the "leave one out" cross-validation process, where kriging is run once for each known data value in the field

but leaving out the value under consideration. As each data value is comprised of the two components *u* and *v*, overall best performance is achieved by minimization of the magnitude of the resulting error vector.

3.4 Model Configuration

3.4.1 The Multilayer Perceptron Model

In the multilayer perceptron model, inputs propagate forward through the weighted network of simple processing elements comprising the MLP. From the input nodes, through the layer of hidden nodes, to the output nodes, the input values are combined and transformed through a series of relatively simple operations into output values. At each hidden- and output-layer processing unit, weighted incoming values are summed and then transformed via some activation function. Such a model requires training to produce correct outputs, which is managed through a process called *backpropagation*. In this process output errors are propagated backwards through the model, modifying all weighted connections in order that outputs at the end of subsequent feed-forward cycles converge towards the desired results. This illustrates a main advantage of using an MLP for modeling a spatial-temporal process – training the model on recent data will automatically integrate spatial and temporal inputs in appropriate proportion for current conditions. Using backpropagation to induce an observed phenomenon's dynamics into a finite collection of coefficients in this way results in a relatively compact, transportable and disposable representation usable by devices of limited computational capability, such as independent WSN nodes in a network.

3.4.2 MLP Model Specification

All MLP models in this paper were implemented with the MATLAB Neural Network Toolbox (v5.1), and have a 32 – 3 – 2 architecture (i.e., 32 input nodes, 3 hidden nodes and 2 output nodes). Two output values were required because the interpolated surface ocean current readings are 2-D vectors. Two eight-hour time series of 2-D vectors comprise the input vector presented to the model's 32 input nodes. Given a tidal period *T* of 12.5 hours, an eight-hour series ensures that appropriate signal harmonics (e.g., *T*/2, *T*/4) are included in the input set. The number of hidden nodes was determined by an evaluation of *normalized RMSE* values (Lee 2004)

$$normRMSE = \sqrt{\sum_k (y_k - \hat{y}_k)^2 \Big/ \sum \hat{y}_k^2} \tag{3.1}$$

on models with one to six hidden nodes, where three hidden nodes provided the smallest *normRMSE*. Many variations on gradient descent exist, but our models employ the Levenberg-Marquardt variant for its adaptive learning rate α, and property of guaranteed convergence.

Because model inputs consisted of two 2-D time series, the spatial interpolation was accomplished primarily by the *spatial time series* approach, extrapolating missing values from a weighted sum of the series of prior known values for that location (Bennett 1979). The first time series (TS1) recorded the prior eight hours of surface velocity vectors for the location (not including the current time's vector reading, which was considered missing), while the second time series (TS2) recorded the prior seven hours of GRI radial values for the location, in addition to the radial value at the time of approximation. While the first time series communicated a purely temporal aspect to the data, the second incorporated some neighboring spatial influences as well due to the

weighted-mean radial processing described in Section 3.3.1. This spatial time series approach of essentially using twin arrays of temporal readings is reflective of the current direction of research in large-scale spatiotemporal data processing, where array databases are seen as the best support for such applications (Camara, Egenhofer *et al.*, 2015). An advantage of taking this approach with an MLP is that more than a single time-series (2 in this case) may be used as the model's input vector, as well as an arbitrary number of other related inputs that may be available to the model. This capability will be explored further in subsequent chapters.

Although the spatial component to the MLP model described here is relatively subtle, it will become much more significant in subsequent chapters 4 and 5. The described MLP model could contain a more explicit spatial component by integrating inputs of a more overtly spatial nature, such as "the reading from the node(s) to my immediate west/east/north/south." This would of course limit the model's use to only those locations with a neighbor in the appropriate cardinal direction. Since the MLP implements the s*patial time series* approach by providing results for every location within its monitored region *M*, these more explicit (but more limiting) spatial inputs were left off in favor of a more straightforward comparison of results with competing models.

Further pre-processing specific to this model included the short-term removal of temporal trend from the surface-velocity vector time series (though not the radial series). The dominant forcing mechanism for tidal currents in the Gulf of Maine is the 12.42-hour M_2 semidiurnal tide (Ku *et al.* 1985), so a 12.5 hour simple moving average signal was subtracted from all surface current vector values in the MLP model's input vector. This had the effect of imposing some temporal stationarity to the time series, while providing

only the residuals to the model for processing (note that while claims are made that ANN models adapt better to nonstationary data better than standard linear-based modeling approaches, they can still realize a performance benefit from using preprocessed data to mitigate or remove nonstationarity (Virili and Freisleben, 2000). Once the model produced its output, the temporal trend was reincorporated into the result as the model's final prediction.

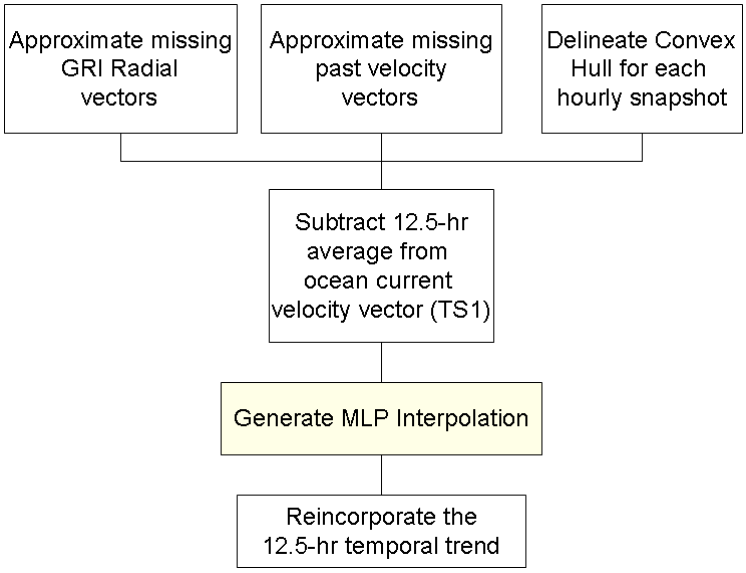


Figure 3.2: Data preprocessing steps involved in this MLP interpolation

A single randomly-initialized MLP model was trained using two random subsets (each approximately 16%) of the dataset, one for training, one for validation. The training set was employed to modify the MLP's internal connection-weights as described in Table 2.1 where each exposure to the complete contents of the training set is termed an *epoch*. Between epochs, the model's performance was gauged against the validation set to ensure that model results were sufficiently generalized, and that *overfitting* (i.e.,

memorization of the training set) was not taking place. Training continued until a satisfactory performance threshold was reached, or until it was determined that further exposures to the training set led to overfitting. After a large number of models were trained, the one with the best RMSE performance was selected as best-fit for the problem-space.

3.4.3 The Ordinary Kriging Model

The baseline spatial interpolation model for assessing MLP performance is an Ordinary Kriging (OK) model. OK interpolation is a well-known technique used to map physical properties of a region for the analysis and interpretation of spatial variation, based on variogram analysis. Standard variogram models include: spherical, exponential, gaussian and linear. Variogram modeling is used to identify the spatial autocorrelation structure in geostatistical analyses. A variogram model $\gamma_z(h)$ is obtained for data readings $z(x_i)$ following the intrinsic stationarity (Goovaerts, 1997) as given by equation 3.2.

$$\gamma_z(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} [z(x_i) - z(x_i + h)]^2 \qquad (3.2)$$

where N(h) is the number of pairs separated by a lag distance *h*. The spatial structure of the variogram is the main factor affecting the accuracy of the geostatistical estimation (Kravchenko, 2003). The ordinary kriging model is expressed as a linear weighted average of observations in the neighborhood of an unsampled location x_o:

$$\hat{z}(x_o) = \sum_{i=1}^{n} \lambda_i \, z(x_i) \tag{3.3}$$

where λ_i is the weight obtained from the ordinary kriging system based on a selected variogram model (Journel and Huijbregts, 1978; Goovaerts, 1997).

For this study the the Matlab DACE Kriging Toolbox (Lophaven *et al.* 2002) was used, which implements semivariance via equivalent *correlogram* modeling rather than variograms directly. Minimum variance was the metric used to select the best fitted correlogram model, which in nearly every case was determined to be the exponential model.

3.4.4 The Temporal Persistence Model

The baseline temporal model for this study is termed the Temporal Persistence Model (TPM), expressed simply by the following equation:

$$\hat{g}(x, y, t+1) = g(x, y, t)$$

In other words, the predicted reading for any given location is simply the reading recorded at that location in the previous time-step. This model has the advantage of being simple while still providing results with high correlation and relatively low error rates as

compared to the actual data it predicts, making the effort to surpass its performance not-inconsequential.

3.5 Results and Discussion

Comparison of model results revealed that both MLP and OK model solutions exceeded the threshold performance of the TPM (Table 3.1).

Table 3.1: Overall performance comparison against the TPM model for a 36-hour test period in June 2005

	SME *u* *(cm/s)*	SME *v* *(cm/s)*	RMSE *u* *(cm/s)*	RMSE *v* *(cm/s)*	Performance Increase
Ordinary Kriging (OK)	0.09	0.01	5.40	6.85	34%
Multilayer Perceptron (MLP)	-1.62	-0.13	7.15	9.05	13%
Temporal Persistence Model (TPM)	0.18	-3.28	7.42	11.03	--

The OK model clearly performed the better of the two competing models in comparing overall magnitude of the error vector to that of the TPM. This is to be expected as the OK model can access the entire data field to generate its variance model, while at best the MLP has access to a much smaller neighborhood of spatial readings (i.e., those explicitly provided within its static set of inputs) as communication costs alone within a WSN would preclude the use of a MLP that required every reading acquired at each time-step in order to provide its own approximation.

Figure 3.3a shows the root mean square error levels, aggregated over the entire gulf for each timestamp, resulting from the three models over a particular 36-hour time period. The dynamics of the ocean current system are seen to be reflected in all three series similarly, as RMSE trends tend to rise and fall in tandem for all three. Of particular interest are those occasions where the three series do not all react in tandem. At 18h00 (*1) we see that the RMSE time series for the OK model, which takes a purely spatial view, moves counter to the MLP and TPM RMSE series, which draw primarily from the temporal aspect of the spatiotemporal field. In this case the temporal structure of data in the spatiotemporal field appears to be more helpful than the spatial structure of the data for that particular time slice as MLP temporarily outperforms OK gulf-wide. Earlier at about 08h00 (*2) however, the temporal structure instead caused the temporal model's RMSE to suddenly increase while its spatial structure (readings in many locations indicating an incoming tide) enabled the OK RMSE results to remain relatively stable.

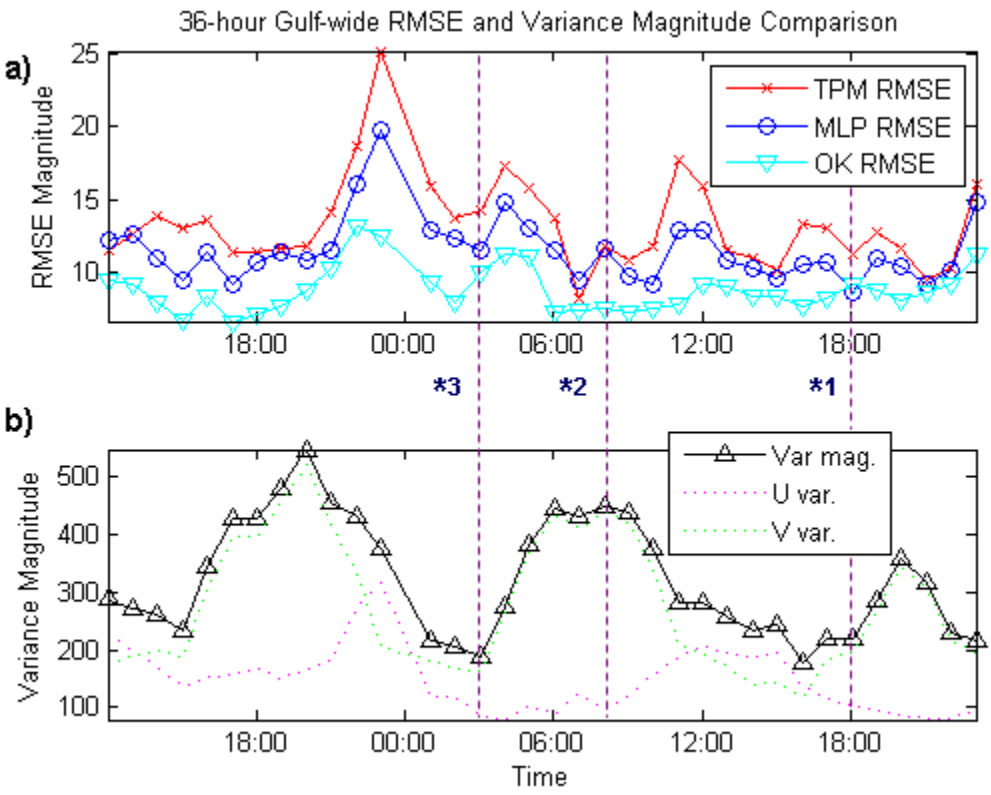


Figure 3.3: a) Gulf-wide Root Mean Square Error of TPM, MLP and OK models compared to b) Gulf-wide surface current variance magnitude (and component variances)

It may be informative to look at the Gulf-wide variance magnitude in surface current readings of the Gulf of Maine for this time period (Figure 3.3b). We notice that at point (*1) on this chart, a temporally-based model might be aware that variance had been increasing over the past two time intervals, and might thus make more informed predictions for the current time point than a model without that information. Whereas at point (*2), one can understand why a temporal model could have been caught off guard, as overall variance magnitude actually *increased* after all prior readings indicated that variance had already peaked and should soon begin to decrease. It is interesting to note that many increases in the OK RMSE series tend to occur during periods of low overall

variance. Sudden increases in RMSE can also be observed in the temporal models when changing from upward trends to downward trends (or vice-versa), followed by a steady RMSE decrease while the current trend continues. From these observations one might be tempted to hypothesize that a temporal model might display better performance than a spatial one after a reasonably long trend of decreasing variance. If so, point (*3) at 03h00 in the variance chart might suggest itself as a likely spot to test that theory. Referring back to Figure 3.3a, the theory appears to bear out as both MLP and TPM RMSE series continue to decrease while OK RMSE has started increasing. So although the spatial structure of the field appears sufficient for OK to generally outperform these basic temporal models for now, there appear to be situations when guidance from the temporal aspect of the data might be more helpful than a strictly spatial perspective. It is likely that with the incorporation of additional relevant spatial variables into the MLP model, its performance would become more competitive with OK's.

Generally comparable results between the OK and MLP models suggest a possible application of the MLP model where OK is less applicable, such as in the domain of wireless sensor networks. Kriging models are much less attractive in a WSN context because of the communication costs involved in assembling all values of the field in a single location to be processed (Jin 2009). Let us assume that each gridded location in the Gulf of Maine contained a fixed wireless sensor node with a trained MLP model onboard. By listening in on neighboring transmissions such a node could keep a temporal record of the last *n* values reported from neighboring locations (analogous to TS1), as well as a time series of spatial trend surface (TS2 analog) synthesized from all captured neighboring transmissions in the past *n* time periods. Should one of these neighbors

subsequently drop out of communication temporarily, one could imagine any of their neighboring nodes being able to fill in with a reasonable approximation based on some local information and the temporal trend in overheard values, at a level of performance not too far from OK, and without the data transmission costs that OK would require.

Analysis of the data variance over the monitored region during this period (Figure 3.4) shows distinct regionalization of data variance, suggesting the possibility of improvement if the space were partitioned and individual MLP models specialized to smaller regions, with access to more relevant local variables. Such regionalization might also allow the partitioned system's performance to approach OK's level of performance more consistently than that of the current global MLP model.

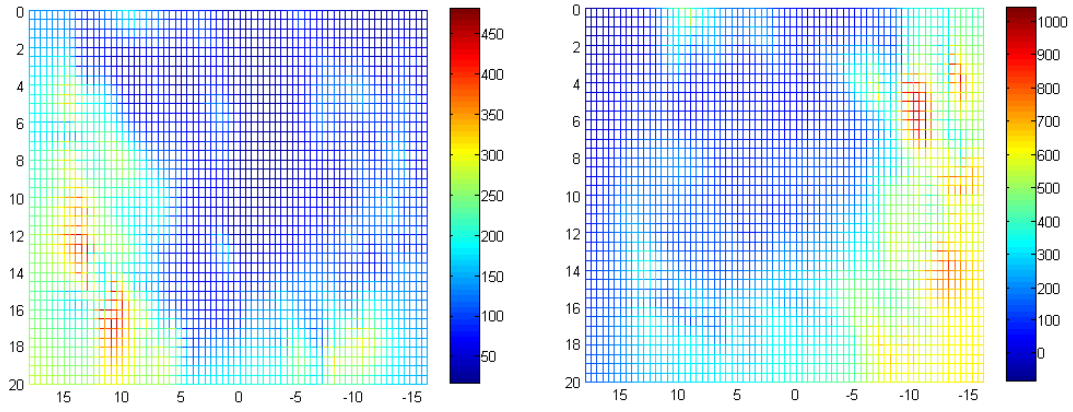


Figure 3.4: Apparent regionalization in variance maps of *u* (left) and *v* (right) component velocity data of monitored region *M*

3.6 Conclusions

This study compared the performance of three interpolation approaches designed to replace missing values in a field of physical readings, using data extracted from a Gulf of Maine CODAR sensor array. Both the OK and MLP models exceeded the minimum performance threshold provided by TPM, the baseline model. Although OK would appear to be the most accurate interpolation method to use in many situations, cases do exist where MLP would be preferable to it, such as for local field interpolation in wireless sensor networks. More attention to spatial structure, in particular spatial nonstationarity in the application of MLP models may lead to improved overall interpolation accuracy. One approach to address spatial nonstationarity is to explicitly identify self-similar spatial clusters in a field of readings and accommodate these with specialized regional MLPs in order to realize a performance benefit.

CHAPTER 4

EVALUATING MLP MODELS ON PARTITIONED SYNTHETIC DATA

4.1 Introduction

Having demonstrated the fitness of the basic, feed-forward Multilayer Perceptron (MLP) artificial neural network model for use in geospatial applications, this Chapter focuses on the investigation of partitioning strategies (both spatially and temporally) for a collection of wireless sensor nodes or platforms to achieve more effective processing and analysis.

A concern in geospatial studies of natural phenomena that evolve over time is that natural processes are frequently inherently nonstationary (although this depends on time and spatial scales chosen – this will be directly addressed in this chapter). In the presence of non-stationarity, we expect some deterioration in the performance of a global MLP model. Previous research in modeling natural spatio-temporal processes has investigated using separable processes (i.e., processing spatial and temporal covariance trend models separately) (Kyriakidis and Journel, 1999), however this desirable property of separability is often an unrealistic assumption in large spatial-temporal domains (Fuentes *et al*. 2003). Fuentes (2005) addressed this problem using spatial clustering to isolate subregions of relative stationarity, as well as temporal segmentation (i.e., temporal "clustering") to maintain a relative constancy in temporal covariance. This concept of temporal segmentation can be presumed valid assuming a relatively-slow rate of change –

for example a process whose period is on the order of days or weeks, rather than minutes or hours. This chapter examines spatial-temporal segmentation, or regionalization (Fuentes *et al*. 2005), approaches in the context of identifying partitions for separable MLP models.

The central research questions this chapter seeks to address are: does the process of approximating missing readings in a WSN-monitored region benefit from partitioning into subregions; and if so, into how many subregions should it be partitioned? To evaluate these questions, this chapter investigates partition-performance on simulated data. K-means is employed as the partitioning technique and partition performance is evaluated by an objective function which is the sum of the root mean squared error (RMSE) of each partition's MLP-model results.

The remainder of this chapter is organized as follows. In Section 4.2 (Methodology) we lay out a context for the problem, and provide specifications for the methods and techniques used to solve it. These approaches are tested on two synthetic benchmark datasets – an orthogonal partition that is described in Sections 4.3-4.5, and a more naturalistic partition in Sections 4.6-4.7. Section 4.8 describes the overall conclusions of this experimentation in preparation for its application to a natural dataset consisting of several months of surface-current readings in the Gulf of Maine in the following chapter.

4.2 Methodology

4.2.1 Application Context

The context for partitioning MLP models includes spatial sensing applications using self-powered sensor-nodes embedded within natural environments, and are not

therefore generally connected to a reliable power-supply. Being unable to replenish energy-stores at will, computation as well as any intra-node communication is kept to the minimum necessary.

Given a wireless sensor network (WSN) of spatially fixed (not to be confused with statistical stationarity) wireless sensor nodes (wsn_o) or platforms embedded within a natural geospatial region *M* such as described in Borgman (2007) or Worboys (2004), self-powered wsn_o, even with renewable-power capacity (e.g., solar panels), will still spend the majority of their time in a minimal-power sleep-mode, waking for particular periods (for instance, during a particular 5-minute slice of each hour).

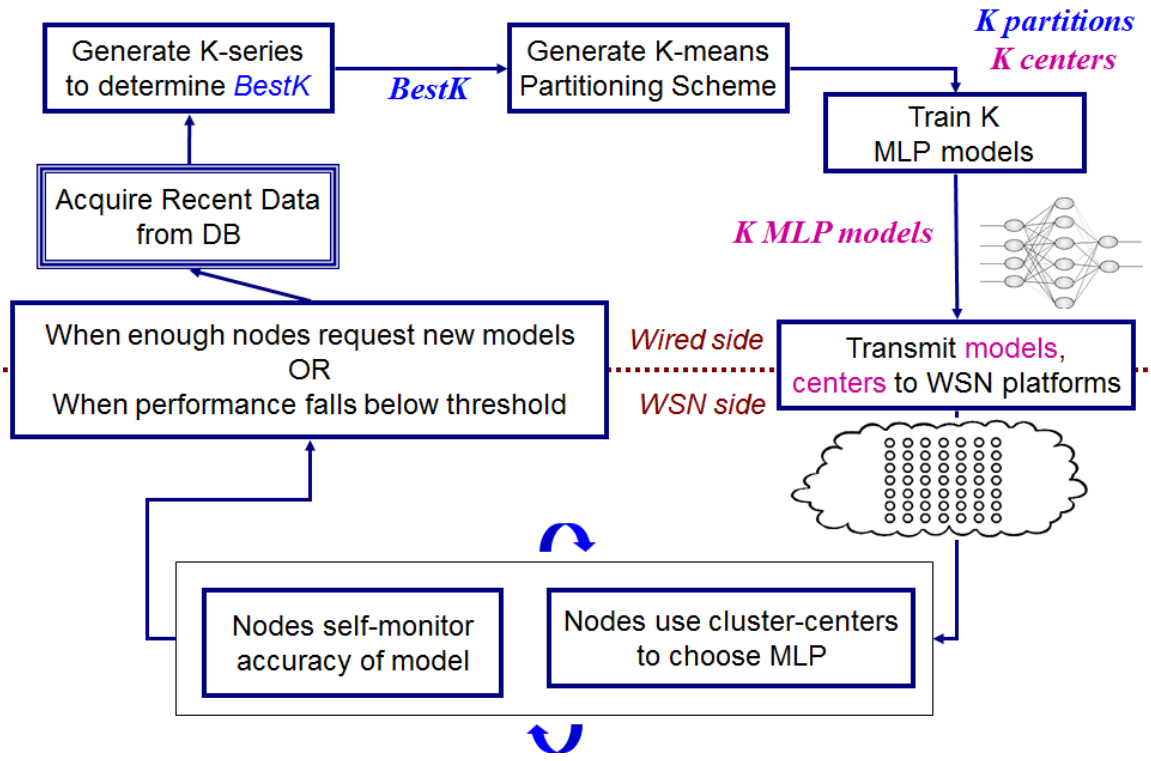


Figure 4.1: Proposed process-chart for proposed CatSTANN framework for partitioned-region MLP model applications

During these wakeful periods they use power as efficiently as possible to: (1) take readings, (2) send and receive communications with neighboring wsn_o, and (3) listen for

communications/updates from some hierarchically superior master-node – possibly another local wsn_o within range, or some global master node or station, wsn_M. A wsn_M is assumed to be hard-wired to a power-source for essentially unlimited power- and computational-capacity (e.g., the transmitter pictured in Figure 4.1), with sufficient transmission-power to reach all its assigned wsn_o during their wakeful periods.

The first step in the process is to determine whether the monitored region *M* would (still) benefit from being partitioned into its current number of subregions or if a new partitioning scheme is required. This decision is determined by a Monte Carlo method on the testbed's most recent collected data (e.g., the process described in (Peeples, 2011)). It is presumed that this processing, as with all demanding processing and analysis, is done on the wsn_M as it is not as power-limited as the wsn_o.

Should *K* multiple partitions be suggested by the Monte Carlo method, *K* MLP models are created and fit to the data-vectors in the subset S comprising one of the *K* detected clusters in the new partitioning scheme. Generally high temporal cross-correlation is assumed, so models trained on near-past data should prove at least adequate to process near-future data vectors as they arrive. The packet of parameters defining each model may then be broadcast to all wsn_o requiring updates as they wake up. Importantly, the K cluster centers determined by k-means are also broadcast to the wsn_o along with the models. Incoming instances of data vectors may then be compared to each of the K centers in order to select which of the K MLP models should process the instance.

Trained MLP models are completely disposable: after a period of time their combined performance degrades. As new surveys of natural clusters in new data take place, new models are generated and distributed to the wsn_o as the system resets to the

new normal (Figure 4.1). This evaluation of obsolescence of the current ensemble of MLP models can be effected either at the level of the individual wsn_o themselves – as they can compare the readings from their sensors against the results of the models meant to simulate them, and thus detect increasing mean error over time at their individual location – or at the level of the wsn_M should the combined performance of the network be found to have dropped beneath a minimum domain-specific threshold (or conversely, should one determine that the system's baseline level of error has become unacceptably high).

Regional signal-surfaces frequently exhibit drop-out zones due to environmental interference with embedded sensor stations (e.g., atmospheric interference, temporary equipment failure). Models are frequently made to approximate such missing readings. Such models are likely to become increasingly inaccurate as time goes by, or as their results are applied further from the center of the region for which they were implemented. These regions may benefit from partitioning, to allow for the application of multiple models – either by mapping a single, distinct model to each spatial, or spatial-temporal, *subregion*; or possibly even one of *several* potential models applied during a *particular period* in a subregion.

4.2.2 The K-Problem: How Many Partitions?

We have elected to use k-means as our partitioning technique, but there is no authoritative way to choose which value of *k* to use. Consequently a wide variety of heuristics exist to do so in just as many particular situations.

To begin, we can imagine a series of k-means partitioning schemes applied to the same multidimensional dataset D containing N instance-vectors, where *k* varies from 1 to K. Standard k-means clustering produces data subsets $D_1 \ldots D_K$. Approximative MLP models $A_1 \ldots A_K$ are generated for each *partition* to process the data subsets, producing K root-mean-squared-error (RMSE) performance measurements $E_{1,K} \ldots E_{k,K}$, one for each of these partitioning-schemes on dataset D, for each particular value K. Total system error SE_K is an aggregation of the individual partition-errors $E_{i,K}$ – essentially a population-weighted average of each partition's $E_{i,K}$. This generated SE series delineates a function *f(K)* ($K \in \mathbf{Z}^+$) which we may assume to be generally *non-increasing*; that is, as more partitions (and more models) are integrated, a somewhat smaller RMSE (i.e., better performance) of the approximation-system emerges. Certainly in theory as K approaches N, system error performance as a whole would approach 0 for the data instances used. However such a system is not feasible, nor is it even desirable as the ensuing gross over-fitting would generate unacceptably large errors as new data-instances were presented to the system. For most practical applications, we anticipate that K would be limited to some relatively small number (e.g., $K < 20$).

Several potential models for *f* may be rejected as unrealistic. The constant function *f(x)=Const*, for example, since observing no performance-improvement as K increases would imply no need for partitioning. We may also reject any negative-slope linear functions of the form *f(x)=-mx+b*, as this would imply that performance could reach 0 and even become negative, both unrealistic situations in the physical, real-world

monitored situations we envision. The best basic model for *f* may be the multiplicative inverse, or reciprocal function,

$$f(x) = c * 1/x + \varepsilon,$$

where ε represents the irreducible error-term associated with the dataset D (also known as the *nugget* in geostatistics). For example, this can include measurement error due to data harvesting methods (e.g., sample rate, measurement error), random noise, various hardware effects, or any other error-source in D that cannot be controlled for. Generally, to explain all variance in natural data is not feasible.

The limit of *f* as *x* approaches infinity is therefore ε, implying that in the extreme case of K = N (i.e., one model per instance-vector in D) our very best performance could be achieved. We may also observe that f is monotonically decreasing, and exhibits the properties of what economists describe as *diminishing returns* (Samuelson and Nordhaus, 2001). Naturally generating a plethora of models is clearly inefficient and would result in overfitting, thus not be reflective of the actual processes that we wish to capture. Assuming that partitioning M could render any discernible benefit, a smaller *k* than N must be found in order to implement a manageable, efficient solution for a wireless sensor network embedded within M.

Assuming that training an MLP model A_i consumes *C* resources (i.e., CPU cycles, processor-time, clock-time, etc.), we can arrive at an efficiency ratio *eff(x)* such that for a system of *x* MLP models, the average error per unit *C* is:

$$eff\,(x) = \frac{f(x)}{C \cdot x} = \frac{\frac{1}{x}}{Cx} = \frac{1}{Cx^2}$$

(Note: here the function's c_1 and ε parameters are ignored as they would become irrelevant as either C or x approach infinity).

As it happens, the actual value of *C* is largely moot as the *shape* of eff(x) will always remain the same (i.e., while the range of values generated will vary, their distribution remains identical) – a monotonically-decreasing function much like *f*, also displaying the property of diminishing returns. As *x* is a stand-in for *K* in these functions, *K* will tend to be relatively small depending on the actual value of *C*, as values of *K* yielding the best efficiency with respect to invested effort will occur before the elbow in the graph – that is, where $\Delta y > \Delta x$. Again, the Monte Carlo BestK technique described in Section 2.4.1.2 should help determine the optimal value of *K*.

4.3 Synthetic Datasets

In order to develop and test a partitioning approach we first apply it in a context in which the number of partitions is known and can be controlled. An artificial dataset containing different *phenomenological regimes* (hereafter simply referred to as *regimes*) was synthesized to investigate the potential and benefits of partitioning a monitored space *M* for improved estimation performance by a system of MLP models, each trained (or, "fitted") to each particular subregion defined by those regimes.

For both practical and ease of comparison purposes, the MLP models used in all simulations in this chapter and the next employ an identical structural design. Though MLP are capable of integrating a far greater variety of inputs than those shown, and different sets of inputs for different spatial and temporal regions would certainly make sense, such models would be much more challenging to meaningfully compare on an even footing. Accordingly, each model consists of eight input units, eight "hidden" processing units, and two output units. In keeping with the *spatial time series* approach,

the input data (i.e., *features*) provided to the model are u- and v-components of generic vector readings (presumably surface currents) for the given location at different temporal lags. The input features themselves are: *u1*, *v1*, *u2*, *v2*, *u3*, *v3*, *u6*, *v6*, where letter represents component and number represents temporal lag. The two outputs provided by the model are *u* and *v*, the components of the model-approximated reading for that location.

4.3.1 Synthetic Orthogonal Testbed

4.3.1.1 Design of Phenomenological Regimes

Consider a 21x21 unit region *M* embedded with a network of evenly-spaced wireless sensor nodes.

- The northeast quadrant (i.e., quadrant I of the Cartesian plane) is composed of an orthogonal collection of 2-dimensional vector values generated by a sinusoidal function with parameters that define a particular *regime*: vectors begin a 72 time-step period with a general heading of 0°, they rotate counter-clockwise, and they have an average magnitude of 10 units.
- The northwest quadrant (and quadrants 3 and 4) are similarly defined, except where the following phenomenological properties differ: the 72-step period starts at a general heading of 120°, rotates clockwise, and have an average magnitude of 20 units.
- The Southwest quadrant: the 72-step period starts with a general heading of 220°, rotates clockwise, and has an average magnitude of 10 units.
- Southeast quadrant: 72-step period; beginning general heading of 315°; CCW rotation; average magnitude of 20 units.

These regimes are illustrated in Figure 4.2

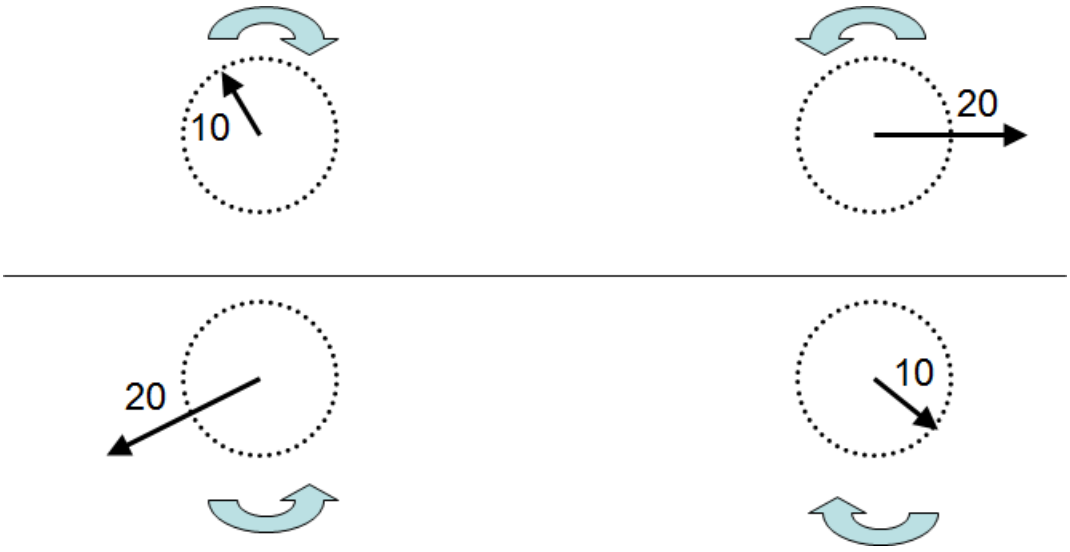


Figure 4.2: Illustration of phenomenological regimes induced into synthetic datasets

4.3.1.2 Stochastic Signal Perturbation

The data generated by the four regime-functions described above are not quite as regular as the descriptions might suggest, however. Random noise of ±5 units is added to each component of the 2-dimensional vector as it is generated. Furthermore, although each vector-heading progresses by an average of 5° in its direction of rotation, there is a 10% chance during each *individual* progression that a vector actually remains one time-step behind; there is also a 10% chance that the vector produced is actually one time-step ahead of its theoretic schedule. In this way the data produced should contain reasonable individual variability while still maintaining predictable aggregate properties.

The simulation is run for 1533 time-steps. Assuming 72 time-steps represents one day of real time (i.e., readings taken three times per hour), this represents approximately 3 weeks of operational data and generates just over 676,000 data instances upon which to fit and test our models. Given this known regime structure we next apply k-means to

evaluate the ability to detect these regimes. The data features chosen to implement clustering were: location (*x*, *y*), 2-D components from two consecutive readings (*u*, *v*) and (*u2*, *v2*), and the angular change that occurred between the two readings (*angchg*). The *u1*, *v1*, *u2*, *v2* features were chosen to their presumed importance as readings immediately preceding the model's approximated output reading; while *angchg* is a parameter derived from these four latter features which was found to be helpful in yielding empirically effective partitions.

To choose the most-appropriate k we apply a variation of the Monte Carlo approach described by (Peeples 2011) which has the advantage of providing an intuitive statistical basis for its results (Tan *et al.* 2006).

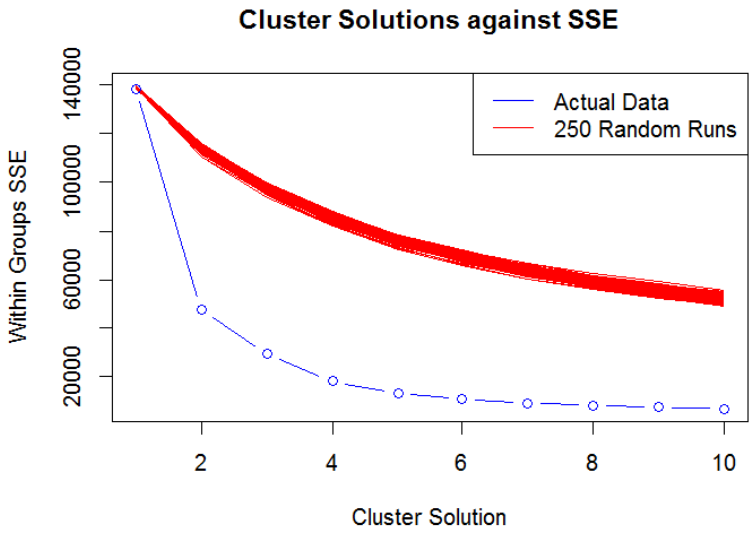


Figure 4.3: Illustration of intracluster sum of squared error vs. K as generated by (Peeples 2011)

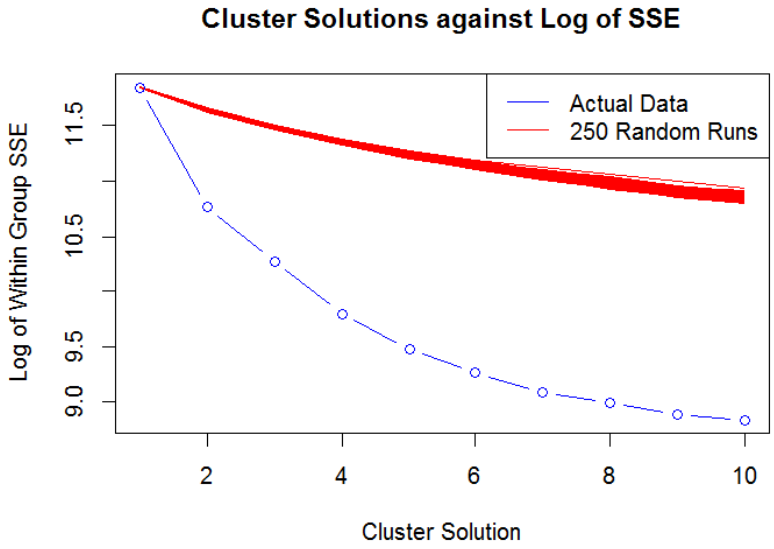


Figure 4.4: Intracluster sum of squared error vs. K as generated by (Peeples 2011) on a logarithmic scale

Figure 4.3 and Figure 4.4 illustrate the within-cluster sum of square (WCSS) error (called "Within Group SSE" in the figures) generated by Monte Carlo simulation for a potentially partition-ready dataset for the various discrete values of k where $k \in [1\ldots10]$. Figure 4.3 displays the raw results whereas Figure 4.4 shows the same information on a logarithmic scale. These function-curves are of the same form as those of timings of parallelized algorithms as additional computational cores are contributed to the process (e.g., (Guan *et al.*, 2011)).

The presumed "best" K (i.e., apparent number of "natural" clusters) by this Monte Carlo bootstrapping procedure occurs at the location in the above graphs where a distinct change in slope (referred to as an "elbow", or "knee") appears in the clustered data's SSE time-series; or stated another way, where the change in the graph's x-axis becomes greater than the change in its y-axis, rather than the other way around. Equivalently, the appropriate K is that discrete value at which the slope of the line tangent to the

normalized graph passes from a slope of *less than* negative one, to *greater* than negative one from one discrete value of K to the next. In this example, the appropriate value is 4. The vast difference in scale between the x and y axes in Figure 4.3 obscures the appropriate solution, and though much closer in scale in Figure 4.4, when comparatively scaled, the -1 slope near K=4 is much more apparent.

A visualization of a k-means clustering of values in the first five time-steps of this dataset yields the partitioning-scheme shown in Figure 4.5, which matches our expectations based on the description given above on how the "orthogonal" synthetic data was generated.

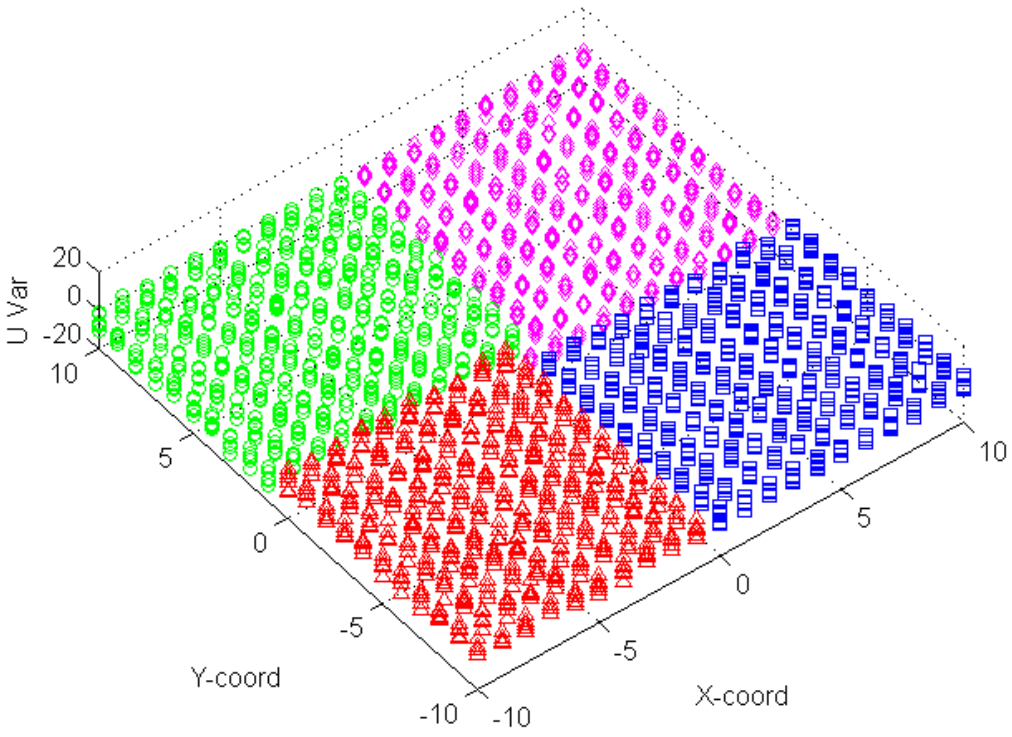


Figure 4.5: Result of a short-term k-means partitioning of the dataset with K=4

In operational terms, however, we would probably not be clustering over such a small timescale; we might instead determine a clustering-scheme for the entirety of a training-period's worth of data (let us say over the preceding three-week period, in order to apply the resulting scheme for the following three week period). Then, as new readings come in, they are assigned to the appropriate model according to this predetermined clustering scheme. A few snapshots of a K=4 clustering scheme, partitioning a three-week span of data, are illustrated in the sequence in Figure 4.6.

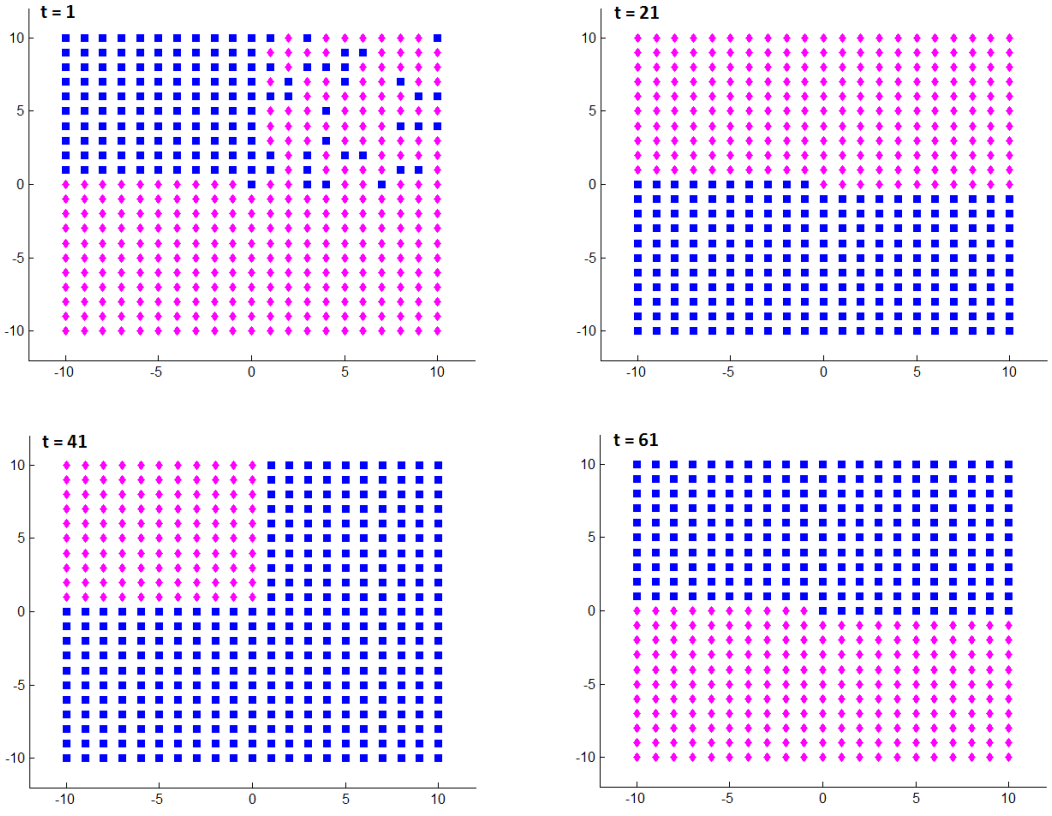


Figure 4.6: Synthetic Orthogonal dataset clustering at time-steps 1, 21, 41, 61

Like a snake swallowing its tail, the two largest-population cluster IDs (1 and 3, respectively) appear in Figure 4.6 to pursue each other in a counterclockwise fashion as time goes on. Indeed, the partitioning that is taking place here will be seen to be due

largely to a particular location/measurement's state – particularly current-bearing – within the context of its local regime's periodic variation.

But what happened to the other two cluster IDs?

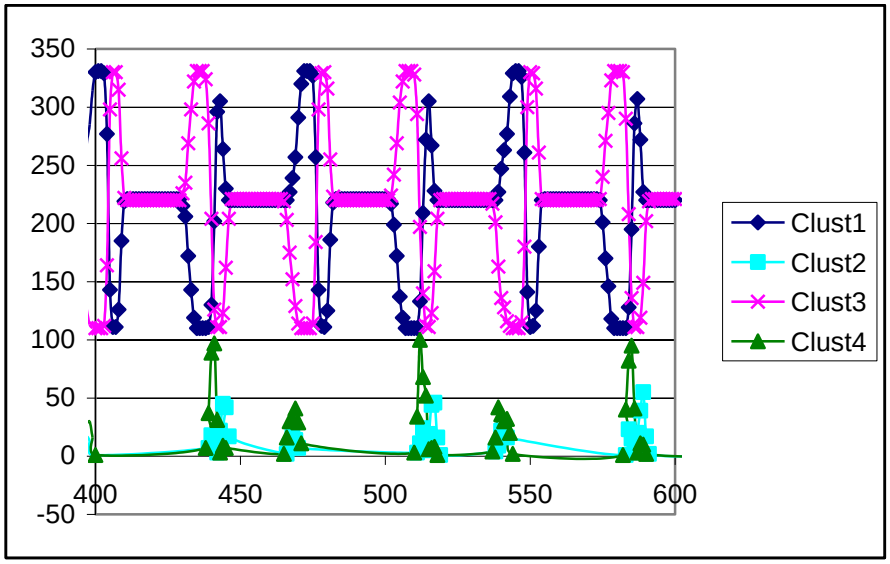


Figure 4.7: Time-series of cluster populations over time

The other two clusters can be seen to appear only periodically (i.e., interstitial clusters), usually in transitionary periods when the two major clusters change state, as can be observed in the waxing and waning cluster-population series presented in Figure 4.7. One might be tempted to do away with these relatively-tiny interstitial clusters as the mass of data instances are classified within clusters 1 and 3. However if we did so, our resulting value of *K* would be situated prior to the natural elbow in the Monte Carlo BestK graph (as illustrated in Figure 4.4), and thus be suboptimal.

Our intuition resulting from the initial short-timescale clustering in Figure 4.5 was that physical location would dominate in making the cluster assignment. However,

determining partitions over such a limited time-scale in effect removes the influence that *time* brings to the situation. The results in Figure 4.6 show k-means converging onto an admittedly less-intuitive scheme, but which incorporates time's effects on the partitioning-scheme: by minimizing member-instance distances while maximizing inter-cluster distances in 7D space.

Unfortunately, though each data instance is classified as one of the k classes, and the k cluster-centers are returned by the kmeans process, no intuitive explanation is provided for *why* a particular instance received the classification it did. This is where the random forests and treebagging techniques enter the picture, as they develop a (linear) regression-fit between the set of instances and the resulting class-IDs. From this fit, the apparent weight of each instance's parameters may be determined in what is described as out-of-bag (OOB) influence factors. The relative influence of the chosen parameters (i.e., *features*) to the k-means model's results (as determined by this *TreeBagging* factor-analysis) appear in Figure 4.8.

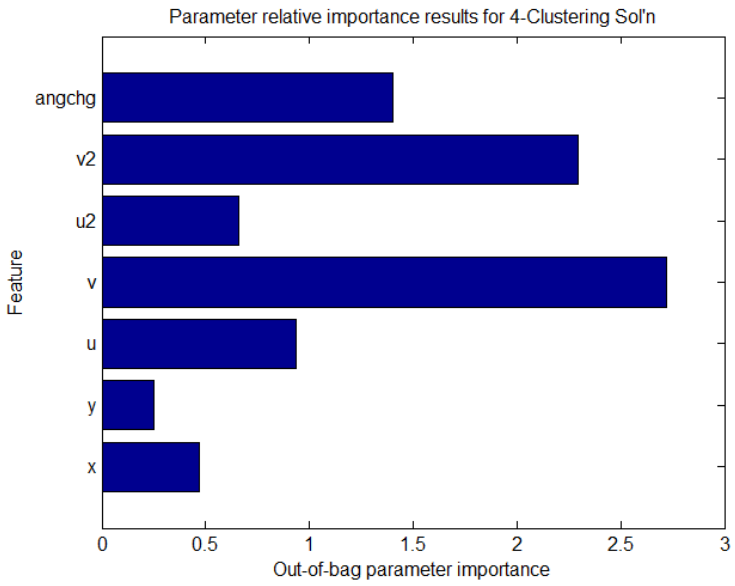


Figure 4.8: TreeBagging results of dominant terms in orthogonal dataset's K=4 clustering solution

4.4 Synthetic Orthogonal Results

Having generated over 676,000 data instances as described above, a *time series* of readings is extracted from the database that include records containing u- and v-components of the 2-dimensional synthetic readings at location (*x*,*y*) and for times steps occurring 1, 2, 3 and 6 time-steps previously

These records are then divided into three groups for artificial neural network model training, testing and simulation, as is traditionally done for these models. In short, the *training* dataset is used to train randomly-initialized MLP models; the *test* set consists of data from the same distribution as the training set, and is fed through the models-in-training to verify that they are not *overfitting* to the training data. Once the models are found to no longer be generalizing to the underlying processes represented by the data instances, and instead beginning to overfit to the training data (i.e., the observation that

mean model error for the test set is significantly greater than mean model error for the training set, for example), training stops.

In this particular case, records prior to time=100 were randomly allocated either into the training set or the testing set, leaving the rest – some 631,000 records – for the *simulation* set. This simulation set was run through a single MLP model trained on the entire data set ("Single Model") to generate a root mean square error result, shown in the second row of Table 4.1.

The results of the K=4 *kmeans* process described by Figure 4.8 are then used to partition all three of the aforementioned MLP datasets into four subsets, corresponding to clusters 1, 2, 3, and 4. Having trained the four models using the partitioned training and test datasets, the RMSE results found in the third row ("Partitioned Model") are generated by the evaluation of the similarly-partitioned simulation sets, for the overall aggregate RMSE found in the last column. The final row in the table contains the populations of the simulation set's four partitions, and is shown simply for informational purposes.

Table 4.1: Results of Single model error vs. Partitioned model error on Orthogonal Dataset

	Clust I	Clust II	Clust III	Clust IV	Overall
Single Model error					2.951
Partitioned Model error	2.851	1.907	2.770	2.013	2.790
Population	306,599	6,965	309,839	8,109	631,512

The results in Table 4.1 indicate an average RMSE difference of 0.161 units of error between the Single MLP Model's performance in processing the entirety of the dataset on its own, and the joint result of four MLP models each trained for one particular

partition of that same dataset. Though the difference is small, a two-tailed 2-sample t-test pairing each result from the Single Model with the corresponding result provided by the 4-part Partitioned Model system rejects the same-mean null hypothesis (i.e., $\mu_{Partitioned} - \mu_{Single} = 0$) at a 99% significance level for N = 631,512 instances (indeed, the *p-value* for this null hypothesis is less than 10^{-4} for this paired test). As this difference is negative, the partitioned model is shown to generate somewhat smaller error overall than the single model.

Table 4.2: Sample readout of Hypothesis test results

```
Overall Paritioned-model improvement: -0.161


***Non-normal Distribution!***
 sng vs. part: 2-sample nonParam KolmogorovSmirnov test(.99):


Null Hypo (that Global and Part datasets come from the same distributions)
rejected at .99 certainty! (no normal distr assumed; p-score: 0.0000)


 Mean Per-Location Difference: -0.16


Standard Paired t-test:
Null Hypo (of: Part-Global mean = 0) rejected at .99 certainty!


2-sample Null Hypo (that SNGerr and PARTerr come from same distributions, same
means) rejected at .99 certainty! (normal distr assumed; p-score: 0.0000)
```

Results of the hypothesis tests displayed in Table 4.2 demonstrate that both non-parametric (i.e., Kolmogorov-Smirnov) and the more traditional Student t-tests confirm that the difference is too great between the means of the Single vs. the Partitioned model

approaches to be considered essentially the same at a statistical significance-level of α=.01. Although given the size of the populations involved, the Central Limit Theorem of statistics would find the Student's t-test sufficient, the obvious regionality and non-normality of the results displayed in Figure 4.9 – Figure 4.11illustrate why the confirmation of a non-parametric Kolmogorov-Smirnov hypothesis test was desired in addition to the Student results.

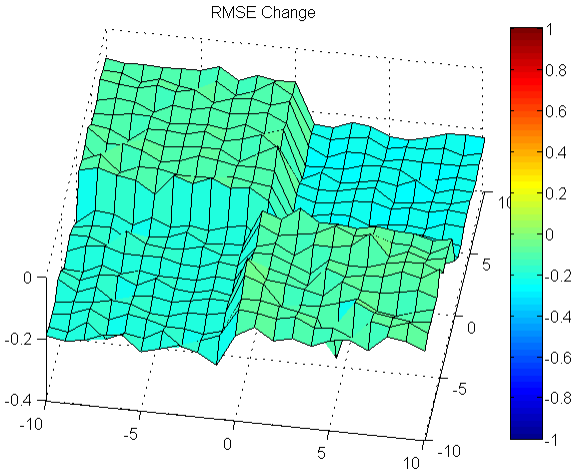


Figure 4.9: Spatial distribution of mean error-reduction resulting from use of partitioned model

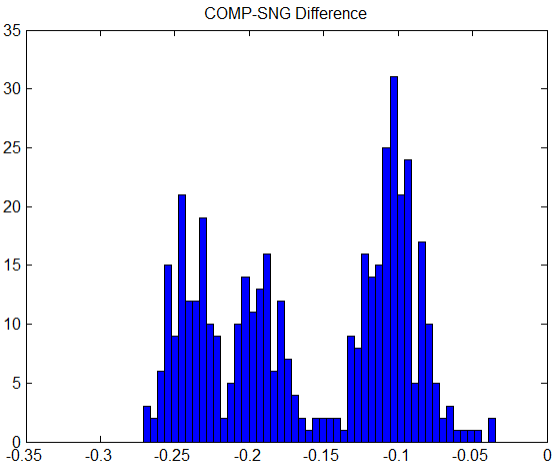


Figure 4.10: Non-normal histogram of partitioned-model's mean, per-location error reduction

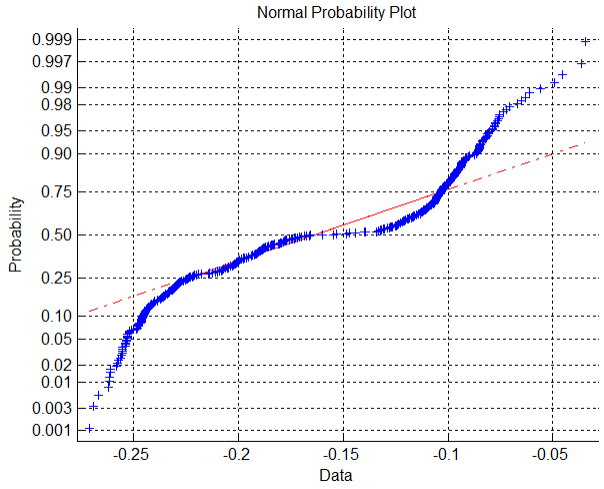


Figure 4.11: Non-normal distribution of partitioned-model's mean, per-location error-reduction

As the error-distribution was non-normal, results of the standard Student's t-test may technically not be applicable. However as all mean errors are negative, it is clear that use of the partitioned model was a significant improvement over the single global model.

These results, although satisfactory, beg two questions: are these the best results that could have been achieved? And perhaps more pertinently, are these the best results achievable for the effort expended? After all, generating more clusters nearly always improves overall WCSS, but the amount of incremental improvement tends to be very small when partitioning into more than the number of natural clusters. In other words, had this artificial dataset not come with the foreknowledge that only four 'natural' clusters existed, leading one to commission a K=4 partitioning scheme, would K=3 not have worked just as well? Might not K=5 possibly have done better? Having no prior knowledge of this dataset, why should we have chosen K=4? Support for the decision to set K=4 is delivered by the MCBestK analysis, an intuitive heuristic for finding an appropriate K for use in k-means (Peeples, 2011) (Tan *et al*. 2006).

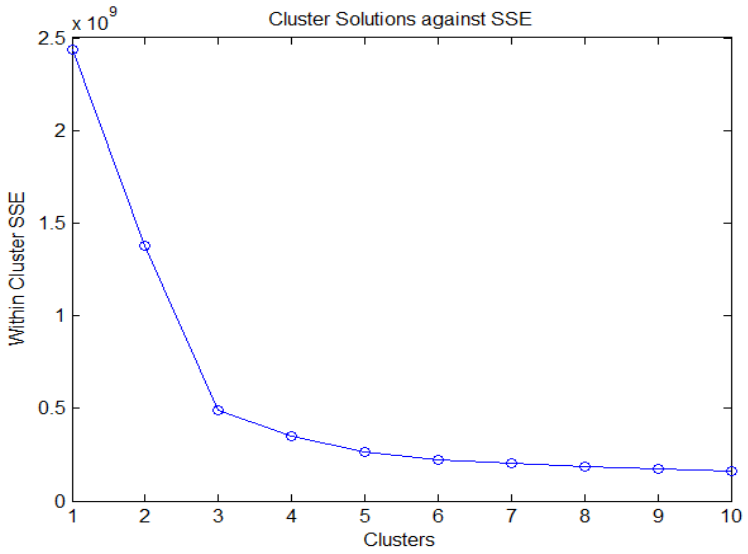


Figure 4.12: Spatial-temporal WCSS vs. Number of Clusters

Figure 4.12 illustrates how the within-cluster sum of squared (WCSS) errors drops as K increases (i.e., more partitions/clusters are created out of the dataset of readings). It is generated by successively subdividing the dataset of readings into K clusters, where K ranges from 1 to some maximum value presumably larger than the "natural" number of underlying processes, and looking for the "bend" in the graph (also occasionally referenced in the literature as the "elbow" or the "knee"). It should be apparent that if this same graph existed within a normalized unit-space (i.e., both axes range over: [0…1]), that bend would occur approximately where the slope of the tangent line (i.e., first derivative) is equal to –1. The location where that slope occurs is thus the critical point which will properly determine our "best" value of K: BestK. Normalizing the axes allows for an accurate localization of this critical point. A simple scaling example for Figure 4.12 follows.

Table 4.3: Normalizing the axes to approximate BestK in Figure 4.12

Normalize x (i.e., K) and y.

	x	y	nx	ny	m
	1	2.436E+09	0	1	
	2	1.380E+09	0.111111111	0.536360566	-3.852812457
	3	4.871E+08	0.222222222	0.143819454	-2.032823527
	4	3.523E+08	0.333333333	0.084622004	-0.441317058
	5	2.639E+08	0.444444444	0.045748997	
	6	2.246E+08	0.555555556	0.028480578	
	7	2.045E+08	0.666666667	0.019658532	
	8	1.854E+08	0.777777778	0.011287627	
	9	1.705E+08	0.888888889	0.00471989	
	10	1.597E+08	1	0	
min	1	1.597E+08			
max	10	2.436E+09			

Formulae:		
	$norm_x$ (nx)	$= (x-min_x) / (max_x - min_x)$
	$norm_y$ (ny)	$= (y-min_y) / (max_y-min_y)$
	m_x	$= (ny_{(x-1)} - ny_{(x+1)}) / (nx_{(x-1)} - nx_{(x+1)})$

The presumed "best" *K* for this case is between 3 and 4, since we traverse m = -1 between those two values. Forced to choose, we should select 4 in this case. Again, although a visual inspection of the graph shown in Figure 4.12 might tend to suggest that BestK is 3, the approximate m in Table 4.3 suggests that at $K = 3$ the goal of $m = -1$ may not quite have been achieved.

Although the choice of $K = 4$ may have appeared obvious in the table due to $f'(4)$ being closer to -1 than $f'(3)$, in many cases we might be tempted to select the smaller of the two discrete x-values due to its $f'(x)$ being closer to the critical slope value of -1. This might be a correct impulse were the graph linear. But careful analysis shows that the model-function's nonlinearity causes far more error when using this otherwise intuitive-seeming latter method, than the sum of over- and under-estimates of *K* produced by simply choosing the larger of the two every time. It can be shown relatively easily that for a graph of the form $f(x) = C * 1/x$, due to Δy decreasing monotonically (and nonlinearly) as *x* increases, the correct choice of *K* will be the larger of the two discrete values over 50% of the time, and thus is most often the better choice.

To prove this, we shall first demonstrate that the slope (i.e., $\Delta y/\Delta x$) of functions of the type $f(x) = 1/x$ changes more quickly in the *first* half of the unit interval (i.e., bounded by $[d \ldots d+1]$, where d is a positive integer) than in the second; therefore, more than half of the interval yields slope-values that are closer to $f'(d+1)$ than $f'(d)$. Then as a corollary, since more than half of the unit interval is claimed by $d+1$ we propose that, of the two endpoints, $d+1$ should most often be selected as BestK.

Lemma 4.1 *For any unit interval between two integers d and d+1, $x_{cv} < d+0.5$ where $f'(x_{cv})=(f'(d)+f'(d+1))/2$ for functions of the form: $f(x) = c * 1/x + \varepsilon$.*

As a first example, consider a function f where $f(x) = c * 1/x$, and an unit interval $[d \ldots d+1]$ where $d \in \mathbf{Z}^+$. To keep matters relatively simple for now, let both c and $d = 1$, and $\varepsilon = 0$.

Now $f'(x) = -1/x^2$, resulting in slopes at either end of the interval of -1 and -1/4 respectively. As these two slopes are separated by a distance of 3/4, the midpoint between these is: -1 + 3/8 = -5/8. If $f'(x)$ were linear over the interval, d + 0.5 would have a slope of -5/8. However according to f', that critical value occurs where $f'(x) = \frac{-1}{8/5}$, or where $x_{cv} = 1.6^{0.5} \approx 1.265$, which is less than 1.5 . Because the midpoint of f'(x) over the interval occurs earlier than the midpoint of d and $d+1$, we know that $\Delta y/\Delta x$ is greater toward the beginning of the interval than toward the end. Since the slope changes more quickly at the beginning of the interval than towards the end, we cannot simply choose the endpoint x with the smaller distance to our desired critical-value x_{cv} and always be

correct that it is the "closer" of the two endpoints. Most of the interval [1, 2] (~73.5% in this case) maps to *f'(x)*-values that are *actually* closer to *f'(2)*.

Now consider the general interval [*d*…*d+1*].

Slopes at either end of this interval are: $\left[-1/d^2 \ldots -1/(d+1)^2\right]$.

The midpoint between these is: $-1/d^2 + \frac{2d+1}{2d^2(d+1)^2} = -\frac{2d^2+2d+1}{2d^2(d+1)^2}$.

Thus if $f'(x) = -\frac{1}{\frac{2d^2(d+1)^2}{2d^2+2d+1}}$, the critical point where Δy = -Δx (i.e., the slope Δy/Δx = –1) is situated where $x_{cv} = \sqrt{2}\sqrt{\frac{d^2(d+1)^2}{2d^2+2d+1}}$.

Consequently, if *c* were different from 1 (and positive), the critical point would occur where: $x = \sqrt{2}\sqrt{\frac{d^2(d+1)^2}{2d^2+2d+1}} \cdot \frac{1}{\sqrt{c}}$, or: $x\sqrt{c} = \sqrt{2}\sqrt{\frac{d^2(d+1)^2}{2d^2+2d+1}}$. Because the slope-value of –1 occurs in Figure 4.12 beyond x=1 (where it would have occurred for *f(x) = 1/x*), we know that *c* will be both positive, and less than one.

This property of *f(x)* may be represented as the expression: $\sqrt{2}\sqrt{\frac{d^2(d+1)^2}{2d^2+2d+1}} < d + 0.5$.

Where is this expression true? To determine that, we simplify the inequality as follows:

$$\sqrt{2}\sqrt{\frac{d^2(d+1)^2}{2d^2+2d+1}} < \frac{2d+1}{2}$$

$$\frac{2d^2(d+1)^2}{2d^2+2d+1} < \frac{(2d+1)^2}{4}$$

$$8d^2(d+1)^2 < (2d^2+2d+1)(4d^2+4d+1)$$

$$8d^4 + 16d^3 + 8d^2 \;<\; 8d^4 + 16d^3 + 14d^2 + 6d + 1$$

$$0 \;<\; 6d^2 + 6d + 1 \quad \therefore$$

The final inequality is obviously true for all d ≥ 1. An inequality plot (Figure 4.13) shows the situation graphically, suggesting an effective range of -0.211325 < x.

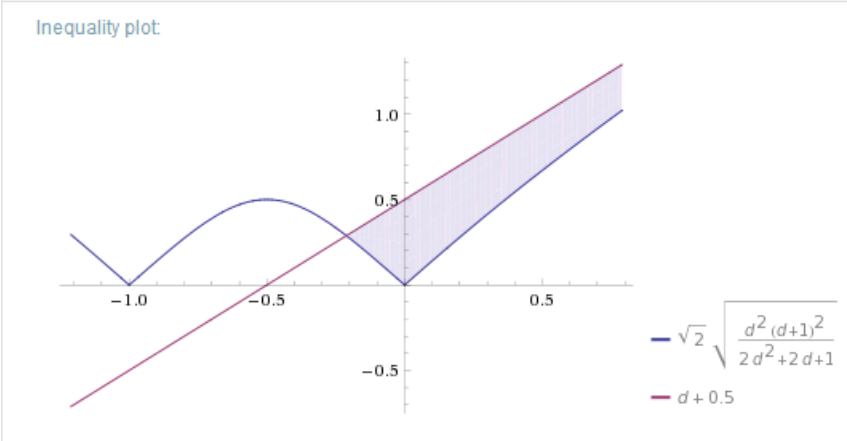


Figure 4.13: Inequality plot of: f(x) < (d + 0.5)

Corollary 4.2 *Where K falls between two integers d and d+1, correct discrete values of K are more likely the larger discrete endpoint of the unit interval than the smaller for functions of the form: f(x) = c * 1/x + ε.*

As $K \in d \in \mathbf{Z}^+$, Lemma 4.1 shows that d+1 is always the more probable choice of the two interval endpoints for K, as more of *f'(x)* over the unit interval is closer to *f'(d+1)* than to *f'(d)* for all positive K.

The main takeaway of this corollary is simply that within the normal operating parameters of one's monitoring system, should one be disinclined to fit the empirical function derived via the MCFindBestK process, one should choose to preferentially round K upwards rather than downwards, as it is shown that more than 50% of the unit space between any two discrete x-values is closer to *f'(d+1)* than *f'(d)*. Let us assume that the average parameters for an operational system consistently work out to a BestK confined within the unit-space between the values d and d+1 of the function f where 55% of the unit's f'-values were closer to f'(d+1) rather than f'(d); that is, rounding K upwards to d+1would be correct only 55% of the time. What would be the effect for the remaining 45% of the time, were we to round *K* to *d+1* when the "true" value of *K* is actually closer to *d*? According to the empirical function modeled by 1/*x*, the result of using a higher *K* than required would generally be somewhat *lower* system-wide error, in exchange for having created one more partition and trained one more MLP model than strictly required. So as long as the time and processing used for the additional partitioning/training process is not somehow a critical factor, the effect of simply rounding *K* upwards 100% of the time should not be detrimental to system performance as it should still generally *improve*, albeit at the cost of some additional time and processing, although probably not to any degree that the resulting likely increase in performance would be deemed significant.

As a final piece of evidence that the presumed *f(x) = 1/x* error model is appropriate – as well as that the MCBestK process has resulted in an appropriate value with *K*=4 (see Table 4.3) – we may observe in Figure 4.14 that, although the increase in number of

partitions does reduce overall error, the error-reducing effectiveness of each new partition (i.e., the effort-ratio) – or in other words, the effectiveness of training each new MLP-model participating in the system – is almost entirely dissipated after *K*=4, the determined BestK. Indeed, it is nearly flat beyond *K*=5. Although additional partitions/models will tend to reduce overall RMSE further, the effort involved to do so may not be worth the rapidly-diminishing marginal returns.

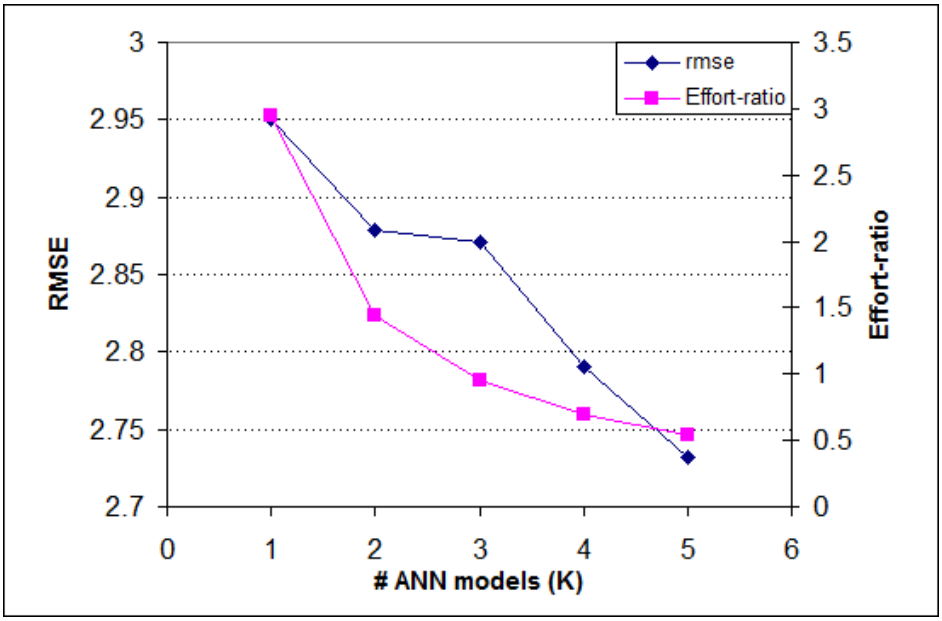


Figure 4.14: Root Mean Square Error and Effort-ratio as K increases

Figure 4.16 a shows a time-series of *instantaneous* BestK results over 300 timesteps. Here the best *K* is selected only for the subsets of readings generated at each time *t*. The disadvantage of this approach is that the influence of time is taken out of consideration, and so the best *K* found is a *spatial* bestK, which may be different from the *spatial-temporal* bestK. One advantage to this approach, however, is that one can detect a point in time when the most appropriate value of *K* may have switched from one value to

another. For example, in the situation represented in Figure 4.15, between timesteps 200 and 300 the process generating the values for quadrant III of the orthogonal synthetic

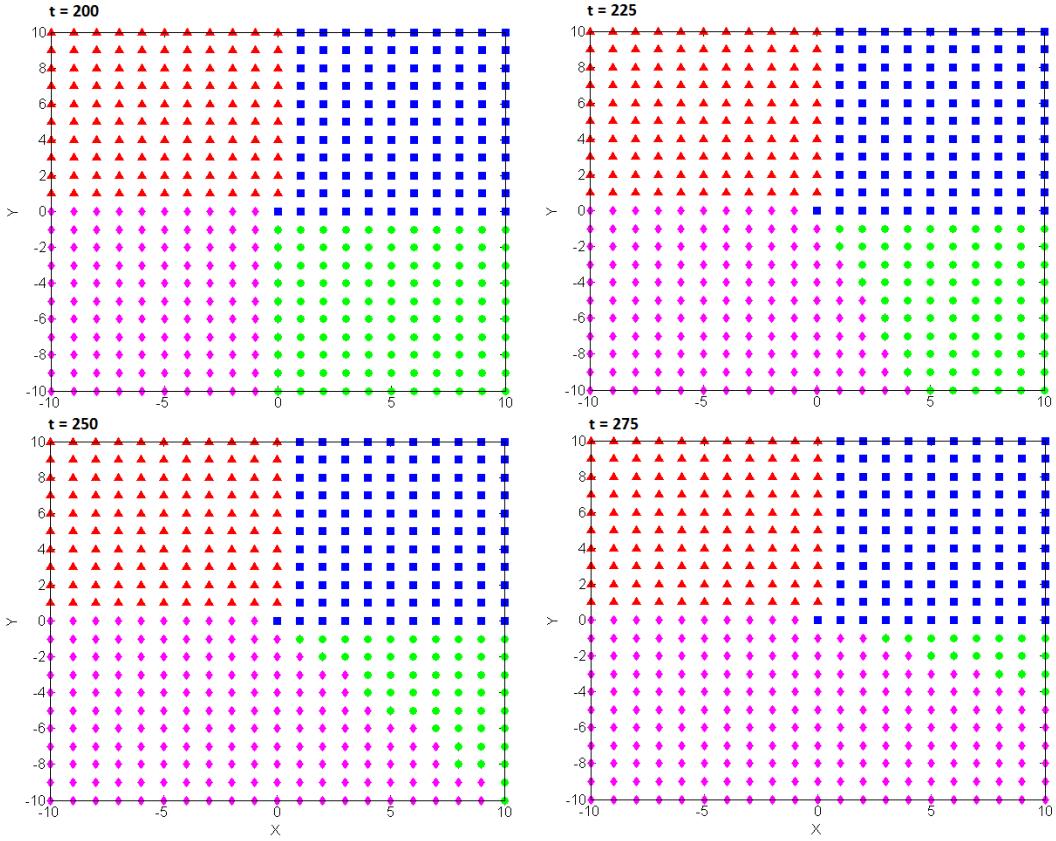


Figure 4.15: Example of change in K

dataset is gradually extending its spatial range into quadrant IV. By t=300 process III has wholly subsumed process IV, and is generating all synthetic readings for both quadrants III and IV. Whereas the instantaneous approximations of bestK vary between 3, 4 and 5 over the first 200 steps, 5 has disappeared as a possibility by time t=300; and whereas 4 was the most-frequent result returned over the first 200 steps, the mode of any *n* consecutive values of the time-series after *t*=250 is much more likely to (correctly) return 3. Such a sustained change in the central tendency is a clear indication in this situation that if the running model-system is still 4-part, it may be worthwhile to check if it should

be reconfigured as a 3-part model-system. Usually, it will not be worthwhile, as graphs such as Figure 4.12 consistently suggest that more partitions yield lower error (although if the "natural" *K* is *d*, *d+1* partitions will not be expected to yield a *significantly* lower error performance).

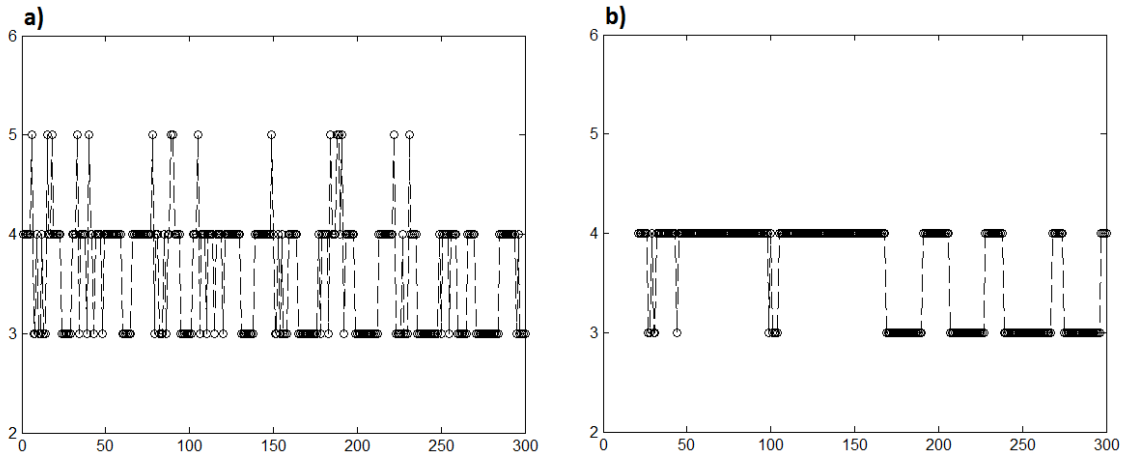


Figure 4.16: a) Time-series of instantaneous Best-K determinations, and b) the $Mode_{20}$-smoothed series

As may be expected of any large collection of readings generated with occasionally-imperfect sensors (or in this case, a synthetic dataset with some added stochasticity), contradictory signals within the dataset leads to a noisy output in Figure 4.16a. Having generated the data, we know that (before process III's generated readings start encroaching into quadrant IV from *t*=200 onwards) there should be only four distinct natural clusters detected before *t*=200. Of course, it is possible that our synthetic processes are not as distinct as we thought and so perhaps three *could* be a reasonable result (note: this is why a confirmatory run of the Monte Carlo process using the entirety of the dataset at once, rather than a series generated by using only those data occurring at each value of t, as seen in this figure, may be necessary). A measurement of the central

tendency of the resulting time-series, such as the mean or mode over a moving window of ten or twenty consecutive results at a time, can be useful in processing the results of our generated instantaneous-BestK series (Figure 4.16b).

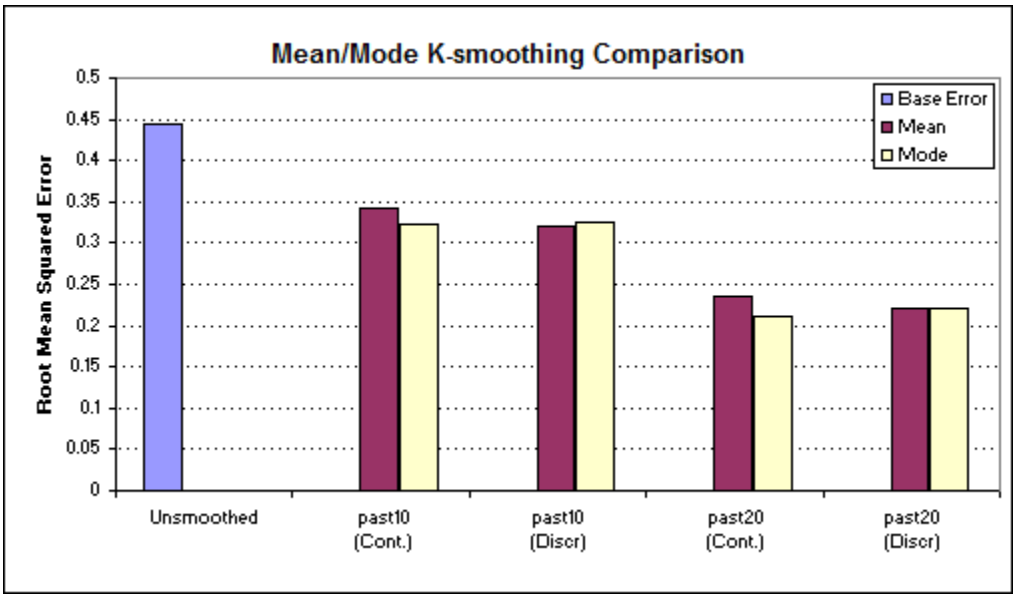


Figure 4.17: *Mean*- and *Mode-Smoothed* Best-K error-rates

Figure 4.17 displays the RMSE improvement over instantaneous results ("Unsmoothed") resulting from smoothing the time-series in Figure 4.16a using either mean or mode, over either the prior ten or twenty readings. A distinction is made here between *continuous* and *discrete* values of K (as a continuous fuzzy-logic approach might evaluate, when process III has overtaken half of IV's territory, that $K = 3.5$). What we can take from this figure is that while both mean- and mode-smoothed results reduce error on average as compared to the unsmoothed result, the mode may perform slightly better in the continuous case.

4.5 Orthogonal Dataset Conclusions

Through use of the synthetic orthogonal dataset, the prior sections demonstrate that partitioning a region does lead to a significant – if small – reduction in the average error produced by a system of models over that of a single model assigned to the entire area.

The temporal extent of the analyzed dataset has a potentially significant influence on the result of a spatio-temporal k-means clustering. Where the effect of time is minimized (i.e., clustering over short time-periods, as in Figure 4.16, the best *K* may not necessarily be the same as when a clustering is executed for a body of readings taken over the *entire* time-period (In our orthogonal test example, however, BestK was still determined to be 4 in either case).

The next section investigates experiments on a more naturalistic 4-process synthetic dataset. Similar analysis as above of instantaneous Monte Carlo clustering results suggest K=4 at each discrete time t, but a partitioning over the entirety of the data (i.e., all timesteps and thus including the effect of time) yields, instead, a BestK result of five.

4.6 Synthetic "Natural" Dataset

4.6.1 Creating the Synthetic "Natural" Testbed

For this more naturalistic synthetic data set, the same four distinct behavioral regimes from the Synthetic Orthogonal dataset were applied within more naturalistic spatial boundaries. The initial color-coded situation (at time t=0) may be described as: two oval-shaped spatial regimes sharing the same 21x21 unit region *M* with a third regime limited to the region behind a linear boundary; and a fourth regime filling in the spaces between the first three. This situation is illustrated in Figure 4.18.

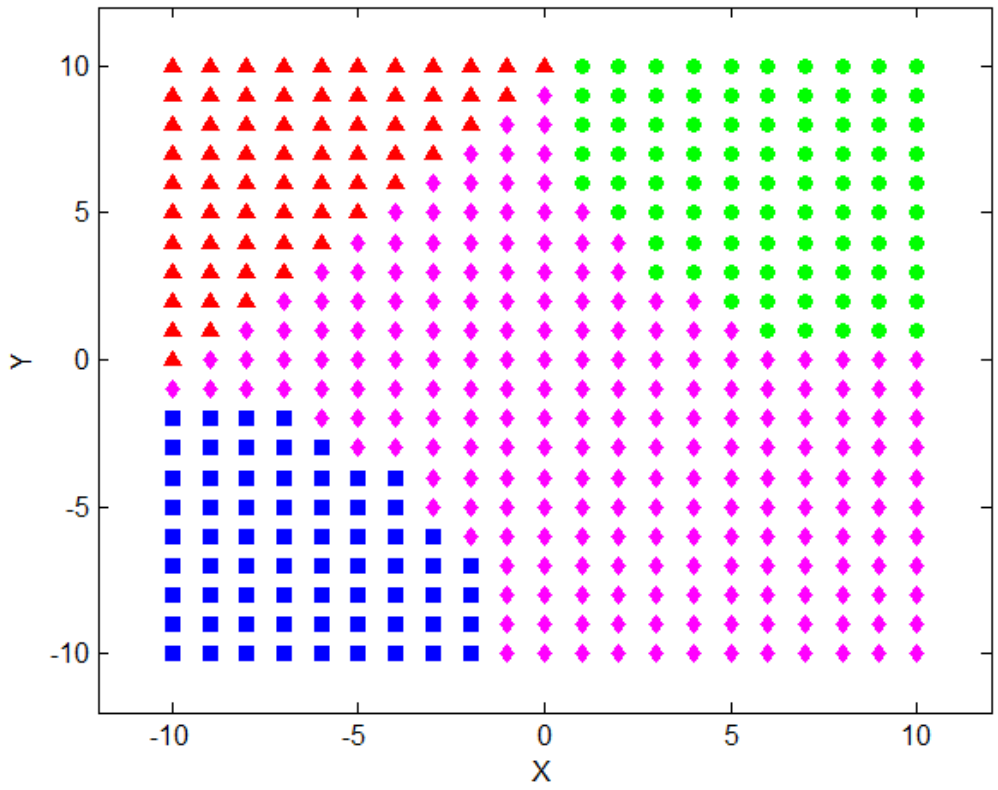


Figure 4.18: Snapshot of regimes in region *M* at time t=0

The four behavioral regimes in this dataset are identical to those in the first synthetic ("orthogonal") dataset (except that they are no longer confined to quadrants of the Cartesian plane – Figure 4.2). That is:

- a 72-step period 2-dimensional vector signal that starts at heading 0° and rotates counterclockwise, with average magnitude of 10 units
- a 72-step period signal that begins at heading 120°, rotates clockwise, and with average magnitude of 20 units
- a 72-step period signal that starts at heading 220°, rotates clockwise, and has an average magnitude of 10 units
- and a 72-step period signal beginning at heading 315°, with counterclockwise rotation, and average magnitude of 20 units.

Stochastic perturbations again ensure that magnitude and bearing of all members of the same region exhibit the identical magnitude and bearing at all times, yet are generally similar enough to be classified within the same cluster. A sample of the code generating these readings can be found in Appendix A.

Over the course of the simulation, the oval regime (denoted in the figure by green circle markers) beginning in the northeast corner of monitored region *M* migrates gradually southwards, changing in shape and topology as it descends. It can be seen to transition through a toroidal (doughnut-like) shape, temporarily enclosing a portion of regime 4 within it, before coalescing back into an closed, oval region and remaining static throughout the remainder of the simulated timespan (this topological progression too is determined by the code in Appendix A). Again the simulated timespan is meant to

approximate a 21-day period with readings taken three times per hour. In total, some 676,000 synthetic instances are generated for subsequent clustering and analysis. Snapshots of the aforementioned progression are shown in Figure 4.19.

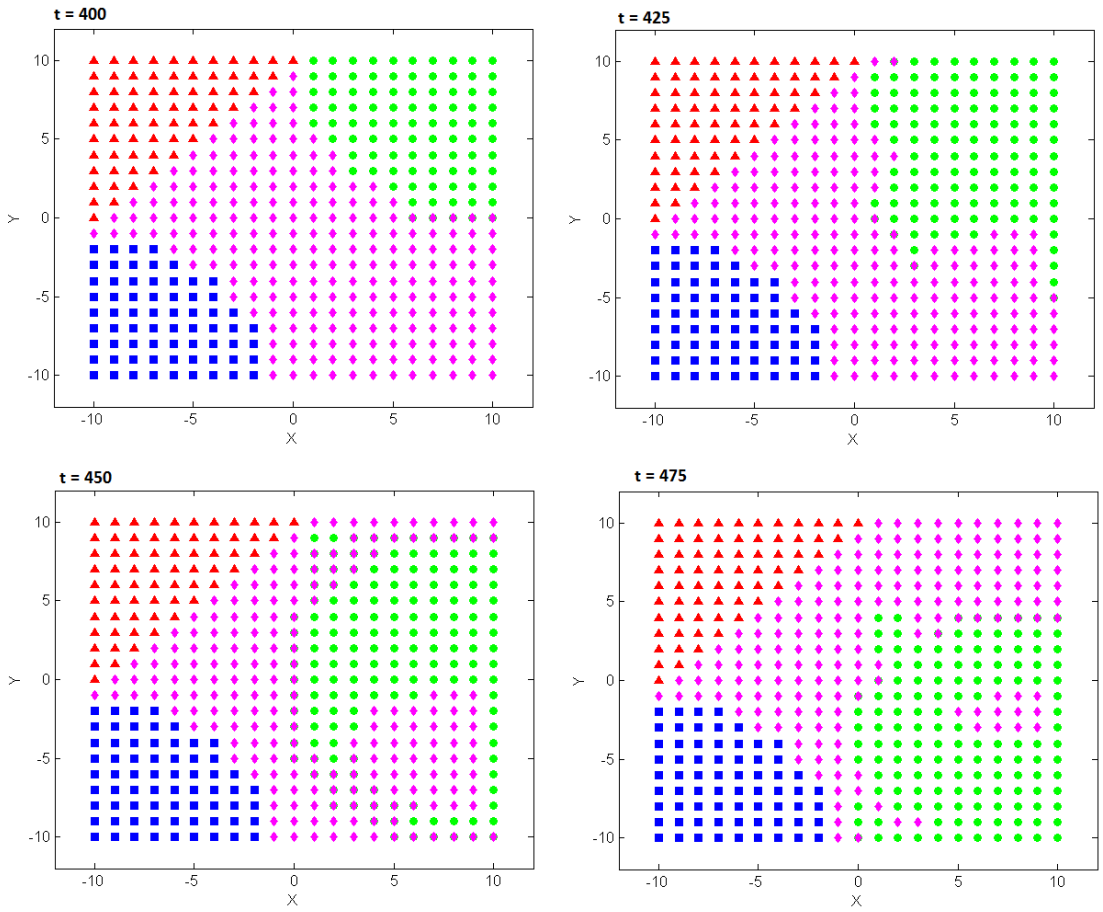


Figure 4.19: Snapshots of the progression of the green-circle partition over time

Figure 4.19 displays, at four particular points in time, which regime is responsible for populating a particular location with a synthetic datum. As there are only four regimes then a perfect analysis of the spatial signal-set over time would result in an optimal *K* of 4, with each partition mapping to exactly one of the color-groups in this figure. Without access to perfect information, however, this will not be the case.

As was done with the orthogonal dataset earlier, the optimal number of partitions (*K*) for this new "Natural Partition" (NP) dataset is determined via the Monte Carlo simulation method, clustering all available data instances on the same basis of location (*x*, *y*), two prior pairs of consecutive surface-current velocity components in that location (*u*, *v*, *u2*, *v2*), and the resulting angular delta-change of the velocity vector in that Δt (*angchg*, or angleΔ). Again as in the preceding section, *TreeBagging* is used to approximate the weight of each parameter (or feature) in each instance in determining its assigned class-id. These relative weights are shown in Figure 4.20.

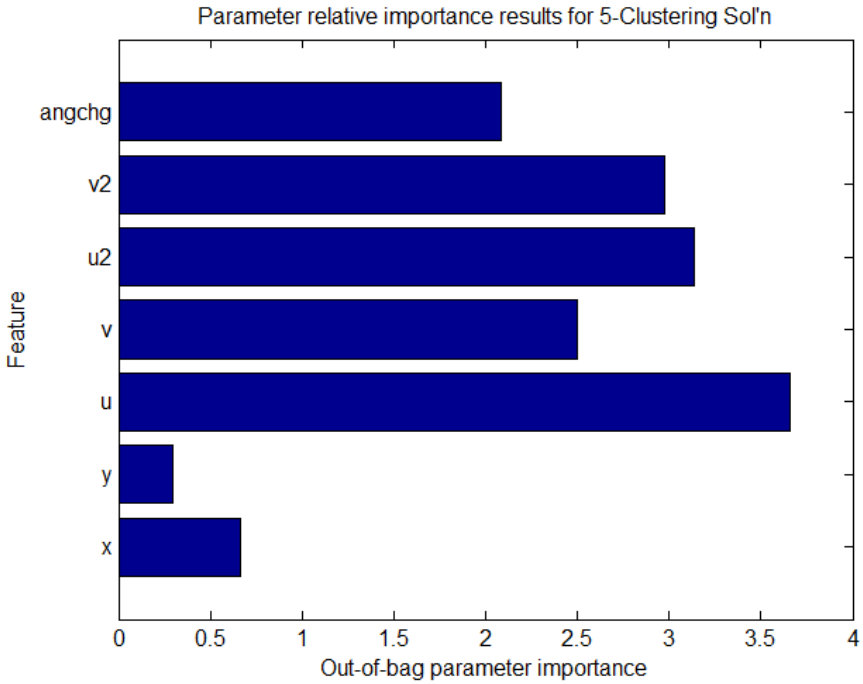


Figure 4.20: Feature-weights of the K=5 clustering solution as determined by random-forest tree-bagging

Interestingly, as with the orthogonal dataset in the prior section, the x-y location of the data instance is greatly de-emphasized by the results in Figure 4.20 in relation to the importance of temporally-changing magnitudes of the velocity-vector's components, and resulting angular rotation. This would seem to indicate a partitioned model whose

components specialize on vectors of a particular *bearing* – regardless of location – as time evolves. This conclusion is supported by biplots of primary component analyses (PCA) of the data set with its resulting cluster centers (Figure 4.21). Note that PCA analysis almost mirrors the TreeBagging technique's findings of relative importance of velocity vectors and *angchg* with respect to location (observe that *x* and *y* are near the origin, contributing almost nothing to the instance's classification). We can observe that in PCA's estimation, the *angchg* feature's contribution happens to be of greater magnitude than raw velocity component values, contrary to TreeBagging's conclusion. Different conclusions resulting from linear (PCA) and nonlinear (Random Forest) analyses is fairly common, simply due to base assumptions of each approach. But this change in relative valuation between *angchg* and the vector components is of little practical concern, as the former is derived from the latter in any case.

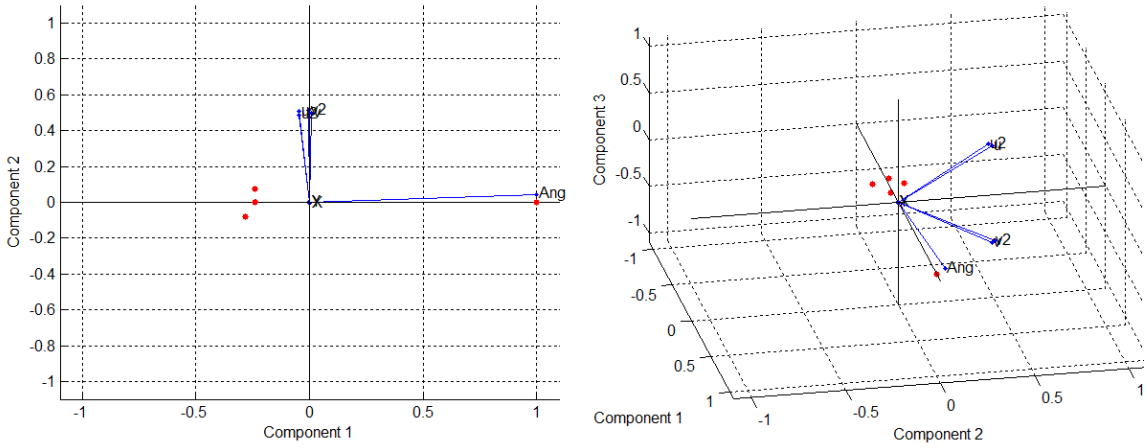


Figure 4.21: 2D (left) and 3D (right) representations of kmean cluster centers, and feature-contributions to instance classification. Note the 90° CCW horizontal rotation effected between left and right views to aid visualisation, with the z-axis now expressed along the main vertical axis in the 3D view.

Figure 4.21 also shows the K=5 cluster-centers within the same unit-space. As predicted, the four largest clusters are largely bearing-influenced – visualized as the four

points orbiting the negative x-axis – as well as a tiny cluster of instances (part2) located along the positive x-axis, mainly determined by the *angchg* feature. We theorize that this partition was created due to the stochastic nature of the generated data, wherein any particular vector had a 10% chance of being either early or late, causing glitches in an otherwise regular progression of angular delta-changes.

As may be seen in the figures above, the seven parameters chosen for the k-means clustering process are spatial (*x*, *y*), temporal (*u*, *v*, *u2*, *v2*), as well as one derived from the temporal readings (for the reader's convenience, the derived parameter – denoting the angular change between the readings (*u*, *v*) and (*u2*, *v2*) – is determined in Matlab by the following formula: `angleΔ = (atan2(v, u) - atan2(v2, u2)) * 180/pi`).
For K=5, the cluster-centers in Table 4.4 were returned:

Table 4.4: Cluster centers resulting from kmeans processing of second synthetic dataset for K=5

cluster	**x**	**y**	**u**	**v**	**u2**	**v2**	**angleΔ**
c1	0.125	0.017	-10.153	11.633	-10.549	11.287	-1.661
c2	-0.598	-0.734	-16.718	1.793	-16.682	-1.829	346.862
c3	0.118	0.013	-11.358	-10.588	-10.965	-10.821	-14.490
c4	-0.082	0.049	10.179	-11.352	10.544	-10.998	-1.563
c5	-0.123	-0.037	11.378	10.090	11.022	10.479	-1.697

A qualitative analysis of the values in Table 4.4 suggests that spatial inputs (x, y) appear to have less weight in cluster-assignments, as original values ranged from -10…10 (on a similar order of magnitude to u and v parameters) yet they are rendered at two levels of magnitude below the temporal parameters in the results. The temporal

parameters themselves appear to segregate readings largely on the basis of the four quadrants of the Cartesian plane (*c1*: quadrant II; *c2*, *c3*: quad III; *c4*: quadIV; *c5*: quadI), as almost all of the temporal parameter values fall squarely in the center of one of these quadrants. The relative weightings suggested by these centers serve to confirm the relative weightings shown in Figure 4.20, where temporal parameters appear to be of primary importance, angleΔ is secondary, and spatial properties are of tertiary import, if any. Cluster2 distinguishes itself by its angleΔ value, but appears to be otherwise largely similar to cluster3 in terms of its other parameters. This may be an artifact of how the synthetic data was generated, for example the trigonometric functions used to do so. Given the constantly-changing nature of natural data signals, it often does not pay to descend too deeply into the details or possible meanings of cluster-centers that were found, after all, by simply determining the minimal distances to k-centers for a given subset of data. The main takeaway we might draw from the kmeans analysis of this synthetic dataset, therefore, is that kmeans found general vector *bearing* to be the most effective partitioning property for overall classification.

Figure 4.22 demonstrates how the output from four generative processes over time is not clustered as intuitively as we might have thought (i.e., as four clusters). Three main partition classes (1, 2 and 5) alternate regularly over time, with two interstitial classes occurring primarily during the transition periods between the three main classes.

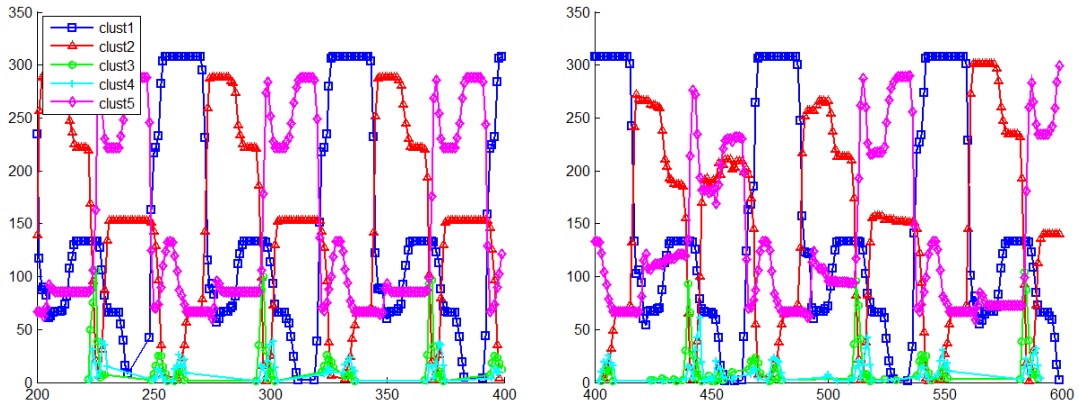


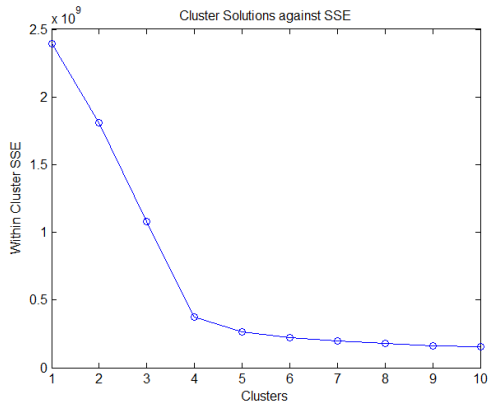
Figure 4.22: View of 5-part membership before and after critical point t=400. Note the regularity of all cluster population series on left as compared to the right, occurring as a distinct regime (green-circle from Figure 4.19) traverses the region M.

Although Figure 4.19 displays the region *M* as classified by the four actual processes generating the data, the Monte Carlo simulation technique found that the optimal number of clusters required to model this region over time to be *five*, rather than four (see Figure 4.23).

Again, K=4 might seem to be the more intuitive visual choice as the sought-after "elbow", but as the automated process finds the slope of the line between K=3 and K=5, the approximated slope for K=4 does not quite reach the desired threshold. It is worth remembering that even if K=5 is an overestimation, it only costs a little extra processing-time up front (to train the extra model) and storage-space on the sensing platforms, and according to the discussion surrounding Lemma 4.1 and Corollary 4.2, will still tend to marginally *improve* model-accuracy if anything.

The possibility remains that such an overestimation error may be due in part to the fact that this data is synthetic, not natural. Indeed, the sharp angle that can be seen in the graphs displayed in both Figure 4.12 and Figure 4.23 appears suspicious to experienced

eyes, especially in comparison to that generated by truly natural data in the following chapter.



Normalize x (i.e., K) and y (WCSS).

K	WCSS	nx	ny	m
1	2.396E+09	0	1	0
2	1.810E+09	0.1111	0.7386	-2.6358
3	1.082E+09	0.2222	0.4143	-2.8855
4	3.717E+08	0.3333	0.0974	-1.6370
5	2.665E+08	0.4444	0.0505	-0.3058
6	2.193E+08	0.5555	0.0295	
7	1.955E+08	0.6667	0.0189	
8	1.774E+08	0.7778	0.0108	
9	1.640E+08	0.8889	0.0048	
10	1.532E+08	1	0	
min	1	1.532E+08		
max	10	2.396E+09		

Formulae:

$norm_x$ (nx) = $(x-min_x) / (max_x - min_x)$

$norm_y$ (ny) = $(y-min_y) / (max_y-min_y)$

m_x = $(ny_{(x-1)} - ny_{(x+1)}) / (nx_{(x-1)} - nx_{(x+1)})$

Figure 4.23: Result of MCBestK heuristic (left), with normalized mean slope-approximation techniques (right)

A smoother graph would not introduce quite the potential for unintuitive choices of K as the current graphs do. A simple column-wise statistical Z-scoring of each instance in the dataset, for instance, enables the kmeans process to work on *normalized* values, allowing each field or parameter in the instance to have equal weight in determining the resulting classification regardless of its range relative to those of the instance's other fields. This process results in a smoother graph (although a somewhat longer k-means rendering-time as well). For this dataset, this generates the curve displayed in Figure 4.24. Whereas the choice for K is no longer as obvious, the BestK selected in this case would be four (as we compare results for both Z-scored and non-Zscored datasets, we shall use K=5 for both – again, overshooting the presumed "optimal" value for K, should it happen, is of no great import).

Whatever the clustering-scheme used, it is unlikely to exactly replicate the actual shapes of the data-generating regimes shown in Figure 4.19. We can observe in Figure 4.25 that the color-coded clusters can generally outline the regimes, but the subset of data provided on which to base cluster determinations does not allow kmeans to map regimes to unique cluster IDs over time. However, this turns out not to be strictly necessary if the goal is simply to improve the accuracy of approximations of a spatial-temporal field via partitioning. Should the partitioning parameters not confer complete information or total separability of underlying processes, they can still provide enough additional information to significantly improve approximation-accuracy of a partitioned field over a non-partitioned one – and the extent of that improvement will be in relative proportion to the relevance of the data provided to kmeans. The same partitions shown in Figure 4.19 may be generally distinguished in Figure 4.25, even if borders separating distinct regimes may occasionally disappear (and though the colors used may seem questionable, it should be noted that this color scheme was kept in order to match partitions with their appropriate series color in Figure 4.22).

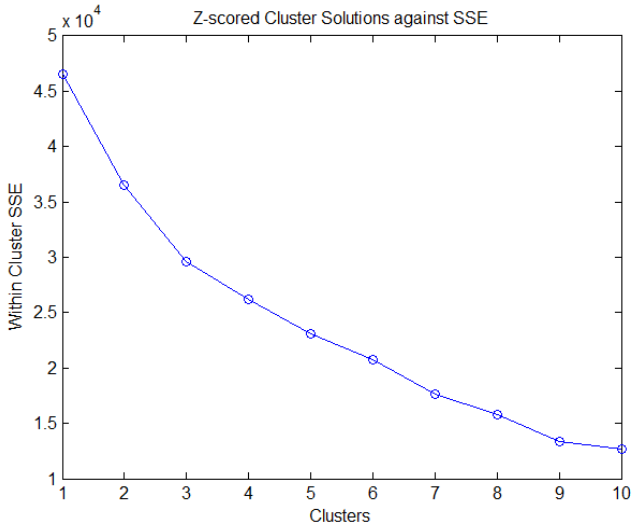


Figure 4.24: WCSS solutions for Z-scored data for varying number of clusters K

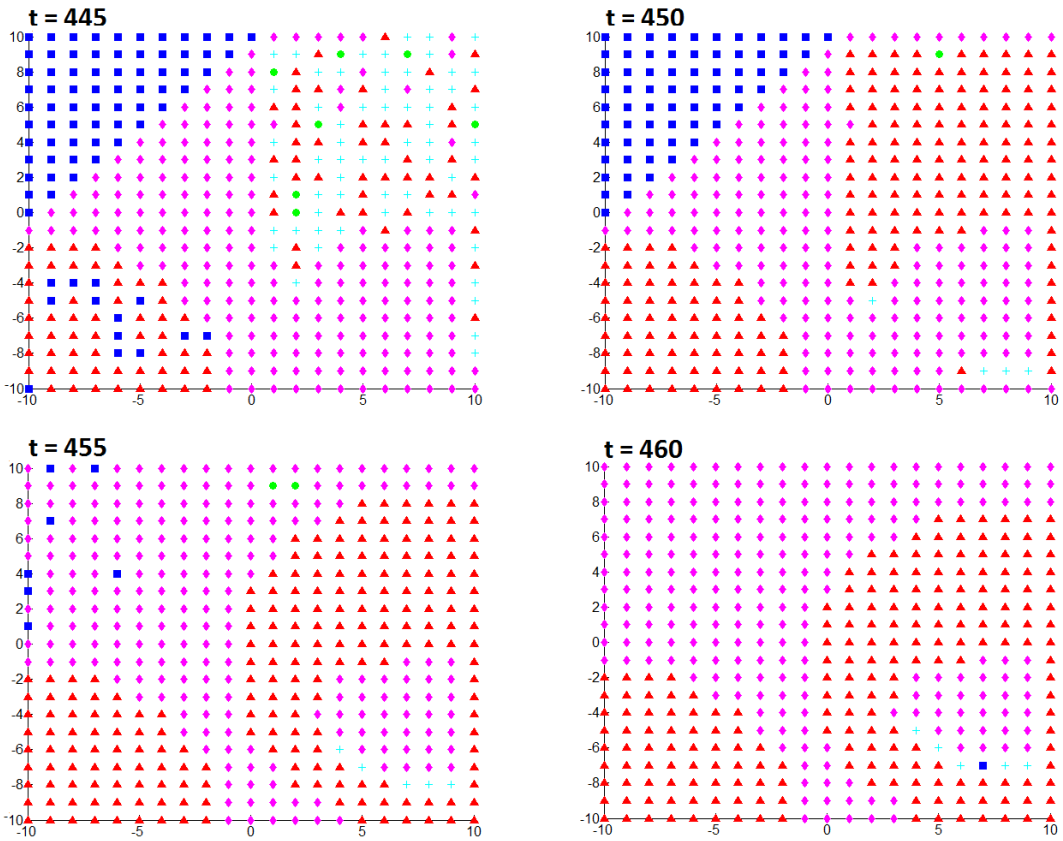


Figure 4.25: Actual cluster self-identifications over a selected time-period of the NP simulation (see *Figure 4.22*)

The results of these clustering schemes may be found in Table 4.5. The mean difference between individual results of the single model and the partitioned (unprocessed) model is -0.22, which is found to be a significant difference at the 99% level for a paired t-test of over 600,000 instances, with a resulting p-score of less than 10^{-4} (MATLAB's default level of accuracy).

In some cases, normalizing attribute-values to z-scores before the k-means process can result in clusters that ultimately yield better model performance. The thinking behind this is that some processes exert their effects over a very small range of readings (e.g., ranging only over a few units, such as barometric pressure readings); these readings will tend to be discounted when included together with inputs that vary over wider ranges by one or more orders of magnitude (e.g., wind-velocity readings). Normalizing such datasets to corresponding z-scores allow data values to be compared on relatively equal footing. A similar approach is taken with inputs provided to MLP models, and for similar reasons (Reed and Marks, 1998).

Table 4.5 shows results of partitioned models trained for use with clusters generated from both non-processed and z-scored datasets. The model using z-scored clusters can be seen to exhibit a small, yet significant, increase in performance over the non-processed cluster model.

The Student paired t-test was used to test the significance of all model-result comparisons. It was considered a valid measure in these instances since the histogram of differences between paired errors (Figure 4.27) in result-sets of such magnitude (i.e., over 700,000 records) is still generally normal, as predicted by the Central Limit Theorem. As a precautionary measure however, the 2-sample Kolmogorov-Smirnov hypothesis test

(one which does not assume normal distribution of results) was used in all cases to confirm the Student test results, which it did in every case. As such, since the z-scored version of the same test data set shows a greater improvement overall, it may be considered a significant improvement over the Global model, in addition to being a statistically-significant improvement over the non-processed partitioned model as well.

Table 4.5: Results of Single model vs. Partitioned model

	Clust I	Clust II	Clust III	Clust IV	Clust V	Overall
Global Model						**3.1074**
Partitioned Model (non-processed)	2.7966	2.6371	2.6831	2.5349	2.4993	**2.6874**
Partition Pop.	209028	205671	202951	8654	5649	
Partitioned Model (Z-scored)	2.5790	2.5246	2.7254	2.6239	2.7052	**2.6324**
Partition Population	154701	8653	152111	157734	158754	631953

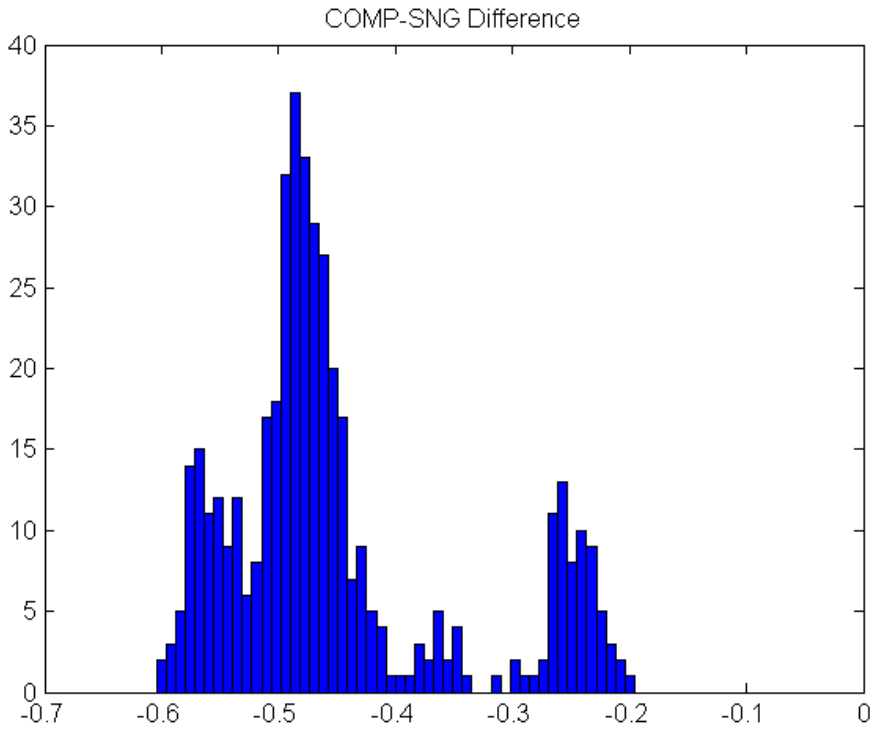


Figure 4.26: Histogram of differences between results of Single model, and Partitioned (K=5) model

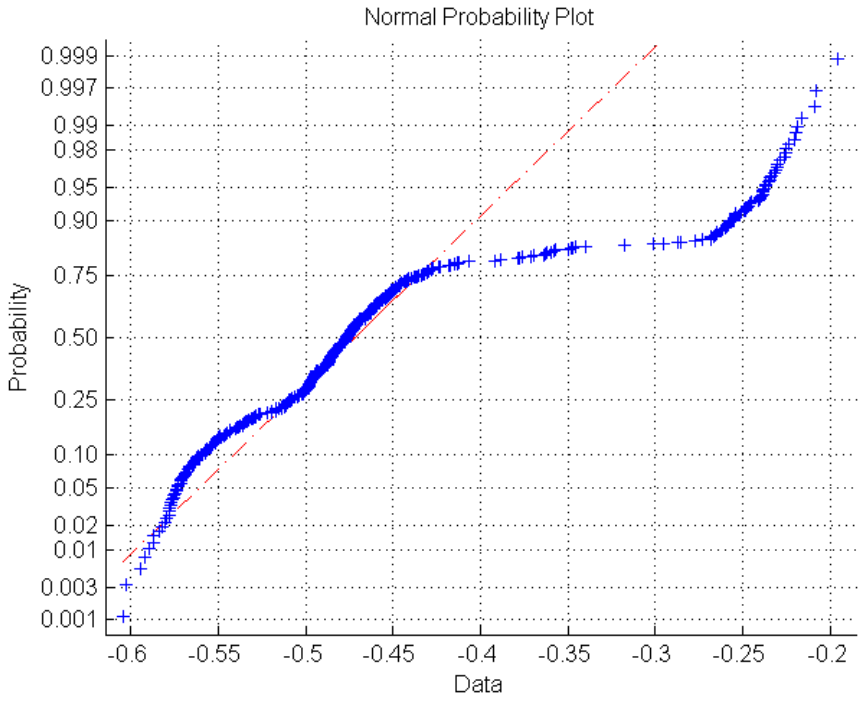


Figure 4.27: Normality plot for paired-differences histogram in Figure 4.26

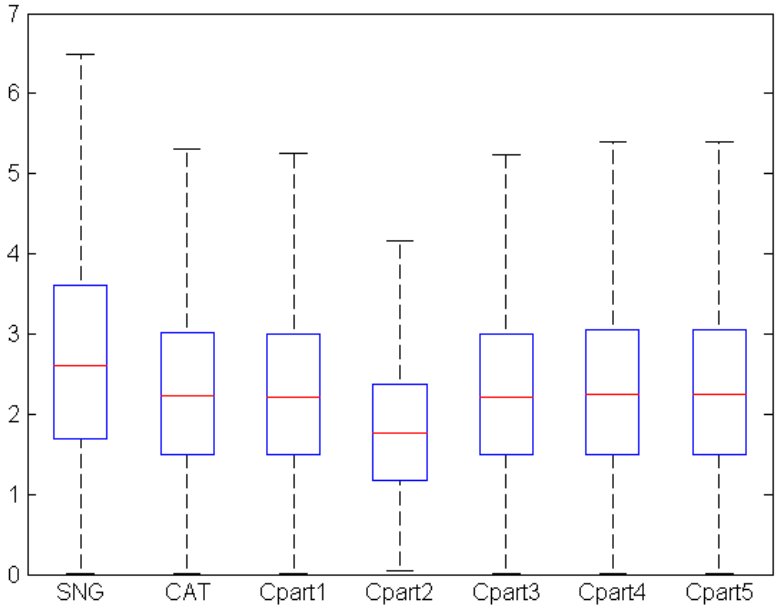


Figure 4.28: A typical boxplot of Z-scored Partitioned model comparison for naturalistic synthetic dataset

4.7 MLP Design and Feature Relevance

As mentioned earlier in section 4.3, all MLP models used in this study accept the same eight input features: *u1*, *v1*, *u2*, *v2*, *u3*, *v3*, *u6*, *v6*. In the course of the training process, MLP models determine for themselves the importance of the provided input features. In the case of partitioned models, all *k* MLPs base their individual determinations of the value of their input features as a result of the data instances populating their assigned partition.

Just as we used TreeBagging to determine how the k-means process valued the features provided to it, we can use the same technique to qualitatively determine the value of the input features provided to our MLP models. In essence, we match data features provided to the model with its provided result and determine a correlation-type

weighting of each feature to the generated resultset. The TreeBagging results for the two synthetic data sets follow.

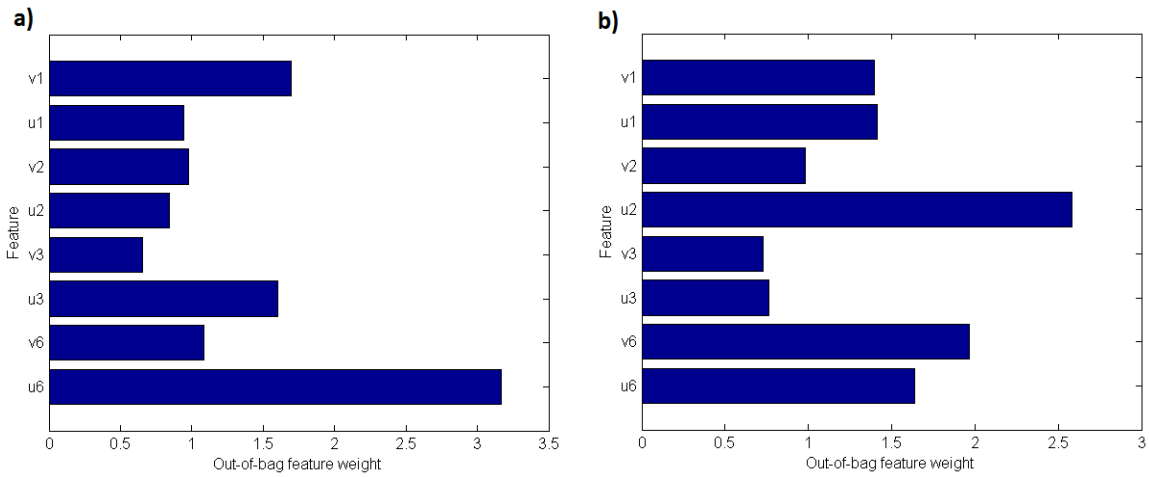


Figure 4.29: Relative feature-weight valuations of Orthogonal synthetic dataset for a) Global model, and b) Partitioned model

It is interesting to note that the ranks of features valued by the single Global model in Figure 4.29a (i.e., readings at 6-lag, 3-lag and 1-lag) are largely the same as those valued by the four MLP comprising the partitioned model in Figure 4.29b (i.e., readings at 6-lag, 2-lag and 1-lag). Note that as each component at a particular lag contains roughly the same information content as the other, it is useful to consider both components as a block – perhaps by comparing the mean of both bars at a given lag to the corresponding means at other lags.

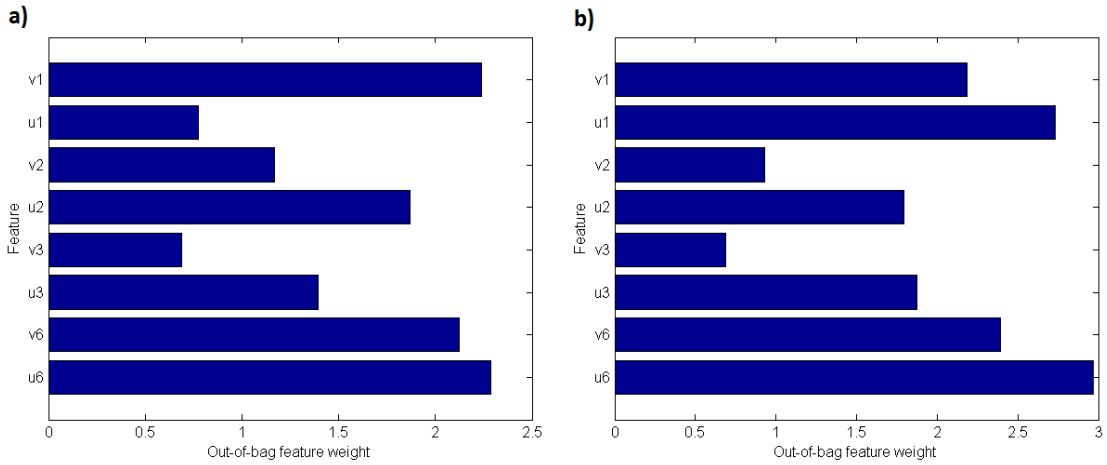


Figure 4.30: Relative feature-weight valuations of "Natural" synthetic dataset for a) Global model, and b) Partitioned model

Similarly, the five MLP of the partitioned model of the "natural" synthetic dataset (Figure 4.30b) made roughly the same evaluation of the value of the provided input features as the global model (Figure 4.30a) did. Such *a posteriori* analyses can be helpful in fine-tuning our MLP models in order to determine which data features are of low value and could potentially be left out, as fewer parameters might potentially make for a simpler, more efficient model.

4.8 Conclusions

This chapter explored the application of the basic k-means clustering technique to a spatial-temporal domain of synthetic readings, presumably gathered over a period of time from a monitored region M. Synthetic data was generated such that there would be four spatial regimes present, and that some of these regions would have dynamically changing boundaries over time. A Monte-Carlo heuristic was used to determine an appropriate value of K for the k-means partitioning process. A small collection of raw and derived signals comprised of both spatial and temporal (i.e., Δt-differenced) readings were found to return K spatio-temporal groupings that were capable of delivering statistically-

significant reductions in overall approximation errors between the global model's results and those of the partitioned model applied to a dynamically changing field of values over time. Though generating k-means clusters using z-scores of input data provided some increased performance with these synthetic datasets, the difference in accuracy may well be deemed not to be worth the extra processing effort, especially as the benefits of the heuristic of z-scoring k-means inputs is often found to be problem-specific. An example of this can be seen in the following chapter, where these same methods are applied to natural surface-current velocity readings harvested from the Gulf of Maine, as opposed to synthetically generated ones, and with similar success.

CHAPTER 5

MLP PERFORMANCE ON NATURAL DATA SET FROM THE GULF OF MAINE

5.1 Introduction

From the introduction in chapter 3 of the concept of using standard feed-forward Multilayer Perceptrons (MLP) to approximate time series, we proceeded in chapter 4 to apply those MLP to spatial regions as they evolve over time. These spatial-temporal (ST) clusters were determined by a standard k-means approach, using spatial as well as 1-hr difference readings as temporal components. The relative importance of these components was determined by a statistical tree-bagging technique, and the number of clusters K for k-means was determined by a Monte Carlo simulation which approximated a break-even point between the concepts of "more clusters than necessary", and "not quite enough." The conjunction of these techniques have allowed us to build a systematic approach to determining whether and how to subdivide a sensor-monitored spatial region M into multiple spatiotemporal partitions, each serviced by an MLP model trained to the characteristics of that particular ST partition, in order to extract additional accuracy from the modeling system for the entire monitored region.

This chapter documents the application of these techniques to testbed data located in the Gulf of Maine (GoM). Establishing a fine-grained model of local ocean currents is important since currents carry nutrients and organisms which affect ecosystems in coastal regions. For example, researchers are interested in establishing current models for the

Gulf of Maine since they can distribute a specific type of algae to shellfish off the coast of Maine during the warm summer months; the shellfish consuming the algae turn toxic for humans (i.e., "red tide" phenomenon) (Pettigrew *et al*. 2005). In addition, shipping traffic in sea lanes continues to increase due to a variety of economic factors, including the regular delivery of everyday commodities such as heating oil, gasoline and natural gas to near-coastal communities otherwise without ready, inexpensive access to them. There is thus a growing need to have a system in place to track drift-based phenomena such as oil spills, search-and-rescue operations, and other lost cargo events, given the increasing likelihood that these events might occur.

Today, major ocean currents are established using coastal radar; however, the information can be spatially and temporally too coarse. We conceive of a system comprising a wireless sensor network of power-limited computational platforms (or nodes) deployed within the marine environment, capable of delivering accurate, fine-grained local current approximations based largely if not solely on local computation and at-need communication with similar, nearby nodes. Although hypothetical systems incorporating mobile sensing stations are being explored (Nittel et al. 2007), we assume fixed nodes moored to a static location. As these WSN nodes are power-constrained, any RF intra-node communications are infrequent if necessary at all, and any needed internal computations should likewise be finite and kept to a minimum.

5.2 Experimental Setup

5.2.1 Gulf of Maine Dataset

The ocean surface current data was provided by the University of Maine's Physical Oceanography Group, covering five consecutive months (December through April) of 2013-2014. It was measured by a 4.3-5.4 MHz SeaSonde HF radar system, which is deployed to observe sea surface currents in the Gulf of Maine. In our simulations, we use current direction and current speed data measured hourly at the center of cells in a 36x24 grid. The size of each grid cell is approximately 16km x 16km.

5.2.2 Proposed WSN Operation

Although no embedded WSN currently exists within this region as such, we consider that the surface current readings returned by the SeaSonde radar system were actually produced by a WSN physically embedded within the region. Then, as proposed in chapter four, we simulate the generation of ST clusters (along with their cluster-centers) and develop their respective MLP models at a non-power-constrained coastal site. ST cluster and MLP model parameters are then transmitted to nodes in the Gulf. In the normal course of operation, a node should wake up at its assigned time, collect and store a reading from its onboard sensor, then return to a sleep state until its next wakeful period. Should the sensor not be functional to take the required reading, the node would compare its most recently stored readings to its saved set of cluster centers, determine from these the least-distance (and thus most appropriate) dynamic regime, and use the stored MLP model associated to that regime to generate an approximation of the desired

sensor reading. The node would continue in this approximation mode until the sensor returned to normal operation.

5.2.3 Determining K

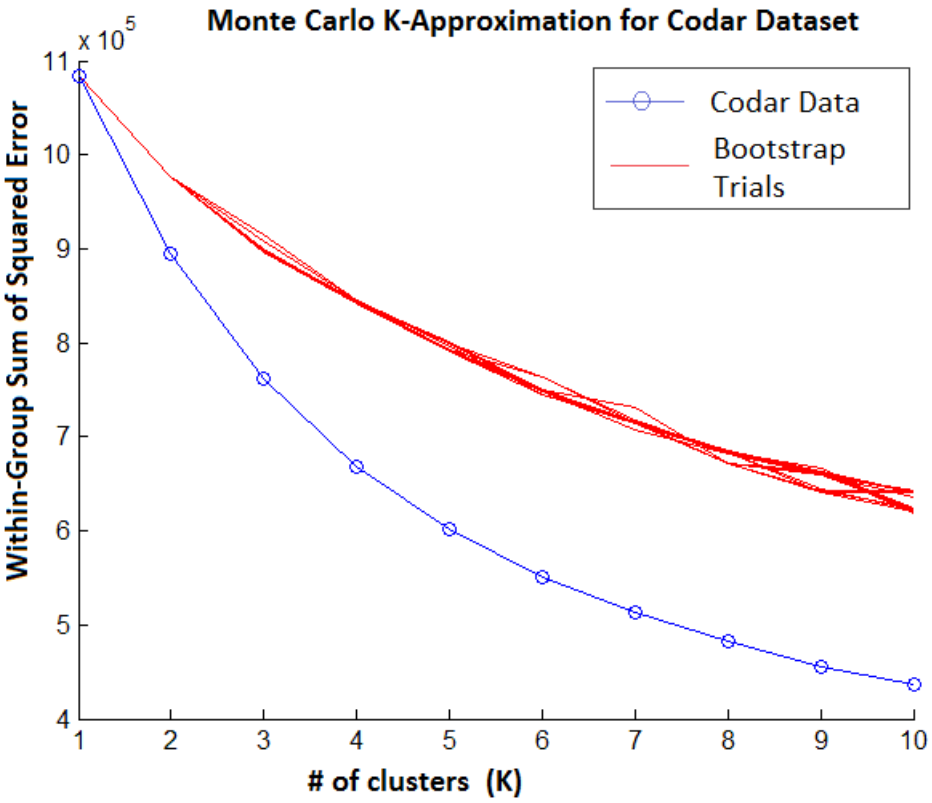


Figure 5.1: Monte Carlo results of raw Codar data compared to 100 randomly permuted versions of dataset

The Monte Carlo approach adapted from (Peeples, 2011) delivers the graph in Figure 5.1 when a series of averaged k-means results of the GoM data for values of K ranging from 1 to 10 is compared to 100 individual k-means runs on column-permuted variations of the original dataset. Given that none of the permuted series results (red lines) approaches or

crosses the original dataset's result series (blue line), we may assume that the latter is not a result of chance at a 99% confidence-level.

It now remains to find where a tangent line to this series changes state, passing the threshold from slope values *less than* -1, to slope values between -1 and 0.

x	y	norm_x	norm_y	m
1	1084342	0	1	
2	895035.2783	0.111111111	0.707615544	-2.23961755
3	762106.7091	0.222222222	0.502307211	-1.58042576
4	667644.2426	0.333333333	0.35640982	-1.119112706
5	601089.2137	0.444444444	0.253615499	-0.810120452
6	551084.431	0.555555556	0.176383052	
7	514000.186	0.666666667	0.119106392	
8	482554.3038	0.777777778	0.07053819	
9	455439.2979	0.888888889	0.028659031	
10	436883.7725	1	0	

As was set out in the preceding chapter, since the slope of -1 occurs between x-values of 4 and 5, we choose K = 5 for our k-means clustering scheme, as no significant additional performance benefit would be expected from a larger number of subpartitions of M, whereas four clusters may still be too low to capture the number of natural clusters present within M in the period of time considered.

Over a one month span of time in the CODAR data record beginning in December 2013, approximately sixty-three thousand data vectors were extracted consisting of the following attributes: latitude, longitude, u0, v0, u1, v1, and Δangle. The first two attributes provide the spatial components of this spatiotemporal clustering; two 2-

dimensional current vectors (u measuring east-west flow, and v measuring north-south) follow, expressed in cm/sec and separated by one hour providing a time-differentiated measure of both magnitude and direction; and Δangle is a derived measure representing the change in heading in decimal degrees, computed by equation 5.1. As Z-scoring the inputs before clustering may potentially improve results (as seen in the previous chapter), two clustering schemes were generated: one Z-scored, one not.

$$\Delta\text{angle} = (\arctan(u1, v1) - \arctan(u2, v2)) * 180/\pi \qquad (5.1)$$

The tree-bagging analysis of the resulting schemes is shown in Figure 5.2. The effect of z-scoring the raw data values is immediately obvious in the comparison between the two graphs, in that the non-Zscored scheme shows one attribute, Δangle, dominating most of its accompanying attributes, whereas the Z-scored version shows a less differentiated distribution of parameter relevance. Interestingly, the non-Zscored scheme appears to largely reverse the effects seen in the Z-scored version: spatial coordinates (lat, long) and east-west flow (u) seem more influential in the latter, whereas Δangle and north-south flow (v1) appear most relevant to the partitioning scheme in the former. Which of the two schemes is most appropriate within the composite-MLP model under current conditions will be determined empirically by the results of the MLP model trained on the cluster groups generated here.

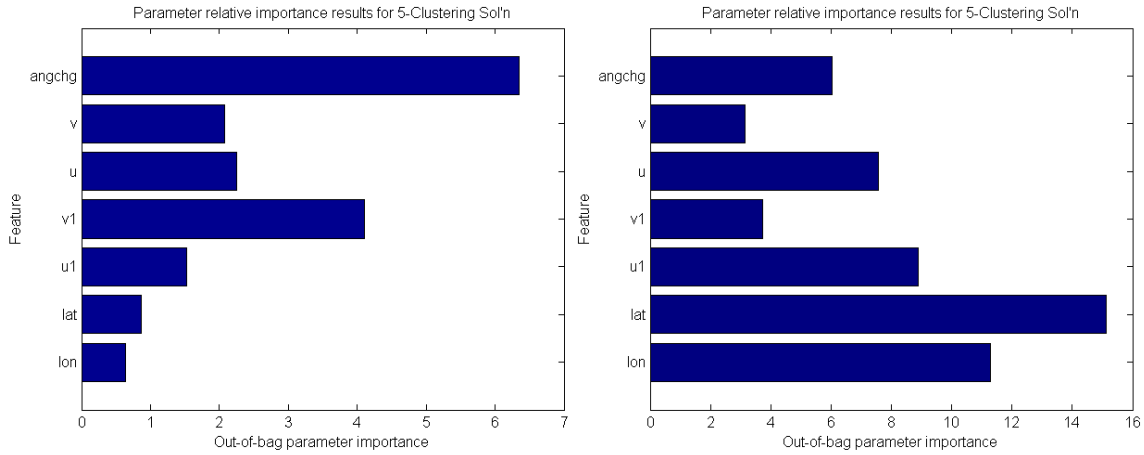


Figure 5.2: Parameter relevance results of two clustering schemes of one month of CODAR data; one using raw values (left) and the other using z-scored values (right)

As an additional check to verify the benefits of the k-means partitions, a third clustering scheme was generated upon this data set: a completely random partitioning of the data set, assigning each instance-vector to one of the five clusters with equal probability, irrespective of timestamp or location. Though a parameter-relevance chart would make little sense for this scheme, three consecutive snapshots of the resulting partitions are shown in Figure 5.3, revealing cluster assignments as random as one might expect.

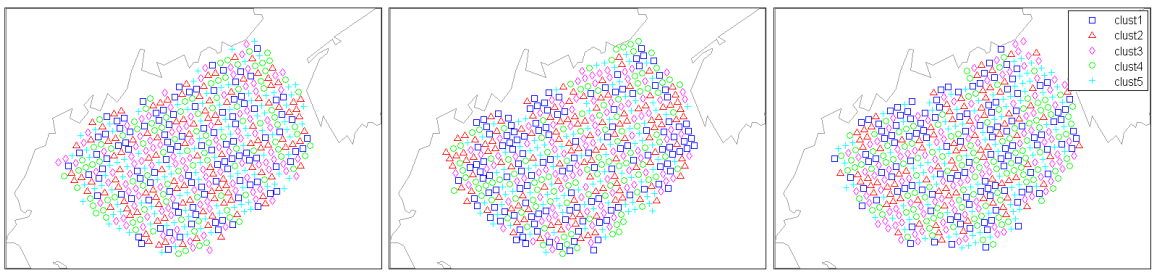


Figure 5.3: Random partitioning of data instance vectors in the Gulf of Maine taken over three consecutive hours (3/06/14 17h00 – 19h00 UTC)

In comparison, both the Z-scored and the non-Zscored partitioning schemes show distinct, persistent, generally self-connected regions over time (though the Z-scored version, with its heightened emphasis on the spatial location of the instance vector, would appear to have more spatial coherence than the latter).

Figures showing a three-hour series of snapshots of both Z-scored (Figure 5.5) and non-Z-scored (Figure 5.7) versions of the partitioning scheme are actually quite similar. Both portray a general partitioning of the Gulf of Maine into two major parts east and west, with a short-lived eruption of a third cluster within the middle of the gulf. The two remaining, so-called "interstitial," clusters appear around the edges of the main three. Peaks corresponding to these main three clusters can be located in the respective charts of cluster populations (Figure 5.5 and Figure 5.7) at approximately 19h00 on 3/06/14. As surface current readings in the center of the Gulf basin tend to be of low magnitude compared to the edges, this eruption could signal the changing direction of tidal flow, from inbound to outbound, for example.

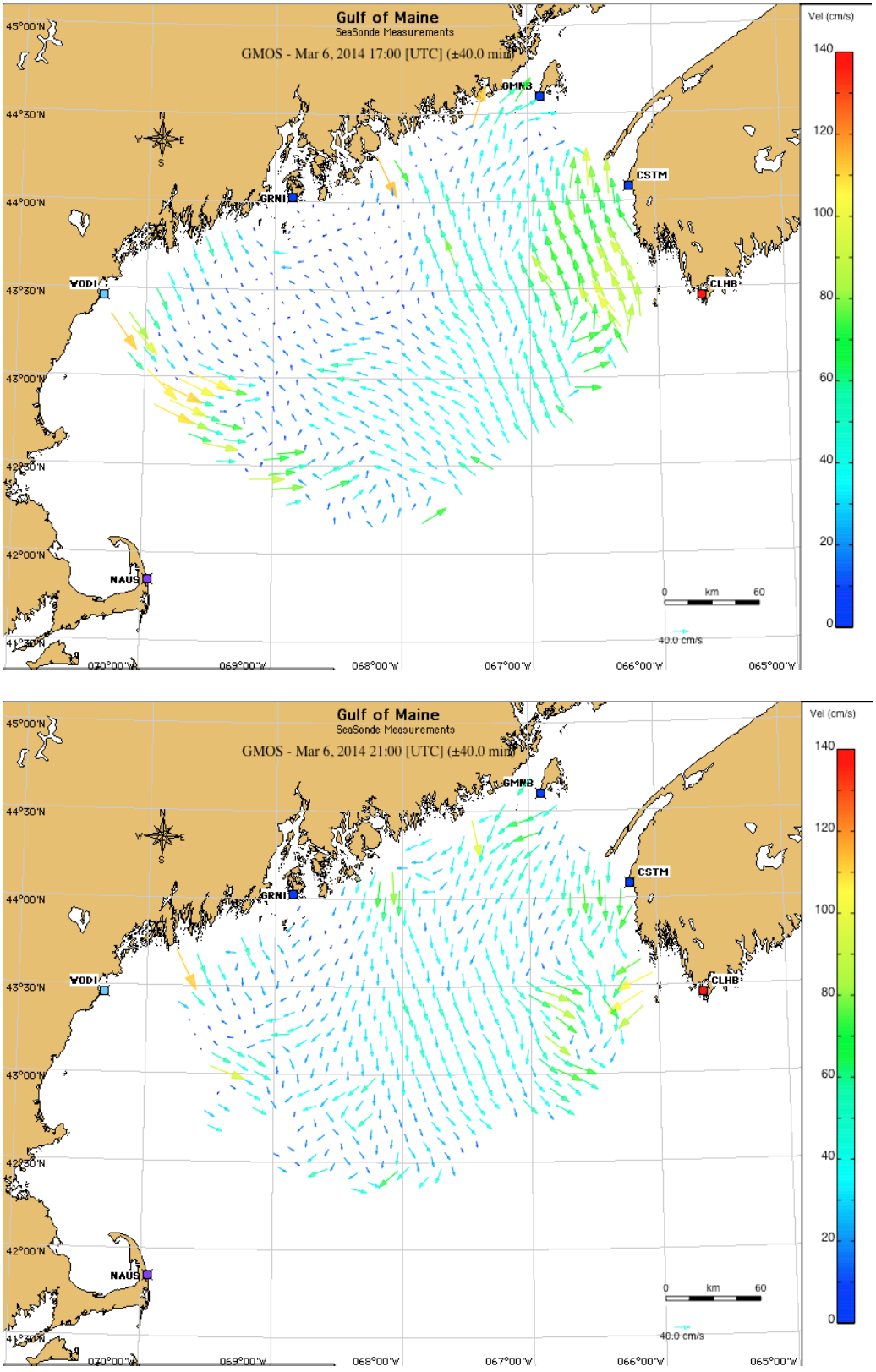


Figure 5.4: Total vectors remotely sensed in GoM on 03/06/2014 at 17h00 (top) and 21h00 UTC (PhOG, 2014) (bottom)

Change in tidal flow does indeed appear to be the cause here, as relatively strong inbound flow appears to be occurring at 17h00 in Figure 5.4 (top), especially through the eastern half of the gulf. Successive snapshots freely available at the University of Maine's GoMOOS website show these inbound currents gradually lessening and eventually reversing direction over the next several hours until 21h00 as seen in Figure 5.4 (bottom). The reader may find it easier to connect this 17h00 surface-current snapshot to the first cluster figure in Figure 5.5, rather than the corresponding first one in Figure 5.7, as the spatial structures of the data seem more readily represented in the former. These are not simple spatial clustering schemes, however, but spatiotemporal ones. Clusters should not be evaluated by a simple look at the assignments and relationships within a single snapshot, but with corresponding assignments in previous and succeeding snapshots as well. As will be seen, the latter (i.e., non-Z-scored) partitioning scheme will actually be determined as the more effective – its predilection for Δangle and v-component values apparently conferring some advantage over the scheme based more heavily on spatial location and u-component values.

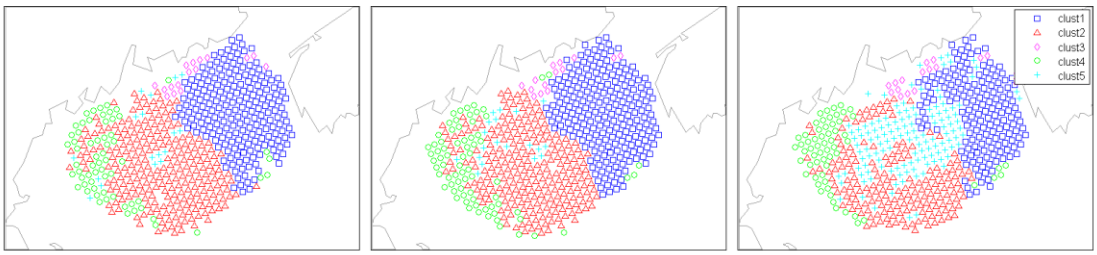


Figure 5.5: Z-scored partitioning scheme over three consecutive hours (3/06/14 17h00 – 19h00 UTC)

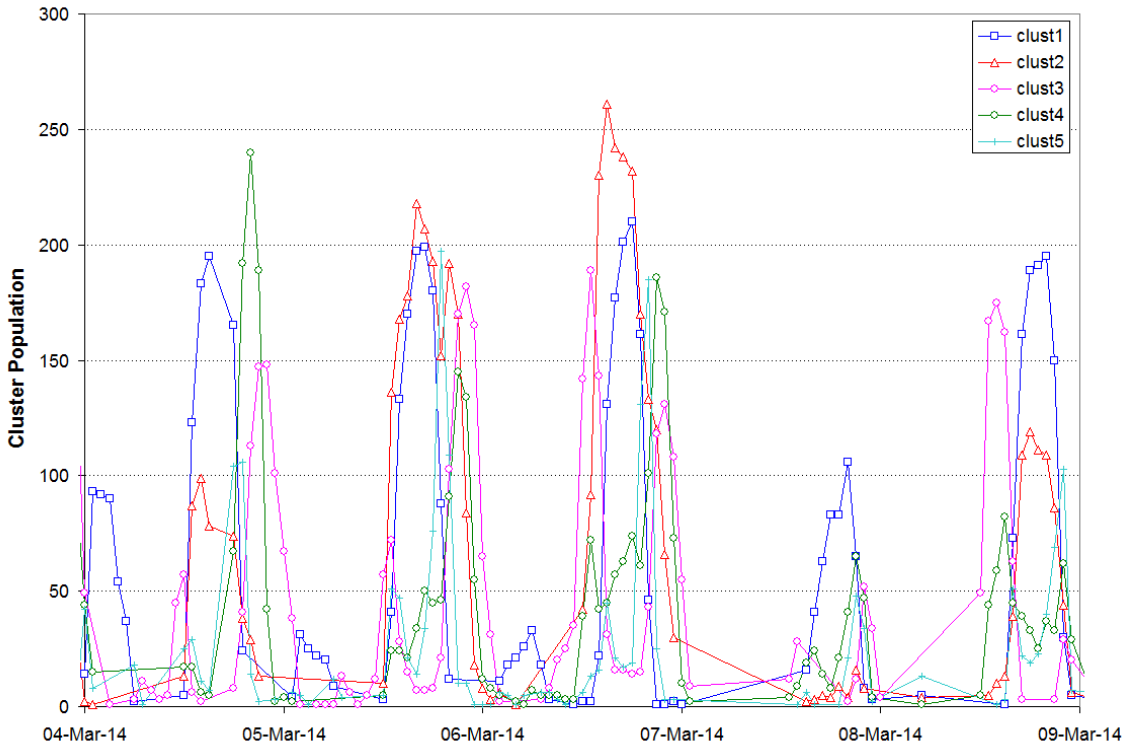


Figure 5.6: Population of Z-scored clusters over five days' time

These cluster population charts demonstrate the difficulty of data collection in a natural environment, as natural atmospheric interference would occasionally hinder the radar's remote sensing of data between approximately 12h00 and 24h00 GMT, and readings can be seen to decline in quantity outside that sensitive period.

They also highlight the relatively large parts four of the five clusters play over time. The general regimes identified by clusters one, two three and four each seem to succeed each other over roughly comparable portions of the gulf as time progresses.

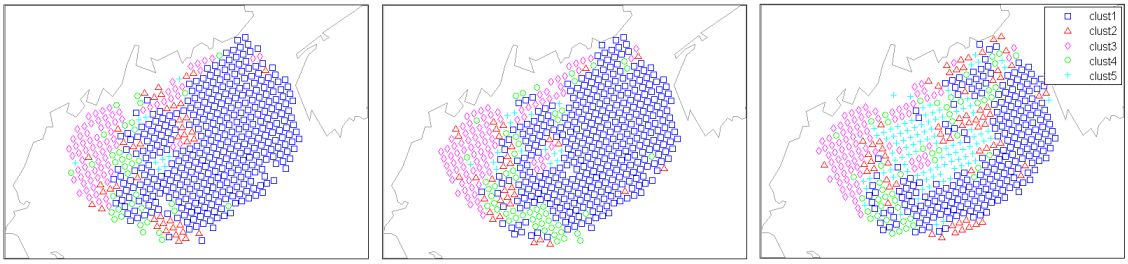


Figure 5.7: Non-Z-scored partitioning scheme over same three consecutive hours (3/06/14 17h00 – 19h00) [Note: no direct relationship should be assumed between these cluster symbols and those of Figure 5.5]

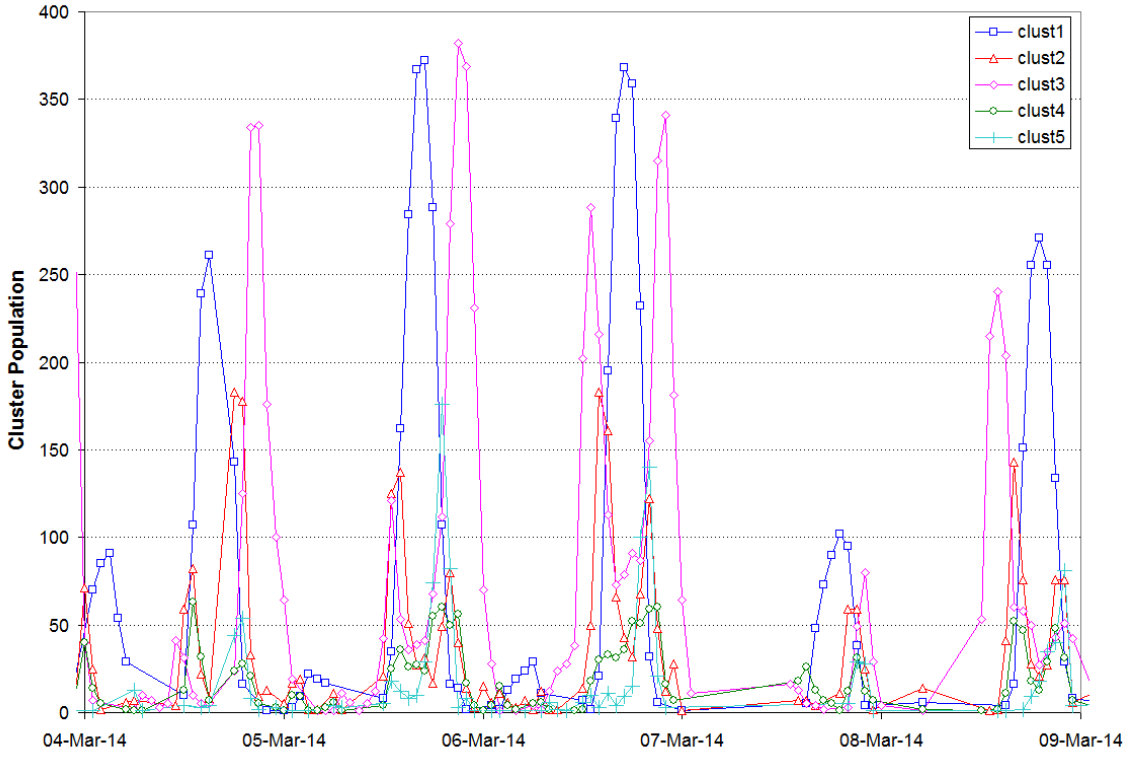


Figure 5.8: Cluster populations of non-Z-scored clusters over five days' time

What, then, are we to make of the appearance of cluster5 symbols in the middle of the Gulf in both (non-random) versions of the partitioning scheme? A small but recurring event, its appearances tend to be short-lived, but it seems to represent the same general physical process in both schema. We propose that this cluster is a representation of the

still surface waters – captured here poised between the incoming tidal flow (visible in Figure 5.4's 17h00 UTC snapshot) and the outflowing tide (at 21h00 UTC). The snapshot of gulf readings represented by the third image in both Figure 5.5 and Figure 5.7 is perhaps less telling without the surrounding context, but is shown in Figure 5.9. The outline of the onset of cluster5 symbols may be traced by the low-magnitude vectors in the center of the gulf, signaling nearly non-existent current flow as the tide begins to reverse. This is a powerful illustration of what we have chosen to call the "interstitial clusters": collections of instances of low population that might otherwise have been written off as insignificant, if their temporal (and spatial) manifestations did not so aptly capture significant events within their spatial-temporal context – significant events benefitting so well from MLP modeling that overall system error rates are reduced thereby.

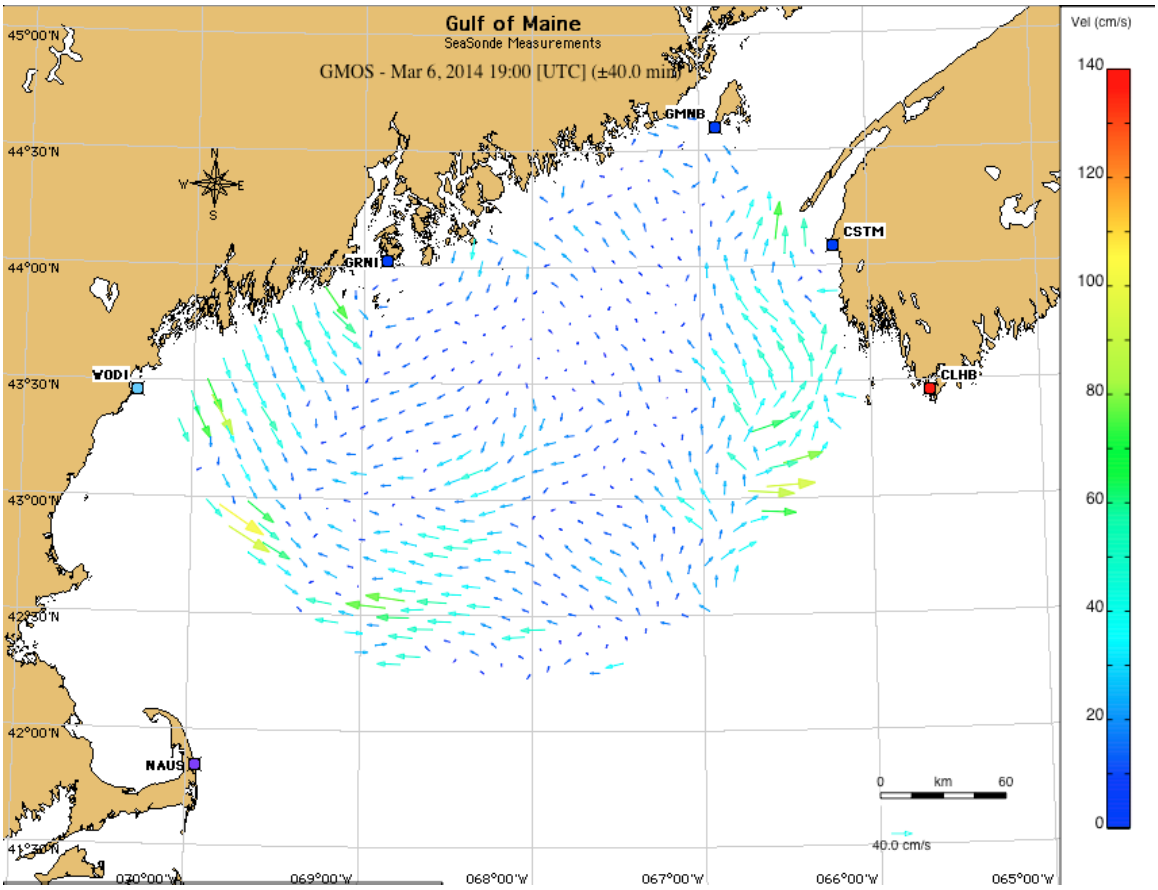


Figure 5.9: Snapshot of reversing tide in Gulf of Maine at 19h00 UTC (PhOG, 2014)

Table 5.1 displays the cluster centers generated by the k-means process. We may note that represented within these centers are vector-templates from all four quadrants of the Cartesian plane, as well as instances of both clockwise and counter-clockwise rotation. The most interesting of these may be the fifth row, representing cluster five – the emergent cluster of crosses in the cluster-figures above. Interestingly, the mean u and v magnitudes of this cluster's members are quite small, supporting the conjecture that this cluster emerges preferentially at times of tidal/surface current flow-reversal.

Table 5.1: Cluster centers resulting from non-Z-scored k-means processing

longitude	latitude	u0	v0	u1	v1	Δangle
-67.76	43.49	3.17	-4.81	4.70	1.91	-77.85
-67.76	43.65	-16.09	10.53	-12.82	-11.28	274.79
-67.63	43.62	-4.07	-29.67	-4.65	-29.48	2.02
-67.48	43.62	-0.46	28.53	0.77	27.86	3.60
-67.56	43.56	-0.13	-1.44	-1.05	-0.07	95.42

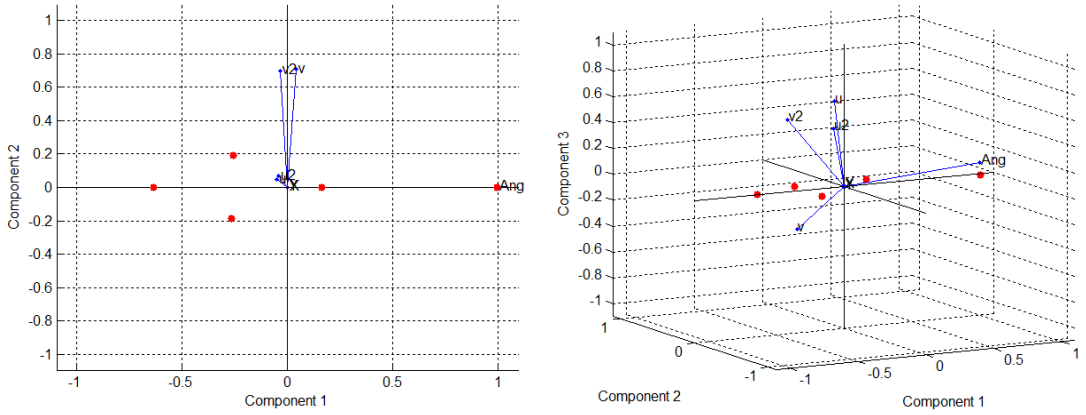


Figure 5.10 : Cluster-centers and attribute influence displayed in 2D (left) and 3D (right) digraphs

Figure 5.10 visualizes the cluster centers of the non-Zscored partitioning scheme, as well as the vector contributions of each of the attributes used to generate them, as 2D and 3D digraphs in the same PCA component space. Again, the 2D graph may be recognized in the 3D version as the horizontal axes. This figure illustrates the importance of Δangle to the scheme, as it contributes to approximately 95% of the first PCA component. Note also how the 3D version shows a truer representation of the contribution of the u-components to the partitioning scheme.

5.2.4 Partitioned MLP Simulation Results

Training separate MLP models to each of the populations identified by these cluster-centers, we generate a combined resultset of approximations to subsequent members of each partition. MLP models are trained on approximately one month of data instance-vectors (approximately 139,000). Each instance was comprised of: lon, lat, u6, v6, u3, v3, u2, v2, u1, v1; the models are trained to return approximations for u0, v0 (where the integers *x* in u*x*, v*x* represent number of hours previous to the approximated readings). Trained MLP models are then tested on three subsequent months of data readings (approx. 240,000) unseen in the training process.

Comparison of the partitioned-model approximations to the actual sensed (u, v) values (Table 5.2) shows reductions to estimation-error as compared to single-model MLP results. Level of improvement, however, depended on whether partition-generation was based on the raw or Z-scored data. In this case, instances categorized based on non-processed (i.e., non-Zscored) inputs resulted in less overall error.

Table 5.2: Non-Zscored Partitioning Scheme Simulation Results

	Clust I	Clust II	Clust III	Clust IV	Clust V	Total
Naïve (single-lag) error (cm/s)						18.970
Single model error (cm/s)						14.383
Random model error (cm/s)	14.410	14.562	14.393	14.484	14.554	14.480
Population	*45678*	*45699*	*45837*	*45586*	*46001*	
Partitioned model (non-Zscored) error (cm/s)	13.863	13.559	12.450	11.948	12.874	**12.512**
Population	*27975*	*20502*	*68383*	*103214*	*8727*	
Partitioned model (Z-scored) error (cm/s)	13.130	13.185	13.798	13.857	13.430	13.436
Population	*56526*	*42172*	*16344*	*53075*	*60684*	

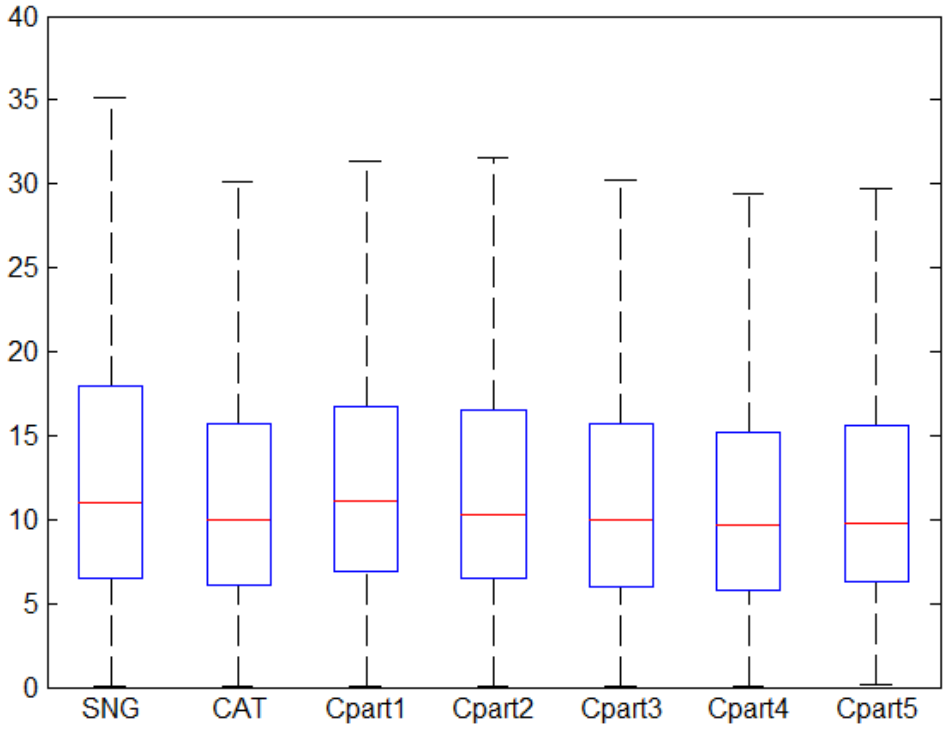


Figure 5.11: Boxplot of (non-Zscored) Partitioned model error comparison for CODAR testbed data

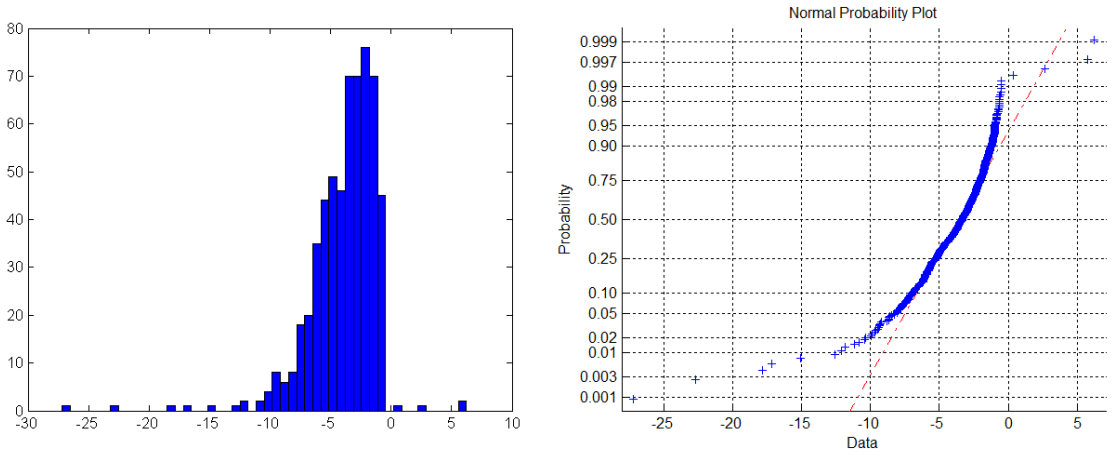


Figure 5.12: Histogram of paired differences of results from the Single and 5-Partitioned (non-Zscored) model (left), and the normality plot of said histogram (right)

Effecting a paired t-test on over 200,000 results each of the 5-Partitioned (non-Z-scored) model compared to the Single global MLP model provokes the rejection of the null hypothesis (of Part5_mean – Single_mean >= 0) to the .99 level of significance with a p-score below the 10^{-4} threshold (i.e., effectively zero). Though we may observe from Figure 5.12 that the distribution is skewed and thus not technically normal, the direction of its skew supports the paired t-test conclusion, as the vast majority of model differences are negative. This conclusion is reinforced as well by a non-parametric (i.e., normal distribution not assumed) paired Kolmogorov-Smirnov hypothesis test that also rejects the null hypothesis, again with a p-score below 10^{-4}.

5.2.5 Gulf of Maine Dataset Feature Relevance

Breiman's TreeBagging technique again provides us with estimates of the relative valuation of input features to the Global and Partitioned models, respectively, for the Gulf of Maine dataset in Figure 5.13. Comparing the two models we may observe that again both approaches had a roughly similar assessment of the importance of their input set, both preferring readings at lags 1 and 3 of the four available.

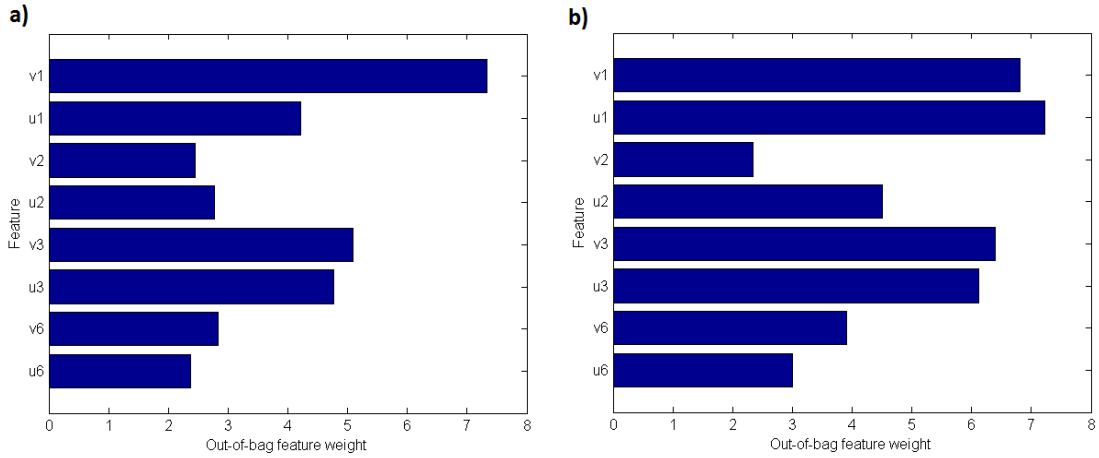


Figure 5.13: Relative feature-weight valuations of 2013-2014 Gulf of Maine dataset for a) Global model, and b) Partitioned model

It may strike some as passing strange that in an ostensible spatial model, no explicitly spatial input features, for example latitude or longitude, are provided to it. Up to this point this decision has been justified by the decision to use the MLP to implement the *spatial time series* spatiotemporal modeling approach. However as we know, the MLP model is vastly overpowered for such a straightforward approach. A signal benefit to using it is in its capability to automatically integrate additional data features – be they temporal, spatial or other – in order to enhance overall model accuracy.

In Section 4.7 we mentioned the utility of the TreeBagging technique to determine which features were less valued by the model and might potentially be precluded with little degradation to its predictive ability. By the same token, the technique can be just as useful for evaluating prospective datastreams for potential inclusion into the input feature set. For example, the latitude and longitude features may be evaluated against the training set of data; a high coefficient of correlation, especially as compared to other input features, would indicate their fitness for inclusion.

In this case, however, it is even more enlightening to compare the latitude and longitude attributes to the trained model results. These TreeBagging valuations are shown in Figure 5.14.

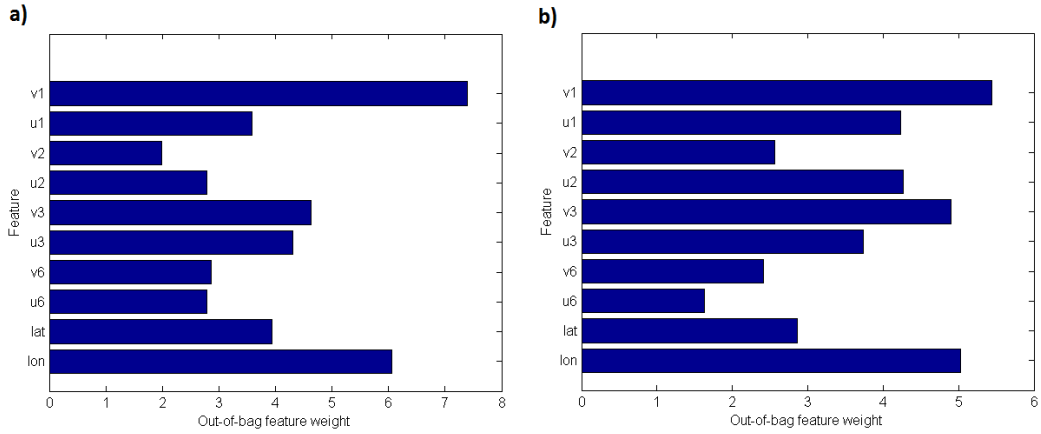


Figure 5.14: Relative feature-weight valuations of 2013-2014 Gulf of Maine dataset for a) Global model, and b) Partitioned model, including spatial attributes latitude and longitude

We observe from the figure that even with the spatial features in play, temporal features at lags 1 and 3 still rank in their same positions relative to the remaining temporal features in their importance to trained model results. Interestingly we also observe that the spatial features rank roughly on par with the main temporal ones – without ever explicitly having been a part of generating said results. This is an example of how relationships between spatial objects are often *implicit* (Shekhar et al. 2003). The combination of temporal features provided to both models in this case seem to have served as a proxy for the spatial features, and may be providing spatial information to the system in a more relevant fashion in the case of surface current maps. An analogy might be that of an expert wine-taster who may be able to determine the year and *terroir* from the qualities (i.e., taste, texture, acidity, etc.) of a sip of wine if not the precise location of

its provenance. Inclusion of these features in our models is thus found to be largely unnecessary. Later experimentation (not shown) suggests that including them does enhance result accuracy somewhat, but not to any impressive degree as their influence was already being felt by way of the other inputs provided.

5.3 Conclusion and Subsequent Work

5.3.1 Conclusions

This chapter has applied the multiple-MLP model developed in previous chapters to a natural dataset of surface ocean currents collected by CODAR radar stations installed around the Gulf of Maine. This dataset is used as a stand-in for a notional network of wireless sensor nodes that we imagine to have one node embedded at each location in the Gulf to which a surface current reading is attached.

The same techniques introduced in the previous chapter (Chapter 4 – Evaluating MLP Models on Partitioned Synthetic Data) are used for this dataset. That is, an appropriate number of partitions (K) is determined by a Monte Carlo analysis of a training dataset. K-means is enlisted to generate a spatiotemporal partitioning scheme using a combination of spatial attributes (i.e., lat, long), and temporal attributes (i.e., u1, v1, u2, v2, Δangle). *K* MLP models are generated to best accommodate the population of data instances that comprise each of the K partitions of the data. This partitioned system is tested upon new, unseen data partitioned by the same predetermined partitioning scheme and compared to the results of a single MLP model (i.e., the *"global"* model). A random partitioning of the same data into K groups will yield results that are no better than the global model's results at nearly *any* level of significance. In comparison, the results returned from the partitioned model consistently displayed significantly lower

error at the 99% confidence level. Whereas in the previous chapter it was found that z-scoring each attribute of the data instances before generating the partitioning scheme would result in further improvement with the synthetic data's approximation results, this was not found to be true for the natural data set. The usefulness of the z-scored k-means partitioning heuristic is thus found to be application-dependent, as it will not consistently deliver improved results.

5.3.2 Subsequent Work

Any model fitted to a particular set of data will contain a certain amount of bias towards data instances similar to those it was fitted upon. That is, data instances dissimilar to the training set will often produce larger error levels when evaluated with this model. A partitioned model composed of individually biased models will thus display a bias resulting from the additive aggregation of the individual biases of its components. What kind of second-order effects may we expect to see as a result of our present composite, spatiotemporally-partitioned model?

Also, any model fitted to a finite set of recent historical data can be expected to degrade in performance over time, given any relatively dynamic system; and its performance with nonstationary dynamic systems can be expected to degrade even more rapidly. How long, therefore, can we expect our partitioned model's results to remain valid, and how can we tell when that tipping point has been reached?

We investigate these last two questions in the following chapter.

CHAPTER 6

EVALUATION OF AUTOCORRELATED RESIDUALS, AND REFRESHING THE SYSTEM

6.1 Introduction

The previous chapters of this work evaluated the appropriateness of the Multilayer Perceptron (MLP) artificial neural network model for use in approximating two-dimensional readings in a field of the same, using a combination of both spatial and temporal readings as inputs. Chapter 4 expanded on the single MLP model application by fitting a system of MLP models to a partitioned spatial extent over an arbitrary period of time. The partitioned model allowed us to control to some degree for the nonstationarity that is typical to natural datasets with significant extents in both space and time (Fuentes *et al*., 2003; Fuentes *et al*. 2005). A Monte Carlo based kmeans approach was used to partition the dataset into subsets upon which to train and test the individual MLP models in the partitioned system. Partitioned MLP systems were found to have improved performance over a single global MLP fitted to the entire dataset when significant clusters or partitions were detected.

Both global and partitioned models leave spatial fields of residuals at every timestep at which they have computed approximations of missing (or imminent) readings. Like any model fitted to a finite set of training data, both global and partitioned MLP models display some bias in their results, based on the similarity (or dissimilarity) of the new data instances to which they are exposed *vis à vis* the data on which they were fitted.

The purpose of this chapter is to examine the change in those residual fields between the global and partitioned models, in order to determine whether the bias in residuals grows as the representative power of the model increases (e.g., does a partitioned model yield more spatially autocorrelated error than a global one). This is important because a model can be judged not only on its operational accuracy but also its level of operational bias. For instance, a model whose performance increases at the cost of increasing its operational bias could induce higher levels of artificial structure in the residual fields, possibly obscuring other important naturally-occurring second-order effects, thereby somewhat offsetting the value of the "improved" model. We employ a well-known spatial statistics measure called the Moran's I to test for any such increases to autocorrelated error in residual fields between the two MLP model results.

As these models are fitted at one point in time to be used for an indeterminate length of time using new incoming data that will likely prove to be increasingly dissimilar to the original training dataset, we propose a method to determine when the incoming data is no longer similar enough to the training set, and thus the models must be refitted. This functions as another control for the nonstationarity of natural systems over time, and provides a reasonable basis for a decision to be made to refresh the system of models in order to maintain their operational effectiveness under the constant incremental change of monitored conditions.

6.2 Spatial Autocorrelation of the Residual Fields

We apply a local Moran's I autocorrelation statistic to residual field values. The weights matrix, w, used in this chapter is a 3x3 matrix, where all elements but the center are set to ⅛, and the center element is 0.

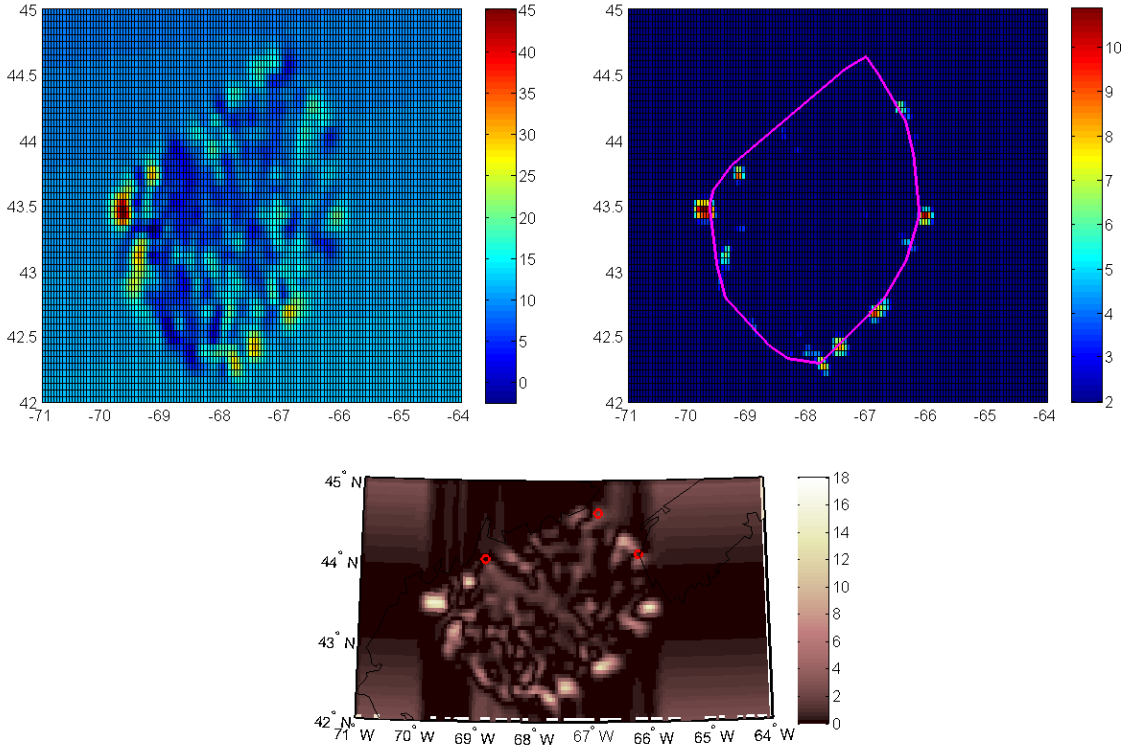


Figure 6.1: Three views of averaged V-component fields on March 20, 2014 – residual V-component values (left), significantly correlated V-component Z-scores (right), and I-field of V-comp autocorrelations (bottom)

The three snapshots presented in Figure 6.1 illustrate a typical process in the generation of a field of Moran's I statistics for each location containing a CODAR reading on March 20, 2014 in the Gulf of Maine. The first snapshot (left) presents the field of v-components to the residual vectors produced by the global MLP model. The second (right) takes the Moran's I field and converts it to z-scores to highlight the significant instances of autocorrelation (hereafter referred to as the *Z-field*). Note that the

minimal z-score in the range is 1.96; therefore only values significant at the 95th percentile and beyond distinguish themselves from the background. Note also the boundary of the convex hull, denoting the extreme limits of the field of readings. Finally the third snapshot (bottom) displays the full range of computed Moran's I values in the context of the coastline of the Gulf of Maine, including the locations of the three main CODAR stations responsible for the original field of readings.

The context of the convex hull of original readings is particularly important because, in order to generate a rectangular snapshot, all values outside the convex hull are extrapolated from the values within, and should thus be discounted A kriging process was used to implement the extrapolation, using a Gaussian correlation model for smoothing.

6.2.1 Rationale for Excluding Edges of the I-Field

The majority of z-scored significance values in the top-right snapshot of Figure 6.1 are located on or outside the boundary of the convex hull. The fact that these peaks often occur at these locations can be attributed to two factors: extremities of CODAR range, and Gaussian extrapolation. As noted in the prior chapter, the real-world dataset that notionally represents an embedded wireless sensor network in the Gulf of Maine is actually generated by several coastal radar installations, currently ranging from Maine to Nova Scotia, Canada. The extent of the radar signals projected by these stations may vary from one hour to the next due to atmospheric conditions, and so the greatest uncertainty occurs in those readings that are gathered at the extreme leading edge of one radar station or another. Since these readings are nearly always found at the current edge of the spatial

field, it is easy to understand that the corresponding field of error magnitudes also slopes upwards toward the edges of the field. The choice to use a Gaussian correlation model in the kriging model rendering a continuous surface for display of the field of Moran's I values (referred to hereafter as the *I-field*) means that an upward slope at the edge of the actual field would create a spurious peak just outside the convex hull of the actual field of readings. This being the case, we treat significant peaks located upon the convex hull boundary with skepticism (due to the high uncertainty associated with the signals comprised within those readings), and ignore values outside the convex hull completely. Later in this chapter, when comparing populations of significant I-field values between models, we only consider readings a minimum distance within the boundaries to avoid as much as possible the most-compromised instances.

6.2.2 Comparison of Moran's I in the U/V fields between Global and Partitioned Models

To display results of a model's autocorrelated errors over a given length of time, the squared residuals of all the model's results for the time period were queried from the database, and averaged by location. As individual residuals may be either positive or negative, squaring their values prevents any cancellation from occurring prior to averaging. The I-field was then generated by passing the chosen weight matrix (i.e., 3x3 averaging filter described above) over the field of values, extrapolated to fit an $m \times n$ region fit for display.

A second copy of the *mxn* region was z-scored by subtracting the matrix's mean from each value and dividing by the matrix standard deviation, according to the following standard formula:

$$Z = \left(X - \overline{X}\right) / \, stdev(X) \tag{6.1}$$

This Z-field allows for the determination of significant instances of error autocorrelation in the variable of interest X.

Examples of Z-fields and I-fields of the U and V components of the residuals for both the Global and Partitioned models are shown in Figure 6.2 and Figure 6.3.

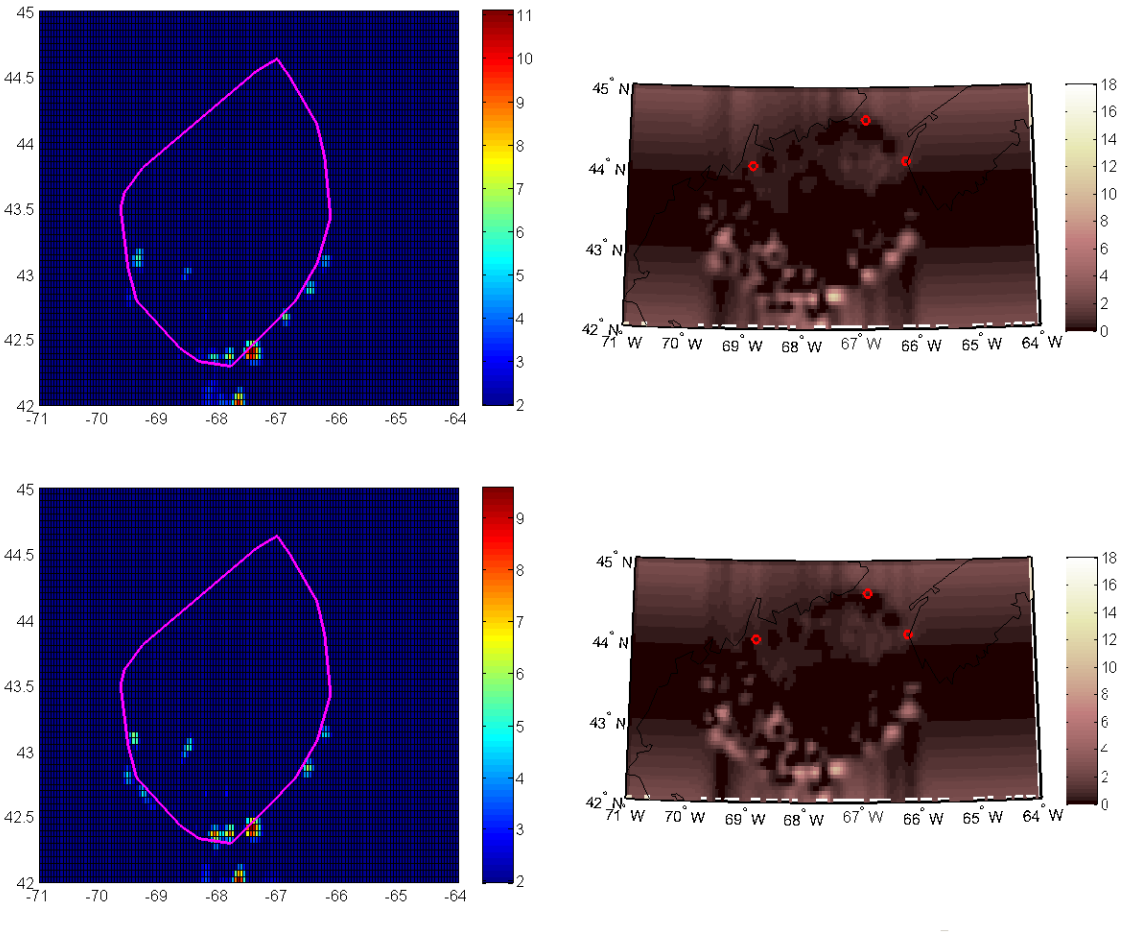


Figure 6.2: Comparison of the U-component's Z-fields (left) and I-fields (right) resulting from the averaged results of the Global model (top) and Partitioned model (bottom) on March 20, 2014

The U-component results show relatively little significant autocorrelation within the convex hull delimiting the field of approximated values (and nearly none at all in the south-central region of the Gulf, interestingly). Most of the significant instances of autocorrelation are found along the bounds of the convex hull for both residual components. Intriguingly, the partitioned model's residuals display generally shallower peaks than the single Global model's residual field for the U-component. The structure of the I-field in both models' results appears otherwise generally the same.

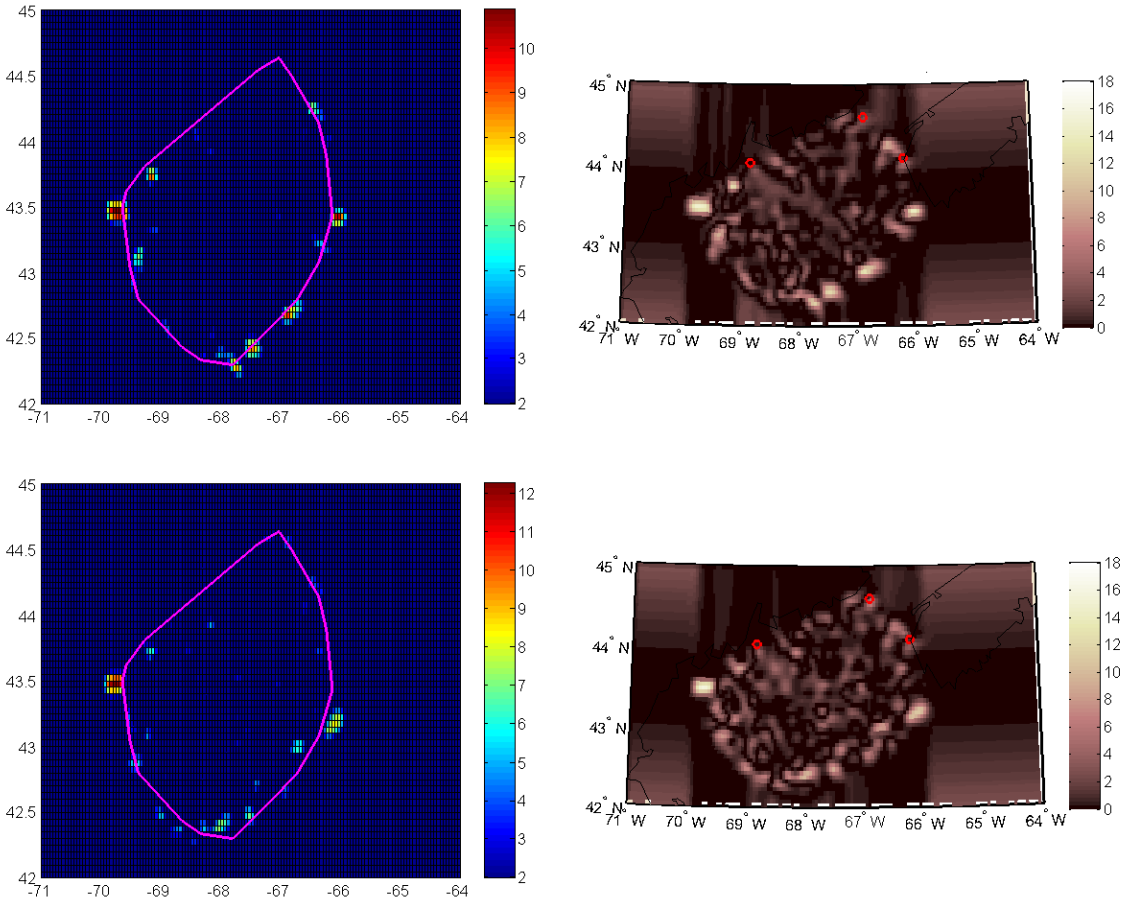


Figure 6.3: Comparison of the V-component's Z-fields (left) and I-fields (right) resulting from the averaged results of the Global model (top) and Partitioned model (bottom) on March 20, 2014

The surface of the V-component's I-field presents as significantly choppier than that of the U-component. This may be because much of the natural system's energy is often found in the V-component in this particular testbed (as suggested by the importance ascribed to the V-component in the selected clustering scheme in the previous chapter). Though it might be argued that the choppiness in the residual field is a result of the use of the clustering scheme to build the partitioned model, it should be noted that the Global model's I-field also shows similar structure without the model having used the clustering scheme. If significant system energy was indeed generally concentrated into the testbed's V-component, however, we may expect that both MLP model systems would have naturally adapted in that direction as they sought to model the axes of greatest variation within the system.

Like the U-component results, most significant residual component autocorrelation results are found along the convex hull, where the largest and most consistent errors are expected by default due to physical signal attenuation. Relatively more significant instances of autocorrelation are found inside the hull for the V-component however, and the Partitioned model's peaks appear to be generally higher than those of the Global model.

6.2.3 Determining the Change in Mean Autocorrelation Between Models

But however illuminating, results for a single day are not representative of a model's performance over its lifetime. To capture the difference in autocorrelated error between the two models, we employ once again the paired-value t-test technique, pairing for each unique data instance each of the Moran's I values returned by the Global and

Partitioned models. Histograms of residual-differences were generated separately for the U-component, V-component and Total Magnitude of each model's residual set, and paired t-test hypothesis tests were run, with the null hypothesis H_0: $\mu_{PART} - \mu_{GLB} = 0$ (i.e., the means of each model's Moran's I distribution are not significantly different).

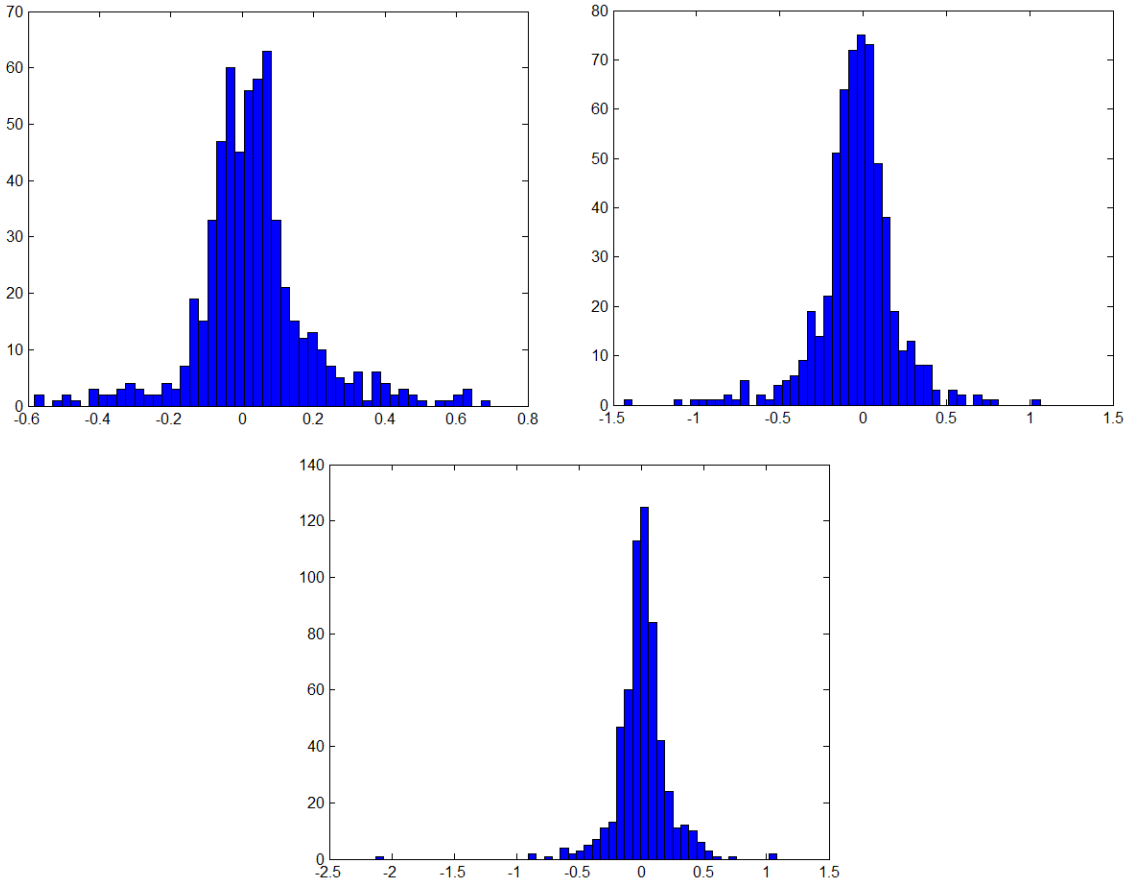


Figure 6.4: Histograms for the differences of paired Moran's I values (PART – GLOBAL) produced by U-component (left), V-component (right) and total residual magnitude (bottom) for the month of April 2014

Two approaches were taken for each of the paired t-tests: a month's worth of data could be aggregated to location-averages for the month, yielding approximately 512 pairs of values; or daily averaged values for each location could be retrieved, yielding approximately 12,000 pairs. These population sizes are large enough that Student's t-

tests should be applicable, by the Central Limit theorem. Ultimately either method returned roughly similar histograms, so the daily-averaged histograms are not duplicated here. The t-test results are shown in the following table

Table 6.1: Paired t-test results for April 2014 Moran's I fields as produced by Global and Partitioned models, aggregated both monthly and daily by location

		Mean Difference	**H_0 Rejected**
U-component	monthly-averaged	**+0.03**	TRUE
	daily-averaged	**-0.0023**	FALSE
V-component	monthly-averaged	**-0.04**	TRUE
	daily-averaged	**-0.04**	TRUE
Total Magnitude	monthly-averaged	**0.00**	FALSE
	daily-averaged	**-0.01**	TRUE

Executing separate paired t-tests by component-fields had mixed results, especially when aggregating in either monthly or daily terms. The monthly-averaged U-component test indicated a significant increase in the Partitioned model's (u-component) error autocorrelation, while the daily-averaged Moran's I values failed to reject the null hypothesis. Both V-component tests indicated significant decrease in the Partitioned model's v-component residual autocorrelation. Whereas for the autocorrelation of the total magnitudes of the residuals, tests indicated either no significant change (monthly-averaged) or a very small reduction in the Partitioned model (daily-averaged). This latter test's results do not much surprise since, as the u- and v-components are combined to obtain the magnitude, the overall effect would favor a slight reduction in the Partitioned model's results. As the majority of tests reject the null hypothesis to indicate a

statistically significant reduction in error autocorrelation in favor of the Partitioned model, the weight of the evidence perhaps should be taken to support that conclusion.

A well-known downside of the t-test technique, however, is that given a large enough number of data instances, a signal is virtually certain to be detected within the set sufficient for rejection of the null hypothesis. The magnitude of the significant differences in this case (i.e., on the order of 10^{-2} – two orders of magnitude below that of the data itself) does not quite sit comfortably with us. Although the Partitioned model may be said to have significantly less autocorrelated error in the statistical sense as a result of these tests, the amount of autocorrelated error produced by either model may in a practical sense be considered to be virtually identical.

We may nonetheless represent these I-value differences spatially in a set of bubble-charts, one for each component-field, located over the Gulf of Maine. In the following bubble charts, size of the blue bubbles represent better relative performance on the Moran's I scale (i.e., less autocorrelated error – that is, Moran's I is closer to zero) by the Partitioned model, and size of white bubbles represent better relative performance in favor of the Global model.

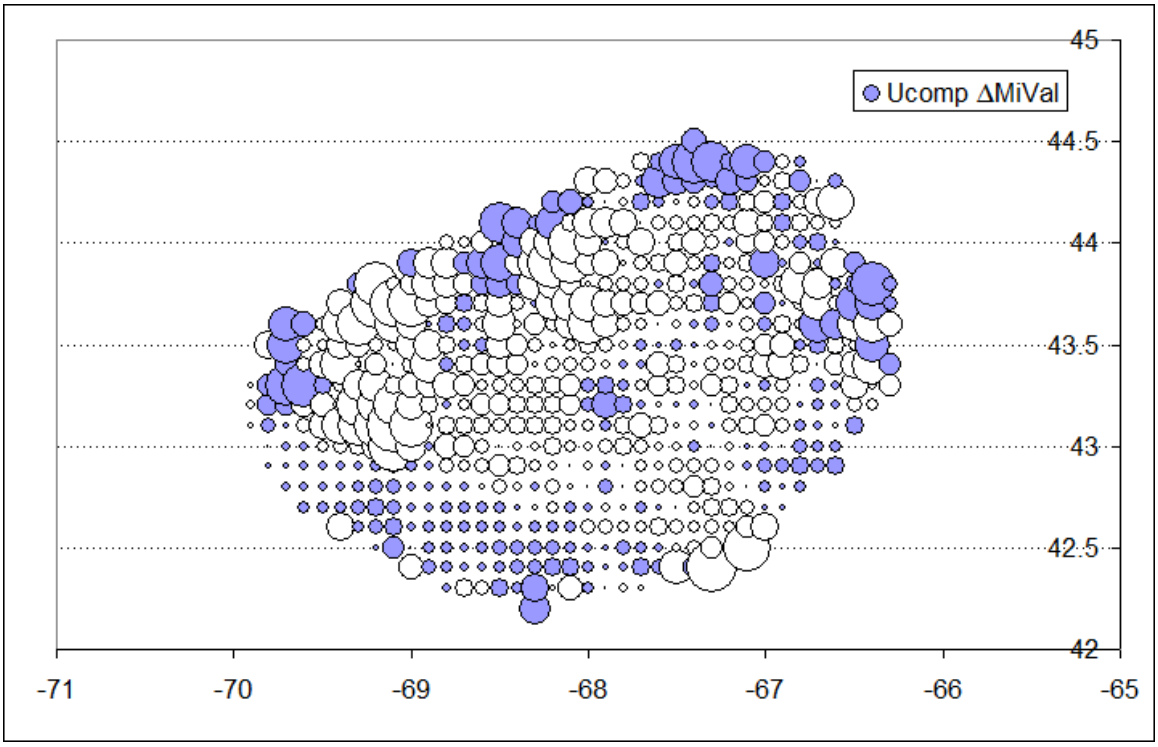


Figure 6.5: Relative differences in I-values of U-component residuals between Global and Partitioned models (Better performance by the Partitioned model is represented by blue bubbles, and by white for the Global, while bubble-area represents magnitude of relative improvement)

Figure 6.5 shows a relative dominance by the Partitioned model in the center of the field for U-component approximation, while the Global model appears to do better at the edges. Spatial distribution of each model's bubbles is not random, indicating a difference in the bias of each model, possibly leading to model-driven second-order effects in their residual fields in those areas where each model's bubbles are clustered. However the energy in the U-component is not as significant as that found in the V-component, so Figure 6.6 should be consulted before any final determination on the matter.

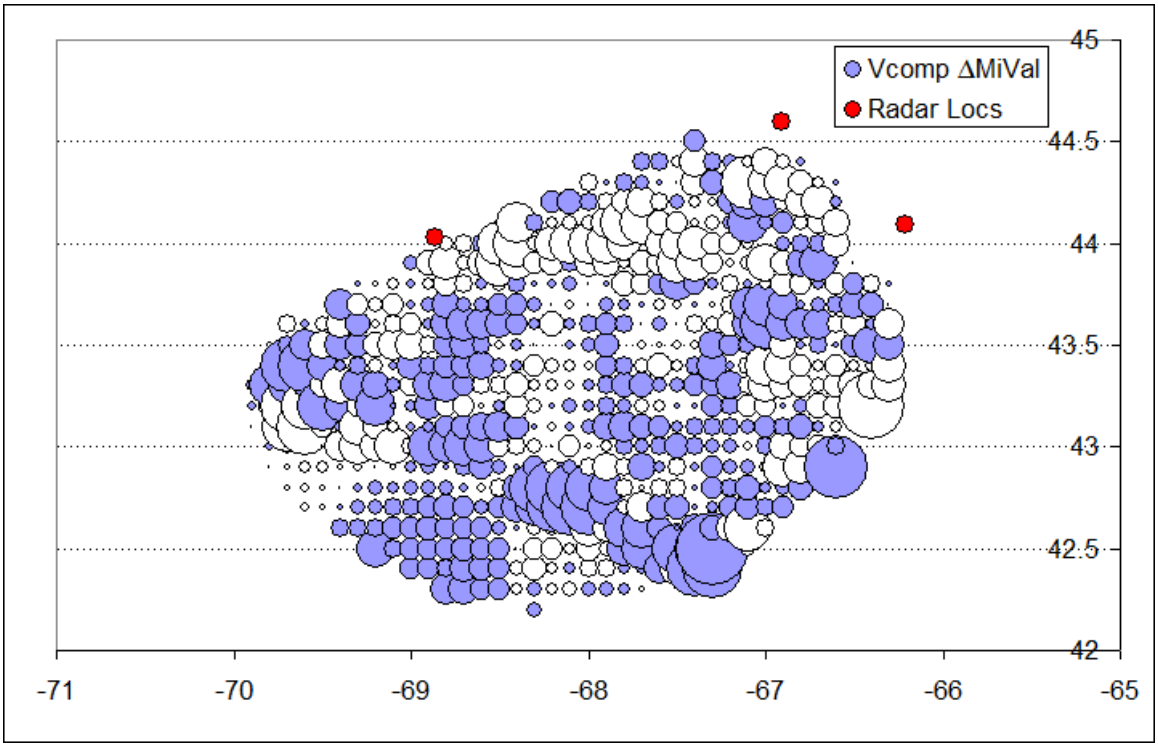


Figure 6.6: Relative differences in I-values of V-component residuals between Global and Partitioned models (Better performance by the Partitioned model is represented by blue bubble, and by white for the Global, while bubble-area represents magnitude of relative improvement)

Figure 6.6 shows a decided advantage for the Global model when the two models are compared. White bubbles dominate nearly everywhere, with the exception of two sections located approximately along the straight-line stretches between the coastal radar installations, where the Partitioned model's results appear to dominate. Note of course that this should not be taken to mean that the latter model's I-values are necessarily low in those regions, only that they are lower on average than the Global model's.

Finally, as the u- and v-components are combined to obtain the error vector's magnitude, a Moran's I field is generated for correlated error magnitudes yielding Figure 6.7, which appears to be a blend of the two preceding figures. The bubbles of each model appear to be slightly more interspersed than in the U-component's Figure 6.5, and the Partitioned model does not appear nearly as dominant overall as in Figure 6.6.

These figures reinforce our feeling first prompted by the paired t-test results: that no practical difference exists between the error correlation in the results returned by either model, even though the Partitioned model in the V-component may technically hold a slight advantage in that regard.

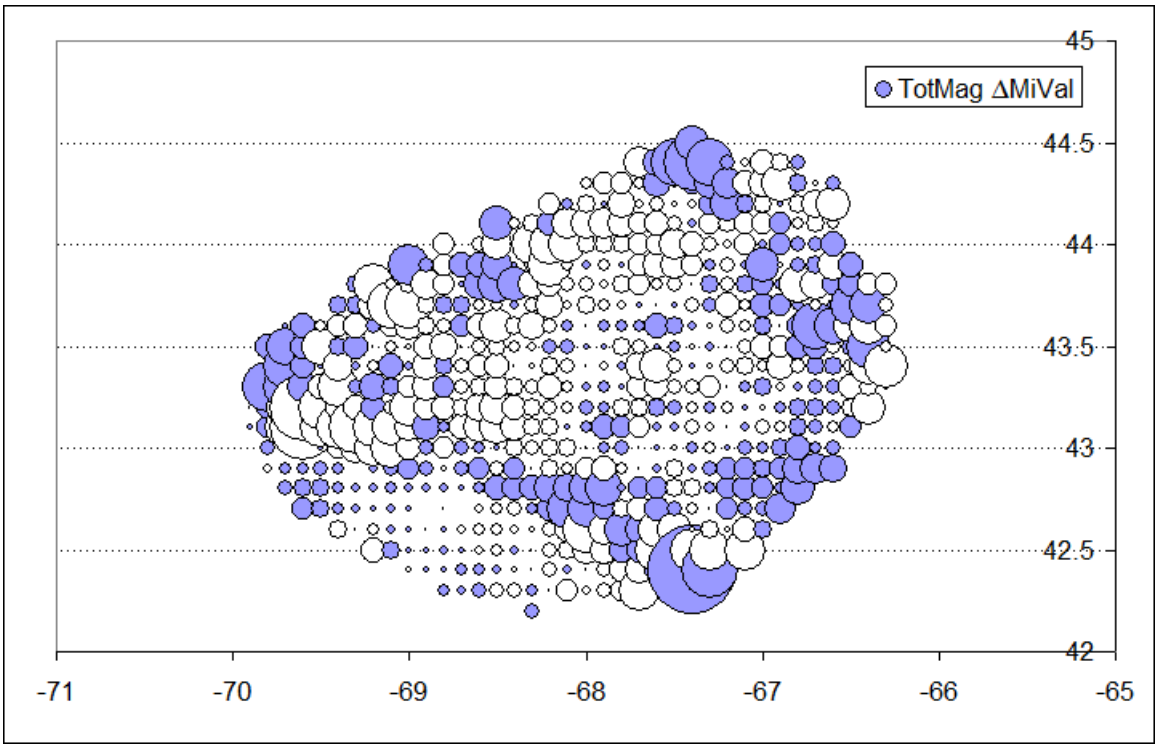


Figure 6.7: Relative differences in I-values of Total Magnitude residuals between Global and Partitioned models (Better performance by the Partitioned model is represented by blue bubbles, and by white for the Global, while bubble-area represents magnitude of relative improvement)

Given that the amount and magnitude of correlated error appears roughly equal between the two models, it appears safe to conclude that the partitioned approach used by the Partitioned model does not lead to additional spatially-correlated error in its results.

6.2.4 Excluding the Possibility of a ST-Leviathan Volume in Residual Fields

Series of two-dimensional spatial graphs are often likened to the pages of a book. Proceeding forwards through the series, like flipping forward through the pages of a picture-book, allows one to witness located values in space as they progress over time. One limitation of the Moran's I statistic as defined is that it does not take into consideration adjacent values in time as well as space, a limitation most recently addressed by (Dubé and LeGros 2013a) and (Dubé and LeGros 2013b). It is possible that a model may leave correlated values along the time axis as well – either as artifacts of the model's bias, or due to the presence of second order processes unexplained by the model. This leaves open the possibility that along the temporal axis a possible 3-D region of significant correlated error values may lurk; a leviathan whose volume we may not detect because we only glimpse it in innocuous cross-sections as we look at the surface maps on each individual page of the book of Time.

Extending the picture-book analogy, we might imagine that some pages within the book are subject to a dry-mold agent causing a particular region of a given page to crumble away to dust over time. This agent might then make contact with neighboring pages as well, causing correlated regions of adjacent pages to also molder away. These correlated voids over a series of pages would create a hollow within the closed book, invisible to anyone viewing it upon the shelf. Taking an X-ray image through the front cover of the closed book, however, would detect this hollow. We have implemented this x-ray analogy computationally as the Maximum Significant-I Persistence (MSIP) technique, to detect the maximum volume of potential correlated errors left by the model along the time axis.

6.2.5 Maximum Significant I Persistence

The purpose of the MSIP technique is to get a sense for how large a volume of persistent significant Moran's I values may potentially exist along the time axis. The MSIP algorithm consists of the following steps:

For each time-step:

1) **Initialize two *mxn* matrices PS and MX to all zero values**
2) **Generate a *mxn* matrix MI of z-scored Moran's I correlation statistics for the given time-step**
3) **For each value above the chosen significance-value *and* within the current time-step's convex hull, convert value to a 1; convert all other values to 0**
4) **Accumulate this matrix into the Persistent-Sum matrix PS such that:**
   a. **any zero-value cells in MI cause corresponding cells in PS to reset to 0**
   b. **any non-zero cells cause corresponding cells in PS to increment by one**
5) **Finally, for each cell i whose value in PS is greater than the corresponding value in MX, let MX[i] = PS[i]**

The resulting matrix *MX* will indicate the maximum length of consecutive significant instances of correlated error for the variable in question (though with no guarantee that any maximal series of values is cotemporaneous with any others). Examples of resulting *MX* images follow.

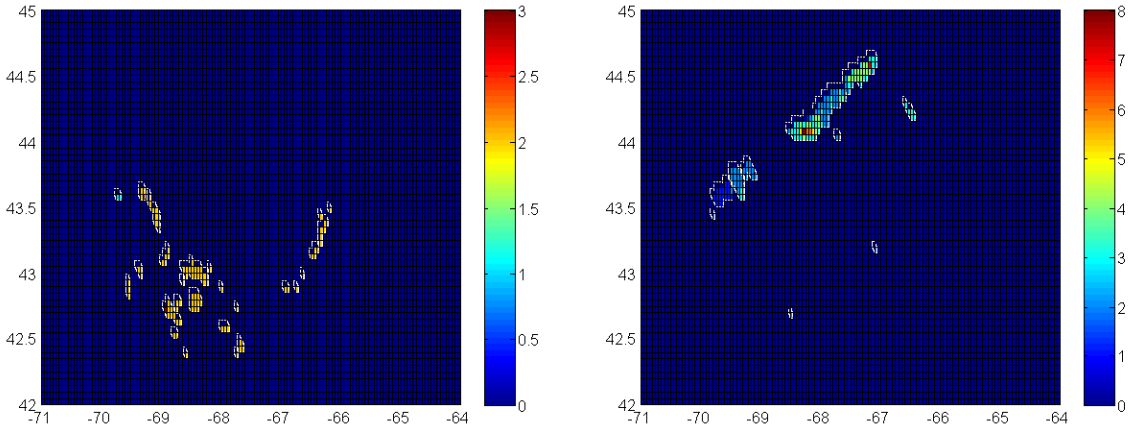


Figure 6.8: Global model MSIP images of the U-component (left) and V-component (right) at 0.01 level of significance for April 2014; aka "beard" and "toupee" signatures

Figure 6.8 illustrates the two signature patterns of significant error occurrence in the MLP model results of the Gulf of Maine dataset. As seen earlier, U-component correlated error tends to occur along the southern edge of the monitored region, while V-component correlated error tends to occur along the main coastal (northwestern) boundary. These tendencies lead to a "beard" and "toupée" nomenclature for the resulting signature patterns. Given that both patterns occur along the edges of the field, both can be explained away. The toupée effect for example may largely be due to coastal interference to the radar signals, such as coastal outcroppings or marine traffic, in addition to the uncertainty created by signal attenuation at the edges of the field. The beard effect can be largely ascribed to signal attenuation at the edges of the convex hull, especially as the southwestern boundary can be seen to be in a near-constant flux of advance and retreat as radar signal range varies with atmospheric conditions. It is not surprising therefore that the U-component's MSIP values do not reach the same maximal values as those along the more-constant coastal boundary. It should also be noted that only those values found to be significant at the 0.01 level or better were noted.

If a hidden leviathan is to be found in this resultset, the threshold of significance may need to be lowered. In order to maximize the size of a potential hidden volume, the significance threshold was lowered to 1.645 (i.e., the 0.1 significance level) for the partitioned model's results.

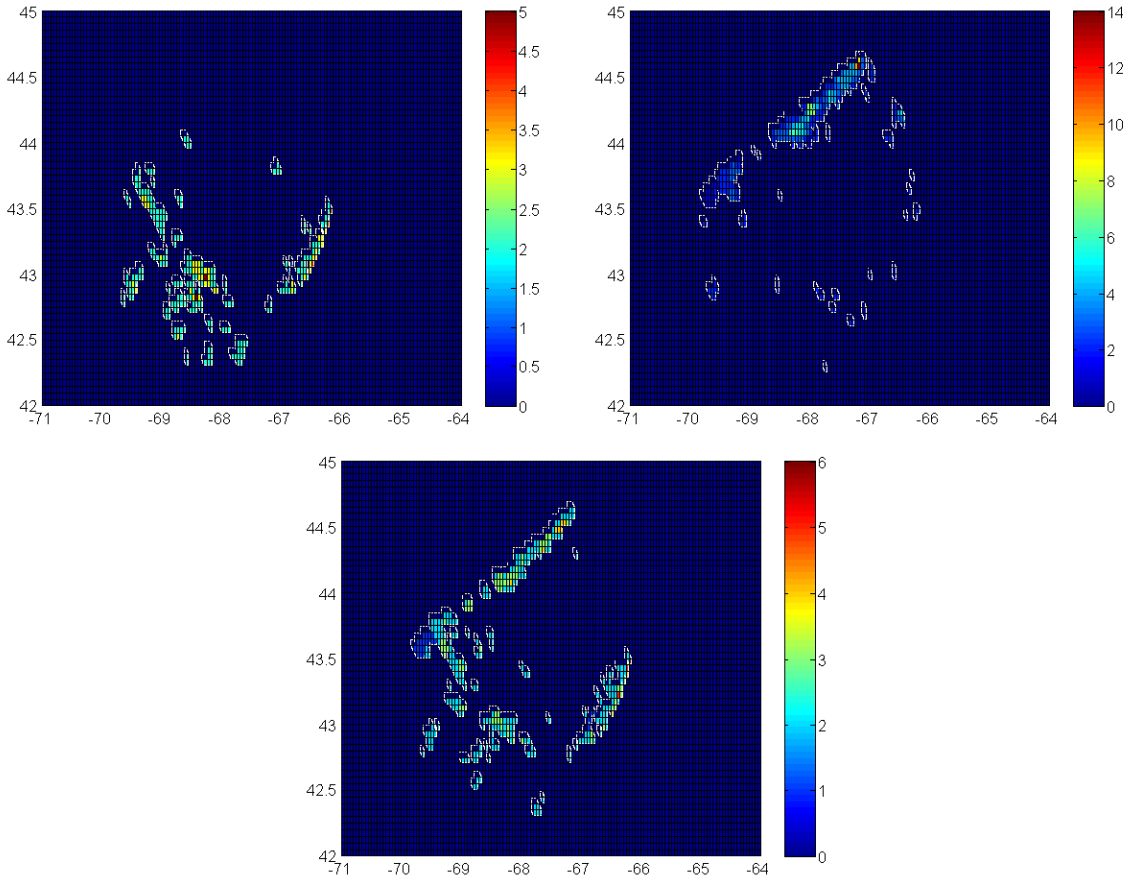


Figure 6.9: MSIP values for U-component error (left), V-component error (right), and residual magnitude (bottom) from Partitioned model results for April 2014

Figure 6.9 shows the signature regions to be significantly enhanced by the lowered significance threshold. Of particular note is that few regions of significantly correlated error can be found inside convex hull boundaries in any component at the 0.1 significance level. The bottom snapshot of error vector magnitude autocorrelation

appears to be a combination of the preceding individual component results, which one might well expect. The only structure of some note would appear to the values arcing to the southeast boundary from approximately (-69, 44), as this region is often within the field's convex hull. As was mentioned before, none of these max-length series is guaranteed to be cotemporaneous with any of its neighbors. But assuming that all of them in this suspicious structure are, the volume of the resulting structure would still struggle to reach even 1% of the total number of cells within the field's convex hull over the course of the month of April 2014 (e.g., an approximate average of 200 cells per day over 30 days). As such, even if this structure does represent a potential second-order effect, and is not conflated with artifacts of the shifting boundary, it does not rise to a great level of significance.

Alternately, this could just as easily be explained as being the high-water mark of the advancing and retreating southwestern bound of the convex hull. Other than this faint possibility, no second order effects of note would appear to be concealed along the temporal axis of this spatiotemporal model's residuals.

6.3 Determining Refresh-Timing

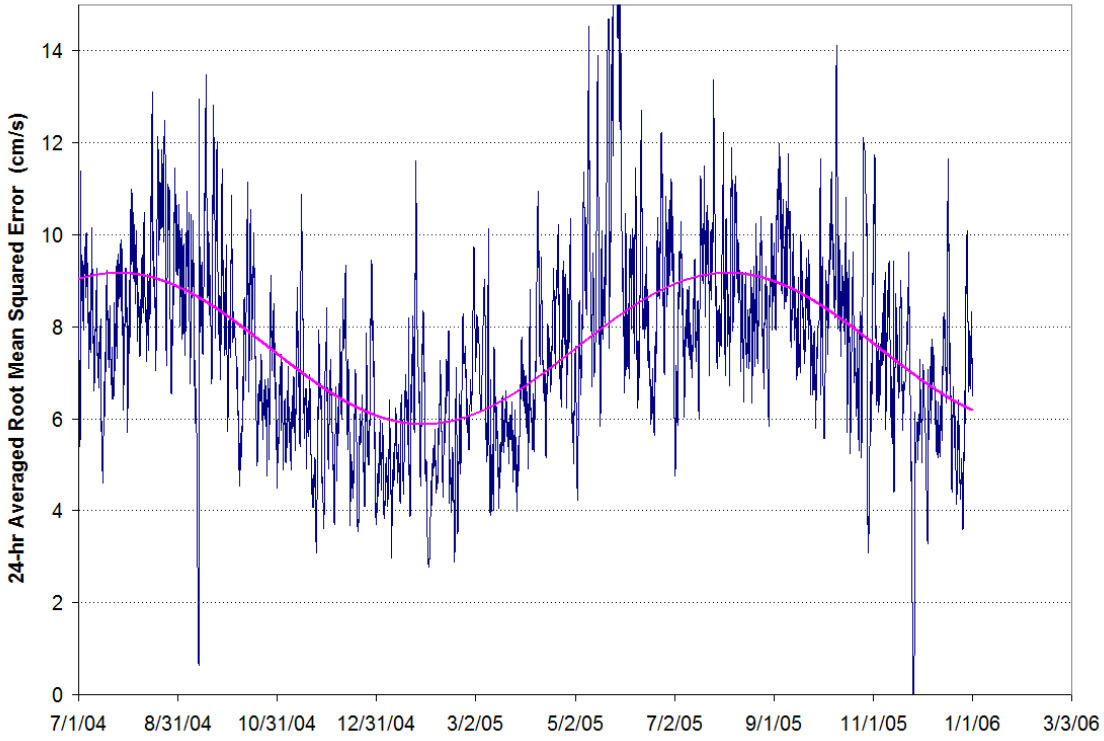


Figure 6.10: Seasonality in error is exhibited by an MLP model fitted to data collected by GoMOOS buoy I in the Gulf of Maine over first half of 2004, 2005

Having developed a viable system for generating a composite reading-approximation model comprised of N MLP models, some final questions remain. Any model trained to mimic a dynamic system can be expected, after some length of time, to see its results diverge from the system it is meant to approximate. This happens because fitted models are trained on data entries from a specific period in time, and the further removed in time from that training period the model becomes, the less valid to the current situation are its base assumptions that were induced in the training process, and therefore the less accurate will its approximations become. How long can this hybrid model be expected to function appropriately before needing to be refreshed? And how can we tell that the refresh-time has arrived?

Focusing on a single sensor-platform (GoMOOS buoy I) in the current Gulf of Maine testbed (Figure 6.10), we can observe that significant seasonality is exhibited in the readings collected at that single location by a simple MLP model trained over the first half of 2004. It might therefore make sense to refresh the compound model at least every three months, in order to readjust its outputs to current seasonal conditions. One might also observe in the figure however that even within a three-month period, extended periods of higher, well-correlated error still occur, suggesting that performance could be optimized further by an automatic refreshing regime that could detect when the current conditions have significantly departed from those to which the partitioned model was originally fitted. To accomplish this as simply as possible, we might conceive of a paired t-test of two populations of collected readings.

6.4 Comparing Populations of Readings for Automated Refresh-Timing

The Gulf of Maine testbed monitoring system collects readings on an hourly basis. Depending on atmospheric conditions over the monitored region, it may sometimes harvest 30 readings, sometimes 500, sometimes zero. Since we cannot count upon gathering a reading from any given location from one hour to the next, we collect instead the entire population of readings gathered over the previous seven day period. This gives us a base set of readings that can be compared to subsequent seven-day periods to determine if the new population's mean has significantly changed. This suggests a moving-window approach where a window seven days long is moved along the timeline one hour at a time, until the significance level of the paired t-test comparing

the current population to the original indicates that they are sufficiently different that a refresh of the current models may be initiated.

Figure 6.11 shows a time-series of p-test values generated by a moving window seven days in length comparing its population of readings to the original seven day population when the operational system of models was put in place. The null hypothesis is H_0: $\mu\_orig - \mu\_current = 0$. The significance threshold is set at 0.005. When this H_0 is rejected at that significance level, we have convincing evidence that the populations have diverged sufficiently that a new set of models should be trained and put in place. According to the figure, that point is reached after approximately 40 days.

A potential complication to this straightforward approach may be glimpsed in the figure in the sudden outcropping between days 55 and 60. If we had chosen to retrain the system on day 55, only to see conditions revert to a state of "not significantly different," might we not regret our decision? What about when another outcropping occurs at day 100? Or 139? Or 150? Using a moving window might generate uncertainty regarding when a signal truly indicates significant divergence, or if a temporary "false" signal is being generated by a short-lived condition in the monitored region, such as a storm system passing through the Gulf of Maine. One might wish for a system more resistant to such temporary anomalies to avoid *model thrashing*, that is refreshing the monitored system's models too early or unnecessarily – especially if the refresh process is expensive in terms of time or stored-power in the WSN expended.

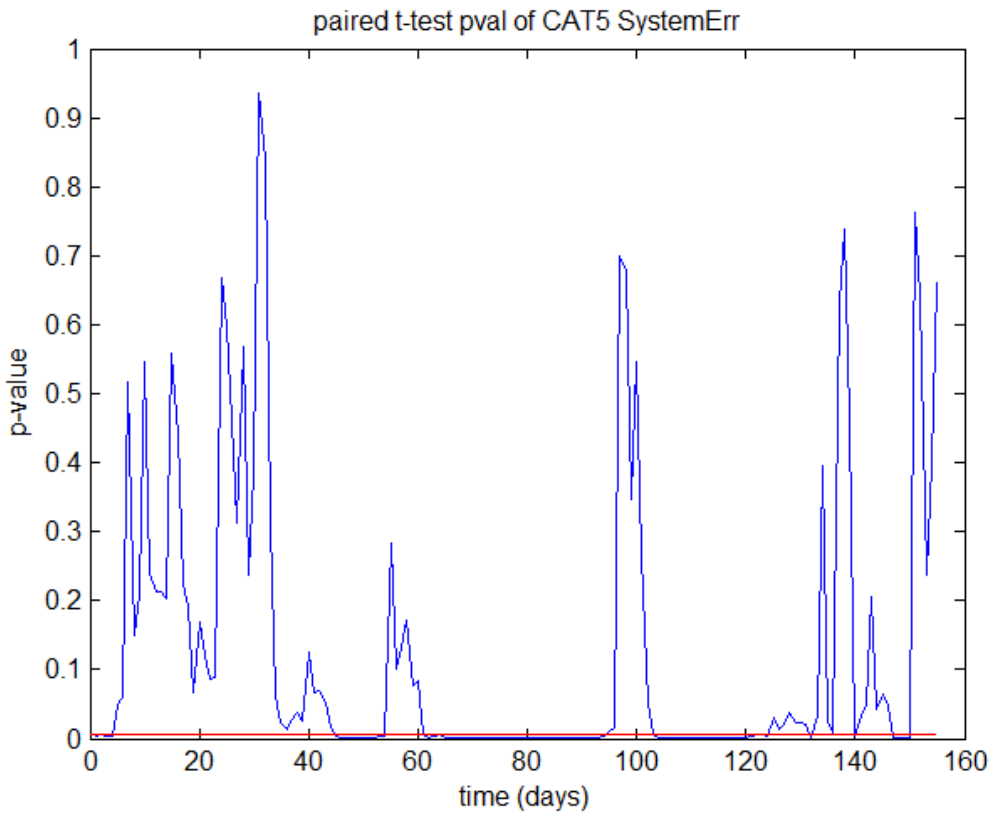


Figure 6.11: Series of p-values resulting from paired-t-tests over 7-day populations of collected readings (red line indicates significance threshold of 0.01)

A potential solution to this concern is illustrated in Figure 6.12. Rather than a moving window of constant size, the window-size is *expanding* instead, comparing an ever-growing population of new readings to the original seven-day population. As new data instances are added to the window's population no comparable number of instances is leaving it, so the overall population's center is much harder to shift by a sudden influx of anomalous newcomers. The resulting silhouette may give the user more confidence in which choice to make as the decision is rendered much more binary. Given the constantly accumulating effect of the expanding window, it may take the signal somewhat longer to collect the weight of evidence needed to cause a drop beneath the threshold under this

approach, but once it does, it will generally have cause to stay below. Should the signal manage to climb back above the threshold, it will generally not have sufficient support to remain above it for long. The overall effect is to instill confidence in the user that once the threshold is breached, it is legitimately time for a system-refresh. Moreover, comparing the two figures we can see the same events of the former represented in the latter; for example, the peaks occurring at 40, 60 and 100 can be found in both. As time goes on, it becomes increasingly difficult for the signal to remain above the threshold, which is as it should be to encourage an appropriate refresh-rate of regional models, without unduly squandering resources by doing so too often.

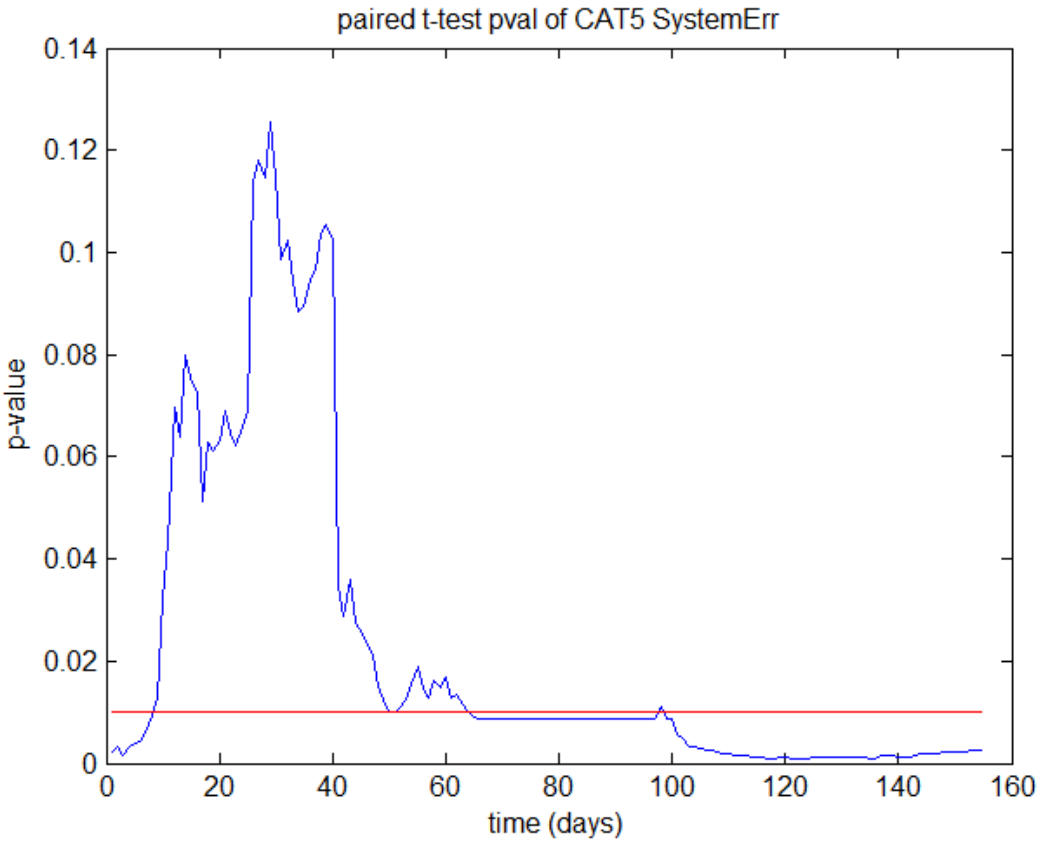


Figure 6.12: Series of p-values resulting from paired t-tests over mean-aggregated populations of collected readings (red line indicates significance threshold of 0.01)

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

7.1.1 Model of Choice: MLP

In this work we have endeavored to shed some light and gain additional understanding on some aspects of spatiotemporal fields, as determined by networks of sensor nodes embedded in natural environments. In order to do so, we have proposed the use of the feed-forward multilayer perceptron (MLP) artificial neural network model, as it meets the criteria desired for a model used within a network of independent wireless sensor network nodes. That is, they are capable of approximating nearly any continuous function; they may accept, process and output parameters of varying measures and types, spatial or temporal; they are adaptive and self-modifying; they may be represented as a finite collection of scalar values for transmission, and may be implemented by a finite number of relatively uncomplicated mathematics, such as multiplication, addition and the logistic function. These properties suggest the MLP as an appropriate model for applications in environments where computation platforms have access to a finite store of energy, communication between sensing platforms is constrained, and processing and storage capacity on these nodes is necessarily limited. As MLPs have been shown to be universal approximators, it makes them appropriate candidates for the approximation of spatiotemporal fields.

7.1.2 Research Questions

This dissertation set out to address several research questions. The first of these was, are MLP models reasonable candidates to apply to spatiotemporal applications. Though most indications listed above would suggest them to be so, we have achieved some confirmation of this through experimentation in Chapter 0, where the MLP's ability to incorporate temporal as well as spatial inputs into its computations demonstrated comparable performance to Ordinary Kriging, a model whose spatial approximation performance is considered a standard of quality, but whose data-access advantages and computation-intensive approach is not generally feasible for a storage- and power-limited wireless network context.

The second question to be addressed was, can the partitioning of space be a benefit. Several studies (Bellman 1957; Brunsdon *et al*. 1998; Fuentes *et al*. 2005) would suggest this to be so due to its potential to reduce Bellman's *curse of dimensionality,* as well as to mitigate the nonstationarity inherent in complex realtime natural systems. Through experimentation in chapters 4 and 5, the performance of a single global MLP model was compared to that of a partitioned system of MLP models, yielding significant improvement in the resulting signal approximations of the partitioned systems within both a synthetic dataset, and one drawn from the Gulf of Maine ocean observing system maintained by the University of Maine Physical Oceanography Group.

Given the non-stationarity of the monitored systems, and despite the MLP's native ability to adapt to changing conditions, we need to consider the possibility of having to reevaluate, retrain and eventually replace working models embedded in the

field with models better fitted to recent data and current dominant conditions in the monitored region. We have explored such a potential process in chapter 6, by monitoring model error-distributions over time and comparing them to error-distributions of the freshly-trained model, we can determine a reasonable threshold after which the initiation of a model-refresh process may justifiably be initiated.

Finally, the thesis explored whether any benefits could be realized by the stacking of models; that is, does the system of partitioned MLP models leave a coherent field of second-order effects that, separated from the first-order effects, could be easily processed by a second model-type? Though such a possibility might certainly exist, the use of expressive MLP models with sufficient degrees of freedom and representational power should render this unnecessary. We found this indeed to be the case with the Gulf of Maine dataset, as the residual field left by the model did not contain regions of significant error correlation throughout the main portion of the monitored region, but only at its edges, where the sensing power of the instruments is limited.

7.2 Future Work

As the field of study comprising wireless sensor networks is still in many ways in its infancy, standards are still being developed to appropriately evaluate network performance, model choice and evaluation, and system maintenance. Certainly, the MLP is not the sole model capable of representing spatiotemporal fields; differential spatiotemporal functions have been proposed (Hofer and Frank, 2009; Weiser and Frank, 2012), as well as fitted Fourier functional formulations (Guan *et al.*, 2011), and either of these may well represent attractive alternative approaches moving forward . In some

ways, a differential equation model could be preferable over neural networks; for example the former is not a black-box process and so its structure can more easily be decomposed and understood. It remains to be seen, however, if these models can exhibit the same representative power, malleability and ease of use for heterogeneous input sets as equivalent neural network models.

Use of the basic k-means technique worked well, but has some limitations. In the absence of much similar work, however, it should provide a useful baseline performance that future constructions may be measured against. The k-means approach comes with a set of base assumptions that will not all be met in every spatiotemporal dataset, nor were they all met in the one presented here. Further, as it readily accepts a multitude of input parameters, problems associated with its dataset violating those base assumptions are likely to worsen due to Bellman's *curse of dimensionality*. In the end, for best performance any particular clustering mechanism will need to either (1) be particularly chosen for use with a dataset based on a fitness measure corresponding to the fewest number of important base assumptions violated; or (2) transform the dataset in order to bring it within closer compliance to the desired clustering technique. Pathological cases can be constructed for any clustering method, illustrating that all such methods have weaknesses and cases for which they find themselves embarrassingly ill-suited. The "no free lunch" theorems of (Wolpert and Macready, 1997), stating the basic (paraphrased) idea that "When averaged across all possible situations, every algorithm performs equally well," means that for any particular dataset, better optimization options than k-means are quite likely to exist, even if when compared over all datasets, k-means fares no worse than any of them.

Even in the field of artificial neural networks, however, this study does little more than scratch the surface. Due to the relative paucity of prior work in this niche, particular effort was made to explore a general, baseline approach which could subsequently be elaborated upon; with a greater variety of inputs to the model for instance, or more attributes taken in consideration during the k-means partitioning process. For instance, although MLP are by their nature capable of continuous learning in the course of their operation, for simplicity's sake none of the models in this work were allowed that capability; primarily because the period of time a model would require to be operational in the environment and receive enough new data instances to significantly modify its initial configuration would generally be far longer than a reasonable refresh period.

Other avenues of interest to be explored within the field of neural network models involve allowing operational models to structurally self-modify, in order to better respond to changes in the monitored dynamic environment (Karimi *et al*. 2014). Although again the time period required for self-modification may be prohibitive in the natural Gulf of Maine testbed described in this work, either or both dynamic restructuring and continuous learning might be viable options in more dynamic monitored systems, or in systems where readings are taken with much greater frequency.

Much of his work was posited on the operation of a simulated wireless sensor network (WSN) embedded in a natural environment. As growing numbers of actual WSN are embedded in such environments, work on spatial databases and data stream engines that process spatial-temporal readings of continuous phenomena (Whittier *et al*. 2013) and discrete events (Beard *et al*. 2008) seem to represent an increasingly relevant step forward in this discipline. As the monitoring via WSN of previously difficult or

otherwise impossible-to-monitor regions becomes more viable both technically and fiscally, investigation of novel stream query approaches and strategies to achieve real-time spatial interpolation of space-time processes such as (Nittel *et al*. 2012) should prove extremely valuable to researchers in the wider geographic information science community.

Our experience with the standard implementation of the MLP model shows that spatial-temporal partitioning techniques are effective and relatively efficient to implement, leading to significantly improved results over single global models monitoring the same region over time. When many potential inputs are available to embedded models, careful determinations will need to be made to ensure the best return given available computational resources and the power expended in an energy-constrained environment. This work demonstrated that even with a limited range of potential inputs both temporal and spatial in nature, significant improvement should be obtainable given relatively modest power and computational expenditures suitable for a wireless sensing network independently embedded within a monitored natural environment.

REFERENCES

Abadi, D. J., Madden, S., and Lindner, W. (2005). REED: robust, efficient filtering and event detection in sensor networks. *Very Large Data Bases, Proceedings of the 31st international conference on*. Trondheim, Norway, VLDB Endowment: 769-780.

Akyildiz, I. F., Melodia, T., and Chowdhury, K. R. (2007). A survey on wireless multimedia sensor networks. *Computer Networks* 51(4): 921-960.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2007). Wireless sensor networks: a survey. *Computer Networks* 38(4): 393-422.

Apiletti, D., Baralis, E., and Cerquitelli, T. (2011). Energy-saving models for wireless sensor networks. *Knowledge and Information Systems*, 28(3), 615-644.

Bailey, T. C. and Gatrell, A. C. (1995). Interactive Spatial Data Analysis. Prentice Hall.

Baker, L. and Ellison, D. (2008). Optimisation of pedotransfer functions using an artificial neural network ensemble method. *Geoderma* 144: 212-224.

Beard, K., Deese, H., and Pettigrew, N. R. (2008). A framework for visualization and exploration of events. *Information Visualization*, 7(2), 133-151.

Bellman, R. E. (1957). Dynamic Programming. Princeton, NJ, Princeton University Press.

Bennett, R. J. (1979). Spatial Time Series: Analysis-Forecasting-Control. London, Pion.

Bischof, H., Leonardis, A., and Selb, A. (1999). MDL principle for robust vector quantisation. *Pattern Analysis & Applications*, 2(1), 59-72.

Blum, A. L. and Rivest, R. L. (1992). "Training a 3-node neural network is NP-complete." *Neural Networks* 5(1): 117-127.

Bollivier, G., Dubois, G., Maignan, M., and Kanevsky, M. (1997). "Multilayer perceptron with local constraint as an emerging method in spatial data analysis." *Nuclear Instruments and Methods in Physics Research A* 389: 226-229.

Borgman, C. L., Wallis, J. C., and Enyedy, N. (2007). Little science confronts the data deluge: habitat ecology, embedded sensor networks, and digital libraries. *International Journal on Digital Libraries*, 7(1-2), 17-30.

Boulesteix, A.-L., Janitza, S., Kruppa, J. and König, I. R. (2012), Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6), 493-507.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123-140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). Classification and regression trees. CRC press

Brunsdon, C., Fotheringham, S., Charlton, M. (1998). Geographically weighted regression. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(3), 431-443.

Bryan, B. A. and Adams J. M. (2002). Three-dimensional Neurointerpolation of Annual Mean Precipitation and Temperature Surfaces for China. *Geographical Analysis* 34(2): 93-111.

Camara, G., Egenhofer, M. J., Ferreira, K., Andrade, P., Queiroz, G., Sanchez, A., and Vinhas, L. (2014). Fields as a generic data type for big spatial data. In *Geographic Information Science* (pp. 159-172). Springer International Publishing.

Camara, G., Freitas, U., Souza, R., Casanova, M., Hemerly, A., and Medeiros, C. B. (1994, December). A model to cultivate objects and manipulate fields. In *Proceedings 2nd ACM Workshop on Advances in GIS* (pp. 20-28).

Cannon, A. J. and Whitfield, P. H. (2002). Downscaling recent streamflow conditions in British Columbia, Canada using ensemble neural network models. *Journal of Hydrology* 259: 136-151.

Castro, J. L., Mantas, C. J., and Benıtez, J. M. (2000). "Neural networks with a continuous squashing function in the output are universal approximators." *Neural Networks* 13(6): 561-563.

Chen, W. P., Hou, J. C., and Sha, L. (2004). Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 3(3), 258-271.

Christakos, G. (2013). *Random Field Models in Earth Sciences*. Elsevier

Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C. (2009) Introduction to Algorithms (3rd ed). MIT Press

Couclelis, H. (1992). People manipulate objects (but cultivate fields): Beyond the raster-vector debate in GIS. *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg. Volume 639/1992: 65-77.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems* 2: 303-314.

Dawson, C. W. and Wilby, R. L. (2001). Hydrological modelling using artificial neural networks. *Progress in Physical Geography* 25: 80-108.

De La Piedra, A., Benitez-Capistros, F., Dominguez, F., and Touhafi, A. (2013, July). Wireless sensor networks for environmental research: A survey on limitations and challenges. In *EUROCON, 2013 IEEE* (pp. 267-274). IEEE

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1-38.

Dimitrakopoulos, R. and Luo, X. (1996). Joint space-time kriging in the presence of trends. *Geostatistics Wollongong 96*. E. Y. Baafi and N. A. Schofield. Wollongong, Australia, Springer. 1: 138-149.

Dube, M. P., and Egenhofer, M. J. (2014, November). Partitions to improve spatial reasoning. In *Proceedings of the 1st ACM SIGSPATIAL PhD Workshop*. ACM.

Dubé, J., and Legros, D. (2013a). A spatio−temporal measure of spatial dependence: An example using real estate data. *Papers in Regional Science*, 92(1), 19-30.

Dubé, J., and Legros, D. (2013b). Dealing with spatial data pooled over time in statistical models. *Letters in Spatial and Resource Sciences*, 6(1), 1-18.

Duckham, M., Nittel, S. and Worboys M. (2005). Monitoring dynamic spatial fields using responsive geosensor networks. *Proceedings of the 13th annual ACM International Workshop on Geographic Information Systems*, pp. 51-60, ACM.

Egenhofer, M. J., and Al-Taha, K. K. (1992). Reasoning about gradual changes of topological relationships. In *Theories and Methods of Spatio-temporal Reasoning in Geographic Space* (pp. 196-219). Springer Berlin Heidelberg.

Estrin, D., Culler, D., Pister, K., and Sukhatne, G. (2002). Connecting the physical world with pervasive networks. *IEEE Pervasive Computing* 1(1): 59-69.

Eynon, B. P. and Switzer, P. (1983). The variability of rainfall acidity. *The Canadian Journal of Statistics* 11(1): 11-24.

Ferentinos, K. P. (2005). "Biological engineering applications of feedforward neural networks designed and parameterized by genetic algorithms." *Neural Networks* 18: 934-950.

Forero, P. A., Cano, A., and Giannakis, G. B. (2008, March). Consensus-based distributed expectation-maximization algorithm for density estimation and classification using wireless sensor networks. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on* (pp. 1989-1992). IEEE.

Forero, P. A., Cano, A., and Giannakis, G. B. (2008, May). Consensus-based k-means algorithm for distributed learning using wireless sensor networks. In *Proceedings of the Workshop on Sensors, Signal and Information Processing*, Sedona, AZ (pp. 11-14).

Fuentes, M., Chen, L., Davis, J. M., and Lackmann, G. (2003). A new class of nonseparable and nonstationary covariance models for wind fields. *North Carolina State University at Raleigh Department of Statistics*.

Fuentes, M., Chen, L., Davis, J. M., and Lackmann, G. M. (2005) Modeling and predicting complex space–time structures and patterns of coastal wind fields. *Environmetrics* 16.5: 449-464.

Funahashi, K. (1989). "On the approximate realisation of continuous mappings by neural networks." *Neural Networks* 2: 183-192.

Gao, J. and P. Revesz (2006). Voting prediction using new spatiotemporal interpolation methods. P*roceedings of the 2006 International Conference on Digital Government Research*, San Diego, CA, ACM.

Genuer, R., Poggi, J. M., and Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, 31(14), 2225-2236.

Gething, P. W., Atkinson, P. M., Noor, A. M., Gikandi, P. W., Hay, S. I., and Nixon, M. S. (2007). A local space–time kriging approach applied to a national outpatient malaria data set. *Computers and Geosciences* 33(10): 1337-1350.

Goodchild, M. F. (1992). Geographical Data Modeling. *Computers and Geosciences* 18(4): 401-408.

Goovaerts, P. (1997). *Geostatistics for natural resources evaluation*. Oxford university press

Gu, D. (2008). Distributed EM algorithm for Gaussian mixtures in sensor networks. *IEEE Transactions on Neural Networks* 19(7), 1154-1166.

Guan, Q., Kyriakidis, P. C., and Goodchild, M. F. (2011). A parallel computing approach to fast geostatistical areal interpolation. *International Journal of Geographical Information Science*, 25(8), 1241-1267.

Haas, T. C. (1995). Local prediction of a spatio-temporal process with an application to wet sulfate deposition. *Journal of the American Statistical Association*, *90*(432), 1189-1199.

Hamerly, G., and Elkan, C. (2004). Learning the k in K-means. *Advances in neural information processing systems*, 16, 281.

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10): 993-1001.

Haykin, S (2008). *Neural networks and learning machines (3rd ed.)*. Prentice Hall

Hofer, B., and Frank, A. U. (2009). Composing models of geographic physical processes. In *Spatial Information Theory* (pp. 421-435). Springer Berlin Heidelberg.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks* 2(5): 359-366.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31(3), 264-323.

Jin, G., and Nittel, S. (2008). Towards spatial window queries over continuous phenomena in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(4), 559-571

Jin, G. (2009). Towards Spatial Queries over Phenomena in Sensor Networks. Spatial Information Science and Engineering. Orono, University of Maine. Ph.D. thesis: 202.

Johnson, M., Healy, M., van de Ven, P., Hayes, M. J., Nelson, J., Newe, T., and Lewis, E. (2009, October). A comparative review of wireless sensor network mote technologies. In *Sensors*, 2009 IEEE (pp. 1439-1442). IEEE

Journel, A. G. (1988). Fundamentals of geostatistics in five lessons. Stanford Center for Reservoir Forecasting, Applied Earth Sciences Department

Journel, A. G. and Huijbregts, C. J. (1978). Mining Geostatistics. London, Academic Press.

Journel, A. G. and M. E. Rossi (1989). When do we need a trend model in kriging? *Mathematical Geology* 21(7): 715-739.

Karimi, F., Dastgir, M., and Shariati, M. (2014). Index Prediction in Tehran Stock Exchange Using Hybrid Model of Artificial Neural Networks and Genetic Algorithms. *International Journal of Academic Research in Accounting, Finance and Management Sciences*, 4(1), 352-357.

Knotters, M., Brus, D. J., and Voshaar, J. O. (1995). A comparison of kriging, co-kriging and kriging combined with regression for spatial interpolation of horizon depth with censored observations. *Geoderma* 67: 227-246.

Kowalczyk, W. and Vlassis, N. A. (2004). Newscast em. In *Advances in Neural Information Processing Systems* (pp. 713-720).

Kravchenko, A. N. (2003). Influence of spatial structure on accuracy of interpolation methods. *Soil Science Society of America Journal*, 67(5), 1564-1571

Ku, L. F., Greenberg, D. A., Garrett, C. J. R., and Dobson, F. W (1985). Nodal Modulation of the Lunar Semidiurnal Tide in the Bay of Fundy and Gulf of Maine. *Science* 230(4721): 69-71.

Kuhnlein, M., Applehans, T., Thies, B. and Nauss, T. (2014). Improving the accuracy of rainfall rates from optical satellite sensors with machine learning – A random forests-based approach applied to MSG SEVIRI. *Remote Sensing of Environment*, 141, 129-143

Kyriakidis, P. C. and Journel, A. G. (1999). Geostatistical Space–Time Models: A Review. *Mathematical Geology* 31(6): 651-684.

Kyriakidis, P. C. and Journel, A. G. (2001). Stochastic modeling of atmospheric pollution: a spatial time-series framework. Part I: methodology. *Atmospheric Environment* 35(13): 2331-2337.

Lam, N. S. N. and Quattrochi, D. A. (1992). On the Issues of Scale, Resolution, and Fractal Analysis in the Mapping Sciences. *The Professional Geographer* 44(1): 88-98.

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

Li, L. and Revesz, P. (2004). Interpolation methods for spatio-temporal geographic data. *Computers, Environment and Urban Systems* 28(3): 201-227.

Londhe, S. N. and Panchang, V. (2007). Correlation of wave data from buoy networks. *Estuarine, Coastal and Shelf Science* 74: 481-492.

Lophaven, S. N., Nielsen, H. B., and Sondergaard, J. (2002). DACE: A MATLAB Kriging Toolbox, Technical University of Denmark.

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

Masters, T. (1995). Neural, novel and hybrid algorithms for time series prediction, John Wiley & Sons.

Matheron, G. (1963). Principles of Geostatistics. *Economic Geology* 58: 1246-1266.

Matheron, G. (1973). The intrinsic random functions and their applications. *Advances in Applied Probability*, 439-468

Murtagh, Fionn. (1985) A survey of algorithms for contiguity-constrained clustering and related problems. *The Computer Journal 28*(1), 82-88.

Nittel, S. (2009). A survey of geosensor networks: Advances in dynamic environmental monitoring. *Sensors*, 9(7), 5664-5678.

Nittel, S., Trigoni, N., Ferentinos, K., Neville, F., Nural, A., and Pettigrew, N. (2007, June). A drift-tolerant model for data management in ocean sensor networks. In *Proceedings of the 6th ACM International Workshop on Data Engineering for Wireless and Mobile Access* (pp. 49-58).

Nittel, S., and Leung, K. T. (2004, June). Parallelizing clustering of geoscientific data sets using data streams. In *16th International Conference on Scientific and Statistical Database Management, 2004. Proceedings.* (pp. 73-84). IEEE.

Nittel, S., Whittier, J. C., and Liang, Q. (2012, November). Real-time spatial interpolation of continuous phenomena using mobile sensor data streams. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* (pp. 530-533). ACM.

Nowak, R. D. (2003). Distributed EM algorithms for density estimation and clustering in sensor networks. *IEEE Transactions on Signal Processing*, 51(8), 2245-2253.

Olden, J. D. and Jackson, D. A. (2002). Illuminating the ‘‘black box’’: a randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modeling* 154: 135-150.

Oliva, G. and Setola, R. (2014) Distributed k-means algorithm. arXiv preprint arXiv:1312.4176 (submitted to *IEEE Transactions on Mobile Computing*).

Openshaw, S. (1977). A geographical solution to scale and aggregation problems in region-building, partitioning and spatial modelling. *Transactions of the Institute of British Geographers*, 459-472.

Openshaw, S. and A. Turner (2001). Disaggregative Spatial Interpolation. *GISRUK 9th Annual Conference*. Wales, U.K.

Openshaw, S. and C. Openshaw (1997). Artificial Intelligence in Geography, John Wiley & Sons.

Pariente, D., Servigne, S. and Laurini, R. (1994). A neural method for geographical continuous field estimation. *Neural Processing Letters* 1(2): 28-31.

Peeples, Matthew A. (2011) R Script for K-Means Cluster Analysis. [online]. Available: http://www.mattpeeples.net/kmeans.html. (June 17, 2014)

Pelleg, D. and Moore, A. W. (2000, June). X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *ICML* (pp. 727-734).

Pettigrew, N. R., Roesler, C.S., Neville, F. and Deese, H. (2008). An Operational Real-Time Ocean Sensor Network in the Gulf of Maine. *Lecture Notes in Computer Science*. S. Nittel, A. Labrinidis and A. Stefanidis, Springer Berlin / Heidelberg. 4540/2008: 213-238.

Pettigrew, N. R., Churchill, J. H., Janzen, C. D., Mangum, L. J., Signell, R. P., Thomas, A. C., and Xue, H. (2005). The kinematic and hydrographic structure of the Gulf of Maine Coastal Current. *Deep Sea Research Part II: Topical Studies in Oceanography*, 52(19), 2369-2391.

Peuquet, D., Smith, B. and Brogaard, B. (1999). The Ontology of Fields. *Report of a Specialist Meeting Held under the Auspices of the Varenius Project*. Santa Barbara, CA, National Center for Geographic Information and Analysis.

Physical Oceanography Group, University of Maine (2014) GoMOOS Total Vector Plots in the Gulf of Maine. [online]. Available: http://gyre.umeoce.maine.edu/data/gomoos/codar/totals_files/plots/2014/03/06/total_NCOS_2014_03_06_1700.png. (March 10, 2015)

Reed, R. D., and Marks, R. J. (1998). Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks. MIT Press.

Rigol, J. P. (2005). Spatial interpolation of natural radiation levels with prior information using back-propagation artificial neural networks. *Applied GIS* 1(2): 1801-1815.

Rigol, J. P., Jarvis, C.H. and Stuart, N. (2001). Artificial neural networks as a tool for spatial interpolation. *International Journal of Geographical Information Science* 15(4): 323-343.

Robinson, W. S. (1950). Ecological Correlations and the Behavior of Individuals. *American Sociological Review* 15(3): 351-357.

Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organisation in the Brain. *Psychological Review* 65: 386-407.

Rouhani, S., and Hall, T. J. (1989). Space-time kriging of groundwater data. In *Geostatistics* (pp. 639-650). Springer Netherlands.

Rumelhart, D. E., Hinton, G. E. and Williams R. J, (1986). Learning representations by back-propagating errors *Nature* 323: 533-536.

Samuelson, P. A. and Nordhaus, W. D. (2001). Microeconomics (17th ed.). McGraw-Hill. p. 110. ISBN 0071180664.

Shekhar, S., Zhang, P., Huang, Y., and Vatsavai, R. R. (2003). Trends in spatial data mining. *Data mining: Next generation challenges and future directions*, 357-380.

Skøien, J. O. and Bloschl G. (2007). Spatiotemporal topological kriging of runoff time series. *Water Resources Research* 43(9).

Snell, S. E., S. Gopal, and Kaufmann, R. K. (2000). "Spatial Interpolation of Surface Air Temperatures Using Artificial Neural Networks: Evaluating Their Use for Downscaling GCMs." *Journal of Climate* 13(5): 886-895.

Stemick, M., Boah, A. and Rohling H. (2008). Over-the-Air Programming of Wireless Sensor Nodes. *Drahtlose Sensornetze am 25. und 26. September 2008 in Berlin*, 11, Freie Universität Berlin.

Stevens, S. S. (1946). On the Theory of Scales of Measurement. *Science* 103(2684): 677-680.

Tan, P., Steinbach, M., and Kumar, V. (2006). Introduction to Data Mining. Addison-Wesley ISBN 0321321367

Tissot, P., P. R. Michaud, and Cox D. T. (2003). Optimization and Performance of a Neural Network Model Forecasting Water Levels for the Corpus Christi, Texas, Estuary. *3rd Conference on Artificial Intelligence Applications to the Environmental Science*. AMS. Long Beach, CA.

Townsend, D. W., Pettigrew, N. R. and Thomas A. C. (2001). Offshore blooms of the red tide dinoflagellate, Alexandrium sp., in the Gulf of Maine. *Continental Shelf Research* 21(4): 347-369.

UNDG (2009). The Tsunami Legacy: Innovation, Breakthroughs and Change - UN Development Group Report, United Nations Development Group: p. 74.

Virili, F., and Freisleben, B. (2000). Nonstationarity and data preprocessing for neural network predictions of an economic time series. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on* (Vol. 5, pp. 129-134). IEEE

Voronoi, G. (1908). Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites. *Journal für die reine und angewandte Mathematik*, 133, 97-178.

Watts, M. J. and Worner, S. P. (2008). Comparing ensemble and cascaded neural networks that combine biotic and abiotic variables to predict insect species distribution. *Ecological Informatics* 3(6): 354-366.

Webster, R. and Oliver, M. (1993). How large a sample is needed to estimate the regional variogram adequately. *Geostatistics* Troia '92. A. Soares, Kluwer, Dordrecht: 155-166.

Weiser, P., and Frank, A. U. (2012). Modeling discrete processes over multiple levels of detail using partial function application. *Proceedings of GI Zeitgeist*, Muenster, Germany.

Werbos, P. J. (1994). The Roots of Backpropagation, Wiley-InterScience.

Whittier, J. C., Nittel, S., Plummer, M. A., and Liang, Q. (2013, November). Towards window stream queries over continuous phenomena. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on GeoStreaming* (pp. 2-11). ACM.

Witten, I. H., Frank, E. and Hall, M. (2011). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

Wolfe, J., Haghighi, A., and Klein, D. (2008, July). Fully distributed EM for very large datasets. In *Proceedings of the 25th international conference on Machine learning* (pp. 1184-1191). ACM.

Wolpert, D. H., and Macready, W. G. (1997). No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1), 67-82.

Worboys, M. and Duckham, M. (2004). GIS: a Computing Perspective, CRC Press LLC.

Worboys, M., and Duckham, M. (2006). Monitoring qualitative spatiotemporal change for geosensor networks. *International Journal of Geographical Information Science*, 20(10), 1087-1108.

Xu, R., and Wunsch, D. Survey of Clustering Algorithms. *Neural Networks, IEEE Transactions on* 16.3 (2005): 645-678

Younis, O., and Fahmy, S. (2004). HEED: a Hybrid, Energy-Efficient, Distributed Clustering Approach for ad hoc Sensor Networks. *IEEE Transactions on Mobile Computing*, 3(4), 366-379.

Zealand, C. M., Burn, D. H. and Simonovic, S. (1999). Short term streamflow forecasting using artificial neural networks. *Journal of Hydrology* 214: 32-48.

Zhong, X., Kealy, A., Sharon, G., and Duckham, M. (2015). Spatial Interpolation of Streaming Geosensor Network Data in the RISER System. In *Web and Wireless Geographical Information Systems* (pp. 161-177). Springer International Publishing.

Zubin, D. (1989). Oral presentation, NCGIA Initiative 2 specialist meeting. *Languages of Spatial Relations: Researchable Questions & NCGIA Research Agenda.* D. Mark. Santa Barbara, CA, NCGIA.

APPENDIX A: MONTE CARLO BESTK MATLAB CODE

(as adapted from (Peeples 2011))

```
clear;

conn = database('dbName', '', '');
setdbprefs('DataReturnFormat','numeric')
dbmeta = dmd(conn);

BEGIT = 350;
SERLEN = 200;
    stopnum = 7;
    AVGCNT = 10;
    KLUSTMAX = 10;
kser = zeros(BEGIT+SERLEN,1);
for it=BEGIT:(BEGIT+SERLEN)
    sql = ['SELECT b.x, b.y, b.u, b.v, a.u, a.v ' ...
      'FROM orthDat a INNER JOIN orthDat b ON a.x=b.x and a.y=b.y AND
a.impno=(b.impno-1)' ...
      'WHERE b.impno=' num2str(it)];
    dd = fetch(conn, sql);

% Determine delta-change in angles in dd
    d2 = (atan2(dd(:,4), dd(:,3)) - atan2(dd(:,6), dd(:,5))) * 180/pi;

    numA = [dd d2];
    dvec = zeros(AVGCNT,KLUSTMAX);

    d = numA(:, 1:end);

    %% Scale data to Z-scores
    blnScale = true;
    if blnScale
        mn = mean(d,1);
        for j=1:size(d,2)
          d(:,j) = (d(:,j) - mn(j)) ./ std(d(:,j));
        end; % for j
    end; % if blnScale

    %% Generate ActualData series (dvec)
    for i = 1:AVGCNT
        for KCLUST = 1:KLUSTMAX
          [cidx,cmeans3,sumd3] = kmeans(d,KCLUST,
'dist','sqEuclidean','replicates',5);

          dvec(i, KCLUST) = sum(sumd3);
        end  % for KCLUST
```

```
    end  % for i
    avgvec = mean(dvec);

    %% Use averaged data-vec
    vec = avgvec;

    %% Generate MCITS series of permuted-column data sets
    MCITS = 5;  % 50
    pvec = zeros(MCITS,KLUSTMAX);
    dperm = zeros(size(d));
for i = 1:MCITS
 for j = 1:size(d,2)           % For each col...
   ix = randperm(size(d,1));   % randomly permute the order of its
contents
   dperm(:,j) = d(ix, j);      %  so each col still has same: mean,
stdev
 end;  % for j

      for KCLUST = 1:KLUSTMAX
        blndone = false;
        REPS = 2;
        while (not (blndone))
            try
              [cidx,cmeans3,sumd3] = kmeans(dperm,KCLUST,
'dist','sqEuclidean','replicates',REPS);
              blndone = true;
            catch excptn
              REPS = REPS + 10;
              blndone = false;
            end;  % try
        end; % while
        pvec(i, KCLUST) = sum(sumd3);
      end  % for KCLUST
    end  % for i

    blnFigs = false;
    if it == stopnum
        it=it;
        blnFigs = true;
    end;

    %% Determine "best" K by normalizing values, seeking a slope of -1
    bestK = 0;
    minx = 1; maxx = KLUSTMAX;
    miny = min(vec);  maxy = max(vec);
    nx = ([1:KLUSTMAX] - minx)/(maxx-minx);
    ny = (vec - miny)/(maxy-miny);
    m = zeros(size(ny));
    for i = 2:KLUSTMAX-1
      m(i) = (ny(i-1) - ny(i+1)) / (nx(i-1) - nx(i+1));
      if (m(i)>=-1  &&  m(i-1)<-1)
          bestK = i;
          break;
      end;
    end;
```

```
    kser(it) = bestK;

end;  % for it

close(conn);

fig3 = figure(3);
plot(BEGIT:(BEGIT+SERLEN), kser);
```

APPENDIX B: SYNTHETIC DATA GENERATIVE CODE

```
  Set wb = ThisWorkbook
  Set ws1 = wb.Worksheets("NatData")      ' more "naturalistic" data

  myfil = FreeFile
  Open "natdata.csv" For Output As myfil

  Const BEGIT = 400

  For da = 1 To 21
      For t = 0 To 72

    '' READ what's on the worksheet
    i = i + 1
    For ro = -10 To 10
      myrow = -ro
      For col = -10 To 10
        c = ws1.Range("O13").Cells(myrow, col)
        m(ro, col) = c
        Select Case c
          Case 1:
            v = Vec1(i, col, ro)
            cat = 1
          Case 2:
            v = Vec2(i, col, ro)
            cat = 2
          Case 3:
            v = Vec3(i, col, ro)
            cat = 3
          Case 4:
            v = Vec4(i, col, ro)
            cat = 4
        End Select

        Print #myfil, CStr(i) & "," & col & "," & myrow & "," &
CStr(v.u) & "," & CStr(-v.v) & "," & CStr(cat)
      Next 'col
    Next 'ro

    '' CHANGE what's on the worksheet
    Dim n(4) As Long
    For ro = -10 To 10
      myrow = -ro
      For col = -10 To 10

'''         If ro = -1 And col = 5 Then Stop

        n(1) = CountNeigh(m, ro, col, 1): If n(1) = 9 Then GoTo Skip
        n(2) = CountNeigh(m, ro, col, 2): If n(2) = 9 Then GoTo Skip
        n(3) = CountNeigh(m, ro, col, 3): If n(3) = 9 Then GoTo Skip
```

```
        n(4) = CountNeigh(m, ro, col, 4): If n(4) = 9 Then GoTo Skip

      Dim git As Long
      If i >= BEGIT And i < BEGIT + 150 Then
        centerX = Round(10 + ((i - BEGIT) * -0.01)): centerY = Round(10
+ ((i - BEGIT) * -0.2))
      ElseIf i < BEGIT Then
        centerX = 10: centerY = 10
      Else
        centerX = 30: centerY = -30
      End If


      If Sqr((myrow - centerY) ^ 2 + (col - centerX) ^ 2) < 10 Then  '
(center: (10, 10) )
        Range("O13").Cells(ro, col).Value = 2
      ElseIf Sqr((myrow + 11) ^ 2 + (col + 11) ^ 2) < 10 Then   '
(center: (-11, -11) )
        Range("O13").Cells(ro, col).Value = 1
      ElseIf col - myrow + 10 > 0 Then
        Range("O13").Cells(ro, col).Value = 3
      Else
        Range("O13").Cells(ro, col).Value = 4
      End If

Skip:
      Next 'col
    Next 'ro


      Next 't
  Next 'da

  Close #myfil


Function Vec1(ByVal t, x, y, Optional MAG As Double = 10) As Vec
  '' Returns a Regime1-vector
    Dim delt As Double
    Dim DIR As Long, PER As Long, step As Long
    Dim ans As Vec

    rand = Fix(Rnd * 10 + 1)
    If rand = 1 Then
      t = t - 1
    ElseIf rand = 10 Then
      t = t + 1
    End If
    noiseu = Rnd * 5 - 2.5   ' noise signal from -2.5 ... +2.5
    noisev = Rnd * 5 - 2.5   ' noise signal from -2.5 ... +2.5

    startat = 0
    PER = 72
    DIR = -1

    step = t Mod PER
```

```
  delt = 360 / PER
  rad = (startat + (step * DIR) * delt) * WorksheetFunction.Pi / 180

  ans.u = Cos(rad) * MAG + noiseu '' values perturbed by +- 5/2
  ans.v = Sin(rad) * MAG + noisev
  Vec1 = ans
End Function
```

BIOGRAPHY OF THE AUTHOR

François Neville was born in North Platte, Nebraska in 1969. He graduated from Pius X High School (Lincoln, NE) in 1988, from Nebraska Wesleyan University in 1992 with a B.S. in Computer Science, and from the University of Nebraska-Lincoln with a M.S. in Computer Science in 1995. He has taught college-level Math and Computer Science courses in the Minnesota State Colleges and Universities (MnSCU) system since 2001. He is a current member of the Association of Computing Machinery.

François is a candidate for the Doctor of Philosophy degree in Spatial Information Science and Engineering from the University of Maine in August 2015.