

© Copyright by David J. Hill, 2007

DATA MINING APPROACHES TO COMPLEX ENVIRONMENTAL PROBLEMS

BY

DAVID J. HILL

B.S., Cornell University, 1999

M.S., University of Illinois at Urbana-Champaign, 2002

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Environmental Engineering in Civil Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2007

Urbana, Illinois

CERTIFICATE OF COMMITTEE APPROVAL

University of Illinois at Urbana-Champaign
Graduate College

July 23, 2007

We hereby recommend that the thesis by:

DAVID J. HILL

Entitled:

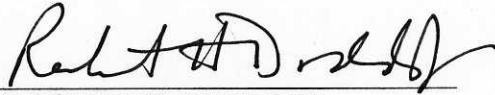
**DATA MINING APPROACHES TO COMPLEX ENVIRONMENTAL
PROBLEMS**

Be accepted in partial fulfillment of the requirements for the degree of:

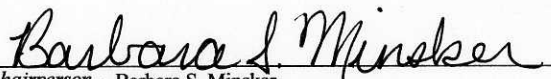
Doctor of Philosophy

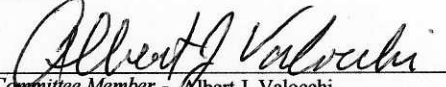
Signatures:



Director of Research - Barbara S. Minsker

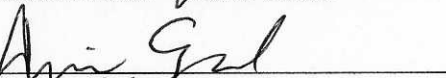

Head of Department - Robert H. Dodds

Committee on Final Examination*


Chairperson - Barbara S. Minsker


Committee Member - Albert J. Valocchi


Committee Member - Praveen Kumar


Committee Member - Eyal Amir

Committee Member - _____

Committee Member - _____

* Required for doctoral degree but not for master's degree

Abstract

Understanding and predicting the behavior of large-scale environmental systems is necessary for addressing many challenging problems of environmental interest. Unfortunately, the challenge of scaling predictive models, as well as the difficulty of parameterizing these models, makes it difficult to apply them to large-scale systems. This research addresses these issues through the use of data mining. Specifically, this dissertation addresses two problems: upscaling models of solute transport in porous media and detecting anomalies in streaming environmental data.

Upscaling refers to the creation of models that do not need to explicitly resolve all scales of system heterogeneity. Upscaled models require significantly fewer computational resources than do models that resolve small-scale heterogeneity. This research develops an upscaling method based on genetic programming (GP), which facilitates both the GP search and the implementation of the resulting models, and demonstrates its use and efficacy through a case study.

Anomaly detection is the task of identifying data that deviate from historical patterns. It has many practical applications, such as data quality assurance and control (QA/QC), focused data collection, and event detection. The second portion of this dissertation develops a suite of data-driven anomaly detection methods, based on autoregressive data-driven models (e.g. artificial neural networks) and dynamic Bayesian network (DBN) models of the sensor data stream. All of the developed methods perform fast, incremental evaluation of data as it becomes available; scale to large quantities of data;

and require no *a priori* information, regarding process variables or types of anomalies that may be encountered. Furthermore, the methods can be easily deployed on large heterogeneous sensor networks. The anomaly detection methods are then applied to a sensor network located in Corpus Christi Bay, Texas, and their abilities to identify both real and synthetic anomalies in meteorological data are compared. Results of these case studies indicate that DBN-based detectors, using either robust Kalman filtering or Rao-Blackwellized particle filtering, are most suitable for the Corpus Christi meteorological data.

To Kellee

Acknowledgements

I would like to express my gratitude to the many people who have contributed towards the completion of this research and dissertation. First, I would like to thank my advisor, Dr. Barbara Minsker, for her mentorship during my graduate studies at the University of Illinois. I especially appreciate her willingness to approach challenging problems with innovative solutions and encourage me to explore my own directions during this research. These qualities have allowed me to develop multidisciplinary research interests. I would also like to thank Dr. Albert Valocchi, for his continued guidance and support, as well as the technical expertise that he has generously offered me throughout my graduate studies. I would also like to thank the other members of my dissertation committee, Dr. Praveen Kumar and Dr. Eyal Amir, whose comments and suggestions have improved this work.

I would like to acknowledge the many agencies that provided funds to support this work: the National Science Foundation under grant BES 97-34076 CAR, the University of Illinois Research Board, the University of Illinois International Programs in Engineering Office through an International Graduate Travel Fellowship, and the Office of Naval Research under grant N00014-04-1-0437.

I would like to thank Vladan Babovic and Maarten Keijzer for teaching me all things GP; Delft Hydraulics, which hosted me in the Netherlands during part of this study; and Desiree Trujillo, John Perez, Terry Riggs, Cheryl Page, and Jim Bonner of Shoreline Environmental Research Facility, Corpus Christi, Texas, for providing data and knowledge regarding their sensors.

A special thanks goes to my parents for encouraging me to study science and engineering as a child, despite my difficulties with mathematics. It was through their love and guidance that I learned about a fascinating discipline: Environmental Engineering.

I would like to thank Amanda, Francina, David, Charis, Sara's Winos, and all my friends who celebrated with me and consoled me throughout this process. They have truly made Champaign-Urbana not just a place where I lived, but my home. Long hours at the office would have been unbearable if not for my officemate and friend Gayathri and the rest of the EMSA research group: Thanks!

Finally, none of this would have been possible if not for the love and support of my dearest Kellee. Everyday, she continues to help me better understand myself, the world and God.

Table of Contents

List of Figures.....	x
List of Tables	xiii
Chapter 1 Introduction.....	1
1.1 Objectives and Scope.....	4
1.2 Summary of Research Approach	5
1.2.1 Chapter 2: Data Mining Methods	6
1.2.2 Chapter 3: Literature Review.....	6
1.2.3 Chapter 4: Upscaling Models of Solute Transport in Porous Media through Genetic Programming.....	7
1.2.4 Chapter 5: Real-Time Autoregressive Data-Driven Anomaly Detection in Streaming Environmental Data.....	7
1.2.5 Chapter 6: Real-Time Bayesian Anomaly Detection in Streaming Environmental Data	8
1.2.6 Chapter 7: Concluding Remarks	9
Chapter 2 Data Mining Methods.....	10
2.1 Genetic Programming	12
2.2 Clustering.....	16
2.3 Perceptron and Artificial Neural Networks	18
2.4 Nearest Neighbor	23
2.5 Dynamic Bayesian Networks.....	25
2.5.1 Kalman Filtering.....	29
2.5.2 Robust Kalman Filtering.....	36
2.5.3 Particle Filtering	39
Chapter 3 Literature Review	42
3.1 Upscaling Models of Solute Transport in Porous Media.....	42
3.2 Anomaly Detection in Streaming Environmental Sensor Data	44
Chapter 4 Upscaling Models of Solute Transport in Porous Media Through Genetic Programming	52
4.1 Methods.....	52
4.1.1 Genetic Programming	52
4.1.2 Mathematical Formulation.....	53
4.2 Case Study	55
4.2.1 Method of Moments.....	56
4.2.2 Synthetic Aquifers	57
4.2.3 Simplifying the Numerical Formulation.....	59
4.2.4 Generation of Training Data	61
4.2.5 Parameterization of ALP.....	62
4.3 Results.....	65

4.4 Discussion	79
4.5 Conclusion	80
Chapter 5 Real-Time Autoregressive Data-Driven Anomaly Detection in Streaming Environmental Data	83
5.1 Methods.....	84
5.2 Case Study	90
5.3 Discussion.....	101
5.4 Conclusion	105
Chapter 6 Real-Time Bayesian Anomaly Detection in Streaming Environmental Data	108
6.1 Methods.....	109
6.2 Case Study	117
6.2.1 Detector Parameterization.....	118
6.2.2 Detection of Synthetic Anomalies	123
6.2.3 Detection of Observed Anomalies.....	129
6.3 Discussion.....	143
6.4 Conclusion	151
Chapter 7 Concluding Remarks.....	154
References.....	164
Author's Biography	181

List of Figures

2.1	Schematic of Dynamic Bayesian Network	26
4.1	Schematic of a two-dimensional perfectly stratified aquifer indicating both the fine and block-scale computational grids as well as the fine-scale velocity distribution	59
4.2	Comparison of the MoM and SGA upscaled models for predicting the vertically averaged time evolution of a pulse input in the synthetic aquifer with parabolic flow	71
4.3	Comparison of the maximum absolute errors between ALP derived upscaled models and the MoM upscaled model	72
4.4	Comparison of the MoM and H1 upscaled models for predicting the vertically averaged time evolution of a pulse input in the synthetic aquifer with parabolic flow	76
4.5	Comparison of the MoM and H1s upscaled models in predicting the vertically averaged time evolution of an instantaneous finite width input of solute in the synthetic aquifer with parabolic flow	77
4.6	Comparison of the MoM and H2 upscaled models for predicting the vertically averaged time evolution of a pulse input in the synthetic aquifer with cos-cos flow	78
5.1	Autocorrelation of the SERF windspeed data calculated during the period of January–May 2004. Note that r^2 values of 0.8 and higher are considered highly correlated	91
5.2	Performance of different data-driven methods for predicting the Corpus Christi Bay windspeed data stream.....	93
5.3	Data exhibiting errors resulting from short duration faults	94
5.4	Data exhibiting errors resulting from long duration faults	95
5.5	Q-Q plot of ANN model residual distribution and standard normal distribution. Vertical lines indicate bounds of 99% PI.....	96
5.6	False positive rates for detecting June 2004 windspeed data errors using 95% and 99% PIs	99

5.7	False negative rates for detecting June 2004 windspeed data errors using 95% and 99% PIs. The bars for the cluster-based methods have been truncated. These methods have false negative rates greater than 89%.....	101
6.1	Schematic of DBN used in BCI-based anomaly detection. Vector X represents the continuous-valued system variables and vector Z represents the continuous-valued observations. Subscripts indicate time	111
6.2	Q-Q plot of ANN model residual distribution and mixture of two Gaussian components with means of 0 and 0, variances of 1 and 12, and mixture ratios of 0.75 and 0.25, respectively.....	113
6.3	Schematic of DBN used in MAP-ms anomaly detection. Vector X represents the continuous-valued system variables, vector Z represents the continuous-valued observations, and scalar MS indicates the discrete measurement status. Subscripts indicate time	116
6.4	Location of SERF CC003 and CC009 sensor platforms within Corpus Christi Bay	119
6.5	False positive rates for the BCI-kf, BCI-rkf, MAP-ms, and AR_ADET detectors for classifying transient synthetic anomalies.....	125
6.6	False negative rates for the BCI-kf, BCI-rkf, MAP-ms, and AR_ADET detectors for classifying transient synthetic anomalies.....	126
6.7	Comparison of uncoupled and coupled MAP-ms anomaly detection methods for classifying transient synthetic anomalies in the windspeed data	129
6.8	Time required by AR_ADET, BCI-kf, BCI-rkf, and MAP-ms methods to evaluate a new measurement vector. Times are averaged over five replicates ..	130
6.9	Scale up of MAP-ms method, with respect to the number of particles used in the Rao-Blackwellized particle filter. The error bars indicate one standard deviation, and the dashed line indicates the linear least squares regression of the points ($r^2=1$)	131
6.10	December 15-16, 2007 sensor measurements from platform CC003.....	133
6.11	December 15-16, 2007 sensor measurements from platform CC009. These data show the effects of a barometer failure at approximately 02:00	134
6.12	Classification of the December 15-16, 2007 sensor measurements from platform CC003 by the MAP-ms-50k detector.....	135

6.13	Classification of the December 15-16, 2007 sensor measurements from platform CC009 by the MAP-ms-50k detector.....	136
6.14	December 1, 2007 sensor measurements from platform CC003	139
6.13	December 1, 2007 sensor measurements from platform CC009	140
6.14	Classification of the December 1, 2007 sensor measurements from platform CC003 by the MAP-ms-50k detector	141
6.15	Classification of the December 15-16, 2007 sensor measurements from platform CC009 by the MAP-ms-50k detector.....	142

List of Tables

5.1	Values for data-driven time-series models	92
-----	---	----

Chapter 1: Introduction

Understanding and predicting the behavior of large-scale environmental systems is necessary for addressing many challenging problems of environmental interest, such as (1) the design of groundwater remediation strategies, (2) the development of early warning systems for natural disasters, like hurricanes or tsunamis, and (3) the understanding of conditions that cause natural events of concern, like hypoxia in the Gulf of Mexico.

One popular method of predicting the behavior of environmental systems is through the use of mathematical models of the system. Unfortunately, there are several obstacles to applying such models to large-scale systems. One such obstacle is that mathematical models often do not scale well to large systems. Models of environmental systems are composed of hierarchical sets of coupled processes for which sub-models exist. These sub-models are often in the form of ordinary or partial differential equations (ODEs or PDEs). Because of the complexity of these equations, and particularly because of the complexity caused by their boundary/initial conditions, they cannot be solved directly; instead, numerical methods must be employed. Such numerical solutions require discretization of the space-time domain and are expressed as a number of simultaneous equations proportional to the number of units in the spatial discretization. These equations must be solved as many times as there are points in the temporal discretization. The resolution of the discretization is not arbitrary, but linked to the stability bounds of the numerical approximation of the differential equation, as well as the scale of the heterogeneity of salient properties of the spatial domain. The result is that the equations

governing the behavior of a large environmental system may require an immense number of simultaneous equations to be solved repeatedly, which may render the model incapable of addressing large-scale systems, especially when a large number of predictions must be made (as in the case of a remediation design problem) or when the predictions must be made quickly (as in the case of an early warning system for natural disasters).

The most straightforward approach for addressing the computational demands of solving models of large-scale systems is to use parallel computing to distribute the computational burden over many processors, thus decreasing the time required for solutions to be reached (Gwo *et al.* 2001, Steefel 2001). Another approach involves developing models that do not need to resolve all the heterogeneity of the spatial domain. The development of such “upscaled” models continues to be a central concern of researchers and is the focus of the first research component of this dissertation.

A second obstacle to using mathematical models of large-scale environmental systems is that the values of the model parameters are often not readily available. These parameters include spatial properties, boundary and initial conditions, and process-related coefficients. Characterizing these parameters often requires detailed measurements at high spatial and/or temporal resolution, the collection of which can be prohibitively expensive and sometimes impossible, without altering the system itself. Some techniques, such as geostatistics (Goovaerts 1997) and stochastic modeling (Rubin 2003), have been developed to improve the accuracy and confidence of model predictions, but these methods would still benefit from higher resolution data.

Recent advances in sensor technology are facilitating the deployment of sensors into the environment that can produce measurements at high spatial and/or temporal resolutions (NRC 2006). Not only can the data be used to better characterize the system for improved modeling, but they can also be used to produce better understanding of the mechanisms of environmental processes. Furthermore, these data can be used to improve model predictions, through the use of data assimilation and real-time forecasting.

Both data assimilation and real-time forecasting use measured field data to improve the prediction accuracy of a model of an environmental system. Data assimilation is a method of combining field-measured data with models of system dynamics, such that model predictions are always made within the predictability limit of the system. Furthermore, data assimilation provides a framework for estimating the prediction error of the model, which can be used to refine the prediction. Data assimilation has long been used in weather forecasting and oceanographic modeling (Ghil & Malanotte-Rizzoli 1991) and has recently been applied to coastal area modeling. In particular, Madsen and Cañizares (1999) and Bertino *et al.* (2002) presented applications of Kalman-filter-based assimilation schemes to 2- and 3-dimensional hydrodynamic modeling in an estuarine system, respectively.

Real-time forecasting uses the most recent measurements of a system state as the initial and boundary conditions of a predictive model. Several case studies (e.g Elliott & Jones 2000, Guinasso *et al.* 2001) have demonstrated that predictions made with real-time

forecasting are more accurate than predictions made with more traditional estimates of initial and boundary conditions.

Unfortunately, the benefits of having increased high-resolution data to characterize large-scale environmental systems come at the cost of needing to manage, archive, clean, aggregate, and derive meaning from the large quantities of data collected by these sensors, so methods must be developed to meet this task. The second research component of this dissertation focuses on automated real-time methods for identifying anomalous measurements (i.e. measurements that deviate markedly from the historical pattern) in environmental sensor data streams. Anomalous measurements can be caused by sensor or data transmission errors or by infrequent system behaviors that are often of interest to scientific and regulatory communities; thus real-time anomaly detection can be used for data quality control and analysis, adaptive sampling, and anomalous event detection.

1.1 Objectives and Scope

As discussed previously, there are many open problems in the field of modeling large-scale systems. These problems have eluded solution by traditional techniques, because of the inability of traditional techniques to scale up to large systems and the uncertainty inherent in these systems. Data mining employs computational techniques from statistics, machine learning, pattern recognition, and other disciplines to extract knowledge from data (Han & Kamber 2006) and is well-suited for developing economical models of complex systems and deriving understanding about the fundamental processes of these systems. This research explores how emerging data mining methods can be used to

address complex environmental systems problems. Specifically, the objectives of this research are:

Objective 1: To explore the upscaling of models of solute transport using data regarding both the fine-scale and the large-scale transport processes.

Objective 2: To explore data-driven methods for identifying anomalous data (i.e. data that deviate markedly from the historical trend) within environmental data streams in real time.

Objective 3: To explore the efficacy of different data-driven anomaly detection methods for use in environmental observation networks.

1.2 Summary of Research Approach

The primary objective of this research was to explore how emerging data mining methods can be used to address complex environmental systems problems. To achieve this objective, several data mining methods are applied to two case studies. The first case study addresses the creation of upscaled models, and the second case study addresses the challenge of identifying anomalous data in environmental sensor data streams. Chapter 2 reviews the existing literature and background information on upscaling of solute transport models and anomaly detection. Chapter 3 summarizes the data mining methods that will be used in the two case studies. Chapter 4 presents a new data-driven method for upscaling models of solute transport in porous media. Chapters 5 and 6 present new anomaly detection methods that can address streaming environmental data in real time. The remainder of this chapter presents more detailed summaries of the chapters of this dissertation.

1.2.1 Chapter 2: Data Mining Methods

Chapter 2 offers a general introduction to data mining. As data mining is a broad discipline, the discussion is quickly narrowed to focus on the data mining methods that were used in this research. These methods include genetic programming, clustering, perceptron, artificial neural networks, nearest neighbor methods, and dynamic Bayesian networks. The discussion of each method covers its most common variants as well as strategies for determining method parameters. The implementation details of the specific variant of each method used in this research are also given.

1.2.2 Chapter 3: Literature Review

Chapter 3 provides a review of the approaches used to address both the upscaling of models of solute transport and the detection of data anomalies. The discussion of upscaling methods covers stochastic, spatial filtering, and statistical moment methods, and explains the motivation for exploring new upscaling methods. The discussion of anomaly detection methods begins by introducing traditional methods suitable only for detecting anomalies in historical data records. Then, methods designed to address real-time detection of anomalies in streaming data are discussed, along with their limitations. These methods include redundancy methods, Bayesian methods, rule-based approaches, and hybrid methods.

1.2.3 Chapter 4: Upscaling Models of Solute Transport in Porous Media through Genetic Programming

Chapter 4 investigates the development of upscaled solute transport models using genetic programming (GP), a domain independent modeling tool that searches the space of mathematical equations for one or more equations that describe a set of training data. An upscaling methodology is developed that facilitates both the GP search and the implementation of the resulting models. A case study is performed that demonstrates this methodology by developing vertically-averaged equations of solute transport in perfectly stratified aquifers. The solute flux models developed for the case study are analyzed for parsimony and physical meaning, resulting in an upscaled model of the enhanced spreading of the solute plume, due to aquifer heterogeneity, as a process that changes from predominantly advective to Fickian. This case study not only demonstrates the use and efficacy of GP as a tool for developing upscaled solute transport models, but it also provides insight into how to approach more realistic multi-dimensional problems with this methodology.

1.2.4 Chapter 5: Real-Time Autoregressive Data-Driven Anomaly Detection in Streaming Environmental Data

Chapter 5 develops a real-time anomaly detection method for environmental data streams that can be used to identify data that deviate from historical patterns. The method is based on an autoregressive data-driven model of the data stream and its corresponding prediction interval. It performs fast, incremental evaluation of data as they become available; scales to large quantities of data; and requires no *a priori* information

regarding process variables or types of anomalies that may be encountered. Furthermore, this method can be easily deployed on a large heterogeneous sensor network. Sixteen instantiations of this method are compared based on their ability to identify measurement errors in a windspeed data stream from Corpus Christi, Texas. The results indicate that a neural network model of the data stream, coupled with replacement of anomalous data points, performs well at identifying erroneous data in this data stream.

1.2.5 Chapter 6: Real-Time Bayesian Anomaly Detection in Streaming

Environmental Data

Chapter 6 develops two automated anomaly detection methods that employ dynamic Bayesian networks (DBNs). Dynamic Bayesian networks are Bayesian networks with network topology that evolves over time, adding new state variables to represent the system state at the current time. Filtering (e.g. Kalman filtering or Rao-Blackwellized particle filtering) can then be used to infer the expected value of unknown system states, as well as the likelihood that a particular sensor measurement is anomalous.

Measurements with a high likelihood of being anomalous are classified as such. The methods developed in this chapter perform fast, incremental evaluation of data as it becomes available; scale to large quantities of data; and require no *a priori* information regarding process variables or types of anomalies that may be encountered. Furthermore, these methods can be easily deployed on large networks of heterogeneous sensors. Unlike the method presented in Chapter 5, the methods presented in Chapter 6 consider several data streams at once, using all of the streams concurrently to perform coupled anomaly detection. This study investigates these methods' abilities to identify anomalies in eight

meteorological data streams from Corpus Christi, Texas, and compares the methods to those developed in Chapter 5. The results indicate that DBN-based detectors, using either robust Kalman filtering or Rao-Blackwellized particle filtering outperform both a DBN-based detector using Kalman filtering and the autoregressive data-driven anomaly detection method developed in Chapter 5 for identifying synthetic anomalies. These methods were also successful at identifying data anomalies caused by two real events: a sensor failure and a large storm.

1.2.6 Chapter 7: Concluding Remarks

The final chapter of this dissertation summarizes the findings of the studies contained therein and suggests ways that the developed methods could be fruitfully extended in the future, in order to better address large-scale problems of environmental interest through the promising methods of data mining.

Chapter 2: Data Mining Methods

Data mining employs computational techniques from statistics, machine learning, pattern recognition, and other disciplines to extract knowledge from data (Han & Kamber 2006). In general, data mining tasks can be divided into two categories: descriptive and predictive. Descriptive data mining tasks characterize the general properties of data and include tasks such as finding frequent patterns, associations, or correlations among the data. Predictive data mining tasks use data to infer future events. For example, historical precipitation records can be used to predict the expected rainfall during future months. As this research will make use of only predictive data mining tools, descriptive data mining tools will not be discussed in detail here. For more information about these types of tools see Han and Kamber (2006).

This chapter will be devoted to a description of the predictive data mining methods that will be used in this research. In order to facilitate the description, a simple example of predicting the flow rate of a river, based on measurements of the stage (flow depth) and linear velocity, will be employed throughout.

Predictive data mining methods learn concepts from data through a process referred to as “training,” which is performed using “examples,” or sets of features that describe the target concept (e.g. the system behavior corresponding to the examples). In our river flow example, the target concept is the relation between the features stage and the measurements of velocity and flow. Each example can be expressed as an n -dimensional vector of features; thus, each example exists as a point in an n -dimensional coordinate

system, commonly referred to as “feature space.” In our river flow example, the feature space is two-dimensional (i.e. stage and velocity). If the outcome of the concept, the “target value” (i.e. the river flow rate in our example), is known for one or more of the examples, it may also be used for training; however, there are tools that can learn concepts without knowing the target value. If the target value is known for every training example, then the training is considered to be supervised. If the target value of one or more training examples is unknown, then the training is considered to be semi-supervised, and if no target values are known, then the training is considered to be unsupervised. Training can either take place “on-line,” where modification of a learned target concept takes place after presenting an individual example to the tool, or in “batch,” where a set of examples, the “training set,” is presented at once to the tool, followed by modification of the concept.

Before data mining can be performed, the data must be prepared for mining. This preparation includes three steps: (1) data cleaning, (2) feature transformation, and (3) feature selection. Data cleaning involves removing faulty or unreliable records from the data archive. Feature transformation involves performing mathematical operations on the features in the data archive. Some common data transformations involve normalizing the features to some finite range or creating new aggregated features. Many data mining tools benefit from normalizing all the features in the examples to a uniform range (e.g. zero to one), when the numeric range of different features varies widely in magnitude. Creating new features through aggregation is beneficial for learning complex concepts with tools with limited expressive abilities. In our river flow example, aggregation can

be used to create new features that are unitless, such as the Reynolds number (Fischer *et al.* 1979). Unitless features are valuable in modeling physical processes, because they ensure that the resulting equations are consistent in terms of units. Feature selection involves selecting only salient features from which to compose training examples. This step defines the dimensionality of the feature space and, thus, has a major impact on the complexity of the concept being learned. If the feature space is unnecessarily large, then it will be more difficult to learn a particular concept than if the feature set had been selected correctly (Kohavi & John 1997). Selecting the optimal feature set, however, is non-trivial, Kohavi and John (1997) review several popular methods for feature set selection.

The data mining techniques used in this research include genetic programming (GP), clustering, artificial neural networks (ANNs), nearest neighbor, and Bayesian methods. The remainder of this chapter summarizes each of these techniques.

2.1 Genetic Programming

Genetic programming is a domain independent method that creates a model based on input data, by searching the space of possible models. This search uses operations inspired by natural evolution, which allow GP to cultivate a diverse set of approaches to solving the problem (Banzhaf *et al.* 1998). Genetic programming has been used successfully for many applications, ranging from electrical circuit design and molecular biology (Koza *et al.* 1999) to rainfall-runoff modeling (Savic *et al.* 1999). Babovic and Abbott (1997b) present four applications of GP in the field of hydrology. The results of

each of these applications illustrate different strengths of GP: (1) its to model “emergent phenomena,” (2) its to find models of data that match human derived models, (3) its to find models of phenomena that are of higher quality than human derived models, and (4) its ability to find models of complex phenomena that are equally accurate, yet simpler to solve, than many human derived models.

GP can create models composed of any set of elementary units whose behavior is well defined (e.g. circuit components). In this research, however, GP will be used to suggest models in the form of mathematical equations, a task referred to as symbolic regression. Regression is the most familiar method of determining relationships between data and known parameters. In traditional regression methods, first a model structure is selected. Then, the coefficients of that model are estimated, based on available data, using a model-fitting algorithm. This method builds the user's bias into the resulting relationship through the functional form of the model chosen for regression. Symbolic regression, however, is a less biased method of determining a relationship between data and known parameters because it determines, based on the available data, not only model coefficients, but also the functional form of the model itself (Babovic & Bojkov 2001).

The process of symbolic regression begins with the establishment of a population of models that has been randomly generated from sets of independent variables and mathematical operators. Each model can be conceptualized as a hierarchy of building blocks connected via mathematical operators, each of which is a valid mathematical statement. These building blocks will hereafter be referred to as clauses. The search for

models that best fit the data is directed by one or more objectives that describe the desired qualities of the model. The fitness of a candidate model is based on its fulfillment of these objectives. The search progresses as a series of iterations known as epochs, and the population in each successive epoch is generated by selecting some of the models for propagation. Selection favors models with higher fitness. Models are propagated into the next epoch, either without modification or with modification, through the operations of crossover or mutation. Crossover is performed by swapping clauses between two equations, whereas mutation is performed by altering an independent variable, constant, or mathematical operator in an equation.

In this research, a symbolic regression implementation known as adaptive logic programming (ALP) was used. ALP employs the concise language of logic programming to facilitate the search through the space of possible mathematical equations. This language enables convenient performance of crossover and mutation and avoidance of syntactically incorrect equations via these operations. More information regarding the ALP system can be found in Keijzer *et al.* (2001).

While other data-driven methods exist that will create black box models that map input data to outputs (e.g. artificial neural networks), symbolic regression provides the benefit of expressing the models in the language of mathematics; hence, they can be analyzed for information regarding the underlying processes that created the data. This information can lead to new understandings of the physical processes being modeled.

While symbolic regression provides the advantage of constructing models without domain specific knowledge, the field of application or desired use of the model may impose constraints. In the case of this research, three goals required the imposition of constraints on the symbolic regression task, based on the desire to create: (1) physically meaningful models, (2) parsimonious models, and (3) models that are expressed as partial differential equations (PDEs).

Models of the physical domain must be dimensionally consistent if they are to be considered meaningful; thus, it is necessary to constrain the GP search to only dimensionally consistent equations. While this can be accomplished in many ways (e.g. Keijzer & Babovic 1999), it is most easily accomplished by converting the model parameters into dimensionless values—the strategy used in this study.

In addition to dimensional consistency, model parsimony is desired, because it removes parameters that add to model uncertainty without compromising predictive ability, and because it renders models that are easier both to analyze for semantic meaning and to implement numerically. Symbolic regression will not necessarily find the most concise form of a mathematical statement. In fact, theoretical studies have shown that GP has a tendency to construct models with many extraneous clauses in an effort to protect salient clauses from the destructive effects of crossover and mutation (Banzhaf & Langdon 2002), a phenomenon commonly referred to as “bloat.” Therefore, it is often necessary for the user to simplify the resulting models into statements that are easier to implement and analyze. Useful strategies for the user to manually address model simplification

include converting mathematical operators to equivalent series representations (e.g. using a Maclaurin series to represent an exponential function) and replacing clauses that approximate constant values with constant-valued parameters. Furthermore, domain knowledge can be used to modify the model to address shortcomings in its predictive ability.

2.2 Clustering

Clustering is a method of partitioning data into subsets (clusters), such that the data within each subset are more similar to one another than they are to data in other subsets. Returning to our river flow scenario, examples with large values of stage and velocity would be partitioned into one cluster, examples with small values of stage and velocity would be partitioned into another cluster, and examples with small stage values and large velocity values would be partitioned into a third cluster. Usually, example similarity is defined based on proximity within the feature space, according to some predefined distance metric. Clustering algorithms can be divided into two types: hierarchical and partitioning. Hierarchical algorithms construct clusters, either by iterative aggregation (called “agglomerative clustering”) or by iterative division (called “divisive clustering”) of previously constructed clusters. Agglomerative algorithms begin by considering each data point as an individual cluster and then iteratively merging the closest clusters until some stopping criteria (e.g. minimum distance between clusters, number of clusters) has been reached. Divisive methods begin by considering all the data points as a single cluster and then iteratively dividing the most distant clusters until some stopping criteria

has been reached. Unfortunately, hierarchical clustering algorithms do not scale well to large data bases.

Partitioning clustering algorithms, such as k -means or k -medoids, partition the data into k clusters, according to the cluster center to which the data are closest. In k -means, the cluster center is defined as the average position in the feature space of all the data in the cluster, whereas in the k -medoids algorithm, the cluster center is defined as the most central data point in the cluster. Each of these algorithms begins by randomly selecting k cluster centers and then partitioning the data and refining the estimate of the cluster center. This process of partitioning and finding the cluster center is repeated iteratively until the location of the cluster center converges. The k -means algorithm scales better to large data sets, but it is more sensitive to outliers in the feature space than the k -medoids algorithm. Furthermore, the clusters found via the k -means algorithm may vary, depending on the initial choice of cluster mean, so it is often necessary to perform k -means clustering several times to find the best clusters. Unfortunately, if the number of clusters is not known *a priori*, then there is little guidance regarding how to select k , the number of clusters. Hastie *et al.* (2001) suggest a method that uses the within-cluster scatter, which indicates the similarity of the points to their assigned cluster center, to give an indication of an appropriate number of clusters. This method posits that if k^* clusters exist in the attribute space, then as the number of clusters is increased but is still less than k^* , examples that belong in different clusters will be less likely to be assigned to the same cluster, resulting in a marked decrease in the within-cluster scatter. However, as the number of clusters is increased and becomes greater than k^* , true clusters will be

broken up, resulting in only a marginal decrease in the within-cluster scatter. Thus, an appropriate value for k can be selected by plotting the within-cluster scatter (in log scale) versus k and selecting the value of k at which the slope of the curve decreases, causing an inflection.

The clustering algorithm used in this research is a k -means algorithm (Han & Kamber 2006) that was implemented in C++ by the author, following Han and Kamber (2006). The number of clusters can be specified *a priori* or learned from data using within-cluster scatter (Hastie *et al.* 2001).

2.3 Perceptron and Artificial Neural Networks

The perceptron maps an input vector \vec{x} to an output scalar value y through the transform:

$$y = \vec{w}^T \cdot \vec{x} + b \quad (2.1)$$

where \vec{w} is a weight vector that defines the relationship between \vec{x} and y . In our river flow example, y would be the river flow rate, and \vec{x} would be the feature vector composed of the values of stage and velocity. The weight vector \vec{w} is learned iteratively, by applying a small correction to each element w_i in the weight vector, proportional to both the perceptron's error on a training pattern and the product $w_i x_i$. This learning algorithm, called the "perceptron learning rule" (Rosenblatt 1958), traverses the error surface to find the point where the model error is minimized. The perceptron can either be used as a classifier or as a regression tool. The perceptron classifier employs a threshold function that maps the linear output to a binary output. It can be seen from

Equation 2.1 that the perceptron is limited to linear functions of the input vector; thus, the perceptron is a linear classifier or a linear regression model. This result led Minsky and Papert (1969) to suggest that the perceptron was too limited to be of practical use, resulting in decreased interest in perceptrons. However, it was later shown that networks constructed of multiple layers of inter-connected perceptrons were capable of representing any non-linear function (Rumelhart *et al.* 1987). These networks have commonly been referred to as either multi-layer perceptrons (MLPs) or artificial neural networks (ANNs).

The structure of an ANN is best described as consisting of several layers, each consisting of one or more perceptron-like nodes. These layers are usually referred to as input, hidden, or output layers, depending on their ordering. The “input layer” is the first layer in an ANN and serves to pass the elements of the input vector to the next layer in the network. Its name, however, is a misnomer, as it is not really a layer of perceptron-like nodes. In our river flow example, the input layer would consist of two nodes corresponding to stage and velocity. The final layer in an ANN is referred to as the “output layer,” because each of its nodes produces one element in the output vector, according to Equation 2.1. In our river flow example, the output layer would consist of one node corresponding to the river flow rate. Between the input and output layers are the “hidden layers,” so named because the inputs and outputs of these layers are neither the network inputs nor the outputs. The output of each node in the hidden layers is calculated by applying a non-linear transformation to the node output calculated by

Equation 2.1. The non-linear transformation commonly takes the form of a logit, sigmoid, or radial basis function (Bishop 1995).

The most common form of ANN is one in which the connections proceed from the input layer to each successive hidden layer, and finally to the output layer, without cycles. This type of network is called a feed-forward network. Recurrent neural networks include one or more cycles. These networks are often better at tasks such as sequence prediction, in which knowledge of previous network states is necessary (Elman 1990). One simple recurrent network configuration, the Elman network (Elman 1990), includes cycles that connect each node in the single hidden layer back to the input layer. Networks in which every node is connected to every other node are referred to as fully connected networks. Other types of ANNs include Hopfield networks (Bishop 1995) and Kohonen Self-Organizing Networks (Han & Kamber 2006).

The simplest approach for training feed-forward or recurrent networks is the error backpropagation method (Rummelhart *et al.* 1986), an iterative method similar to the perceptron learning rule. Given a training example, this learning strategy attempts to approximate the error gradient with respect to the elements of each node's weight vector, without taking into account the dependencies between nodes in adjacent layers.

Although error backpropagation is easy to implement, it may require a large number of iterations to converge to a solution, and it may converge to locally optimal solutions rather than the global minimum in the error surface.

Another learning strategy involves posing the error minimization problem as a set of simultaneous equations, with one equation for each training example. Assuming that there are more training examples than network weights, this set of equations is over-determined, and the network weights can be solved using methods such as conjugate gradient or Levenberg-Marquardt (Bishop 1995). These methods are often quicker than backpropagation and are less likely to converge to local optima. It is important to note, however, that these methods are, by definition, batch learning algorithms.

The most challenging aspect of applying ANNs is to select the number of hidden layers and the number of nodes in each hidden layer. These parameters are referred to as the network's architecture, and they determine the range of functions that the network can express; this range grows with the network size. Unfortunately, under certain conditions (e.g. few training examples), more expressive networks are capable of memorizing the training examples, a phenomenon commonly referred to as "overtraining," which leads to poor network generalization to previously unseen examples. Thus, the task of selecting network architecture requires selecting architecture that is expressive enough to characterize the data, without being so expressive that it is prone to overfitting.

While the network architecture is commonly selected by trial and error approaches (Bishop 1995, Hecht-Nielsen 1990), some theoretical guidance exists. Bishop (1995) indicates that often, more than one hidden layer is not necessary. Furthermore, the Kolmogorov Theorem states that any continuous function of n variables (i.e. the features in the training set) can be constructed by the superposition of $2n+1$ linear models of one

variable, thus indicating that $2n+1$ nodes in the hidden layer should be used (Bishop 1995). Unfortunately, this theorem does not specify what form the linear function of one variable should take, nor does it require that this form be constant for all of the $2n+1$ functions. Furthermore, this large number of hidden nodes can often result in overfitting of the training data, resulting in an ANN that does not generalize well. Thus, Hecht-Nielsen (1990) suggests that architectures with $2\sqrt{n} + m$, where m is the number of outputs to $2n+1$ nodes in the hidden layer, should be tried. Duda *et al.* (2001), suggest that the number of weights should be related to the number of available training data points and recommend that the number of weights in the network should be equal to $1/3$ the number of training examples. This guidance ensures that enough information is contained in the training set to learn the network weights; however, this method may recommend too many weights if the number of training examples is large.

Due to the complexity of selecting an appropriate network architecture, some researchers have suggested methods for determining the architecture during the learning process. For example, the optimal brain surgeon algorithm (Bishop 1995) decreases the expressivity of a network by removing connections between nodes that have weights below a particular threshold, while the multiple resource allocation neural network (Markus 2005) increases/decreases the number of hidden layers, if the network appears to be under/overfitting the training data.

Like perceptron, ANNs can be used for either classification or regression; however, unlike perceptron, ANNs are capable of producing non-linear output. Feed-forward and

recurrent neural networks have been shown to be successful for many applications, including tidal prediction (Huang *et al.* 2003, Balas *et al.* 2004), wind speed prediction (More & Deo 2003), rainfall-runoff modeling (Amenu *et al.* 2007), and other time-series data (Zhang & Qi 2005).

The perceptron/ANN algorithm used in this research addresses only feed-forward architectures and was implemented in C++ by the author. Network architecture must be specified *a priori*. Network weights are learned from a training set using the standard backpropagation algorithm. Training was terminated after a fixed number of training epochs were completed or when further training caused a decrease in the model performance on a testing set (Bishop 1995; Hastie *et al.* 2001). This latter condition discouraged overtraining of the network.

2.4 Nearest Neighbor

The nearest-neighbor method (Duda *et al.* 2001, Hastie *et al.* 2001) is a method for predicting the classification or outcome of a system state based on observable features. The target value of an unseen example is predicted to be the same as that of the nearest neighbor in the feature space. Training the predictor only requires archiving the training set so that distances between new examples and the training examples can be computed quickly. Prediction is performed by calculating the distances between the feature vector for prediction and each feature vector from the training set. The training example with the shortest distance is the nearest neighbor, and the target value of the nearest neighbor is predicted to be the output of the system to the new example. In our river flow example,

the flow rate for a particular combination of stage and velocity would be predicted to be the flow rate of the most similar (in feature space) previously observed example.

As the number of training examples increases, this predictor will become more computationally demanding; thus, many optimizations have been suggested that attempt to reduce the number of distances actually computed. For example, the feature space can be partitioned so that distances will only have to be computed within specific partitions known to contain the nearest neighbor. Furthermore, the accuracy of the algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the numeric range of the features varies widely. Thus, feature selection and normalization (i.e. scaling the features such that they have the same numeric range) are very important preprocessing steps when using this algorithm.

This method can be extended to consider more than one of the nearest training examples, where the classification of a new feature set is taken as the majority vote, and predictions can be made as the average response. Furthermore, confidence in the prediction can be inferred from the homogeneity/heterogeneity of the target values of the training examples determined to be nearest to the new example. This extension is called the k -nearest neighbor algorithm.

The nearest neighbor algorithm used in this research was implemented in C++ by the author, using Euclidean distance as the distance metric. This implementation is based on the nearest neighbor algorithm presented by Duda *et al.* (2001).

2.5 Dynamic Bayesian Networks

Bayesian networks are members of the family of probabilistic graphical models (Jordan 2002). They represent a set of variables and their dependencies as directed acyclic graphs, whose nodes represent the variables and whose arcs represent the conditional dependencies between the variables. The variables within the graphs can be either random or deterministic; thus, given a set of values of deterministic variables, the networks can be used for Bayesian inference (i.e. inference that is updated for a particular set of observed variable values) regarding the belief state (i.e. the joint probability distribution of the encoded variables conditioned on the values of the deterministic variables) of the random variables they encode. Furthermore, the variables encoded in the network can be either discrete (e.g. night/day) or continuous (e.g. flow rate), and they can also be either scalar or vector quantities.

Dynamic Bayesian networks (DBNs) are Bayesian networks whose network topology evolves with time; thus, they are well suited for modeling data sequences because new variables can be added incrementally to the network to represent new members of the sequence. DBNs are generalizations of common state space models, such as hidden Markov models and Kalman filters (Murphy 2002), and they are well suited for time-series modeling because of their flexibility in handling multivariate data and non-stationary processes (Spall 1988).

Figure 2.1 illustrates a DBN model of time-series data. In this DBN, the system state (\mathbf{X}) at any given time is considered to be unknown and only observable through noisy measurements (\mathbf{Z}). The system state at any time t is represented by \mathbf{X}_t , and the measurements of the system state at any time t are represented by vector \mathbf{Z}_t . Each pair $(\mathbf{X}_t, \mathbf{Z}_t)$ is referred to as a time slice. In our river flow example, the state variables (\mathbf{X}) would be vectors representing the river stage and flow rate at any time t , while the observed variables (\mathbf{Z}) would be stage measurements corresponding to time t . The graphical structure of the DBN indicates that \mathbf{X}_t is conditionally independent, given \mathbf{X}_{t-1} , and that \mathbf{Z}_t is conditionally independent, given \mathbf{X}_t . The former conditional independence relationship is known as the first-order Markov condition. Given the Markov assumption, the belief state completely captures all the information from the past (Boyen & Koller 1999); thus, the DBN assimilates new measurements into its belief state, allowing it to model non-stationary processes.

To fully specify a DBN, not only must the graphical structure of the network be specified, but the transition model (i.e. the model describing the evolution of the state variable distribution from time t to $t+1$) and the observation model (i.e. the model describing the

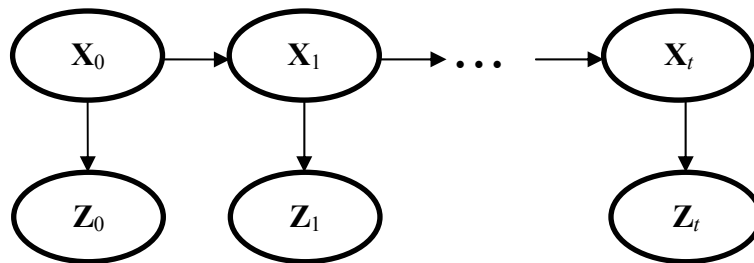


Figure 2.1 Schematic of Dynamic Bayesian network.

relationship between the state variables and the observed variables) must also be specified. If the state variables are Gaussian, and the transition and observation models are linear, then the required model parameters are the (1) transition matrix, (2) transition covariance matrix, (3) observation matrix, and (4) observation covariance matrix. From the state-space modeling perspective, these models can be represented respectively as follows:

$$X_{t+1} = A \cdot X_t + N(0, Q) \quad (2.2a)$$

$$Z_t = C \cdot X_t + N(0, R) \quad (2.2b)$$

where A is the transition matrix, Q is the transition covariance matrix, C is the observation matrix, R is the observation covariance matrix, $N(0, Q)$ represents zero mean Gaussian noise with covariance matrix Q (the system noise), and $N(0, R)$ represents zero mean Gaussian noise with covariance matrix R (the observation noise). Note that A and C are not functions of time, and thus the process and measurement model's dynamics are assumed to be time stationary. This assumption is not inherent to state-space modeling in general, but it will be used throughout this research because of the added complexity of specifying time-varying dynamics models. Often, the transition and observation models are known *a priori*. However, if the models are not known, the expectation maximization (EM) method can be used to learn these models from a sequence of observed data (Ghahramani & Hinton 1996, Digalakis *et al.* 1993, Shumway & Stoffer 1982); the last reference suggests modifications to account for missing observations in the training data. The EM method is a two-step iterative process. In the first step (E) of EM, an estimate of the model parameters is used to calculate the expected value of the likelihood that the models describe the observed system behavior. The second step (M) analytically

calculates the maximum likelihood estimate of the model parameters, using the expected likelihood function found in the E step. The parameters found in the M step are then used to begin another E step, and the process is repeated. For a more thorough description of the EM method, see Duda *et al.* (2001). If the state variables are discrete, then the transition and observation models take the form of discrete conditional probability tables, which can be learned using the Baum-Welch algorithm (Baum *et al.* 1970). In this research the EM algorithm was only applied to the Kalman filter, and more details are given in Section 2.5.1. The EM algorithm is significantly more complex for DBNs with more complex state distributions, such as those used for robust Kalman and particle filtering, and the data considered in this research (see Chapter 6) were not sufficient to adequately characterize these distributions. Therefore this approach is not discussed further for the more complex DBNs in Sections 2.5.2 and 2.5.3.

Given the transition and measurement models, and a sequence of measurements of the system, DBNs can infer something about the true state of the system via filtering, prediction, smoothing, or most-likely-explanation. Filtering refers to the inference of the current system state, given all measurements to date. Prediction refers to the inference of some future system state, given all measurements to date. Smoothing refers to the inference of a previous system state, given all measurements to date. Most-likely-explanation refers to the inference of the most likely sequence of system states that will result in the observed series of measurements. Since this research focuses on real-time forecasting, only filtering will be discussed in this section. For more information about prediction, smoothing, and most-likely-explanation, see Russell and Norvig (2003).

DBN filtering has a long history. The most notable early work is attributed to Kalman (Kalman 1960). Since then, many researchers have generalized this work and associated it with state-space models and hidden Markov models (e.g., Jordan 2002). The result of this study is a well-known recursive algorithm for filtering DBNs (Murphy 2002). In a recursive algorithm, data are processed sequentially, rather than in batch form; the complete set of data never needs to be stored prior to calculating the filtered estimate; and reprocessing of existing data is unnecessary if new measurements become available. Thus, recursive algorithms are well-suited for real-time forecasting of streaming data. Unfortunately, it is generally not possible to obtain analytical solutions for the Bayesian recursion relations required for exact filtering (Sorenson 1988). The remainder of this section is devoted to describing Kalman Filtering, the most notable exception to this rule, as well as to discussing two approximate methods for filtering: robust Kalman filtering and particle filtering.

2.5.1 Kalman Filtering

Kalman filtering is an analytical method of filtering a DBN, in which the system state is a continuously valued vector or scalar, and the state transition model is linear. This algorithm can be extended to non-linear state transitions through data transformations, as in the case of the extended Kalman filter (e.g. Sorenson 1966, Anderson & Moore 1979). During the Kalman filtering recursion, the state vector at time t is propagated forward in time by using the state transition model to calculate the prior probability distribution of the state vector at time $t+1$. Then, the available measurements are used to calculate the

posterior probability of the state vector at time $t+1$ using the observation model. The Kalman filtering equations can be written as:

$$\hat{X}_{t+1} = A\hat{X}_t + K_{t+1}(Z_{t+1} - CA\hat{X}_t) \quad (2.3a)$$

$$\hat{\Sigma}_{t+1} = (I - K_{t+1}C)(A\hat{\Sigma}_tA^T + Q) \quad (2.3b)$$

$$K_{t+1} = (A\hat{\Sigma}_tA^T + Q)C^T [C(A\hat{\Sigma}_tA^T + Q)C^T + R]^{-1} \quad (2.3c)$$

where \hat{X}_t is the filtered estimate of the posterior mean of the state vector at time t , K_{t+1} is the Kalman gain matrix, $\hat{\Sigma}_t$ is the filtered estimate of the posterior covariance matrix, I is the identity matrix, and the symbol (T) indicates the matrix transpose. The Kalman gain matrix blends the Bayesian prior of the state vector at time $t+1$ with the error between the prior and the observation. The smaller the error covariance (R), the more the Kalman filter “believes” the observation, thus resulting in a larger gain matrix. Larger error covariance results in less weight being given to observation, thus resulting in a smaller gain matrix.

The Kalman filtering equations (Equation 2.3) can be adaptively modified during the recursion to accommodate missing or partially missing observations. Missing observations occur when none of the measurements in the observation vector are available, while partially missing observations occur when a subset of the measurements in the observation vector are available. In the case of a missing measurement, the state mean and covariance matrix of the Bayesian prior are used to define the belief state. In the case of a partially missing measurement, the calculation of the Kalman gain matrix must be modified, such that the gain matrix elements corresponding to the missing

measurements are equal to zero (Liu & Goldsmith 2004, Schumway & Stoffer 1982). This is accomplished by setting the missing measurements, as well as the elements of matrices C and R that correspond to the missing measurements, equal to zero. For example, if there are two measurements, and the measurement stored in position 1 of the measurement vector is missing, then the element in position 1 of the measurement vector and the elements in positions (1,1), (1,2), and (2,1) of matrices C and R would be set to zero.

To learn the Kalman filter parameters A , C , R , and Q in Equations 2.3 a-c from a sequence of data, the EM method discussed previously is used. The EM method is a general iterative optimization method that seeks to find the value of an unknown parameter that maximizes the joint probability of the state and measured variables given a set of measured variables. This method proceeds in two steps: an E step and an M step. Application of the EM method to the task of parameterizing a Kalman filter begins with the E step, which uses an estimate of the model parameters to calculate the expected value of the likelihood function. The initial values of the model parameter estimates are set by the user, who makes a best guess at their values. In the M step, the maximum likelihood estimate of the model parameters are calculated using the expected likelihood function found in the E step. The parameters found in the M step are then used to begin another E step, and the process is repeated.

The E step for Kalman filter parameterization begins by calculating the expected likelihood function of the states and their observations. The joint probability of the

predicted states at each time t and the observations that correspond to these states can be expressed as:

$$P(\{X\}, \{Z\}) = P(X_1) * \prod_{t=2}^T P(X_t | X_{t-1}) * \prod_{t=1}^T P(Z_t, X_t) \quad (2.4)$$

where $\{X\}$ denotes the set of state vectors, $\{Z\}$ denotes the set of all observation vectors, X_t is the state at time t , Z_t is the observation at time t , $P(X_t | X_{t-1})$ is the probability of X_t conditioned on X_{t-1} , and $P(Z_t | X_t)$ is the probability of Z_t conditioned on X_t . Since the Kalman filter assumes that the joint distribution of the state and observation vectors is Gaussian, substituting the equation for the multivariate Gaussian into Equation 2.4 and taking the natural logarithm retains the behavior of the likelihood function and reduces Equation 2.4 to a sum of quadratic terms:

$$\begin{aligned} LL = & -\sum_{t=1}^T \left[\frac{1}{2} (Z_t - CX_t)^T R^{-1} (Z_t - CX_t) \right] - \frac{T}{2} \log_e(|R|) \\ & - \sum_{t=2}^T \left[\frac{1}{2} (X_t - AX_{t-1})^T Q^{-1} (X_t - AX_{t-1}) \right] - \frac{T-1}{2} \log_e(|Q|) \\ & - \frac{1}{2} (X_1 - \mu_1)^T V_1^{-1} (X_1 - \mu_1) - \frac{1}{2} \log_e|\Sigma_1| - \frac{T(p+k)}{2} \log_e(2\pi) \end{aligned} \quad (2.5)$$

where p is the dimensionality of the state vector; k is the dimensionality of the observation vector; μ_1 and V_1 are the mean and the covariance matrix of the initial state distribution, respectively; the operator $(^T)$ indicates the matrix transpose, and the operator $(| |)$ indicates the matrix L2-norm. The expectation of this log-likelihood, given the sequence of training observations is:

$$\begin{aligned}
E[LL | \{Z\}] = & -\frac{1}{2} \log_e |V_1| - \frac{1}{2} \text{tr} \left\{ V_1^{-1} \left[\tilde{\Sigma}_1 + (\tilde{X}_1 - \mu_1)(\tilde{X}_1 - \mu_1)^T \right] \right\} \\
& - \frac{n}{2} \log_e (|Q|) - \frac{1}{2} \text{tr} \left\{ Q^{-1} (\gamma - \beta A^T - A \beta^T + A \alpha A^T) \right\} \\
& - \frac{n}{2} \log_e (|R|) \\
& - \frac{1}{2} \text{tr} \left\{ R^{-1} \sum_{t=1}^n \left[(Z_t - C \tilde{X}_t)(Z_t - C \tilde{X}_t)^T + C \tilde{\Sigma}_t C^T \right] \right\}
\end{aligned} \tag{2.6a}$$

$$\alpha = \sum_{t=2}^n (\tilde{\Sigma}_{t-1} + \tilde{X}_{t-1} \tilde{X}_{t-1}^T) \tag{2.6b}$$

$$\beta = \sum_{t=2}^n (\tilde{P}_{t,t-1} + \tilde{X}_t \tilde{X}_{t-1}^T) \tag{2.6c}$$

$$\gamma = \sum_{t=1}^n (\tilde{\Sigma}_t + \tilde{X}_t \tilde{X}_t^T) \tag{2.6d}$$

where \tilde{X}_t and $\tilde{\Sigma}_t$ are the mean and variance of the posterior state distribution given the full sequence of data (both past and future observation), not just the previous data; $\tilde{P}_{t,t-1}$ is the posterior covariance of the system state at times t and $t-1$ given the full data sequence; n is the number of measurements in the training sequence; and tr indicates the matrix trace. Since these quantities depend on the full sequence of data (both past and future observation), not just on the previous data, the Kalman filtering relationships are insufficient to calculate them; however, Kalman smoothing can be used to calculate these quantities.

Kalman smoothing is performed using a recursive algorithm often referred to as the forward-backward algorithm (Russel & Norvig 2003). Given a set of observations, this algorithm performs two sequential processing steps. In the first step, the data are processed in chronological order, and Kalman filtering is performed with the filtered state

estimate (i.e. \hat{X}_t and $\hat{\Sigma}_t$) being stored at each time slice. In the second step, the data are processed in anti-chronological order, and the Kalman smoothing recursions are performed, as detailed in the following equations:

$$\tilde{X}_{t-1} = \hat{X}_{t-1} + J_{t-1}(\tilde{X}_t - A\hat{X}_{t-1}) \quad (2.7a)$$

$$\tilde{\Sigma}_{t-1} = \hat{\Sigma}_{t-1} + J_{t-1}(\tilde{\Sigma}_t - \hat{\Sigma}_t)J_{t-1}^T \quad (2.7b)$$

$$J_{t-1} = \hat{\Sigma}_{t-1}A^T(A\hat{\Sigma}_{t-1}A^T + Q)^{-1} \quad (2.7c)$$

where \tilde{X}_t and $\tilde{\Sigma}_t$ are the smoothed state distribution mean and variance, respectively.

Shumway and Stoffer (1982) have shown that during the backwards smoothing recursion,

$\tilde{P}_{t,t-1}$ can be calculated as:

$$\tilde{P}_{t,t-1} = \hat{\Sigma}_{t-1}J_{t-1}^T + J_t(\tilde{P}_{t+1,t} - A\hat{\Sigma}_{t-1}J_{t-1}^T) \quad (2.8)$$

Equations 2.3, 2.7, and 2.8 provide the values necessary to compute the expected log-likelihood function in Equation 2.6, thus concluding the E step.

In the M step, the maximum likelihood estimates of the model parameters are calculated by setting the partial differential of Equation 2.6 to zero. For the parameters A , C , and Q of Equation 2.3, the updated parameter estimates are:

$$A(r+1) = \beta\alpha^{-1} \quad (2.9a)$$

$$Q(r+1) = n^{-1}(\gamma - \beta\alpha^{-1}\beta^T) \quad (2.9b)$$

$$R(r+1) = \sum_{t=1}^n \left[(Z_t - C\tilde{X}_t)(Z_t - C\tilde{X}_t)^T + C\tilde{\Sigma}_tC^T \right] \quad (2.9c)$$

where r indicates the EM iteration counter.

An update equation for the measurement model is not given, since the relationship between the system state and the sensor's output is often known (as was the case in this work: see Section 6.1); however, this update equation can be derived using the method described above, as well. After updating the parameter values in the M step, the sequence of E and M steps is repeated until further updates do not increase the likelihood function. It can be shown that each EM iteration increases the observed data likelihood function. However, there is no guarantee that the sequence converges to a maximum likelihood estimator; in fact, since the likelihood function converges to infinity as the joint probability distribution narrows to a degenerate distribution centered on one of the training data points, care must be taken to find a local maximum of the likelihood function that provides good performance on the test data. Thus, many EM trials, starting from different initial conditions, may be necessary. For example, in Chapter 6, four trials were carried out before a result that was not caused by this degeneracy was found.

Because the state and observation vectors are normally distributed, and because the state transition and observation models are linear, the prior and posterior distributions can be calculated analytically, resulting in a compact representation of the belief state of the DBN (i.e. mean and covariance matrix), as well as in efficient updates to accommodate new measurements. While the linear Gaussian assumption is a strong assumption, Kalman filters have been successful at modeling time-series data in a wide variety of applications, such as aerospace studies (e.g. Schmidt 1981), economics (e.g. Johnson &

Sakoulis 2003), computer graphics engineering (e.g. Welch *et al.* 2001), and hydrodynamic modeling (e.g. Yonekawa & Kawahara 2003).

The Kalman filtering algorithm used in this research was implemented in C++ by the author, following Maybeck (1979). Missing and partially missing observations are addressed using the modification of the Kalman gain matrix adapted from Liu and Goldsmith (2004) and Schumway and Stoffer (1982). Transition and observation model parameters can either be learned via EM (Ghahramani & Hinton 1996, Digalakis *et al.* 1993, Shumway & Stoffer 1982) or be specified *a priori*.

2.5.2 Robust Kalman Filtering

Not only does Kalman filtering require the strong assumption of linear Gaussian distributions, but it is also not robust, since, as can be seen in Equation 2.3, the mean of the Bayesian posterior distribution is an unbounded function of the discrepancy between the observation and its Bayesian prior, while the variance does not depend on the observed data. Thus, the model cannot adapt for changes in the measurement variance, and outlying observations will adversely affect inferences about the belief state (Schick & Mitter 1994, Meinhold & Singpurwalla 1989). Robust Kalman filtering is an approximate method of filtering a DBN that retains the advantages of Kalman filtering (e.g. fast updates for new measurements and compact representation of the belief state), while still addressing the problem of outliers explicitly within the DBN structure by permitting the use of heavy-tailed distributions in place of the multivariate Gaussian distributions required for Kalman filtering. The assumption of linear transition and

observation models is also used in robust Kalman filtering. The remainder of this subsection will describe two approaches for robust Kalman filtering that employ Student's t-distributions and mixture-of-Gaussian (MoG) distributions, respectively.

Student's t-distribution is similar to the Gaussian distribution, except that Student's t-distribution uses the estimator for the population standard deviation given the population size (N), instead of using the true standard deviation (Note: the quantity $N-1$ is often referred to as the number of degrees of freedom). Thus, Student's t-distribution is considered to be the distribution that best describes a variable, if the variable's standard deviation is uncertain (Devore 1995). With one degree of freedom, Student's t-distribution is equivalent to the standard Cauchy distribution. As the number of degrees of freedom increases, the Student's t-distribution converges to the standard normal distribution. Student's t-distribution can be used for robust Kalman filtering if the distribution of the state and observed variables are specified to be Student's t-distributions with n and m degrees of freedom, respectively, where n and m are chosen to attain the desired level of robustness (e.g. Meinhold & Singpurwalla 1989).

Unfortunately, the use of multivariate Student's t-distributions violates the behavior of a robust Kalman filter because the posterior distribution calculated after observing an outlying measurement does not converge to its prior (Dawid 1973), so it is necessary to approximate the multivariate Student's t-distribution by assuming independence of the variables and approximating the multivariate posterior distribution of the belief state as a poly-t (i.e. a mixture of Student's t-distributions) distribution (Broemeling 1985). Since the number of components in the poly-t distribution will grow as an exponential function

of the number of time slices evaluated by the robust Kalman filter, it is necessary to approximate the poly-t distribution itself to limit the number of mixture components (e.g. see Jeffreys 1961).

The probability density function of the MoG distribution is represented as a weighted average of the probability distribution functions of Gaussian mixture components:

$$f(x) = \sum_i \alpha_i f_i(x) \quad (2.10)$$

where α_i is the mixture ratio – the weight of the i^{th} mixture component, $\sum_i \alpha_i = 1$, and $f_i(x)$ is the probability density function of the i^{th} mixture component. For example, a MoG distribution with two components with means and variances of (0,1) and (0,10), respectively, will have heavy-tailed behavior, due to the second mixture component. The amount of probability mass residing in the tails of this distribution can be further controlled by adjusting the α values. Robust Kalman filtering using MoG representations of the state and observed variables can be envisioned as using a mixture of Kalman filters, each specific to one Gaussian mixture component. The mixture ratios for the state and observed variable distributions are specified as priors, which are updated given new observations, as shown by Frühwirth (1995) and Guttman & Peña (1988), respectively. As in the case of the poly-t distribution, the number of components of the MoG distribution will increase exponentially with the number of filtered time slices. Therefore, it is necessary to approximate the MoG distribution to limit the number of components. This is done by approximating the MoG posterior distribution as a Gaussian distribution with mean:

$$\mu = \sum_i \alpha_i \mu_i \quad (2.11)$$

and variance:

$$\Sigma = \sum_i \alpha_i (\Sigma_i - \mu_i^2) \quad (2.12)$$

where μ_i and Σ_i are the mean and variance of the i^{th} mixture component, respectively.

This approximation minimizes the Kullback- Liebler distance between the MoG and its Gaussian approximation and has been shown to be the optimal Gaussian approximation of the MoG (Peña & Guttman 1989).

The robust Kalman filtering algorithm used in this research was implemented in C++ by the author using MoG distributions to accommodate outliers (Frühwirth 1995, Guttman & Peña 1988). Missing and partially missing observations are addressed using the modifications to the Kalman gain matrix and the matrices C and R described in Section 2.5.1. Transition and observation model parameters of the mixture components, as well as the mixture ratio priors, must be specified *a priori*.

2.5.3 Particle Filtering

Both Kalman filtering and robust Kalman filtering are restricted to system state distributions of only continuously-valued variables. If the system state is composed of a hybrid of continuous- and discrete-valued variables, then filtering must be performed with approximate methods that represent the hybrid distributions through sampling (Arulampalam *et al.* 2002, van der Merwe *et al.* 2000, Gordon *et al.* 1993). These methods are called sequential Monte Carlo methods or particle filters. Particle filters

maintain a cloud of “particles” that are a representative sample of the hybrid distribution of the belief state of the DBN. Thus, particle filters do not have a concise representation of the belief state like Kalman and robust Kalman filters do. The cloud of particles can be used to approximate the likelihood of different states or observations, as well as the expected values of these variables, by enumerating the frequency of the particles with different values. The particles are propagated from one time slice to the next, through transition and observation models that can be based on sampling conditional transition distributions (similar to hidden Markov models) or on analytical relations (similar to Kalman filters). Transition and observation models based on sampling transition distributions allow the particle filters to address non-linear and non-Gaussian state variables, but also result in particle filters that are extremely computationally intensive; thus, these models are intractable for large systems (Doucet *et al.* 2000a).

The Rao-Blackwellized particle filter is a special type of particle filter that uses the Rao-Blackwell formula (Casella & Robert 1996) to separate the continuous and discrete state variables. This separation of variables requires only that the discrete variables be conditionally independent of the continuous variables (both continuous variables and discrete variables can be conditionally dependent on discrete variables) (Doucet *et al.* 2000a). Discrete variables that are conditionally independent from other discrete variables are propagated through time using sampling of the conditional transition probabilities, while continuous variables can be propagated through time using more economical transition models, such as those used in Kalman or robust Kalman filtering, thus resulting in significantly increased computational economy (Doucet *et al.* 2000a,

Doucet *et al.* 2000b). For example, if the transition and observation models of the continuous variables are assumed to be linear Gaussian, then the Kalman filtering equations (Equation 2.3) can be modified to describe the continuous variables as:

$$\hat{X}_{t+1}^i = A_k \cdot \hat{X}_t^i + K_{t+1}^i (Z_{t+1} - C_k A_k \hat{X}_t^i) \quad (2.13a)$$

$$\Sigma_{t+1}^i = (I - K_{t+1}^i C_k) (A_k \Sigma_{t+1}^i A_k^T + Q_k) \quad (2.13b)$$

$$K_{t+1}^i = (A_k \Sigma_{t+1}^i A_k^T + Q_k) C_k^T [C_k (A_k \Sigma_{t+1}^i A_k^T + Q_k) C_k^T + R_k]^{-1} \quad (2.13c)$$

where the superscript i indicates the i^{th} particle, and the subscript k indicates the value of the discrete variables upon which X is conditionally independent. A resampling step during the update is used to ensure that the particles stay focused in high probability regions of the posterior belief state distribution, such that the particles maintain a representative sample of this distribution (Koller & Lerner 2000).

The Rao-Blackwellized particle filtering algorithm used in this research was implemented in C++ by the author using the linear Gaussian assumption described in Equation 2.13 (Doucet *et al.* 2000a). Missing and partially missing observations are addressed using the modifications to the Kalman gain matrix and the matrices C and R described in Section 2.5.1. Transition and observation model parameters of the discrete and continuous variables must be specified *a priori*.

Chapter 3: Literature Review

To explore the efficacy of the tools and techniques described in the previous section for solving complex problems of environmental interest, two illustrative applications will be explored: (1) upscaling models of solute transport in porous media, and (2) anomaly detection in environmental sensor data streams. The remainder of this chapter describes each of these applications.

3.1 Upscaling Models of Solute Transport in Porous Media

Solute transport in porous media is fundamental to many significant engineering problems, such as design of subsurface waste storage facilities, performance and assessment of underground waste repositories, remediation of contaminated groundwater aquifers, and design of packed bed chemical reactors. Thus, modeling this process is an area of active research in many disciplines. One popular method of modeling the movement of solute through porous media involves the use of physically-based mathematical equations based on conservation of momentum and mass. Darcy's Law and the advection-dispersion equation (ADE) are widely accepted as the equations governing flow and transport of groundwater and dissolved substances at the continuum-scale, the length scale at which the heterogeneous aggregation of soil grains can be treated as a homogeneous spatially-averaged material. It is now well recognized, however, that natural porous media exhibit significant spatial variability at the continuum-scale and that this variability has a profound impact upon solute fate and transport at the larger field-scale relevant to environmental and hydrological problems. The effect of this variability

on solute transport is enhanced spreading, a phenomenon referred to as macrodispersion. Detailed measurements at several field sites (Sudicky 1986, Mackay *et al.* 1986, LeBlanc *et al.* 1991) have revealed that the length scale of significant conductivity variations is on the order of a few meters in the horizontal direction but only ten to twenty centimeters in the vertical direction. Therefore, computational limitations prevent the use of a transport model grid fine enough to resolve all of the spatial scales of this variability. Furthermore, many problems of environmental interest require solving the transport models many times (e.g. through the use of Monte Carlo simulations); thus, a need exists for more economical models of solute transport. For this reason, much effort has been directed towards developing models that describe transport processes at a length scale larger than the continuum-scale so that coarse computational grid blocks may be used. These “upscaled” models cannot explicitly resolve all of the salient features of the transport process, yet they should capture the impact of the small-scale heterogeneity in order to provide an accurate prediction of the overall plume evolution.

Traditional methods for upscaling the ADE include stochastics (Gelhar *et al.* 1979, Dagan 1986, Sposito 1997, Rubin 2003, Rubin *et al.* 2003), spatial filtering (Beckie *et al.* 1996, Beckie 1998), homogenization (Mei 1992, Wood *et al.* 2003), and statistical moments (Aris 1956, Frankel & Brenner 1989, Kitanidis 1992, Whitaker 1999).

Unfortunately, although these methods are mathematically rigorous, they usually require restrictive assumptions, such as small variability, large scale separation, or ergodicity or periodicity of the medium, to achieve closure of the upscaled models.

Chapter 4 discusses the development of an upscaling methodology using genetic programming (GP), a promising new tool for modeling complex phenomena whose physics are not well defined (Babovic & Abbott 1997a). For illustration, this methodology is applied to the case of developing vertically averaged models of the transport of a non-reactive solute in confined stratified aquifers. The results are compared with models developed through the method of moments (MoM), a traditional upscaling technique that is well suited for this transport configuration (Güven *et al.* 1984).

3.2 Anomaly Detection in Streaming Environmental Sensor Data

Modeling environmental systems requires many parameters, including spatial properties, boundary and initial conditions, and process-related coefficients. Unfortunately, for complex systems, the necessary information is often not readily available, thus requiring researchers to use approximations and best guesses. For example, decision-makers have traditionally relied on historical or averaged seasonal data to predict the response of environmental systems to events of interest, such as contaminant releases. Unfortunately, reliance on these types of data limits the value of these predictions for coordinating real-time responses to such events. Elliott and Jones (2000) and Guinasso *et al.* (2001) present case studies in which real-time data successfully indicated the trajectory of oil slicks caused by tanker accidents, whereas the trajectory predicted using models relying on averaged historical data differed significantly. In the latter case, decisions facilitated by the real-time data allowed the clean-up effort to be focused on the coastal areas affected by the oil slick, thus increasing clean-up efficiency.

In-situ environmental sensors are sensors that are physically located in the environment they are monitoring. They collect time series data that flow continuously to a repository, creating a data stream. Recently, there have been efforts to make use of streaming data from environmental sensors for real-time applications. For example, draft plans for the Water and Environmental Research Systems (WATERS) Network, a proposed national environmental observatory network, have identified real-time analysis and modeling as a significant priority (NRC 2006). The value of streaming data for real-time forecasting and decision making has been demonstrated using a simulated oil spill (Bonner *et al.* 2002), and continuing efforts are being directed towards facilitating near-real-time hydrodynamic forecasting using these data (Shah *et al.* 2005).

Because *in-situ* sensors operate under harsh conditions, and because the data they collect must be transmitted across communication networks, the data can easily become corrupted. Undetected errors can significantly affect the data's value for real-time applications. Thus, a report from a National Science Foundation-funded workshop on sensors for environmental observatories (NSF 2006) has indicated a need for automated data quality assurance and control (QA/QC) that occurs in real time. Anomaly detection is the process of identifying data that deviate markedly from historical patterns (Hodge & Austin 2004). Anomalous data can be caused by sensor or data transmission errors or by infrequent system behaviors that are often of interest to scientific and regulatory communities. In addition to data QA/QC, where anomalous data are treated as erroneous, anomaly detection has many other practical applications, such as adaptive sampling, where anomalous data indicate phenomena that researchers may wish to investigate

further through increased sampling, and anomalous event detection, where anomalous data signal system behaviors that require other actions to be taken, for example in the case of a natural disaster. These applications require real-time detection of anomalous data, which requires that the anomaly detection method be rapid and be performed incrementally to ensure that detection keeps up with the rate of data collection.

Additionally, successful real-time anomaly detection in environmental streaming data must surmount four additional challenges: (1) continuous collection of streaming data results in a large volume of data, thus, the entire data set cannot be held in memory nor can all existing data be reprocessed when new measurements become available; (2) real-time decisions can only use previous observations, thus, future observations cannot be used for anomaly classification; (3) environmental sensors go off-line frequently because of the harsh environment they are deployed in, thus, if a significant number of specific historical measurements are necessary to process a new measurement, many measurements will not be able to be processed; and (4) sensors deployed in the natural environment behave in unexpected ways; thus, no *a priori* definition of the types of anomalies that may be encountered is available.

Traditionally, anomaly detection has been carried out manually with the assistance of data visualization tools (Mourad & Bertrand-Krajewski 2002), but these approaches are too time consuming for real-time detection in streaming data (e.g., wind data from the sensors used in this study are generated at a rate of one sample per second). More recently, researchers have suggested automated statistical and machine learning approaches, such as minimum volume ellipsoid (Rousseeuw & Leroy 1996), convex

peeling (Rousseeuw & Leroy 1996), nearest neighbor (Tang 2002, Ramaswamy *et al.* 2000), clustering (Bolton & Hand 2001), neural network classifier (Kozuma *et al.* 1994), support vector machine classifier (Bulut *et al.* 2005), and decision tree (John 1995). These methods are faster than manual methods, but they have drawbacks that make them unsuitable for real-time anomaly detection in streaming data. For example, some require all of the data to have accumulated before anomalies can be identified; some are computationally intractable for large quantities of data; some require pre-classified anomalous data, which characterize all anomalies that may be encountered; and some require pre-classified non-anomalous data, which characterize the range of possible non-anomalous data.

Several researchers have suggested anomaly detection methods specifically designed for real-time detection in streaming data. These methods can be divided into four categories: (1) redundancy-based approaches, (2) state-space modeling/filtering approaches, (3) rule based-approaches, and (4) combined approaches.

Redundancy-based approaches can be further divided into two sub-categories: physical and analytical. Physical redundancy-based approaches employ two or more identical sensors at a particular location, resulting in multiple coincident measurements. These measurements can be directly compared. If the measurements deviate significantly, then at least one of the measurements can be deemed erroneous. However, it can be seen that if only one redundant sensor is used (two total sensors), then it is impossible to determine which measurement is erroneous; thus, it has been suggested (Mourad & Bertrand-

Krajewski 2002) that at least two redundant sensors should be used. Physical redundancy-based methods are quite accurate, but are of limited utility for environmental sensors for two reasons. First, because many environmental systems of interest are quite large, many sensors are required to achieve spatially dense measurements and sensor cost may prohibit deployment of redundant sensors. This is further exacerbated by the expense of certain sensors. For example, one sensor deployed in the WATERS Network Corpus Christi Bay testbed, by the Shoreline Environmental Research Facility (SERF), the acoustic Doppler current profiler (ADCP), has a unit cost of around \$10,000. Second, environmental sensors are often deployed in areas with limited power supplies, so powering redundant sensors may not be possible. For example, SERF operates several offshore sensor platforms, which must generate all the energy used by the attached sensors from either wind turbines or solar arrays.

Analytical redundancy-based methods remove the burden of operating redundant sensors. Instead, a model of the sensor data stream is used to simulate a redundant sensor. The classification of a measurement as anomalous is based on the difference between the model prediction and the sensor measurement. Early work on such methods (e.g. Upadhyaya *et al.* 1990, Belle *et al.* 1983) employed statistical time series models (Box & Jenkins 1970) to predict the next measurement in the sensor data stream, using historical values. More recently, other modeling approaches, such as artificial neural networks (ANNs), have been suggested (Nairac *et al.* 1999, Fantoni & Mazzola 1996, Silvestri *et al.* 1994). These methods were designed for use in manufacturing/power plants, and the only researchers to have suggested a method of threshold selection (Belle *et al.* 1983) for

classifying data as anomalous/non-anomalous required detailed process knowledge that is not generally available for natural systems. Krajewski and Krajewski (1989) present an analytical redundancy method for streamflow data that employs model error standard deviations to set the threshold. This method, however, relies on a physically-based real-time model of the natural system—a tool which may not always be readily available.

Bayesian filtering approaches operate similarly to analytical redundancy methods except that they use Bayesian filtering to determine the likelihood of a particular measurement, given all previous measurements in the sensor data stream. Filtering-based error detection has been used most widely in robotics for detecting errors in onboard sensors (Nicholson & Brady 1994, Goel *et al.* 2000), though Lerner *et al.* (2000) also demonstrates a filtering error detection method using a hypothetical process control case study.

Rule-based approaches use knowledge regarding correct sensor operation and likely data sequences to indicate data that may be erroneous. Sensor-related knowledge includes detection limits of the sensor, known failure responses (e.g. unavailable measurements), and maintenance intervals. Data sequence knowledge includes process variable range (e.g. jet engine speeds vary from 0 rpm to a defined maximum rpm), process variable gradient (e.g. due to rotational mass, jet engines cannot accelerate/decelerate faster than a particular rate), and correlated data (e.g. fuel consumption increases with engine speed). From knowledge such as this, rules about properly operating sensors can be made; thus, data errors can be determined by citing violations of these rules. For example,

association rule-based decision making (Depold *et al.* 2003) and fuzzy sets (Ananthanarayanan & Holbert 2004) have been presented as methods for detecting erroneous data from sensors in electric power generation plants, and Bayesian belief networks (Mehranbod 2003) have been presented as methods for detecting errors in chemical manufacturing sensor data. In addition to being able to identify erroneous measurements, these methods are often able to determine the cause of the erroneous measurements and sometimes suggest remedial actions (Ramanathan *et al.* 2006). However, if rules cannot be made regarding correct sensor or process behavior, these methods are infeasible. Furthermore, since no model of the processes being measured is used, estimates of the erroneous data values cannot be suggested.

Due to the various strengths and weaknesses of the different categories of methods described above, some researchers have suggested approaches that combine more than one category of method. Bertrand-Krajewski *et al.* (2003) and Hodge and Austin (2004) provide good surveys of these approaches.

Many of the techniques described above were designed to operate in controlled environments, such as manufacturing/power plants or machinery, where sensors are measuring process variables that are likely to be within a predefined range. In this research, however, the sensors are located in the natural environment, measuring very unpredictable process variables for which a predefined range is not known *a priori*.

Chapter 5 develops a new analytical redundancy methodology for anomaly detection in environmental streaming data that employs a data-driven autoregressive model of each sensor data stream. For illustration, this methodology is applied, using several different data-driven modeling methods, to the case of identifying erroneous windspeed measurements caused by transmission faults within the SERF sensor network. The results from the different modeling methods are compared based on their ability to quickly and reliably identify erroneous measurements.

Chapter 6 develops two methods for anomaly detection via Bayesian filtering, which, unlike the method presented in Chapter 5 consider several data streams at once using all of the streams concurrently to perform coupled anomaly detection. These methods employ a dynamic Bayesian network (DBN) to model the environmental system being monitored by the environmental sensors which are solved using an appropriate filtering algorithm. For illustration, these methods are applied, using DBNs of varying complexity, to the case of identifying both synthetic anomalies and known anomalies in eight data streams from the SERF sensor network. The results from the different anomaly detection methods and DBNs are compared based on their ability to quickly and reliably identify the synthetic anomalies. The performance of the Bayesian filtering methods is also compared with the performance of the autoregressive anomaly detection method described in Chapter 5.

Chapter 4: Upscaling Models of Solute Transport in Porous Media through Genetic Programming

This case study develops an upscaling methodology using genetic programming (GP), a promising new tool for modeling complex phenomena whose physics are not well defined (Babovic & Abbott 1997a). For illustration, this methodology is applied to the case of developing vertically averaged models of the transport of a non-reactive solute in confined stratified aquifers. The results are compared with models developed through the method of moments (MoM), a traditional upscaling technique that is well suited for this transport configuration (Güven *et al.* 1984).

4.1 Methods

The upscaling methodology developed in this study takes advantage of GP's ability to model complex phenomena. This section includes a description of how GP was performed, followed by the mathematical formulation of the upscaling problem addressed in this study.

4.1.1 Genetic Programming

As discussed in Chapter 2, GP is a domain independent method that creates a model based on input data, by searching the space of possible models. This search uses operations inspired by natural evolution, which allow GP to cultivate a diverse set of approaches to solving the problem (Banzhaf *et al.* 1998). Genetic programming has shown success in many applications (e.g. Koza *et al.* 1999, Savic *et al.* 1999). Babovic

and Abbott (1997b) present four applications of GP in the field of hydrology. The results of these applications illustrate the abilities of GP to: (1) model “emergent phenomena,” (2) find models of data that match human derived models, (3) develop models of phenomena that are of higher quality than human derived models, and (4) find models of complex phenomena that are equally accurate, yet simpler to solve, than many human derived models.

In this research GP will be used to perform symbolic regression to induce upscaled models of solute transport in porous media from data describing the upscaled process. Building differential equations via symbolic regression is difficult, because no general differential equation solver exists to evaluate the fitness of the candidate equations. Thus, it is important to find a method of learning differential equations without requiring integration of each candidate differential equation in the population. In this research, the upscaling problem is decomposed into a new problem that does not require the use of calculus to evaluate the objectives, as described in the next section.

4.1.2 Mathematical Formulation

Because data regarding the target phenomenon is presented to ALP as a list of examples containing several descriptive attributes and the observed response of the system, and because it is necessary for the resulting upscaled models to be easily implemented, in this study, the upscaling problem was reduced to a problem of calculating upscaled solute fluxes. The mathematical formulation starts with the ADE, as it is assumed that this

model is valid for continuum-scale solute transport. Using the summation convention for repeated indices, the ADE equation can be expressed as:

$$\frac{\partial C}{\partial t} = -\frac{\partial}{\partial x_i}(u_i C) + \frac{\partial}{\partial x_i} D_{i,j} \frac{\partial C}{\partial x_j} \quad (4.1)$$

where C is the continuum-scale solute concentration, t is the time, x_i is the Cartesian position vector, u_i is the pore water velocity vector, and $D_{i,j}$ is the dispersion tensor. This equation can also be expressed in terms of fluxes as:

$$\frac{\partial C}{\partial t} = -\frac{\partial J_i^A}{\partial x_i} + \frac{\partial J_i^D}{\partial x_i} = -\frac{\partial J_i^T}{\partial x_i} \quad (4.2)$$

where J^A is the solute flux due to continuum-scale advection, J^D is the solute flux due to continuum-scale dispersion, and J^T is the total continuum-scale solute flux. By employing spatial filtering, as shown by Beckie (1998), Equation 4.2 can be upscaled to the block-scale equation:

$$\frac{\partial \bar{C}}{\partial t} = -\frac{\partial \bar{J}_i^T}{\partial x_i} \quad (4.3)$$

where the overbar indicates a spatially filtered quantity equivalent to the convolution integral of the continuum-scale quantity multiplied by a filtering function. Since volume averaging is a form of spatial filtering, the filtered terms can be thought of as block-scale averages (Nitsche & Brenner 1989, Beckie *et al.* 1996). With some algebra, the block-scale total solute flux can be divided such that:

$$\bar{J}_i^T = \bar{J}_i^A + \bar{J}_i^{NA} \quad (4.4)$$

where $\bar{J}^A = \bar{u}_i \bar{C}$ is the solute flux due to block-scale advection, and \bar{J}^{NA} is the remaining non-advective solute flux. A similar decomposition of the block averaged flux was used by Efendiev *et al.* (2000). Unlike the block-scale advective flux, this latter term contains

sub-grid closure quantities, and thus cannot be easily modeled at the block-scale.

However, by collecting data regarding the block-scale non-advective flux, ALP can be used to develop models of this flux in terms of other block-scale parameters, which will allow the solution of Equation 4.3. Thus, the upscaling problem is reduced to the problem of finding a model for the block-scale non-advective flux in terms of resolvable block-scale quantities, and ALP does not have to search the space of PDEs.

Data describing the block-scale non-advective flux can be generated using two numerical grids: (1) a highly resolved grid and (2) a coarse grid representing the block-scale. The ADE is solved numerically on the fine grid, while the coarse grid is used for evaluating block-averaged parameters throughout the simulation. Block-scale parameters for each grid location are calculated by averaging the fine-scale parameters over the entire block, or in the case of vector quantities, such as the non-advective flux, over the appropriate block surface for each vector component.

4.2 Case Study

This case study presents the development of vertically averaged models of the transport of non-reactive solutes in two-dimensional, confined, perfectly stratified aquifers (i.e. vertically varying horizontal flow parallel to the layers). This idealized transport system was selected to allow comparison of the GP-derived vertically averaged equations with those derived using the method of moments (MoM) (Taylor 1953, Aris 1956, Güven *et al.* 1984, Kitanidis 1992), a well accepted approach for deriving vertically averaged equations for perfectly stratified aquifers. This section will proceed by first describing

the upscaled transport equations that can be derived using the MoM. Then, two test cases using different synthetically generated velocity fields will be defined. Next, mathematical simplifications to Equations 4.3 and 4.4, which are made possible by vertically averaging two-dimensional, confined, perfectly stratified aquifers, will be discussed. Finally, the generation of input data for ALP will be explained.

4.2.1 Method of Moments

The MoM aims to describe the solute plume at any point in time, in terms of its spatial moments. Mathematical expressions for the solute distribution's moments as functions of time can be derived from the ADE. For the case of transport in a laminar shear flow (equivalent to the case of horizontal flow in a perfectly stratified aquifer), Aris (1956) demonstrated that the MoM could be used to derive models for the temporal evolution of the spatial moments of the cross-sectionally averaged concentration. Commonly, only the zeroth, first, and second spatial moments are considered, as the models for the higher order moments are more cumbersome. The zeroth moment indicates the total solute mass in the system, the first moment indicates the mean position of the plume, and the second moment indicates the plume spread. The resulting upscaled model of transport has the same form as the ADE, except that an effective velocity vector calculated from the first moment replaces the velocity vector, and the dispersion tensor is replaced with a time dependent macrodispersion tensor calculated via the second moment. Thus, the MoM model of the non-advective flux can be expressed as:

$$\bar{J}_i^{NA} = D_i^{eff}(t) \frac{\partial \bar{C}}{\partial x_i} \quad (4.5)$$

where $D_i^{eff}(t)$ is the macrodispersion coefficient. It can be seen that the assumption of locally Fickian macrodispersion (i.e. the assumption that plume spreading due to hydraulic conductivity variability can be effectively modeled as a random process) is inherent in this model. The resulting MoM model describes the solute distribution at any time as Gaussian, with the same mean and variance as the observed plume.

Because this type of model employs the assumption of locally Fickian macrodispersion, it is only valid when the plume has spread sufficiently, such that all velocities are sampled with the same frequency with which they appear in space. Furthermore, this method requires assumptions regarding the continuum-scale velocity field in order to close the equations for the spatial moments of the solute distribution. One common assumption is that of a periodic medium (Kitanidis 1992, Wood *et al.* 2003); in particular, Aris (1956) showed that for confined, perfectly stratified aquifers with flow parallel to the layers, such as those considered in this study, it is possible to rigorously derive the effective velocity and macrodispersion terms.

4.2.2 Synthetic Aquifers

The process of defining the properties of the synthetic aquifers for this study was guided by the desire to facilitate comparison with the MoM and to maximize the generalizability of the aquifers. Assuming a two-dimensional system where flow is parallel to the x-axis, and the z-axis represents the aquifer depth, any arbitrary velocity profile can be discretized into small sub-layers of constant velocity. Because of this, and because this study is confined to vertically averaged blocks, a two-dimensional computational grid is

necessary to fully resolve the fine scale concentration distribution, whereas the aquifer's vertically averaged counterpart can be resolved with a one-dimensional computational grid. The two-dimensional representation will hereafter be referred to as the fine-scale representation. The fine and block-scale representations are illustrated in Figure 4.1. In order to facilitate comparison with the MoM, the following two velocity distributions were considered:

$$u(z) = h^2 \left(1 - \frac{z^2}{h^2} \right) \quad (4.6a)$$

$$u(z) = 0.5 \cos \left(2\pi \frac{z}{h} \right) + 0.5 \cos \left(4\pi \frac{z}{h} \right) + 1 \quad (4.6b)$$

where h is the aquifer depth. The flow distributions described by Equations 4.6a and 4.6b will hereafter be referred to as parabolic and cos-cos, respectively. Two distributions were chosen in order to demonstrate the generality of the derived upscaled models to solute transported by different velocity distributions. The parabolic distribution was selected because many MoM studies address transport by this distribution (e.g. Aris 1956, Güven *et al.* 1984), while the cos-cos distribution was selected because it varies more sharply than the parabolic distribution. These velocity distributions were applied by creating 100-layer synthetic aquifers and defining the flow rate in each layer, such that the total mass of water passing through the layer was equivalent to the total mass of water that would pass through the same discretized region using Equations 4.6a or 4.6b. The number of layers for the aquifers was selected in order to minimize the differences between the discretized distribution of the aquifers and the continuous distribution functions, thus facilitating comparisons with the MoM upscaled equations, since these equations use continuous velocity distributions. The depth of the aquifers was chosen to

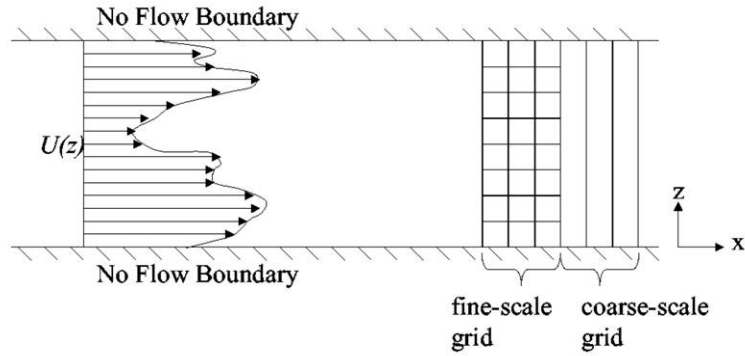


Figure 4.1: Schematic of a two-dimensional perfectly stratified aquifer indicating both the fine and block-scale computational grids, as well as the fine-scale velocity distribution.

be 1 m. The transverse dispersion coefficient within the aquifers was specified to be 0.01 m/s^2 . Since it has been shown that the longitudinal spreading of the plume due to aquifer heterogeneity is significantly larger than that due to continuum-scale dispersion (Gelhar *et al.* 1979), the latter was ignored. The exact values of these parameters, however, are unimportant, because, as will be discussed shortly, dimensionless parameters are used to describe the aquifers. Thus, the numerical results of the transport simulation can be scaled to represent a large number of aquifer geometries and transverse dispersion conditions.

4.2.3 Simplifying the Numerical Formulation

The use of vertically averaged representations of two-dimensional aquifers allows two simplifications to be made to Equations 4.3 and 4.4. First, the subscripts (i) can be dropped from the vector quantities (e.g. \bar{J}_i^{NA}) because the vectors have only one

component (i.e. x-directional). Thus, Equations 4.3 and 4.4 can be combined and simplified to:

$$\frac{\partial \bar{C}}{\partial t} = -\frac{\partial}{\partial x} \bar{u} \bar{C} - \frac{\partial \bar{J}^{NA}}{\partial x} \quad (4.7)$$

The second simplification involves converting from a Cartesian to a Lagrangian coordinate system that moves at the vertically averaged pore water velocity. In this coordinate system, the position vector is $x_R = x - \bar{u}t$, and Equation 4.7 becomes:

$$\frac{\partial \bar{C}}{\partial t} = -\frac{\partial \bar{J}^{NA}}{\partial x_R} \quad (4.8)$$

This simplification allows for convenient implementation of the upscaled solute transport model, because the block-scale advective flux is implicitly accounted for by the Lagrangian coordinate transformation. Therefore, the block-scale solute concentration can be calculated using Equation 4.8, where \bar{J}^{NA} is modeled by ALP.

Since dimensionless training data are used to implicitly constrain ALP to dimensionally consistent equations, the following transformations were used to convert dimensional data into dimensionless data:

$$\phi = \frac{C}{C_0} \quad (4.9a)$$

$$\tau = \frac{tD_T}{h^2} \quad (4.9b)$$

$$\xi = \frac{x - \bar{u}t}{h} \quad (4.9c)$$

$$\psi^{NA} = J^{NA} \frac{h}{D_T C_0} \quad (4.9d)$$

where C_0 is the input concentration, and D_T is the transverse dispersion coefficient.

4.2.4 Generation of Training Data

In order for ALP to develop models of the block-scale non-advective flux, it is necessary to provide a set of training examples that contain block-scale descriptive attributes and the resulting block-scale non-advective flux observed in the aquifer being studied.

Training examples were collected from the aquifer with the parabolic flow distribution for a pulse input of solute. The time evolution of the continuum-scale solute distribution was solved using a numerical finite difference solution to the ADE, which employed operator splitting to separate the modeling of the advective and dispersive processes. A third order explicit total variation diminishing (TVD) method (Leonard 1988) was used to solve the advection term, while an implicit method was used for the dispersion term. This method for solving the ADE was selected to minimize prediction errors of the numerical solution to the ADE and, thus, to minimize errors in the training data. The fine-scale solution used a highly refined computational grid to minimize errors in the training data. The fine-scale solution used a highly refined computational grid to minimize numerical errors. The block-scale computational grid was defined such that each block spanned the entire depth of the aquifer and had the same length as the fine-scale grid.

The descriptive attributes selected to describe the block-scale non-advective flux at the block interface consisted of the block-scale concentration ($\bar{\phi}$) at the upstream and downstream block centers; the position of observed block interface (ξ); the time of

observation (τ); the block-scale concentration gradient ($d\bar{\phi}/d\xi$) at the block interface, as well as at three upstream locations; the second spatial derivative of the block-scale concentration ($d^2\bar{\phi}/d\xi^2$) at the block interface and at three upstream locations; and the mixed space/time derivative of the block-scale concentration ($d^2\bar{\phi}/d\xi d\tau$) at the block interface and at three upstream locations. The values of the block-scale derivatives were estimated using discrete approximations. Unless otherwise indicated, these attributes were recorded at the block interfaces. These attributes were selected because they appeared in upscaled transport equations derived using traditional methods (Gelhar & Axness 1983, Beckie 1998, Efendiev *et al.* 2000). Because there are many block interfaces and time steps in the numerical simulation of the fine-scale transport model, it was necessary to select only a subset of the salient examples for training. Examples from a particular block interface were considered salient only if the solute concentration at some point within a five-block neighborhood surrounding the interface was non-zero. The examples used to train ALP were selected at random from the set of salient examples that were recorded during every tenth time step of the simulation.

4.2.5 Parameterization of ALP

ALP requires many user-selected parameters to define the search for good models of the training examples, including objective, functional set, population size, and number of training epochs. Unfortunately, there is little theoretical work to direct the selection of these parameters, and thus, a large number of experiments were performed with different values in order to find good solutions.

The objective parameter defines the criteria by which to evaluate the quality of the derived equations for predicting the target attribute. ALP implements several types of objectives, including both goodness-of-fit (e.g. sum of squared errors) and parsimony (e.g. equation length) objectives; furthermore, ALP permits multi-objective searches. In this study, the objectives were selected to be the correlation coefficient (r^2) between the candidate equation and the training data and the equation length. The r^2 statistic indicates the degree to which the relationship between two variables is linear; thus, it is insensitive to relational constants, such as scale or shift (Devore 1995). The r^2 statistic was selected for this latter property because it does not require ALP to find the correct value of the relational constants, a task that is generally difficult for GP (Koza 1992). Scale constants refer to constants that are multiplied to or divided from a function, while shift constants refer to constants that are added to or subtracted from a function. When using the r^2 objective, it is necessary to determine both the scale and shift constants *a posteriori*. In this research, the scale and shift parameters were determined by performing linear regression of the data pairs composed from the observed non-advective flux in the training data and the model prediction of this flux. The slope and y-intercept of this line are the scale and shift parameters, respectively. The equation length was selected to encourage ALP to explore equations of varying complexity and to control bloat. Because ALP uses a multi-objective search, controlling bloat in this manner will not eliminate clauses that improve the model's r^2 value.

The functional set defines the mathematical operators that can be used to relate the attributes to the target value. Several functional sets were evaluated, including the set of all arithmetic operators (e.g. $x+y$, where x and y are attributes), the set of all arithmetic and geometric operators (e.g. $\sin(x)$, where x is an attribute), and the set of all arithmetic operators and the exponential function (i.e. e^x , where x is an attribute). It was observed that the latter two functional sets found many large equations that fit the data well and consisted of long chains of sine/cosine terms or exponential functions, respectively. It is well known that any continuous function can be represented by an infinite series of sine/cosine or exponential terms through the formation of Fourier or Taylor series. Because the exponential operator is easier to simplify than the set of geometric operators, the functional set including both arithmetic operators and the exponential function was used in order to retain the expressivity facilitated by exponential functions without overwhelming the GP results with difficult to analyze solutions.

The population size specifies how many candidate equations participate in the search for good equations. Larger populations contain a greater variety of clause building blocks from which to derive new candidate equations; however, larger populations also increase the time it takes to evaluate one epoch of GP. Therefore, it is necessary to have a population that is large enough to represent an adequate number of discrete clauses, yet small enough to allow a reasonable computation time. According to the guideline presented by Sastry *et al.* (2003) the population would have to contain over 11 million individuals in order to guarantee a good supply of building blocks in this research. However, preliminary results indicated that a population size of 1000 individuals was

sufficient to produce good results. For this reason, a population size of 1000 candidate equations was chosen.

The number of epochs specifies how many iterations of the genetic operations the population of candidate equations is subjected to. Langdon and Poli (2002) showed that more epochs result in a larger number of extraneous clauses in each candidate equation in the final epoch. Therefore, it is common to use only a few training epochs but perform GP many times (Koza 1992). In this paper, each run of ALP will be referred to as an experiment. After all the experiments have been completed, the results from each experiment are merged, eliminating all the results that are dominated by results from different experiments, resulting in a “front” of non-dominated (and thus Pareto optimal) equations for modeling the training data. The candidate equations are then evaluated for semantic meaning, as well as for goodness of fit. In this study, hundreds of experiments were performed, during which the candidates were evolved for 50 epochs.

In addition to these parameters, a crossover rate of 0.8, a mutation rate of 0.1, and binary tournament selection were used. These values reflect those recommended by the developers of the ALP system from extensive trials on many different functions (e.g. Keijzer *et al.* 2001, Babovic *et al.* 2001, Keijzer 2002, Keijzer & Cattolico 2002).

4.3 Results

The results from many experiments of ALP compose a Pareto front of non-dominated solutions to the GP task. In this case, the front consisted of many equations with nearly

equivalent r^2 values but widely varying lengths. In general, longer equations tended to have slightly higher r^2 values. Analysis of these models, however, showed that the majority of the equations along the front contained the same clause, along with many irrelevant or nearly-irrelevant clauses that could be removed without significantly reducing the ability of the equations to fit the training data. These extraneous clauses were considered to be the result of GP bloat and, thus, were removed from the equations, resulting in a consensus on a final model for the data.

Three characteristic results from along the Pareto front were:

$$\bar{\psi}_i^{NA} = \bar{\phi}_{i-\Delta\xi/2} \frac{\xi}{\tau} \quad (4.10a)$$

$$\bar{\psi}_i^{NA} = \exp\left(\exp\left(\bar{\phi}_{i-\Delta\xi/2} \frac{\xi}{\tau}\right)\right) \quad (4.10b)$$

$$\bar{\psi}_i^{NA} = \exp\left(\exp\left(0.85 \frac{\partial \bar{\phi}}{\partial \xi}\right)\right) - \frac{\left(\left(\frac{\bar{\phi}_{i-\Delta\xi/2} + \bar{\phi}_{i+\Delta\xi/2}}{2}\right) \frac{\xi}{\tau} - \exp\left(0.85 \frac{\partial \bar{\phi}}{\partial \xi}\right)\right)}{\exp(\exp(\bar{\phi}_{i+\Delta\xi/2}))} \quad (4.10c)$$

where the subscript i indicates the i^{th} block interface. These results fit the training data with r^2 values of 0.95, 0.97, and 0.95, respectively; thus, the models are of similar quality, but they differ greatly with regard to semantics and complexity. Equation 4.10a only contains three parameters, all of which contribute significantly to its quality. However, it is interesting that the model only includes the dimensionless concentration upstream of the block face. If this parameter is replaced with the dimensionless concentration at the block face (calculated as the average of the concentration upstream and downstream of the block face), the resulting equation becomes:

$$\bar{\psi}_i^{NA} = \frac{\bar{\phi}_{i-\Delta\xi/2} + \bar{\phi}_{i+\Delta\xi/2}}{2} \frac{\xi}{\tau} \quad (4.11)$$

which also has an r^2 value of 0.95. Thus, the replacement of the upstream concentration with the block face centered concentration does not improve (or reduce) the quality of the model's fit with the non-advective flux data, but it does improve the performance of this model for prediction of the block-scale solute distribution. The improvement in the solute distribution prediction occurs because, at the downstream edge of a solute plume, the model shown in Equation 4.10a will predict zero solute flux, whereas the model shown in Equation 4.11 will predict a finite non-advective solute flux, the latter case being the physically plausible model response. This discrepancy in flux prediction between the two models occurs because, at the downstream edge of the plume (i.e. at position i), the concentration a bit further downstream (i.e. at position $i + \Delta\xi/2$) is zero, whereas the concentration just upstream (i.e. at position $i - \Delta\xi/2$) is non-zero.

Equation 4.10b can also be reduced to Equation 4.11. Recall that the exponential function is equivalent to the Maclaurin series:

$$\exp(a) = \sum_{n=0}^{\infty} \frac{a^n}{n!} \quad (4.12)$$

Since the magnitude of the product within the exponentials in Equation 4.10b is always less than 1, the terms in the series get smaller as $n \rightarrow \infty$; therefore, all but the first two terms of the series can be ignored. If this procedure is followed for both exponential functions, the resulting model is a linear function of Equation 4.10a. Since the r^2 statistic is insensitive to scale and shift parameters, the r^2 value of the approximation to Equation 4.10b is equal to that of Equation 4.10a: namely, 0.95. In return for the reduction in

performance caused by this approximation, there is a substantial increase in both semantic meaning and ease of implementation of the model.

Equation 4.10c can also be reduced to Equation 4.11 through evaluation of its clauses.

Equation 4.10c can be divided into four clauses such that:

$$\bar{\psi}_i^{NA} = clause_1 - (clause_2 - clause_3) / clause_4 \quad (4.13)$$

Using the training data, the minimum, mean, and maximum values of $clause_1$ can be calculated to be 2.705, 2.718, and 2.720, respectively. Because the clause has a small range, it can be replaced by its mean with little loss of generality. The same is true for $clause_3$ and $clause_4$. Since $clause_2$ is Equation 4.11, it can be seen that the approximation of Equation 4.10c is a linear function of Equation 4.11; thus, the r^2 value is 0.95, which is equivalent to the r^2 value of Equation 4.10c. Therefore, there is no measurable predictive ability lost by using Equation 4.11 to approximate Equation 4.10c.

In the preceding discussion, it was demonstrated that different length models from the set of Pareto optimal solutions could be simplified to the same model without a significant loss of predictive ability. However, if the longer models have a higher r^2 value, why were they not preferred? The answer is twofold. First, there is a precedent in learning theory to prefer simpler models to more complex models with similar predictive abilities (i.e. Occam's razor) (Duda *et al.* 2001). Second, the longer models are often too complex to be implemented numerically. Furthermore, a t-test with a 95% significance level showed that the difference in r^2 values between the longer models and Equation 4.11 is insignificant. Note that due to space constraints, only a few of the shorter equations were

discussed; the same techniques can be applied to the longer equations along the Pareto front, often resulting in Equation 4.11. This consensus between models strengthens the claim that Equation 4.11 best models the non-advective flux.

It should now be clear that many of the Pareto-optimal results of the GP task can be reduced to one common equation shown in Equation 4.11. This model will hereafter be referred to as the sub-grid advective (SGA) model for reasons that will become clear shortly. The SGA model is strongly correlated with the non-advective solute flux from a pulse input of solute in both the aquifer with the parabolic velocity distribution and the aquifer with the cos-cos velocity distribution, with r^2 values of 0.95 and 0.93, respectively, whereas the MoM model (Equation 4.5) is only weakly correlated with the observed non-advective flux, indicated by r^2 values for the parabolic and cos-cos velocity distributions of 0.1 and 0.24, respectively. This result enables two conclusions. First, since flux data from the cos-cos velocity distribution was not used for training, this result indicates that the SGA model generalizes to different flow conditions than those used for training. This generality suggests that the SGA model describes the mechanism of macrodispersion, rather than merely being a concise representation of the training data. Second, because a strong correlation exists between the SGA model and the observed non-advective flux, but not between the MoM model and the observed non-advective flux, this result indicates that the SGA model is a better predictor of the observed non-advective flux than the MoM model. This result may appear surprising because the MoM model should be correct at late times, when the assumption of Fickian macrodispersion is valid. However, the r^2 metric considers the model residuals holistically with respect to

time, and the model residuals are more likely to be large in magnitude at early times than at later times because the magnitude of the non-advective flux is larger at early times. Therefore, the r^2 metric is biased towards early time behavior. Since r^2 was used as the goodness-of-fit metric, this latter conclusion suggests why ALP did not create any models similar to the MoM model.

In order to determine the scale and shift constants, linear regression between the SGA model (converted back into dimensional form) and the observed non-advective flux was performed. This regression indicated approximate scale and shift parameters of 1 and 0, respectively, resulting in the equation:

$$\bar{J}^{NA} = \bar{C} \frac{x_R}{t} \quad (4.14)$$

where x_R is the position in the Lagrangian coordinate system. This model describes the non-advective flux of solute in the aquifer with the parabolic flow profile. The zero value of the shift parameter is expected, because a non-zero shift parameter would indicate that a significant component of the non-advective flux could not be modeled by the SGA model. Equations 4.8 and 4.14 can be solved numerically to predict the time evolution of a pulse input of solute in the aquifer with the parabolic flow profile at times greater than zero (since time appears in the denominator).

Figure 4.2 compares the performance of the SGA upscaled model with the MoM upscaled model for predicting the evolution of the solute plume. Because the SGA model is not valid at very early times, the fine-scale model was used to predict the plume evolution for the first 30 time steps of the simulation (until $\tau = 0.03$), before the SGA and

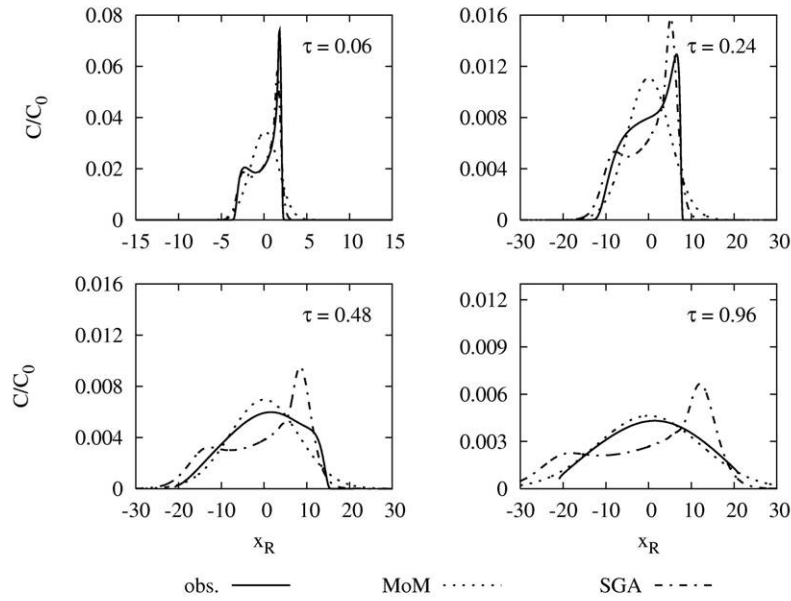


Figure 4.2: Comparison of the MoM and SGA upscaled models for predicting the vertically averaged time evolution of a pulse input in the synthetic aquifer with parabolic flow.

MoM models took over the prediction. It can be seen that, at early times, the SGA model more closely approximates the plume shape, whereas at later times, the MoM model produces a more accurate result. However, it is important to note the y-axis scale when comparing the two predictions, because the maximum absolute error between the MoM model and the fine-scale model is much larger than the maximum absolute error between the SGA model and the fine-scale model, as depicted in case 0 of Figure 4.3. This latter result may be misleading because the MoM model was derived such that the error between the predicted and observed values of the first two spatial moments of the plume is minimized, while the SGA model (Equation 4.11) was developed with the goal of minimizing errors. Thus, a comparison that invokes absolute errors will be biased towards the SGA model, while a comparison based on moments will be biased towards

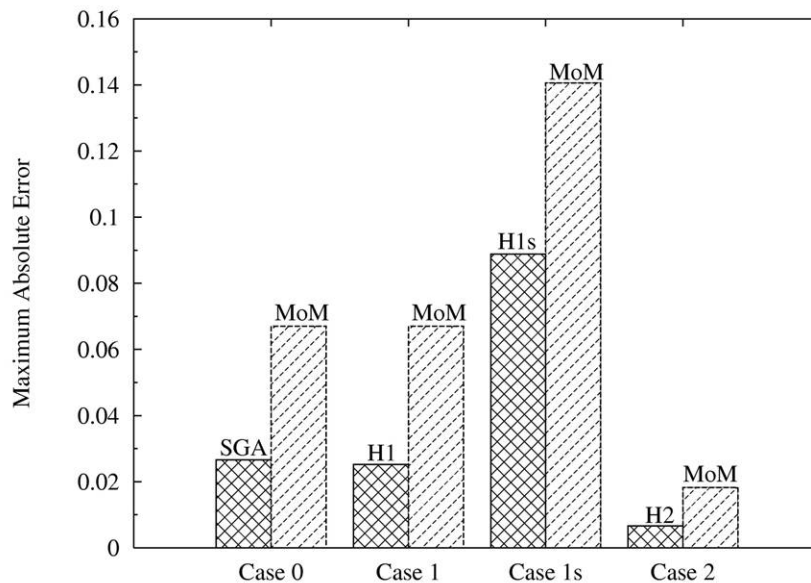


Figure 4.3: Comparison of the maximum absolute errors between ALP derived upscaled models (SGA, H1, H1s, and H2) and the MoM upscaled model. Cases 0 and 1 refer to the transport of a pulse input in the synthetic aquifer with parabolic flow, case 1s refers to the transport of a finite width input in the synthetic aquifer with parabolic flow, and case 2 refers to the transport of a pulse input in the synthetic aquifer with cos-cos flow.

the MoM model. However, a comparison of the time evolution of the first two moments, calculated by the SGA and MoM models, to the first two moments of the plume, calculated using the ADE, indicates that both the SGA and MoM capture the time evolution of the zeroth, first and second spatial moments well with average errors of less than one-half percent. Therefore, in our numerical experiment, the two models perform equally well when compared via spatial moments.

Analyzing the second term in the SGA illustrates that the block-scale non-advective flux can be attributed to solute advection that occurs below the block-scale:

$$\frac{x_R}{t} = \frac{x - \bar{u}t}{t} = u - \bar{u} = u' \quad (4.15)$$

Equation 4.15 relates the position of the solute to u' , the deviation (from the mean velocity) of the unresolved velocity. Thus, the SGA model approximates the solute transported by unresolved velocity variations using block-scale resolvable parameters. In fact, it can be shown that the SGA model is quite capable of reproducing the solute plume evolution of a pulse input in a perfectly stratified system with no transverse mixing.

Figure 4.2 indicates that, at early times, the time evolution of the solute distribution in the aquifer appears purely advective, and that the time evolution of the plume at late times is well described by the MoM model. However, at intermediate times, the solute plume behaves in a manner consistent with a combination of the pure advection and Fickian macrodispersion cases, where the influence of the SGA model decreases with time, and the influence of the MoM model increases with time. Thus, a new model, which is a hybrid of the SGA and MoM models, is suggested:

$$\bar{J}^{NA} = F(t) * \underbrace{\left(D_{\infty}^{eff} \frac{\partial \bar{C}}{\partial x} \right)}_{MoM \ model} + (1 - F(t)) * \underbrace{\left(\bar{C} \frac{x_R}{t} \right)}_{SGA \ model} \quad (4.16)$$

where $F(t)$ is a continuous function over all values of t and has a minimum value of zero that occurs at $t=0$ and a maximum value of 1 that occurs at very late time, and D_{∞}^{eff} is the asymptotic coefficient of macrodispersion suggested by the MoM. The function $F(t)$, which will hereafter be referred to as the mixing function, controls the influence of both

the SGA and MoM models over time, allowing the SGA model to dominate the behavior of the solute distribution at early times and allowing the MoM model to dominate its behavior at later times. This model is consistent with a conceptual model of the transport process in which, at early times, insufficient solute has been exchanged between the layers, such that the process is similar to the pure advection process described by the SGA model. As a larger quantity of solute samples more of the flow paths in the individual layers, a larger fraction of the transport process behaves in a manner consistent with Fickian macrodispersion; once sufficient time has passed for the average solute behavior to be consistent with having sampled all the flow paths, the process is well described by Fickian macrodispersion. Equations 4.7 and 4.16 can be solved to predict the time evolution of a pulse input of solute in an aquifer at times greater than zero.

In the discussion above, the mixing function $F(t)$ was intentionally vaguely defined because the optimal function may vary depending on transport conditions. In this research, a sigmoid function:

$$F(t) = \left[1 + \exp\left(-\left(t \frac{D_t}{h^2} - a\right)/b\right) \right]^{-1} \quad (4.17)$$

was chosen to demonstrate how the model described by Equation 4.16 performs on the aquifers considered in this study. Note that in this equation, time is normalized by the transverse dispersivity and aquifer depth, resulting in an equation that generalizes to other aquifers with similar flow distributions but different depths and transverse dispersion coefficients. The parameters for the sigmoid were chosen via a manual trial-and-error approach using visual inspection of the solute distribution shapes to guide the search.

Nevertheless, the upscaled models created performed well on a variety of transport conditions, including different initial conditions of the solute input into the aquifer with parabolic flow and different velocity distributions. Since, as described previously, the SGA model is not valid at very early times, for the experiments described below, the fine-scale model was used to predict the plume evolution for the first 30 time steps ($\tau \leq 0.03$) of the simulation, after which the hybrid and MoM models took over the prediction.

For the case of a pulse input into the aquifer with parabolic flow (which is equivalent to ALP's training conditions), values of 0.3 and 0.01 were chosen for the sigmoid function parameters, a and b , respectively. It can be seen in Figure 4.4 that the hybrid model with these parameter values, hereafter referred to as H1, performs better when compared to the MoM model. At early times, model H1 preserves the favorable behavior of the SGA model, predicting the bimodal solute distribution with high accuracy. At late times, model H1 retains the benefits of the MoM model, predicting a unimodal, nearly Gaussian plume. At intermediate times, model H1 performs quite well at capturing the peak concentration and shape of plume's leading edge. It also does quite well at capturing the overall shape when compared to the MoM model. Furthermore, the maximum absolute concentration deviation between model H1's prediction and the observed plume shape is less than that of either the SGA or MoM models alone, as illustrated in Figure 4.3.

The predictive abilities of the upscaled model H1, however, are not restricted to the conditions on which it was developed. Using superposition, this model can be extended to the cases of an instantaneous finite width input, a finite duration input, or even a

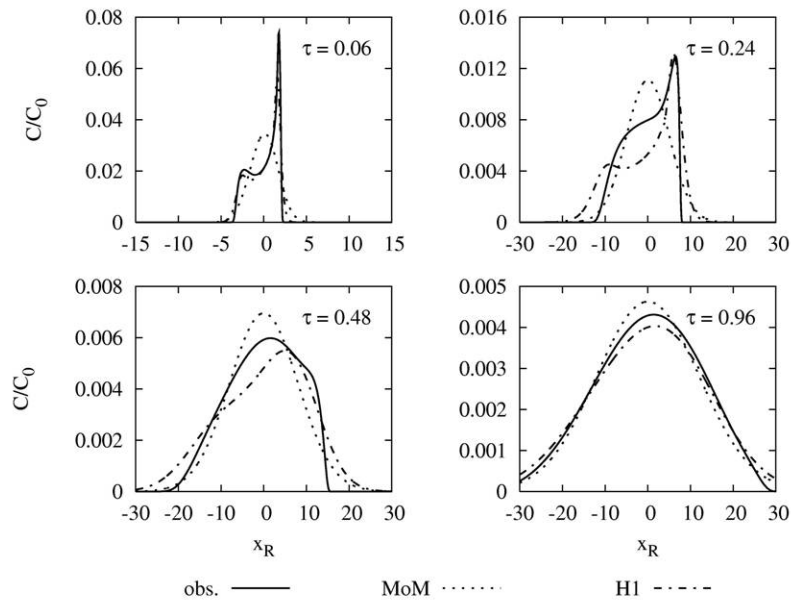


Figure 4.4: Comparison of the MoM and H1 upscaled models for predicting the vertically averaged time evolution of a pulse input in the synthetic aquifer with parabolic flow.

continuous input, by summing the effects of multiple pulse inputs each calculated with model H1. For example, Figure 4.5 shows the superior performance of the superposed H1 model, hereafter referred to as H1s, for the case of an instantaneous input over a finite width of the aquifer with parabolic flow. In particular, model H1s more accurately predicts the shape of the plume's leading edge, as well as the peak concentration location, than the MoM model at all times. Additionally, as shown in Figure 4.3, the maximum absolute error between model H1s and the observed plume shape is smaller than that of the MoM model alone. Thus, though model H1 was developed for particular initial conditions, it generalizes well to other input conditions.

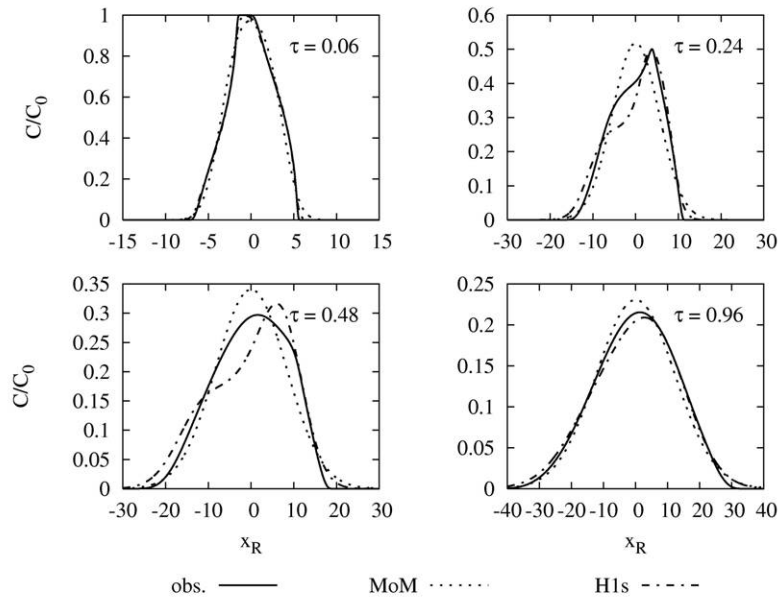


Figure 4.5: Comparison of the MoM and H1s upscaled models in predicting the vertically averaged time evolution of an instantaneous finite width input of solute in the synthetic aquifer with parabolic flow.

Furthermore, the hybrid model can be adapted for use in the aquifer with cos-cos flow simply by changing the parameters a and b of the sigmoidal mixing function to 0.5 and 0.0875, respectively. These parameters were found using another trial-and-error fit. A comparison of the performance of this adapted hybrid model, hereafter referred to as H2, with the MoM model, for predicting the time evolution of an instantaneous pulse input into the aquifer with cos-cos flow is shown in Figure 4.6. Even for velocity distributions on which the SGA model was not developed, the hybrid model outperforms the MoM model in predicting the plume shapes, especially in capturing the shape of the plume's leading edge, as well as in predicting the magnitude and location of the peak concentration, and in minimizing the maximum absolute error, as shown in Figure 4.3.

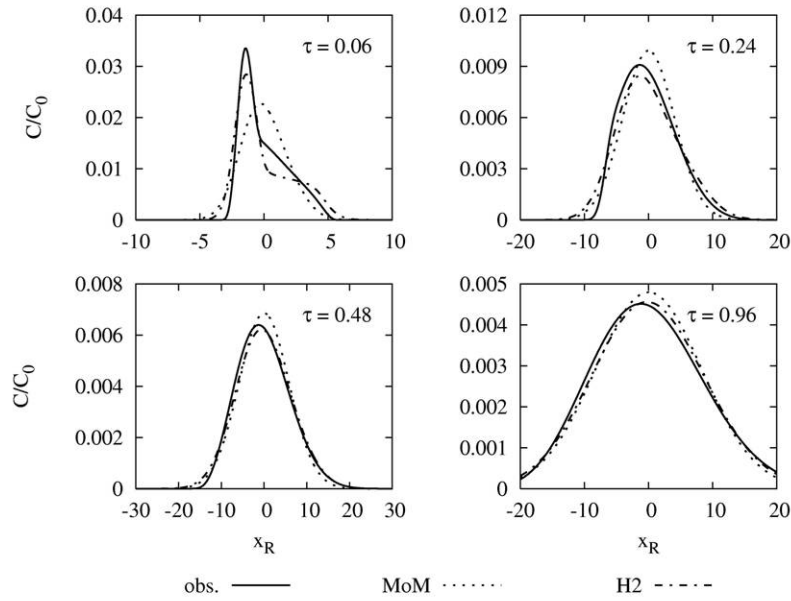


Figure 4.6: Comparison of the MoM and H2 upscaled models for predicting the vertically averaged time evolution of a pulse input in the synthetic aquifer with cos-cos flow.

The applicability of the general hybrid model to a range of initial conditions and velocity distributions suggests that this model does not serve simply as a surrogate for the observed data used to train it, but actually describes the processes that drive the solute's macrodispersion. For that reason, it can be suggested that the behavior of the macrodispersion transitions from a process that manifests itself as advective at the block-scale to a process that manifests itself as Fickian at the block-scale. Furthermore, since $F(t)$ had to be re-parameterized for model H2, but not model H1s, this experiment suggests that the behavior of this transition is a function of the velocity field, rather than of the initial solute distribution. This knowledge could be used to develop a relationship between the mixing function parameters and the velocity distribution, so that a trial-and-error fitting procedure would no longer be necessary.

4.4 Discussion

The case study presented in this chapter illustrates several benefits of using GP as a research tool. First, GP can not only be used to accurately model training data, but also to produce mathematical models that researchers can understand, unlike other data-driven approaches to modeling (e.g. neural networks). This representation facilitates the interpretation of model semantics, as was illustrated in the case study, when the ALP derived model was related to sub-grid advection. Furthermore, the representation of models as equations renders them capable of being modified to incorporate domain knowledge to improve applicability. This is especially beneficial in the case of ill-defined modeling tasks. For example, in the case study, ALP's objective was to find a model that fit the non-advective flux data well using the r^2 statistic. However, groundwater researchers evaluate the quality of upscaled solute transport models based on their ability to predict the time evolution of plumes. This latter objective is difficult to express mathematically and would require the solution of a PDE for each population member during each epoch. This would require both an automated numerical implementation scheme and a long time to evaluate each training epoch. Therefore, significant economy is realized by decomposing the problem and later reconstructing it from its constituent parts. The reconstruction process is facilitated by the mathematical form of the GP models, which allowed the combination of the SGA model with a model of Fickian macrodispersion to improve late-time performance. The mathematical representation of the GP derived models also facilitates the integration of these models into more complex modeling tasks. For example, ALP helped create a model of the

block-scale non-advective flux, which could then be integrated into a PDE describing the solute plume evolution. Finally, because researchers can interpret the mathematical equations produced by GP, these equations can be used to gain insight into the predominant processes that create the training data. For example, the case study shows how the GP search encouraged the development of a conceptual model of macrodispersion that transitions from a predominantly advective to a predominantly Fickian process.

The results of the case study also provide some insight into how to approach upscaling solute transport models to multi-dimensional blocks using GP. This task requires learning a model for a multi-dimensional vector quantity. Therefore, ensuring that mass continuity is conserved will be more difficult than in the one-dimensional case presented here, and new objectives may be necessary to guide the GP search towards methods that conserve mass. Furthermore, a method should be sought to reduce the observed bias of the r^2 metric for capturing early time behavior.

4.5 Conclusion

This case study presents promising initial results from a novel data-driven approach to upscaling solute transport models. A methodology was developed such that the problem of upscaling models of solute transport from the fine-scale to the block-scale was reduced to finding a model of the block-scale non-advective flux. To demonstrate this method, a case study was performed, in which vertically averaged models were developed for the transport of solute in perfectly stratified aquifers by flow parallel to the layers. The many

Pareto optimal equations found by ALP were analyzed to discover a consensus equation that described the advection of solute by fine-scale velocity variations from the vertically averaged velocity that could be expressed entirely in terms of block-scale parameters.

When this model was used to predict the time evolution of the solute distribution, the short-term predictions were of high quality, but this was not the case with the long-term predictions. This result may be due to a bias in the ALP fitness function toward capturing early time behavior. This model, however, was determined by the consensus of many searches to best capture the behavior of the non-advective flux, thus compelling the development of a new hybrid model of the non-advective flux that transitioned from an advective to a Fickian process. This new hybrid model was shown to be applicable to a variety of initial conditions and flow distributions, rather than merely the conditions used to train GP, suggesting that the new hybrid model describes the mechanism of macrodispersion, rather than simply being a surrogate for the training data.

Though the case study develops vertically-averaged models of solute transport under relatively simple flow conditions (i.e. 2-dimensional, steady state flow in a confined, perfectly stratified aquifer of infinite extent), the results presented in this study are promising. Data-driven modeling using GP is a novel approach to the upscaling problem, and to our knowledge, no previous studies exist in which data-driven modeling techniques have been used to develop semantically meaningful upscaled solute transport models. As demonstrated here, GP can be used as a tool to inspire researchers to develop novel solutions that may not be immediately obvious. The success of the hybrid model

for predicting the evolution of the solute plume indicates that the GP upscaling methodology may also be successful for modeling more complex systems. Furthermore, the results of this case study provide insight into how to approach more complex transport conditions, as well as multi-dimensional blocks.

Chapter 5: Real-Time Autoregressive Data-Driven Anomaly Detection in Streaming Environmental Data

Models of environmental systems, such as the upscaled model developed in Chapter 4, require many parameters, including spatial properties, boundary and initial conditions, and process-related coefficients. Unfortunately, for complex systems, the necessary information is often not readily available, thus requiring researchers to use approximations and best guesses. The recent deployment of sensors into the environment provides an opportunity to use real-time data to parameterize these models. This chapter develops an anomaly detection method that can be used for real-time quality control and analysis (QA/QC) in environmental streaming data. This study develops a new real-time anomaly detection method that employs data-driven autoregressive models of the data stream and a prediction interval (PI) calculated from recent historical data to identify streaming data anomalies. The method used to calculate the PI in this study accounts for uncertainty in the data and in the data-driven model's parameters. Data-driven time-series models are employed instead of statistical time-series models, because they are simpler to develop and because they rapidly produce accurate short-forecast horizon predictions. Data are classified as anomalous/non-anomalous based on whether or not they fall outside of the $p\%$ PI. Thus, the method provides a principled framework for selecting a threshold. This method does not require any pre-classified examples of data, scales well to large volumes of data, and allows for fast, incremental evaluation of data as it becomes available. The following section describes this method in detail. Next, it is tested through a case study, in which several different data-driven modeling techniques are used to identify erroneous measurements in a windspeed data stream from the Water

and Environmental Research Systems (WATERS) Network testbed at Corpus Christi Bay, Texas, provided by the Shoreline Environmental Research Facility (SERF) (<http://www.serf.tamus.edu>). Finally, the results of the different instantiations of the anomaly detection method are compared, and implications of the different modeling strategies are discussed.

5.1 Methods

This study proposes a new analytical redundancy method for anomaly detection that employs data-driven autoregressive models and their corresponding PIs for one-step-ahead prediction of the expected value and range of plausible values for the next measurement, respectively. The PIs are then used to delineate the boundary between anomalous and non-anomalous data. Data-driven models are statistical models of the conditional distribution of the output variables conditioned on the input variables, such that the outputs of these models approximate the conditional averages of the target data (Bishop, 1995). Data-driven models were chosen because they can efficiently model the time-series data collected by the sensors without *a priori* assumptions about the structure of relationships in the data.

Autoregressive models are models that predict future measurements in a sensor data stream using only a specified set of previous measurements from the same sensor; they are used because they simplify the anomaly detection process in several ways. First, using only previous values of the same data stream avoids complications caused by different sampling frequencies that can arise if a heterogeneous set of sensor data streams

is used. Second, because of the frequency with which the SERF sensors go off-line, a significant number of gaps of undefined duration exist within each of the sensor data streams, and when comparing the streams, these gaps usually do not correspond with the same time periods. Since time-series models require a defined set of input variables, it is unclear how to make predictions if one or more of the input variables is not available; thus, the use of autoregressive models reduces the number of predictions that cannot be made due to insufficient data. Finally, since anomalous data cannot be expected to produce reasonable predictions when used as inputs into a model, if data from more than one sensor are used for prediction of a particular data stream, then anomalous data from different sensors must be detected in a particular order, such that all of the independent variables of a model have already been processed. For example, if the model for data stream *A* requires the most recent measurements from data stream *B*, then anomaly detection must first be performed on data stream *B*, before it can be performed on data stream *A*. The use of autoregressive models allows anomaly detection on several sensor data streams to take place rapidly and in an arbitrary order.

The anomaly detection method considers the sensor data stream sequentially and in chronological order, classifying measurements one at a time as either anomalous or non-anomalous. A measurement will be classified as anomalous if it deviates significantly from the one-step-ahead prediction of its value calculated by the data-driven autoregressive model of the time series. Significant deviation is defined using a constant threshold calculated via the PI. The PI gives the range of plausible values that the next measurement can take, and the prediction level (p) indicates the expected frequency with

which measurements will actually fall in this range. If it is assumed that the model residuals are normally distributed, then the 100(1- α)% PI can be calculated as:

$$\bar{x} \pm t_{\alpha/2, n-1} * s \sqrt{1 + \frac{1}{n}} \quad (5.1)$$

where $t_{\alpha/2, n-1}$ is the 100(1 - $\alpha/2$)th percentile of a Student's t-distribution with $n-1$ degrees of freedom, s is the standard deviation of the model residual, and n is the sample size used to calculate s . This type of PI is a type of t-interval, because it relies on Student's t-distribution. If the new measurement falls within the bounds of the PI, then the measurement is classified as non-anomalous; otherwise, it is classified as anomalous. Thus, the PI represents a threshold for acceptance or rejection of a data point. The benefit of using the PI instead of an arbitrary threshold is that the prediction level guides the selection of the interval width.

Once an anomalous data point is identified, two strategies for processing future data are compared. The first strategy is to simply flag the data point as anomalous and proceed to calculate the PI for the next chronologically sequential data point using the newly classified anomalous data point as input to the data-driven model of the data stream. The second strategy is to flag the data point as anomalous and proceed to calculate the PI for the next chronologically sequential data point, using the data-driven model prediction of the anomalous data point (instead of the newly classified anomalous data point itself) as an input to the data-driven model of the data stream. The former strategy will hereafter

be referred to as anomaly detection (AD), while the latter strategy will be referred to as anomaly detection and mitigation (ADAM).

This study compares four data-driven methods for creating the one-step-ahead prediction models: naïve, clustering, perceptron, and artificial neural network (ANN). These methods were selected because they have all been demonstrated to successfully model different types of time-series data, as indicated below. Data-driven methods like these develop models using sets of training examples. Each example contains an input vector (i.e. the set of variables used to make a prediction) and an output vector (i.e. the set of desired model outputs). Training these models involves fitting their parameters to minimize error on the training examples, without overfitting to the training data, which can lead to poor predictions. Since the PI requires an estimate of the standard deviation of the modeling error, 10-fold cross-validation (Duda *et al.* 2001, Han & Kamber 2006) was used to train the models. Ten-fold cross-validation divides the training set into 10 non-intersecting subsets. One subset is retained for model validation, while the remaining nine are used for model training. This process is repeated once per fold so that every subset has been used for model validation. The model error mean and standard deviation are then calculated as the average mean and standard deviation of prediction errors for the validation sets over all the folds. This method has been shown to quantify error caused by uncertainty in both data measurement and model parameter estimation (Han & Kamber 2006). The remainder of this section describes the implementation of the four data-driven modeling methods used in this study and indicates their suitability for modeling time-series data.

Naïve Predictor: The naïve predictor is a nearest-neighbor approach (Duda *et al.* 2001, Hastie *et al.* 2001) that bases its prediction of an unseen event on the response of the system to the most similar historical event and defines similarity in terms of temporal distance. Thus, the naïve prediction of a measurement at time $t+\Delta t$ is equal to the value of the measurement at time t . Because this nearest neighbor approach uses temporal proximity, and because time-series data is inherently chronologically ordered, this approach, unlike many nearest neighbor approaches, scales well to large quantities of data. Amenu *et al.* (2007) demonstrated the utility of this method for modeling stream flow time-series data.

Clustering Predictor: The clustering predictor is slightly more sophisticated than the naïve method because, while it also predicts the value of an unseen event based on the observed responses of similar events, it defines similarity by mapping each measurement to a region of input space (i.e. the coordinate system defined by the set of input variables). It then partitions the input space into local regions (clusters) based on the training data and predicts the system response from each cluster to be the mean of the training data target values that mapped to each cluster. The k -means clustering algorithm (Duda *et al.* 2001, Hastie *et al.* 2001, Han & Kamber 2006) was used because it scales well to large quantities of data (Schütze & Silverstein 1997). The number of clusters was specified using within-cluster scatter (Hastie *et al.* 2001), which indicates the similarity of the points to their assigned cluster center. This method of determining the number of clusters is costly, but since it only has to be performed once, it does not significantly affect the

utility of the clustering predictor for real-time applications. Vasquez and Fraichard (2004) demonstrated this method for predicting time-series data from robotic optical sensors.

Perceptron Predictor: The perceptron model (Bishop 1995) predicts the response of a system (y) to be a linear combination of the input variables (\vec{x}) describing the system state through the transform:

$$y = \vec{w}^T \cdot \vec{x} + b \quad (5.2)$$

where \vec{w} is a weight vector that defines the relationship between \vec{x} and y , and b is a constant. The weight vector \vec{w} is learned iteratively by applying a small correction to each element w_i in the weight vector proportional to both the perceptron's error on a training pattern and the product $w_i x_i$, where the constant of proportionality is called the learning rate, and the constant b is learned in a similar manner. This learning algorithm, called the "perceptron learning rule" (Rosenblatt 1958), traverses the error surface to find the point where the model error on the training set is minimized; thus, the perceptron performs a linear least-squares regression of the training data. In this study, the learning rate is selected using a trial-and-error approach. The perceptron model has well documented abilities to predict time-series data (e.g. Bishop 1995).

Artificial Neural Network Predictor: Artificial neural networks (Bishop 1995, Duda *et al.* 2001, Hastie *et al.* 2001) are networks of perceptron-like nodes (i.e. the output of each node is a non-linear function of the weighted sum of the node inputs) that create models of a system state using non-linear combinations of the input variables. The ANN

considered in this study is a feed-forward network that is trained using the standard backpropagation algorithm with gradient descent (Rummelhart *et al.* 1986). Training was terminated after a fixed number of training epochs were completed or when further training caused a decrease in the model's performance on a testing set (Bishop 1995, Hastie *et al.* 2001). This latter condition discourages overtraining of the network. ANNs require that a learning rate, number of hidden layers, number of nodes in each hidden layer, and maximum number of training epochs be specified. These values were selected using a trial and error approach (Bishop 1995, Hecht-Nielsen 1990). The ability of ANNs to model time-series data is also well documented (e.g. Bishop 1995).

5.2 Case Study

To demonstrate the efficacy of the anomaly detection methods developed in this study for data QA/QC, it was applied to a windspeed sensor data stream from Corpus Christi Bay. The sensor is an R.M. Young model 05106 marine wind monitor, which collects windspeed and direction at a frequency of one hertz. The AD and ADAM strategies were tested using all four modeling methods and 95% and 99% PIs. These 16 combinations will hereafter be referred to as anomaly detectors. The features used to model the data stream were selected using correlation analysis, a common approach in time-series modeling (Box & Jenkins 1970). This analysis indicated that the current windspeed is strongly correlated with historical measurements as distant as ten minutes, as shown in Figure 5.1. Because the sensor samples at one hertz (i.e., every second), there are approximately 600 highly correlated measurements. Given that the complexity of data-driven modeling increases with the number of input variables, only the first 30 seconds of

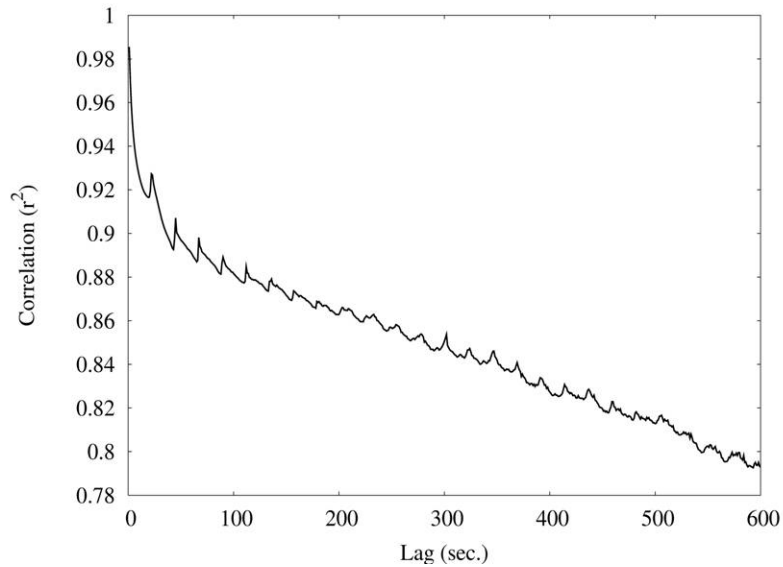


Figure 5.1: Autocorrelation of the SERF windspeed data calculated during the period of January – May 2004. Note that r^2 values of 0.8 and higher are considered highly correlated.

data were selected for the input variables of the data-driven models. This subset of the highly correlated measurements provides a good tradeoff between model complexity and prediction accuracy. The windspeed models were developed using 30,000 training examples selected at random from the period of January–May 2004 and were tested on data collected in June 2004. The user-defined parameters for these models, which were selected as described in the previous section, are shown in Table 5.1.

Figure 5.2 shows the predictive abilities of the generated data-driven models on a segment of the windspeed data collected on June 15, 2004, which is part of the testing dataset. The clustering predictor appears to be the least accurate, while the naïve predictor appears to be the most accurate. The 16 resulting anomaly detectors were then compared based on their ability to identify errors in the testing data.

Table 5.1: Values for data-driven time-series models.

Model	Parameter	Values
Clustering	Number of Clusters (k)	6
ANN	Number of Hidden Layers	1
	Number of Nodes in First Hidden Layer	50

Since the data used in this study were subjected to manual quality control measures before they were archived, it was expected that the detectors would not identify many data anomalies in the archive. However, this was not the case. The detectors identified approximately 6% of the data during the month of June as anomalous. This result encouraged focused inspection of these data. For example, Figure 5.3 shows a two-minute segment of the data stream from June 15, 2004, in which six suspicious events affecting nine data points can be easily identified. All six of these events had been classified as anomalous by one or more of the 16 detectors. Subsequent investigation of these types of data anomalies by the SERF data managers revealed that events such as these were most likely caused by wireless transmission errors. Further analysis of anomalous data revealed other suspicious events that were of significantly longer duration than those shown in Figure 5.3. For example, Figure 5.4 shows a 35 minute segment of the data stream from June 22, 2004, during which a suspicious long-duration event occurs. The windspeed between minutes 8 and 26 in the plot appears to have been offset by a constant 7 m/s. It is the sharpness of the transition from the slower (~ 5 m/s) to the faster (~ 12 m/s) windspeed and back, as well as the existence of data in both the slow and fast regimes that appear to correlate with data in the opposite regime, which suggests that a significant portion of the data presented in this figure does not represent the actual windspeed. This data segment is of particular interest because its behavior is similar to

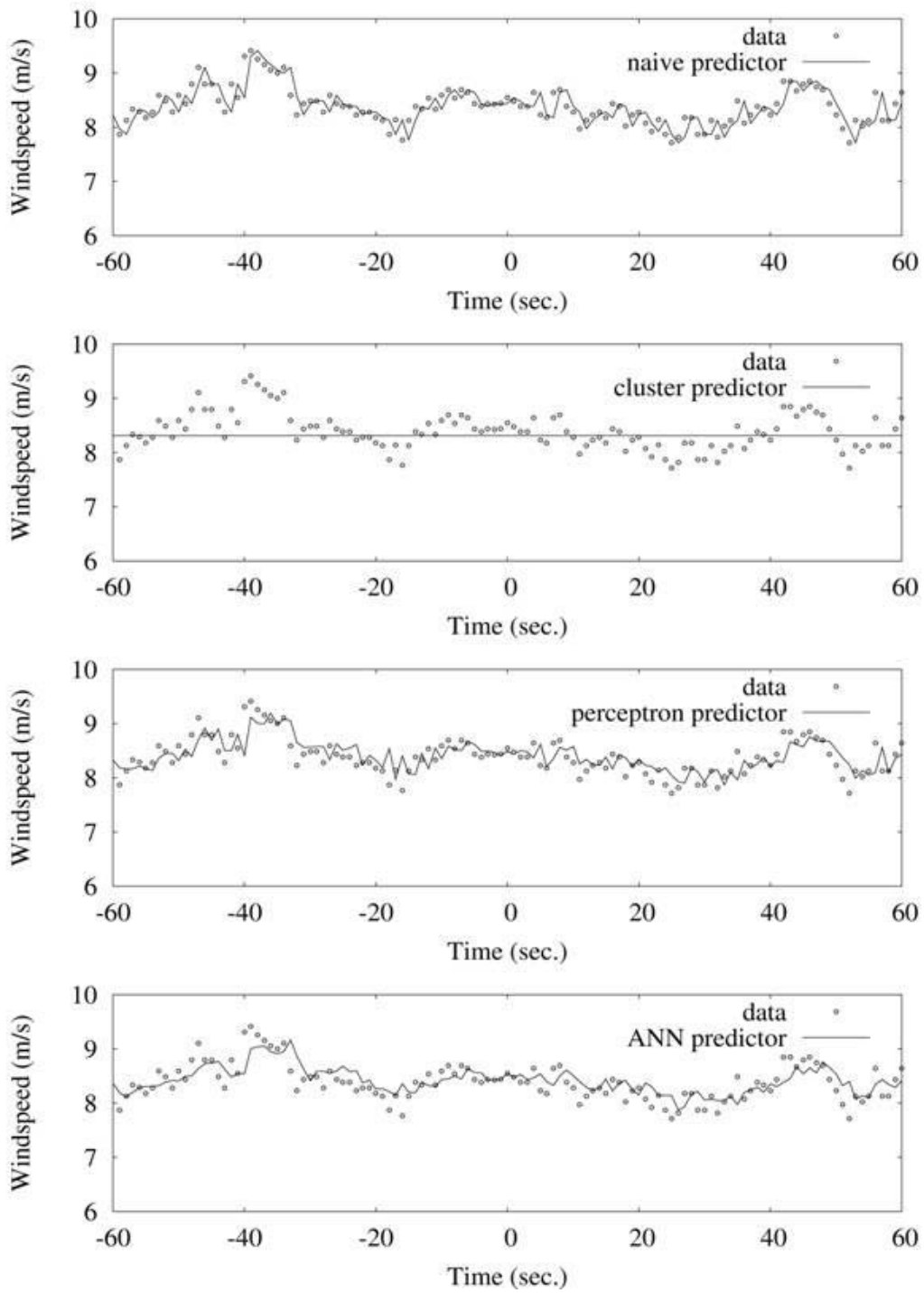


Figure 5.2: Performance of different data-driven methods for predicting the Corpus Christi Bay windspeed data stream.

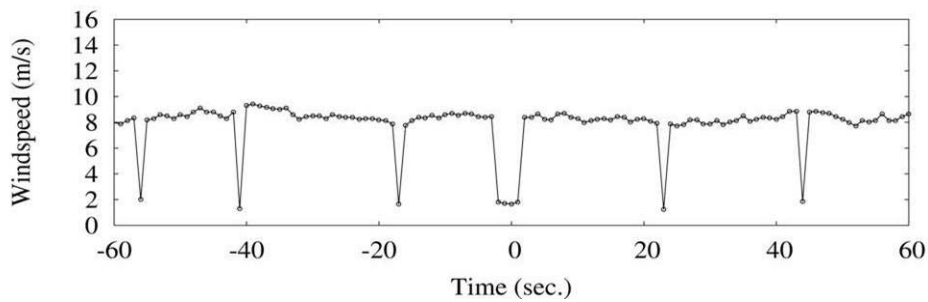


Figure 5.3: Data exhibiting errors resulting from short duration faults.

the behavior of a type of sensor fault (offset bias fault) described by Koushanfar *et al.* (2003), which causes sensor data to be offset by a constant value. Errors resulting from short-duration faults, such as those shown in Figure 5.3, may not have a significant effect on the utility of the data if time averages are used. However, due to the high frequency with which these events are observed in this data stream, even moderate time averages are adversely affected. For example, the two-minute average (a standard granularity for wind data collected by the National Oceanic and Atmospheric Administration’s National Data Buoy Center) of the raw data shown in Figure 5.3 has a value of 7.2 m/s, whereas the two-minute average of the data with the nine anomalous data points removed has a value of 8.3 m/s, a difference of 6%. Long-duration errors, such as those shown in Figure 5.4, are even more worrisome because their effect can only be mitigated if very long time averages are used.

The existence of errors in the June 2004 data indicated that the data from January to May (used for training) also contained errors. Thus, before proceeding with an assessment of the 16 detectors, it was necessary to clean the training data and retrain the regression models. Cleaning was performed using the Naïve-AD detector with a 95% PI. The naïve

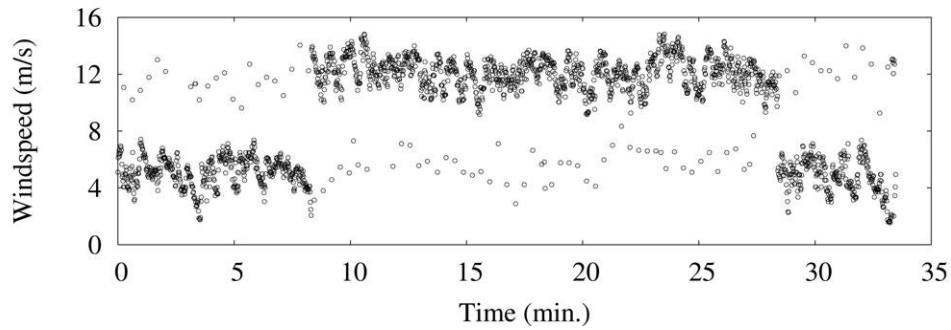


Figure 5.4: Data exhibiting errors resulting from long duration faults.

detector was chosen because it performs well at identifying anomalous data and because it does not rely on a model of the data stream that could have been affected by errors in the uncleaned training data. The AD strategy was used because its long-term performance is not affected by previous misclassifications (as opposed to the ADAM strategy, which replaces measurements classified as anomalous with its own estimate, based on prior data, thus perpetuating the effect of classification errors over the long term). Records containing data classified as anomalous were removed from the training set.

Once the training data were cleaned, the models were retrained, and the residuals were tested for normality using a Lillifors test at the 5% level. The test indicated that although the distributions were symmetric and unimodal, they had thicker tails than a normal distribution. This feature of the residual distributions can be illustrated using a quantile-quantile (Q-Q) plot, such as the one shown in Figure 5.5, which compares the quantiles of the ANN model residuals (y -axis) with the quantiles of a standard normal distribution (x -axis). The Q-Q plot compares the quantiles of one distribution against the quantiles of another distribution and is used to identify differences between the two distributions. A

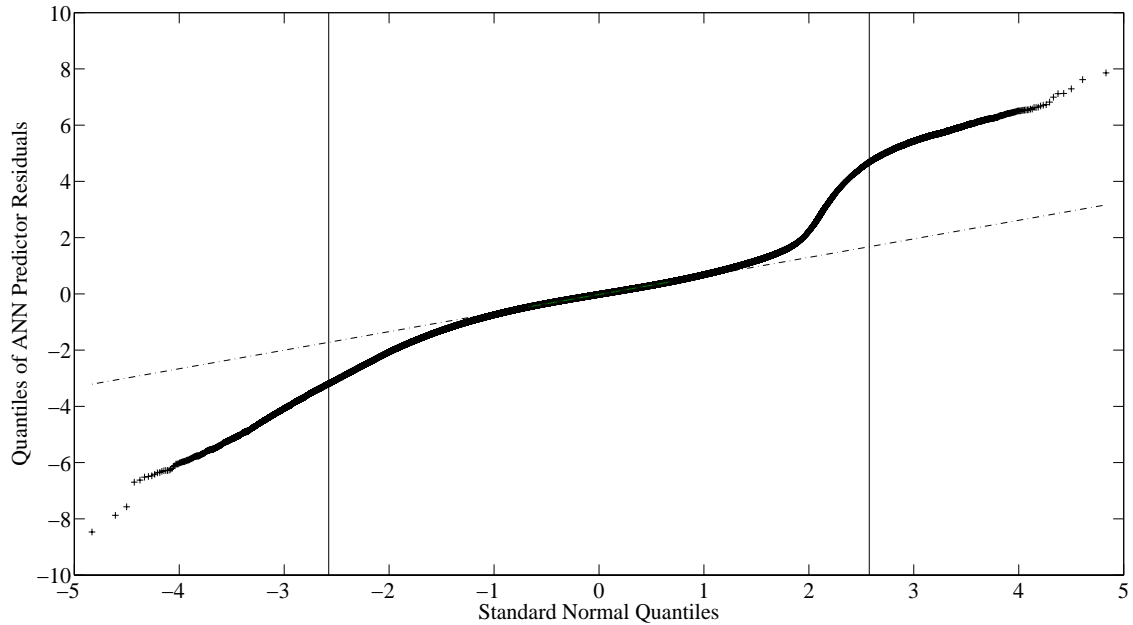


Figure 5.5: Q-Q plot of ANN model residual distribution and standard normal distribution. Vertical lines indicate bounds of 99% PI.

reference line is also plotted. If the two distributions are similar, then the points will fall approximately on the reference line shown in Figure 5.5, whereas departures from this line indicate differences in the distributions. It can be seen in Figure 5.5 that the points fall on the reference line in the central region of the distribution (e.g. around the mean) but depart from the reference line towards the tails. This departure becomes obvious around $x = \pm 1.5$ and is caused by the ANN model residual quantiles having a larger magnitude than the standard normal quantiles in the tail region, thus indicating that the ANN model residuals have thicker tails than those of a normal distribution.

This result affects the validity of the implicit assumption in the PI calculation (Equation 5.1) that the model residuals are normally distributed. However, since the central region of the model residual distribution behaves similar to that of a normal distribution, the PI

calculated using Equation 5.1 will be unaffected. In the case of the ANN model residuals, this region covers approximately $x = \pm 1.5$. Thus, for prediction levels of 85% or less, the PI calculated with Equation 5.1 will accurately bound the region of plausible measurement values that the measurements will take with a frequency of $1-p$. For prediction levels greater than approximately 85%, the PI will be non-conservative (i.e. the frequency that the measurement will fall into the $p\%$ PI will be less than $1-p$), thus resulting in a higher false positive rate than $p\%$. This results from the existence of a higher frequency of extreme (tail) values in the model residual distribution than would be indicated by a normal distribution. Since this non-conservative effect increases as the PI bounds move further into the tails of the model residual distribution, the PI will become increasingly non-conservative as the prediction level increases. For example, the vertical lines in Figure 5.5 are at positions $x = \pm 2.58$ on the standard normal quantiles and indicate the bounds of the 99% PI. Within this region, the quantiles of model residual distribution deviate moderately from the quantiles of the normal distribution; thus, it is expected that the non-conservative effects for prediction levels less than 99% will be moderate. The other model residual distributions exhibit behavior similar to that of the ANN model residuals, indicating that for prediction levels less than 99%, the PI will overestimate the frequency that a measurement will fall within its bounds, thus causing slightly higher than expected false positive rates.

The performance of the 16 anomaly detectors at identifying additional erroneous data was then quantified using a sample of over 2700 other data points from the data archive that had not been cleaned. True/false positives were identified visually, using domain

knowledge provided by the SERF data managers. The data managers indicated that measurements that deviate from their close neighbors by approximately two m/s or more and that break the relatively smooth trend of the windspeed are most likely erroneous. For example, the erratic oscillations noted in Figures 5.2 and 5.3 are erroneous because an air mass cannot plausibly accelerate and then decelerate at the rate required to cause such oscillations.

Figure 5.6 shows the detectors' false positive rate for identifying erroneous data in the windspeed data stream. It can be seen that the ADAM strategy reduces the false positive rates of the perceptron and ANN-based detectors, whereas it increases the false positive rates of the naïve and clustering-based detectors. For the perceptron and ANN-based detectors, without the use of mitigation, previously processed erroneous data adversely affect future classifications. This occurs because using erroneous data as inputs to the perceptron or ANN models requires the model to make a prediction using a type of input data (i.e. erroneous measurements) with which it was not trained (i.e., to extrapolate beyond the training data), thus reducing the accuracy of the one-step-ahead predictions and their corresponding PIs. The naïve and clustering-based detectors, however, appear to be less sensitive to erroneous input data. Unlike the perceptron and ANN detectors, the naïve and clustering-based methods do not predict the future windspeed using a function of the input values. Rather, these detectors predict the future windspeed using a similar, previously observed example. Thus, input measurements to these detectors that vary significantly from the current locally-averaged windspeed (e.g. data errors) will not

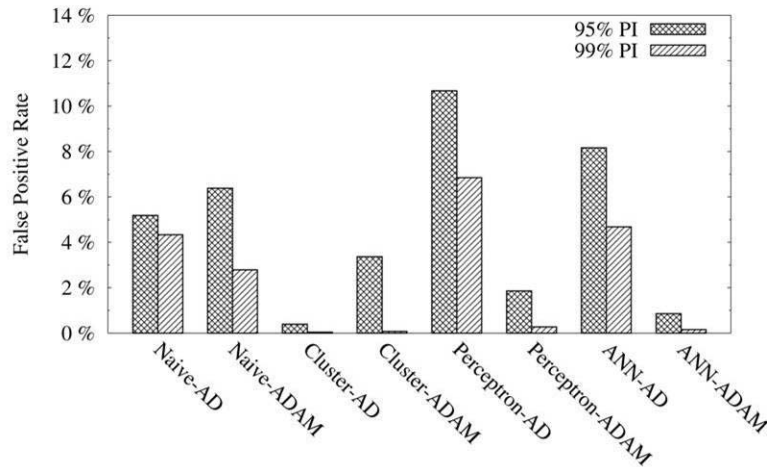


Figure 5.6: False positive rates for detecting June 2004 windspeed data errors using 95% and 99% PIs.

cause a future windspeed prediction that is vastly different from previously observed windspeeds.

However, the use of the ADAM strategy can negatively impact the performance of a detector when it misclassifies a non-anomalous point as anomalous, because this causes the detector to replace the valid measurement with an incorrect value, which sometimes results in the detector continuing to make mistakes and replace valid measurements with incorrect values until the cycle is broken. In the case of the perceptron and ANN-based detectors, this behavior is uncommon and outweighed by the positive benefits of mitigation described previously, whereas in the case of the naïve and clustering-based detectors, this behavior significantly affects the detectors' performance. Since the naïve-based detectors use only one data point to predict the future windspeed, it is understandable that, when using the ADAM strategy, this method perpetuates

misclassifications, because future predictions reflect the incorrect values that replaced valid measurements. However, it is less clear why the clustering-based detectors behave in this manner. Perhaps it is because the cluster membership is more heavily influenced by recent measurements than by distant measurements.

Figure 5.6 also indicates that an increase in the prediction level from 95% to 99% decreases the number of false positives, though this decrease is not as dramatic as the change associated with the use of mitigation, indicating that the appropriate use of mitigation has the most significant effect on the detectors' false positive rate.

The false negative rate for the detectors is shown in Figure 5.7. It can be seen that the naïve, perceptron, and ANN-based detectors misclassify significantly fewer erroneous data than the clustering-based detectors, which misclassify almost all of the erroneous data. Thus, the clustering-based detectors are not useful for detecting erroneous data in the windspeed data stream. It can also be seen that the use of the ADAM strategy significantly improves the ability of the perceptron-based detectors to correctly classify erroneous data, whereas it does not significantly affect the false negative rates of the other detectors. The decrease in the perceptron-based detectors' false negative rate, due to the use of the ADAM strategy, can again be attributed to erroneous data adversely affecting future classifications by requiring extrapolation of the perceptron model. Furthermore, an increase in the prediction level from 95% to 99% results in an increase in the false negative rate, which, for the best-performing detectors, is larger than the corresponding decrease in the false positive rate, indicating that a 95% PI provides a

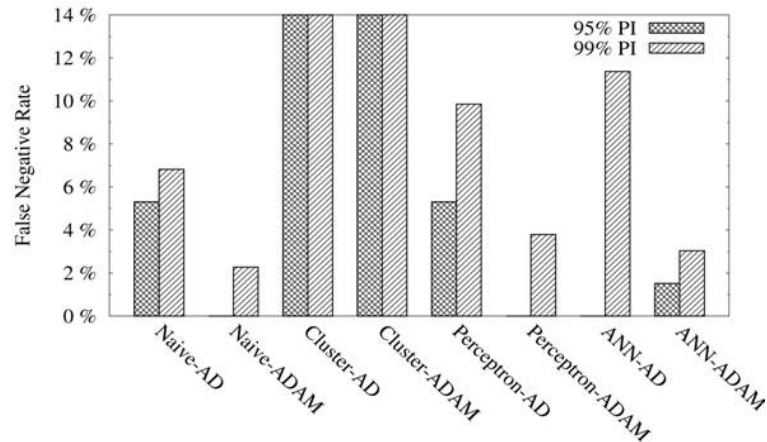


Figure 5.7: False negative rates for detecting June 2004 windspeed data errors using 95% and 99% PIs. The bars for the cluster-based methods have been truncated.

These methods have false negative rates greater than 89%.

reasonable tradeoff between misclassifying erroneous and non-erroneous data in this data stream.

5.3 Discussion

The anomaly detectors used in the case study represent 16 instantiations of the proposed anomaly detection method, which relies on a threshold deviation from a model prediction to delineate the boundary between anomalous and non-anomalous data. This type of method will misclassify both anomalous data that have smaller deviations than the threshold and non-anomalous data that have larger deviations than the threshold. Thus, selecting an appropriate threshold is important. The benefit of using the $p\%$ PI, rather than an arbitrary, user-defined threshold, is that the PI provides guidance (in the form of the prediction level) for the selection of the threshold without requiring any knowledge of the process variables being measured. The $p\%$ PI indicates the region likely to contain at

least $p\%$ of the possible sensor measurements, given both the modeling and measurement errors; thus, we expect that approximately $(1-p)\%$ of the non-anomalous data will be misclassified as anomalous. Despite the anticipation of a slightly higher false positive rate resulting from the heavy-tail behavior of the model residual distributions (Figure 5.5), the false positive rates for the detectors in the case study (Figure 5.6) roughly agree with this expectation, if the effect of mitigation is accounted for. This indicates that the normal assumption implicit in the PI calculation is reasonable for the CCBay windspeed data.

Note that the prediction level, however, cannot give an *a priori* estimate of the false negative rate, because the behavior of the anomalous data is unknown, and thus, it is impossible to estimate how many of these data will fall inside a given PI. For this reason, the prediction level should be set such that the false positive rate is reasonably low (e.g. prediction levels of 95% or 99%). Fortunately, in many practical applications of anomaly detection, such as data QA/QC, setting the prediction level in this way yields good results because anomalous data that fall outside the PI are more important to classify correctly than those that fall inside of the PI, since, given a reasonably accurate model of the time-series data, falling inside the PI indicates that anomalous data are roughly within the measurement error of the sensor. For example, the nine obvious erroneous data points in Figure 5.3 were identified by the ANN-ADAM detector with 95% PI, and cleaning these points changed the average value of all 120 points by 6%.

In addition to providing a principled framework for threshold selection, the proposed anomaly detection method allows a detector for one data stream to operate independently

of the behavior of other data streams or their detectors by utilizing autoregressive models of the data stream. The use of autoregressive models, however, precludes the use of information gathered by other sensors, without which the anomaly detector may misclassify certain types of data anomalies, but the case study demonstrates that this limitation does not significantly affect the success of the anomaly detection method for performing QA/QC on one Corpus Christi Bay windspeed data stream.

Finally, the data-driven autoregressive anomaly detection method developed in this study must accommodate two features of environmental sensor data streams: non-time-stationary data (i.e. data whose pattern changes with time) and gaps in the sensor data stream. Because many environmental processes, such as wind, are not time stationary (Amenu *et al.* 2007), data-driven models must be updated periodically. For batch-trained models, such as those employed in this case study, model updating can be accomplished by retraining the models periodically. Batch retraining can be time consuming, but because of energy constraints, sensor data is usually sent from the sensor to the data archive in small batches, rather than one at a time, so retraining only has to be faster than the frequency of data broadcasts, not faster than the sensor sampling rate. Except for the poorly performing cluster-based detectors, training the data-driven models is relatively quick (in this instance, on the order of one minute). Therefore, since the SERF windspeed data is sent in 5 minute intervals, model updating can occur between data transmissions. On-line training algorithms (i.e. algorithms in which the model is incrementally updated for new data as they arrive from the sensor) could be considered for updating the models if the data were transmitted more frequently. However, with an

on-line training algorithm, it is unclear how best to calculate the standard deviation of model errors needed for the PI calculation (Equation 5.1). Therefore, batch retraining is preferable for updating data-driven models to account for changing temporal patterns in the data. If data arrive faster than retraining can occur, then a dual model approach can be used, in which a new model is trained while the previous model is being used for anomaly detection.

Gaps in the sensor data stream are the result of the sensor going off-line because of the harsh environment in which the sensors operate. The windspeed data set considered in the case study has over 500,000 gaps of durations ranging from 1 second to 90 days. The vast majority (99.99%) of these gaps, however, are less than five seconds in duration. Because the detectors require a defined set of input features, these gaps render the error detectors unable to classify certain future measurements as anomalous/non-anomalous. The naïve-based detectors require the measurement at a time one second previous to the prediction time; thus, they cannot classify the first measurement after a sensor goes back on-line. The remaining methods require measurements at times minus one through 30 seconds in order to classify a new measurement; thus, they cannot classify any data within the first 30 seconds after the sensor goes back on-line. Some of these measurement gaps can be addressed by using the detector to fill in missing values with the predicted value of the measurements. This method will only work for short duration gaps, though, as the accuracy of the model prediction will degrade with successive iterations. However, since only 0.01% of the gaps are longer than four seconds, this method will be useful in the majority of cases. For the few remaining cases, manual

inspection of a maximum of 30 data points per sensor is likely to be feasible, especially since multiple sensors are unlikely to fail at the same time.

5.4 Conclusion

Real-time detection of anomalies in environmental streaming data has many practical applications, such as data QA/QC, adaptive sampling, and anomalous event detection. This research developed a new anomaly detection method based on autoregressive data-driven models of the sensor data stream and the PI. This method performs fast, incremental evaluation of data as it becomes available, can scale up to large quantities of data, and requires no *a priori* information regarding process variables or the types of anomalies that may be encountered. To account for temporal changes in the data pattern, the data-driven model employed by this method can be updated, a process which takes approximately one minute. Since the SERF windspeed sensor data is transmitted in five-minute blocks, model updates can be accomplished between data transmissions. Furthermore, because this method employs autoregressive models of the sensor data stream, it is easy to apply to a network of heterogeneous sensors, since the performance of the anomaly detector on one sensor data stream is independent from detrimental properties of other sensor data streams, such as unavailable data or undetected erroneous data. Finally, because the PI is calculated using 10-fold cross-validation, it accounts not only for uncertainty in the data, but also for uncertainty in the data-driven model parameters.

The value and efficacy of this anomaly detection method for data QA/QC is illustrated using a case study involving a windspeed data stream from Corpus Christi Bay.

Anomaly detectors using different data-driven modeling techniques and both the AD and ADAM anomaly handling strategies identified a significant number of erroneous measurements in the windspeed data that manual QA/QC had failed to detect. The errors had durations ranging from 1 second to several minutes and affected approximately 6% of the data. After cleaning the errors in the training data, an assessment of 8 instantiations of both the AD and ADAM strategies indicated that the performance of the perceptron and ANN-based detectors at detecting errors in the testing data was significantly improved by the use of the ADAM strategy, and that the ANN-ADAM detector using a 95% PI performed best, with a false positive and a false negative rate of 1% and 2%, respectively.

The case study results suggest that the anomaly detection method developed in this study is a useful tool for identifying anomalies in environmental streaming data. However, it should be noted that while the ANN produced the best model for the windspeed data considered in the case study, this model may not be the most appropriate choice for other types of environmental data. The four data-driven techniques considered in this paper have been shown to produce good predictions on different types of time-series data and are a useful starting point for selecting the most appropriate model, but the anomaly detection method developed here is not restricted to these models and can easily accommodate other data-driven modeling techniques. This method is limited, however, because it cannot consider several data streams at once and, hence, cannot take advantage

of information gathered by multiple sensors. This lack of information may cause the anomaly detector to misclassify certain types of data anomalies. This method is also limited because it requires a manual restart if the sensors go off-line, as it cannot continue to classify measurements after several sequential measurements have been missed.

Chapter 6: Real-Time Bayesian Anomaly Detection in Streaming Environmental Data

Chapter 5 presented an analytical redundancy method for detecting anomalies in environmental sensor data and compares its performance using several data-driven modeling approaches, including nearest neighbor, clustering, perceptron, and artificial neural networks. This method, however, is limited, because it cannot consider several data streams at once and because missing values in the data stream render it incapable of classifying measurements that immediately follow the missing values. To address these limitations, this chapter develops three real-time anomaly detection methods that employ dynamic Bayesian networks (DBNs) to identify anomalies in streaming environmental data. DBNs are artificial intelligence techniques that model the evolution of discrete- and/or continuous-valued states of a dynamic system by tracking changes in the system states over time. The methods developed in this study use three different DBN implementations: the well-known Kalman filter; the robust Kalman filter, which, to the author's knowledge, has not found wide application in the field of environmental engineering; and the Rao-Blackwellized particle filter, which has only recently been developed. Furthermore, because of the nature of environmental streaming data, it was necessary for the implementations of each of these DBNs to be modified such that they were robust to missing values in the sensor data as discussed in Section 6.1. The following section describes these methods in detail. The DBN-based methods are then tested through a case study, in which they are used to identify anomalous measurements in eight meteorological data streams from the same WATERS Network testbed considered in Chapter 5. Finally, implications of the results are discussed.

6.1 Methods

This study develops three DBN-based methods for real-time detection of anomalies in environmental sensor data streams. As discussed in Chapter 2, DBNs are well-suited for modeling time series data, such as those composing sensor data streams, because they can easily model multivariate data and non-stationary processes. Furthermore, filtering can be used to infer the values of the DBN system state variables from the available measurements without using any future measurements; thus, anomaly detection can be performed in real time. Unlike the anomaly detection methods presented in Chapter 5, the methods developed in this chapter can process many sensor data streams jointly, by considering the data at each measurement interval in chronological order. Two of the anomaly detection methods developed here use the Bayesian credible interval (BCI) to classify data as either normal or anomalous. The first method uses Kalman filtering to calculate the BCI and thus will be referred to as the BCI-kf method. The second method uses robust Kalman filtering to calculate the BCI and thus will be referred to as the BCI-rkf method. The third method uses the maximum *a posteriori* (MAP) estimate of a variable indicating the status (normal/anomalous) of each measurement to classify the sensor data. Thus, the third method will be referred to as the maximum *a posteriori* measurement status (MAP-ms) method.

The BCI-based methods track the multivariate distribution of the system states and their observed counterparts, which are measured by the environmental sensors, using the DBN shown in Figure 6.1. The system states are assumed to be first-order Markov processes,

so the state at time t only depends on the state at time $t-1$. Filtering is used to sequentially infer the posterior distribution of the state variables and their corresponding observations, as new measurements become available from the sensors. The posterior distribution of the observed variables can then be used to construct a BCI for the most recent set of measurements. The $p\%$ BCI indicates that the posterior (i.e. adjusted for the available observations) probability of the observed state variables falling within the interval is p ; thus, the BCI delineates the range of plausible values for sensor measurements. For this reason, any measurements that fall outside of the $p\%$ BCI will be classified as anomalous. The $100(1-\alpha)\%$ BCI for a new measurement can be calculated as:

$$\bar{x} \pm z_{\alpha/2} * \sqrt{\Sigma} \quad (6.1)$$

where $z_{\alpha/2}$ is the $100(1 - \alpha/2)^{\text{th}}$ percentile of a normal distribution and Σ is the variance of the posterior distribution of the measurement prediction. The BCI method of anomaly detection is similar to the prediction interval-based method presented in Chapter 5; however, because the BCI is based on the posterior distribution (whereas the prediction interval is not), the width of the $p\%$ -BCI changes dynamically with the uncertainty of the modeled system. Furthermore, unlike the cross-validation method for calculating the prediction interval (PI), the method used to calculate the BCI does not take into account uncertainty in the estimation of the model parameters; thus, it differs from the cross-validation-based method presented in Chapter 5.

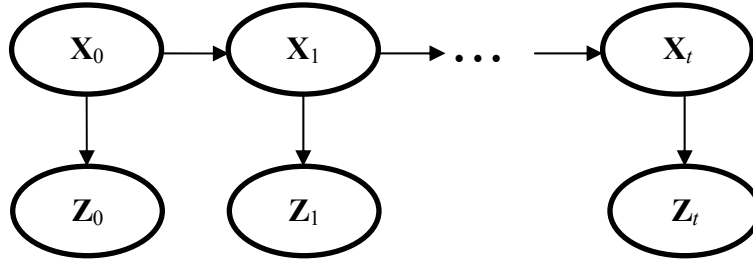


Figure 6.1: Schematic of DBN used in BCI-based anomaly detection. Vector \mathbf{X} represents the continuous-valued system variables and vector \mathbf{Z} represents the continuous-valued observations. Subscripts indicate time.

BCI-kf: In the BCI-kf method, Kalman filtering is used to track the state variables. Thus, the transition and observation models are linear Gaussian. Though this is a strong assumption, as discussed in Chapters 2 and 5, the Kalman filter has been successful for modeling time-series data in a wide variety of applications. The transition and observation models used for Kalman filtering are learned from sensor data using the expectation maximization (EM) method, as described in Section 2.5. Since the measurements correspond directly to the state variables tracked by the DBN, the observation model is constrained, such that the observation matrix (matrix C in Equation 2.2) is the identity matrix, and such that the observation covariance matrix (matrix R in Equation 2.2) is diagonal. These constraints incorporate domain knowledge into the DBN, indicating that the measurements directly correspond to the process being measured and that measurement errors are not correlated between sensors, respectively. Due to the tendency of the EM method to converge to sub-optimal solutions as the joint probability distribution narrows to a degenerate distribution centered on one of the training data points (as discussed in Section 2.5.1), it was necessary to perform four EM

trials starting with different initial conditions, before a suitable parameterization was found

BCI-rkf: In the BCI-rkf method, robust Kalman filtering is used to track the state variables. Again, the transition and observation models are assumed to be linear; however, robust Kalman filtering can account for outliers (i.e. anomalies) in the data through the use of a mixture-of-Gaussians observation model, which has been demonstrated to approximate many heavy-tailed distributions with high fidelity (Blum *et al.* 1999, Efron & Olshen 1978). For example, the model residual data that were shown to have heavier-tails than a normal distribution (Figure 5.5) can be well approximated by a mixture of two Gaussian components with zero mean and variances of 1 and 12, respectively, as demonstrated in the quantile-quantile (Q-Q) plot shown in Figure 6.2. Recall that a Q-Q plot is used to identify differences between two distributions, and that if the two distributions are similar, then the points will fall approximately on the reference line. As shown in Figure 6.2, for this case study, a simple two-component Gaussian mixture is able to model data with a heavy-tailed distribution. In general, since the complexity of robust Kalman filtering increases with the number of Gaussian components used in the mixture model, it is preferable to begin with a two component mixture. A Q-Q plot can then be used to determine how well the mixture corresponds with the data, and the number of components can be increased one-by-one until a sufficiently close fit is achieved.

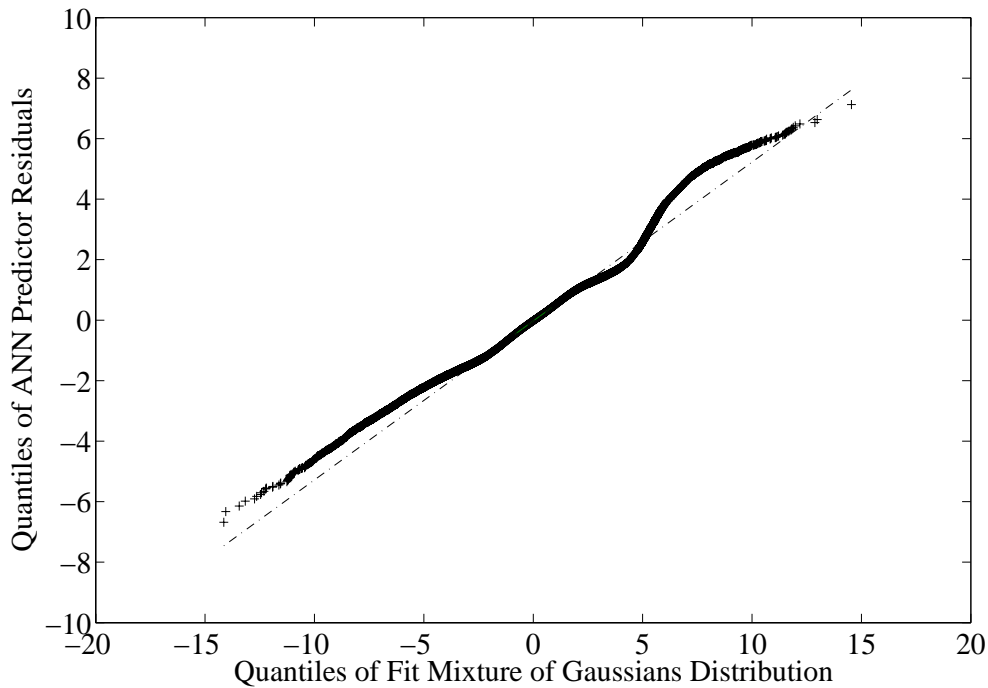


Figure 6.2: Q-Q plot of ANN model residual distribution and mixture of two Gaussian components with means of 0 and 0, variances of 1 and 12, and mixture ratios of 0.75 and 0.25, respectively.

Since measurement anomalies are expected in the sensor data, robust Kalman filtering will more accurately model the system than Kalman filtering. Since the mixture changes dynamically as a function of the observations, the result is an overall non-linear model of the system and observation dynamics. In this method, the distribution of the measurements is represented as a mixture of 2^N Gaussians, where N is the number of measurements, each corresponding to a unique combination of normal and anomalous measurements. For example, if there are two measurements, then the Gaussian mixture will have four components corresponding to the cases (normal, normal); (normal, anomalous); (anomalous, normal); and (anomalous, anomalous). Since the robust

Kalman filter can be conceived as a weighted mixture of Kalman filters (as noted in Chapter 2), setting the observation covariance in this way results in a mixture of Kalman filters, each processing a different subset of the sensors. The transition and observation models developed for the Kalman filter are adapted for use in the robust Kalman filter as follows. The state transition model for each mixture component in the robust Kalman filter is equivalent to that of the Kalman filter. For the observation model, the observation matrix of the robust Kalman filter is the same as that of the Kalman filter, but the observation covariance matrix changes for each of the mixture components. For the case in which all the measurements are normal, the observation covariance matrix is equivalent to that of the Kalman filter, whereas for the cases in which one or more measurements are anomalous, the parameter specifying the measurement variance of the anomalous measurement is set to be a large number (e.g. 1,000), indicating that regardless of the true state of the system, the measurement could take any real value with approximately equal probability. This description of anomalous measurements is used because it indicates that an anomalous measurement is more likely to fall outside the range of plausible measurements than a non-anomalous measurement, without requiring *a priori* knowledge of the types of anomalies that can occur. The mixture ratio used by the robust Kalman filter is set manually, using domain knowledge regarding the frequency of measurement anomalies. Manually setting the parameters for the cases in which one or more measurements are anomalous is necessary because anomalous measurements are, by definition, infrequent; as such, sufficient information may not be available for learning these parameters from the data. Furthermore, learned parameters may define anomalies too narrowly to identify the range of anomalies that may be

encountered. Because the BCI-rkf method does not explicitly track the anomalies through time, it cannot represent any dependency relationships between anomalies; thus, the assumption of time independence of the anomalies is implicit in this method.

MAP-ms: The MAP-ms anomaly detection method uses a more complex DBN that includes a discrete variable that indicates whether or not each measurement, within a given measurement interval, is anomalous. The graphical structure of this DBN is shown in Figure 6.3. If there are N measurements within each measurement interval, and if there are only two possible measurement statuses (i.e. normal/anomalous), then this variable will have 2^N values, each corresponding to a unique combination of measurement classifications. For example, if there are two measurements, then the measurement status variable will have four values: (normal, normal); (normal, anomalous); (anomalous, normal); and (anomalous, anomalous); thus, it can be seen that with a binary (normal/anomalous) classification strategy, the DBNs considered in the BCI-rkf and MAP-ms method are similar because each represents the belief state as a mixture of 2^N Gaussian components. Thus, this DBN will also be able to represent heavy-tailed distributions well, as illustrated in Figure 6.2. Rao-Blackwellized particle filtering is used to sequentially infer the posterior distribution of the state variables and their observations as new measurements become available from the sensors. The MAP estimate (i.e. the most likely value, given the posterior distribution) of the state variable that indicates the measurement status can then be used to classify the sensor measurements as normal or anomalous. As in the case of the robust Kalman filter, the transition and observation models developed for the Kalman filter are used to describe

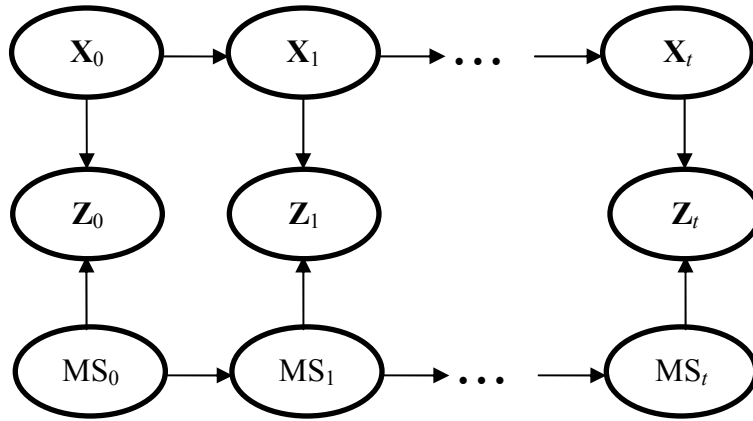


Figure 6.3: Schematic of DBN used in MAP-ms anomaly detection. Vector \mathbf{X} represents the continuous-valued system variables, vector \mathbf{Z} represents the continuous-valued observations, and scalar \mathbf{MS} indicates the discrete measurement status. Subscripts indicate time.

the case in which all measurements are normal; for the cases in which one or more measurements are anomalous, the parameter specifying the measurement variance of the anomalous measurement is set to be a large number. Thus, given a particular value of the discrete measurement status variable, the transition and observation models are linear. However, since the transition model describing the measurement status variable is non-linear, the overall DBN model is non-linear.

Finally, the filtering algorithms (Kalman filtering, robust Kalman filtering, and Rao-Blackwellized particle filtering) were modified to accommodate missing values in the sensor data streams. This modification is necessary because of the harsh environment under which environmental sensors must operate. As discussed in Section 2.5, this modification causes the elements corresponding to the missing measurements in the gain

matrix as well as the system and measurement noise matrices to be equal to zero. Thus, the posterior estimate of the state variable corresponding to the missing measurements is equal to the prior estimate of this variable.

6.2 Case Study

To demonstrate the efficacy of the anomaly detection methods presented in Section 6.1, they were applied to eight meteorological data streams from Corpus Christi Bay. These data streams measure windspeed, wind direction, air temperature, and barometric pressure at two minute intervals at two SERF sensor platform locations: CC003 and CC009. The CCBay platform discussed in Chapter 5 will not be addressed in this study, because it is no longer used. Figure 6.4 shows the location of these sensor platforms within Corpus Christi Bay.

Because the sensors addressed in this case study are under development, there are many missing measurements due to sensor outages in the historical data record. Additionally, even though the historical data were subjected to manual quality control measures before they were archived, an initial application of the anomaly detection algorithms identified several anomalous events that were subsequently confirmed, through investigation by the SERF data managers, to be the result of sensor failures. Data from November 2006, which do not appear to contain sensor failure errors, were used for training the DBNs. This time span was chosen because the EM algorithm requires contiguous blocks of measurements for learning and because these data are reasonably complete (there still were approximately 10 and 800 missing measurements in each data stream from CC003

and CC009, respectively, during this time period). It does, however, reflect the effects of two storm events on November 11 and 31. During November, the approximate ranges of the windspeed, wind direction, air temperature, and barometric pressure at both platform locations are: 0-40 knots (kt), -180°-180° degrees from north (dfn), 5°-28° Celsius (°C), and 1000-1040 millibars (mb), respectively. The performance of the anomaly detectors was then evaluated using data from early December. These data were chosen because the CC003 wind direction sensor went offline around December 15 (subsequent to this study's completion, this sensor was returned to duty).

6.2.1 Detector Parameterization

Parameters for the DBNs were learned from the approximately 21,600 measurements collected during the month of November 2006, as described in the previous section. For this case study, The BCI-kf and BCI-rkf methods were performed using a 99% BCI for anomaly classification. This level of BCI was chosen for two reasons: (1) because it should not misclassify many normal data points as anomalous (the $p\%$ BCI is expected to make this type of misclassification $(1-p)\%$ of the time), and (2) because it produced good results in preliminary trials. The reader may recall that in Chapter 5, a 95% PI was used. This discrepancy can be explained by noting the difference between a PI and a BCI. PIs are based on prior distributions of predicted measurements, whereas BCIs are based on posterior distributions of predicted measurements. Since posterior distributions are updated for actual measurements, they are narrower than prior distributions; thus, for the same level p , the PI will be wider than the BCI. For this reason, the level of the BCI used in this study (99%) does not correspond to the level of the PI used in Chapter 5 (95%).

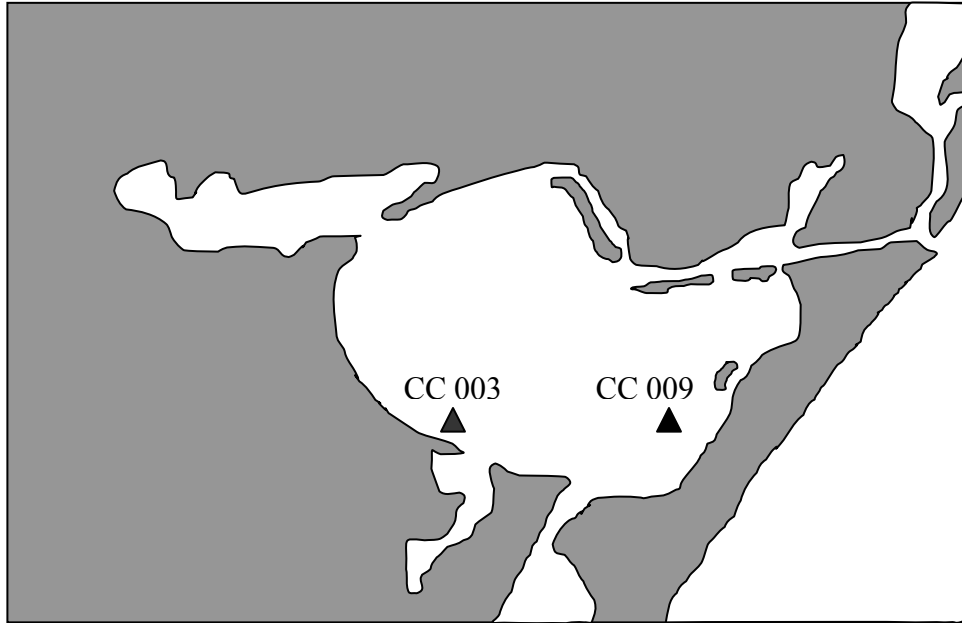


Figure 6.4: Location of SERF CC003 and CC009 sensor platforms within Corpus Christi Bay.

The MAP-ms method was performed using Rao-Blackwellized particle filtering in four trials with 1000; 10,000; 50,000; and 100,000 particles. In the results presented below, the number of particles used in the MAP-ms method is indicated by appending the name with the number of particles (e.g. MAP-ms-10k indicates 10,000 particles).

Two of the data types addressed in this study represent measurements of processes with non-linear dynamics: windspeed, which cannot be negative, and wind direction, which has a discontinuity between -180 and 180 degrees. Despite the non-linearity of windspeed, preliminary work indicated that a linear dynamics model provides a good approximation of this process. Wind direction, however, was not well approximated by a linear dynamics model; thus, this study uses a transformation that converts wind direction into a

two-component vector composed of the cosine of the angle (with respect to north) of the wind and the sine of the angle (with respect to north) of the wind. Thus, this transformation maps the wind direction into two values that vary smoothly from -1 to 1. These values can be envisioned, respectively, as indicating the ratio of the total wind vector blowing in the northerly and easterly directions. Preliminary work also investigated the transformation of the windspeed and wind direction data streams into the two-dimensional wind velocity vector. This transformation addressed not only the non-linearity of wind direction, but also removed the non-negativity condition of the windspeed data. However, this transformation rendered the detectors unable to distinguish between anomalies in the windspeed sensor, the wind direction sensor, or both sensors. For this reason, this latter transformation was not pursued further.

Because of the transformation of the wind direction data, the DBNs in all three anomaly detection methods tracked 10 state variables (one each for windspeed, temperature, and pressure at each location, and two each for wind direction at each location). Thus, the vector X in Equation 2.2 has 10 dimensions referring to the true system states: windspeed at CC003, wind direction (northerly component) at CC003, wind direction (easterly component) at CC003, air temperature at CC003, barometric pressure at CC003, windspeed at CC009, wind direction (northerly component) at CC009, wind direction (easterly component) at CC009, air temperature at CC009, and barometric pressure at CC009. The vector Z in Equation 2.2 also has 10 dimensions referring to the measurements of these 10 states. The linear model parameters A and C in Equation 2.2

are 10x10 matrices, as are the covariance matrices Q and R in Equation 2.2. See Chapter 2 for more details on how these parameters are calculated.

Both the robust Kalman filter and the Rao-Blackwellized particle filter are designed to model $2^8=256$ anomaly combinations, where anomalies in the transformed wind direction vector components are considered jointly. These anomaly combinations result in measurement covariance matrices that are equivalent to the measurement covariance matrix of the no anomaly case, except that the variance of the anomalous measurements is set to 1000 as discussed previously. The probability of an anomaly in each data stream (used to set the mixture ratio for the robust Kalman filter (see Equation 2.10) and the discrete variable transition model for the Rao-Blackwellized particle filter (see Section 2.5.3) was specified to be 5% and independent of current or historical anomalies in any data stream. This value was selected to be representative of the frequency of data anomalies due to measurement or data transmission errors in the Corpus Christi sensor array, based on the results of the case study presented in Chapter 5. The assumption of time-independence of the anomalies was made because there is no information regarding the persistence of anomalies caused by sensor failures, and because the assumption of independence is more general than the assumption of a particular dependency relationship. The assumption of independence of concurrent anomalies in different data streams is made because failure of a sensor rarely affects the capabilities of other sensors, and because there is no indication that such a relationship exists for the SERF meteorological sensors considered in this case study.

To facilitate comparison of the DBN-based anomaly detectors with the autoregressive anomaly detector (AR_ADET) presented in Chapter 5, eight AR_ADET detectors (one for each data stream) were used. These detectors employed an artificial neural network (ANN) model of the sensor data stream, as indicated in Chapter 5. Parameterization of these models was performed as described in Chapter 5, using correlation analysis to determine the salient autoregressive variables. The resulting autoregressive models for windspeed and wind direction use the most recent thirty measurements to predict the next measurement, while the resulting models for air temperature and barometric pressure use the most recent two measurements. The models were then trained using 30,000 randomly selected data points from October through November 2006. These data were selected for training, because data prior to October were not available, and because data from December were to be used for demonstration of the detectors. However, since these data do not span as many months as the training data used in Chapter 5, it was expected that the AR_ADET detectors would not perform as well as indicated in Chapter 5. Since ANNs are suitable for modeling non-linear data, the wind direction was modeled directly (i.e. the transformation described above was not used). Because the measurement interval of the data considered in this study was significantly longer (2 minutes) than the measurement interval of the data used in Chapter 5 (1 second), the variability of the data from one interval to the next was larger than that in the data used in Chapter 5. For this reason, and because the synthetic anomalies (which will be described shortly) were less dramatic (compared to the normal variability of the processes) than the observed anomalies described in Chapter 5, a narrower prediction interval (80%) was used in this study. Furthermore, this narrower PI, coupled with the increased variability between

consecutive measurements (especially in the cases of the windspeed and temperature measurements), resulted in the AR_ADET method exhibiting a higher false positive rate than that which was observed in Chapter 5; therefore, the ADAM method tended to replace valid measurements misclassified as anomalous with its own estimate, based on prior data, which perpetuated the effect of classification errors over the long term and degraded the performance of the anomaly detectors. For this reason, the AD strategy was used for the windspeed and temperature data streams.

To demonstrate the efficacy of these anomaly detection methods, their performance will be discussed using synthetic anomalies, as well as actual data anomalies that were identified within the December meteorological data.

6.2.2 Detection of Synthetic Anomalies

Synthetic anomalies affecting data from December 2-5 are used to compare the performance of the anomaly detection methods developed in this study with each other, as well as with the AR_ADET method. Synthetic anomalies are used for this comparison, because there are not enough known anomalies in the historical sensor data to evaluate the relative performance of the methods, and because it is difficult to know the true classification of observed data, which is necessary for false positive/negative calculations.

The synthetic errors were specified to be transient errors (i.e. they did not persist for long periods of time), and were generated according to the following equation:

$$M^* = M \pm \Delta \quad (6.1)$$

where M^* is the anomalous measurement, M is the true measurement, and Δ is an offset. The offsets for windspeed, wind direction, air temperature, and barometric pressure were selected based on judgement, to be [4-12] kt, [45-180] dfn, [3-9] °C, and [10-30] mb, respectively; when domain knowledge was not available, they were set intuitively. Synthetic anomalies of this type were randomly introduced into each of the eight data streams independently with a frequency of 5%. Since the anomalies in one sensor data stream were independent of other previous and current anomalies, both concurrent anomalies in different data streams and multiple sequential anomalies in a single data stream, could occur sequentially.

Figure 6.5 illustrates the false positive rates of the BCI-kf, BCI-rkf, MAP-ms, and AR_ADET methods, where the false positive rate is the ratio of the number of data misclassified as anomalous to the total number of non-anomalous data. Because of the stochastic nature of particle filtering, the results for the MAP-ms detectors were averaged over five replicates. As can be seen in this figure, the BCI-rkf and MAP-ms detectors performed significantly better than the AR_ADET and BCI-kf detectors. This result is not surprising, because both the BCI-kf method (as discussed in Section 2.5) and the AR_ADET method are highly sensitive to anomalous measurements, and because the AR_ADET method cannot take advantage of information in other data streams that may help the classifier discern between an anomalous and a normal measurement. There is little difference between the false positive rates of the BCI-rkf detector and the MAP-ms detectors that use more than 1000 particles, with the MAP-ms-100k method showing a

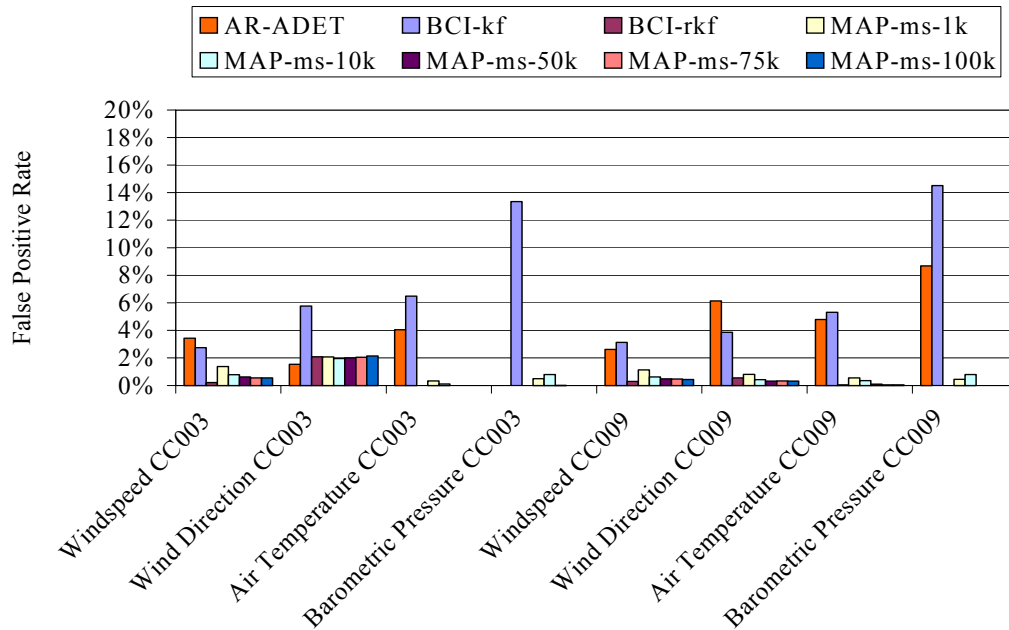


Figure 6.5: False positive rates for the BCI-kf, BCI-rkf, MAP-ms, and AR_ADET detectors for classifying transient synthetic anomalies.

slight advantage over the BCI-rkf method. These results indicate that a conceptual model that accounts for outlying measurements is better suited for describing systems in which measurements can be anomalous.

Figure 6.6 illustrates the false negative rates of the three Bayesian detection methods and the AR_ADET method, where the false negative rate is the ratio of the number of data misclassified as anomalous to the total number of non-anomalous data. Again the results of the MAP-ms method were averaged over five replicates. As can be seen in this figure, the AR_ADET and BCI-kf methods were again outperformed by the BCI-rkf and the MAP-ms methods. However, there is an obvious difference between the BCI-rkf method and the MAP-ms method with fewer than 50,000 particles, which results in increased

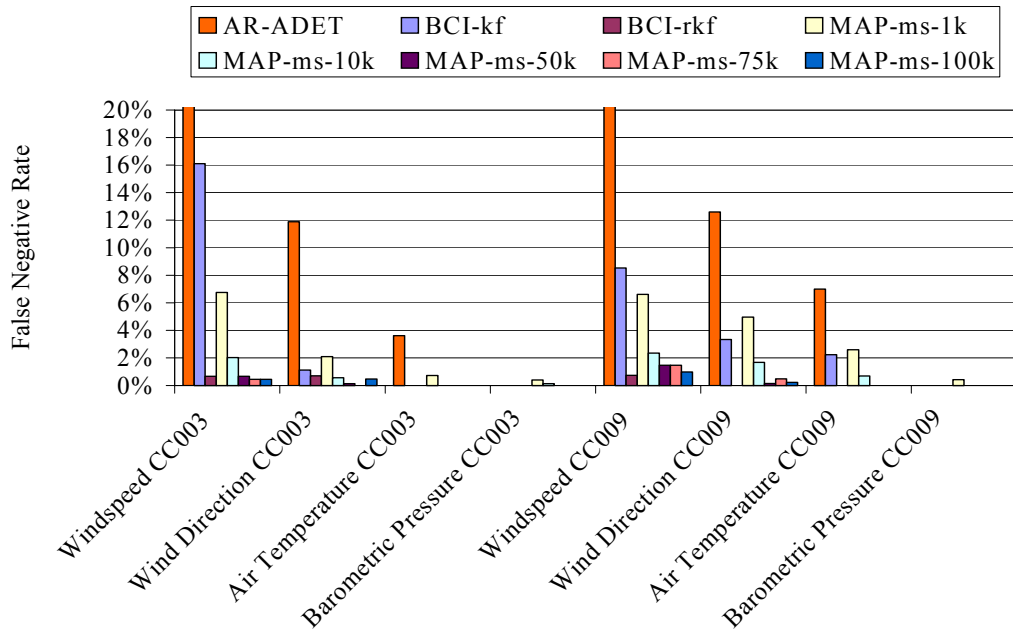


Figure 6.6: False negative rates for the BCI-kf, BCI-rkf, MAP-ms, and AR_ADET detectors for classifying transient synthetic anomalies.

false negative rates of the MAP-ms-1k and MAP-ms-10k detectors. This difference can be explained by the difference between robust Kalman filtering and Rao-Blackwellized particle filtering, with respect to their representations of the non-Gaussian state distribution caused by anomalous measurements. As described in Chapter 2, the robust Kalman filter uses the optimal Gaussian representation of a mixture-of-Gaussians distribution, while the Rao-Blackwellized particle filter uses a cloud of particles to resolve the state distribution. The latter approximation of the non-Gaussian state distribution has more descriptive power, but the quality of this approximation is highly dependent on the number of particles used. As the number of particles increases, the particle filter can better represent regions of low probability within the state distribution. Because data anomalies are, by definition, infrequent events, and because the synthetic

anomalies considered here are independent, the probability of multiple concurrent anomalies is finite but small (e.g. the probability of an anomaly occurring at the same time in three data streams is $0.05^3=0.000125$); thus, anomalous events fall into the low probability regions of the state distribution. Therefore, as the number of particles increases, the particle filter can better represent regions of the state distribution that correspond to multiple anomalies, thus resulting in a lower false negative rate. From this explanation, it would seem that in order to ensure that the particle filter could represent the case of anomalies occurring simultaneously in all eight data streams, the particle filter would need to use at least 25 billion particles. Thankfully, however, these results indicate that the MAP-ms method performs very well with significantly fewer particles, though it does require approximately 50,000 particles to match the performance of the BCI-rkf method on this set of data.

In this analysis, the DBNs considered all eight sensor data streams at once to perform coupled anomaly detection; thus, the detectors could take advantage of correlated data being measured by other sensors. To demonstrate the beneficial effect of considering multiple data streams in the DBN model, three additional detectors using the MAP-ms method of anomaly detection, but addressing only windspeed data, were created. The first DBN considered only the windspeed measurement from the CC003 platform, the second considered only the windspeed from the CC009 platform, and the third considered the windspeed from both platforms to perform coupled detection. Parameterization of these DBNs was performed using the same method described in Section 6.2.1 for the MAP-ms detector; however, because of the reduced dimensionality, 1000 particles were

sufficient. Figure 6.7 shows the false positive and false negative rates of these detectors for identifying synthetic anomalies in the December 2-5 data. From this figure, it is clear that coupling the anomaly detection process significantly reduces the false negative rate.

During the period of December 2-5, there were 14 missing measurements in the data from the CC009 sensors. Because the AR_ADET method requires a particular number of previous measurements to be available in order to process a new measurement, these missing data rendered the AR_ADET detectors for the CC009 sensors unable to classify 427, 427, 44, and 107 measurements from the windspeed, wind direction, air temperature, and barometric pressure data streams, respectively. On the other hand, since the Bayesian anomaly detection methods presented in this chapter do not require any fixed set of measurements to be available, the BCI-kf, BCI-rkf, and MAP-ms methods were able to classify all of the available measurements.

Figure 6.8 illustrates the time (averaged over five replicates) required by each of the Bayesian anomaly detectors and the AR_ADET detector to classify a new measurement. Timing was performed on a Suse Linux workstation equipped with AMD dual core Opteron 1.8 GHz processors and 7 GB of memory. Since the AR_ADET detector only operates on one data stream at a time, the time plotted in Figure 6.8 is equal to the time needed to process a new measurement in a single data stream multiplied by eight (the number of data streams concurrently processed by the Bayesian anomaly detectors). From this figure, it can be seen that the AR_ADET and BCI-kf methods are the fastest (requiring only 0.004 and 0.002 seconds, respectively), followed by the BCI-rkf and the

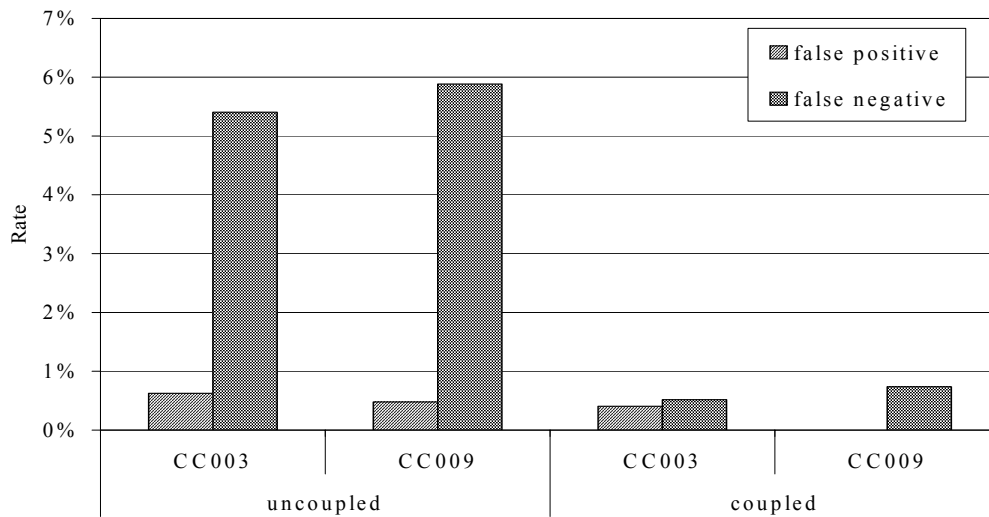


Figure 6.7: Comparison of uncoupled and coupled MAP-ms anomaly detection methods for classifying transient synthetic anomalies in the windspeed data.

MAP-ms methods. The MAP-ms-1k detector has a speed nearly equivalent to the BCI-rkf method (~0.6 sec.); however, as described above, in order to match the performance of the BCI-rkf method, the MAP-ms method must use 50,000 particles, requiring approximately 30 sec. per measurement. As expected, the MAP-ms method scales linearly with the number of particles used in the Rao-Blackwellized particle filter. This result is illustrated in Figure 6.9. Since the measurement frequency of the sensors considered in this study is two minutes, all of the anomaly detection methods are quite viable.

6.2.3 Detection of Observed Anomalies

The synthetic anomalies considered in the previous subsection provide a good comparison of the different methods. This subsection will describe how the MAP-ms-50k method, which performed best on the synthetic anomalies, performs on two real data

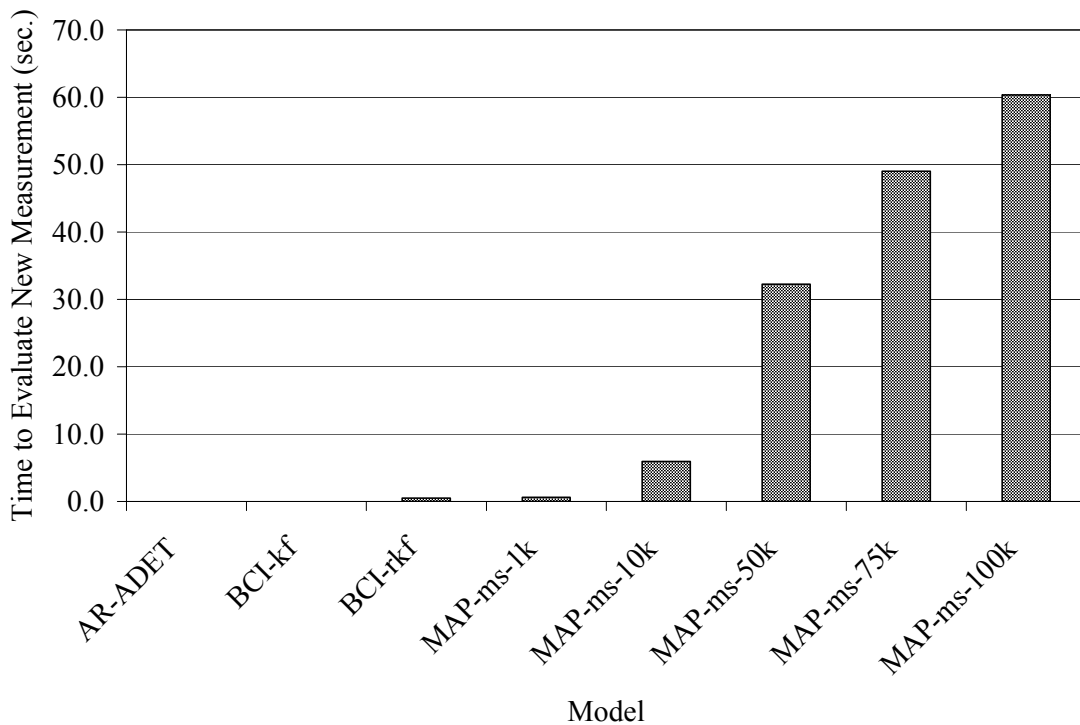


Figure 6.8: Time required by AR_ADET, BCI-kf, BCI-rkf, and MAP-ms methods to evaluate a new measurement vector. Times are averaged over five replicates.

anomalies observed within the December meteorological data. The first anomalous event occurs around midnight on December 16. The anomalous measurements caused by this event were first identified by the anomaly detectors developed in this study and subsequently brought to the attention of the data managers, who suggested that the anomalous data were errors caused by the failure of the CC009 barometer. The second anomalous event, which the SERF data managers attributed to the arrival of a storm front, occurs around 04:00 on December 1.

Figures 6.10 and 6.11 show a 24-hour segment of data spanning from 12:00 December 15 through 12:00 December 16, from the CC003 and CC009 detectors, respectively. From

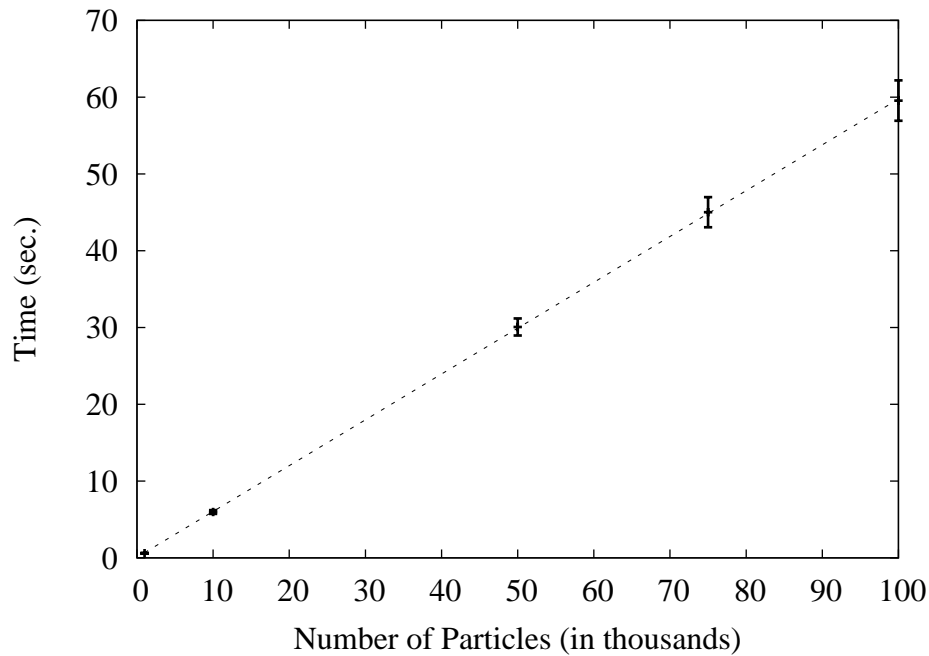


Figure 6.9: Scale up of MAP-ms method, with respect to the number of particles used in the Rao-Blackwellized particle filter. The error bars indicate one standard deviation, and the dashed line indicates the linear least squares regression of the points ($r^2=1$).

these figures, it can be seen that the CC003 wind direction sensor goes offline at approximately 19:30 December 15 and that all the CC009 sensors go offline at approximately 06:00 December 16. Furthermore, the CC009 barometer reports a large transient deviation at 21:00 December 15, as well as a rapid decrease followed by a rapid increase of pressure starting at 02:00 December 15 and continuing until the sensor goes offline. This behavior, according to the SERF data managers, is indicative of a barometer failure on platform CC009. Further evidence that this event was caused by a sensor failure, rather than by a system anomaly, is found by considering the behavior of the CC003 barometer, which does not echo the behavior of the CC009 barometer. A similar

event to this one also occurred at approximately 09:00 on October 2. Neither of these events was identified during SERF's manual QA/QC regimen, but both were identified by all three of the detection methods presented in this chapter during preliminary work.

Figures 6.12 and 6.13 show the MAP-ms-50k classification of the meteorological data from 12:00 December 15 to 12:00 December 16 from the CC003 and CC009 detectors, respectively. These figures show that the MAP-ms detector can effectively identify the anomalous measurements caused by the CC009 barometer failure. This result indicates that the assumption of independence used in the anomaly definition of the MAP-ms method does not adversely affect the detector's ability to identify persistent failures. The assumption of independence does, however, cause the detector to classify a couple of CC009 barometric pressure data points at approximately 03:00 as normal, because these data fall into the expected range of the barometric pressure, given previous CC009 measurements and the current CC003 measurement. These figures also show that the detector makes only a few false positive classifications (i.e. normal data classified as anomalous) on the data, though it does appear that the false positive rate of the CC009 wind direction data has increased slightly from what was expected, given the results of Section 6.2.2. Further inspection reveals that this slight increase in false positive rate occurs after the wind direction sensor on the other platform (CC003) has gone offline and after the barometer on the same platform (CC009) has begun malfunctioning. These sensor failures, however, do not appear to increase the false positive rate in the other seven data streams; thus, since these failures only resulted in a marginal increase in the number of false positives on one data stream, this method appears to be robust to the

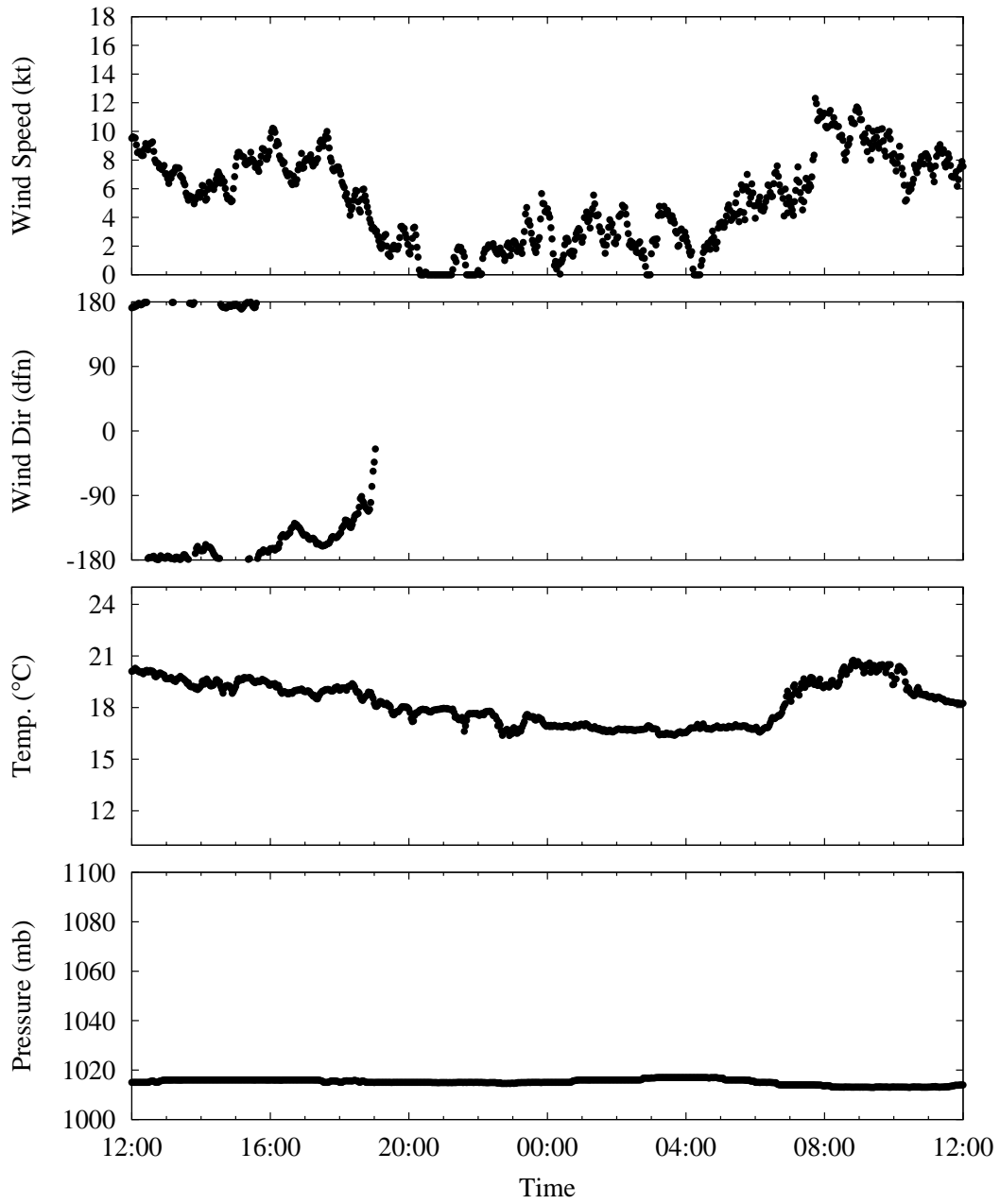


Figure 6.10: December 15-16, 2007 sensor measurements from platform CC003.

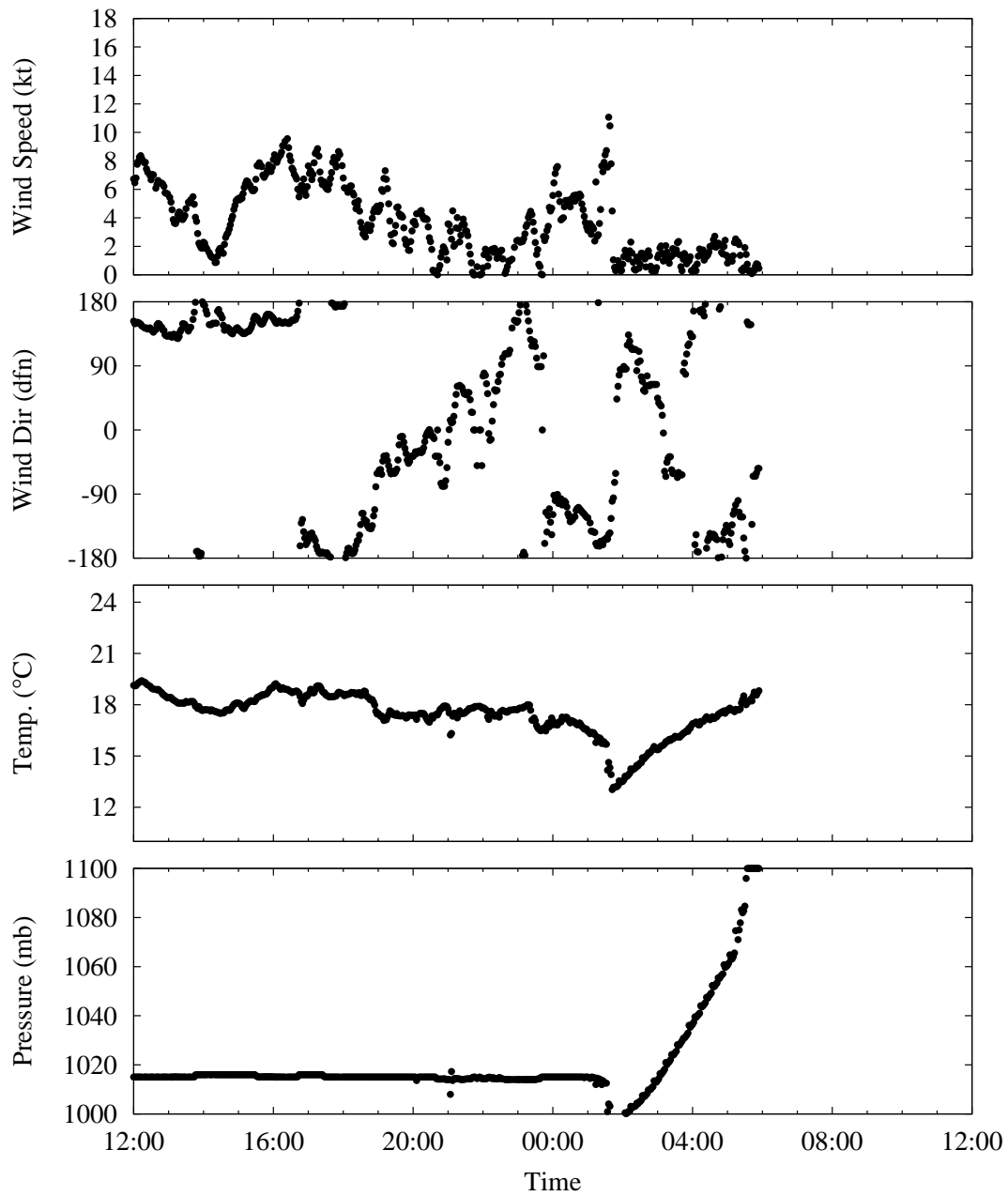


Figure 6.11: December 15-16, 2007 sensor measurements from platform CC009.

These data show the effects of a barometer failure at approximately 02:00.

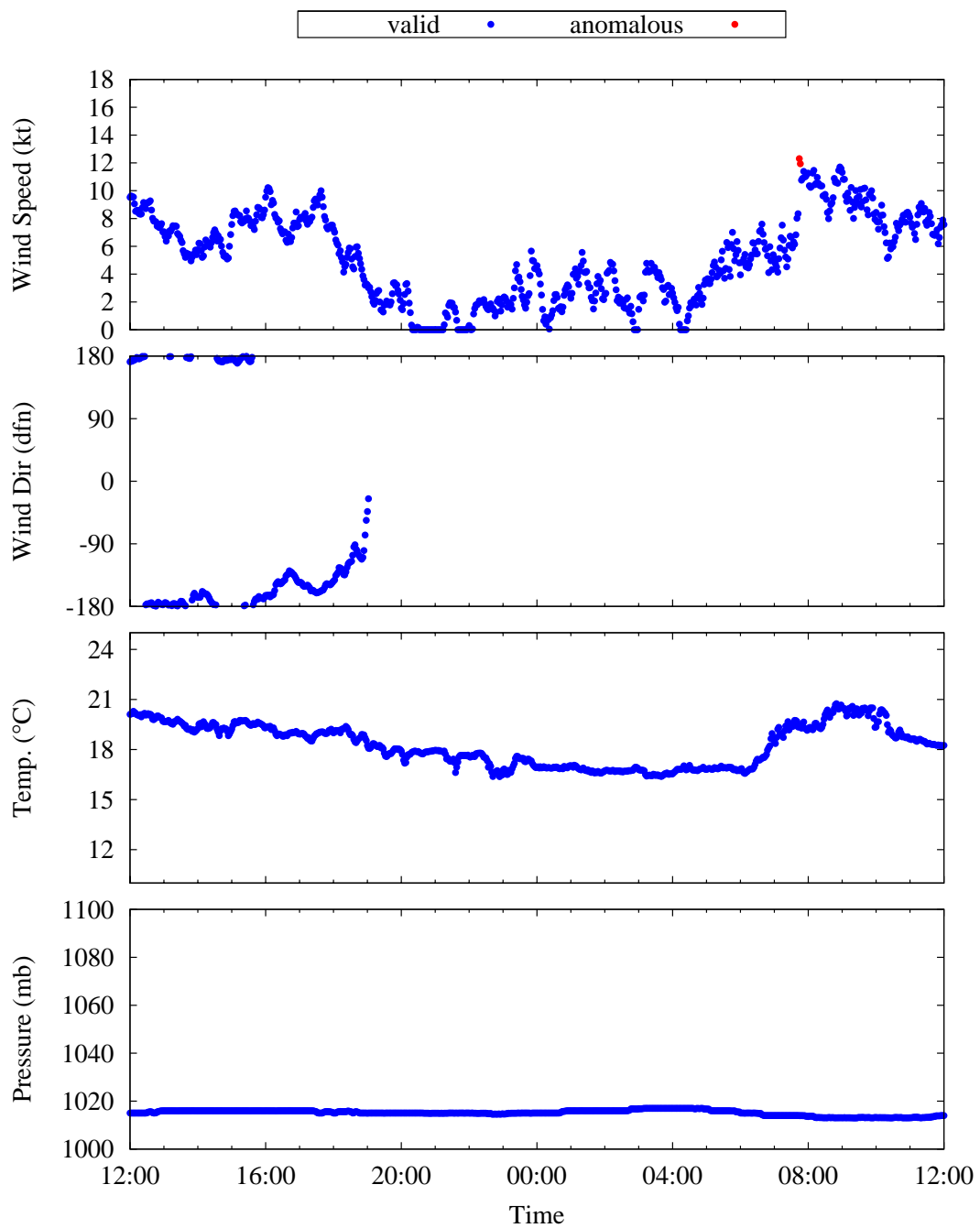


Figure 6.12: Classification of the December 15-16, 2007 sensor measurements from platform CC003 by the MAP-ms-50k detector.

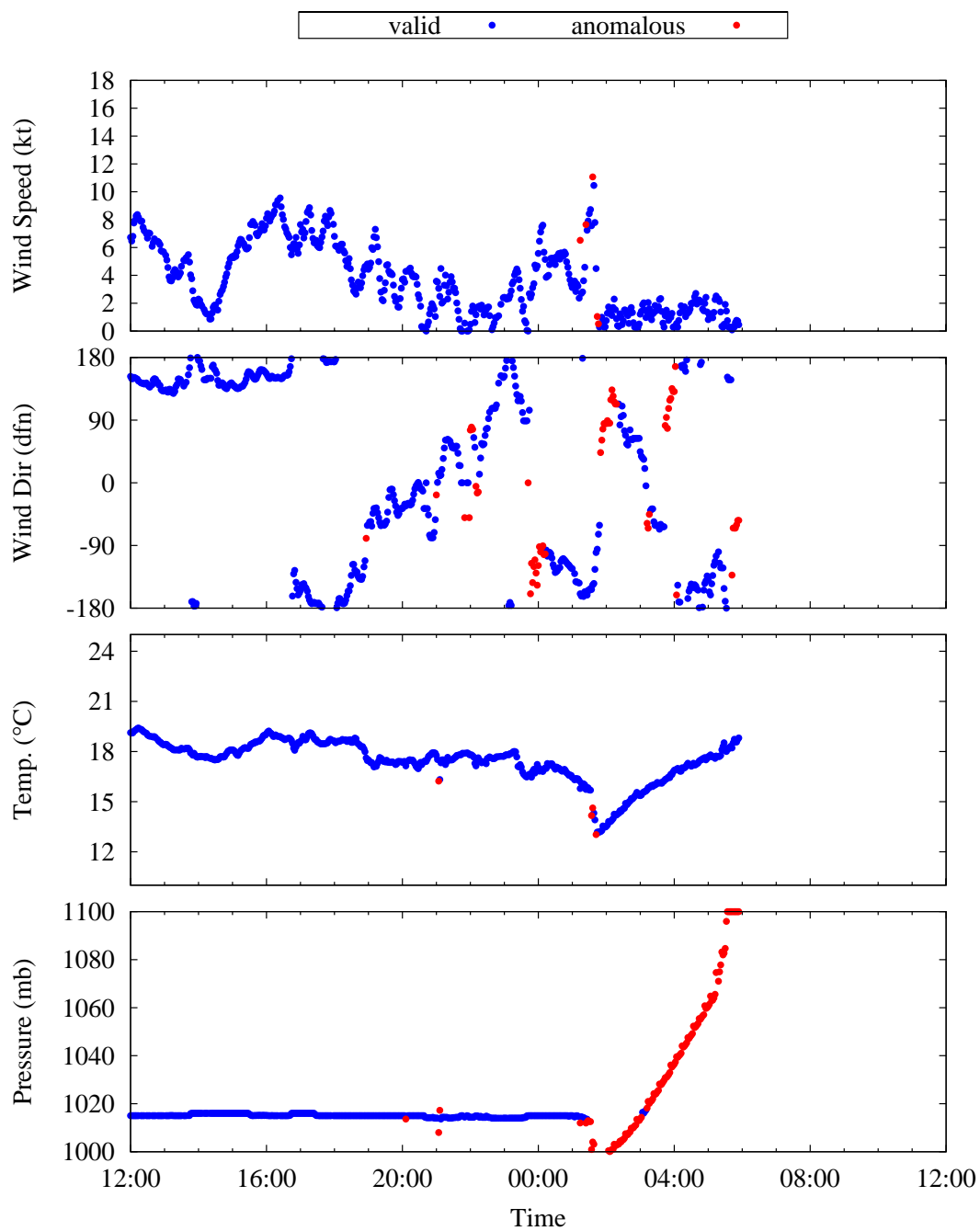


Figure 6.13: Classification of the December 15-16, 2007 sensor measurements from platform CC009 by the MAP-ms-50k detector.

failure of up to two sensors for this case study. Though the results are not shown, the MAP-ms detector with 1k, 10k, and 100k particles also exhibited behavior similar to that shown in Figures 6.12 and 6.13, as did the BCI-rkf detector.

Figures 6.14 and 6.15 show a 24-hour segment of data corresponding to December 1 from the CC003 and CC009 detectors, respectively. From these figures, it can be seen that at approximately 03:40, the windspeed increases dramatically (maximum rate of change of approximately 15kt over 2 minutes), the wind direction changes from southerly to northerly, the temperature drops dramatically (maximum rate of change of approximately 5° C over 2 minutes), and the barometer rises. Furthermore, the corresponding sensors on both sensor platforms report similar observations. This behavior, according to the SERF data managers, is indicative of the arrival of a severe storm front (an infrequent event).

Figures 6.16 and 6.17 show the MAP-ms-50k classification of the meteorological data from December 1 from the CC003 and CC009 detectors, respectively. These figures show that the MAP-ms detector identifies data corresponding to the severe changes in the windspeed and temperature as anomalous. This result indicates that the anomaly detectors are not only able to identify anomalies caused by sensor failures but also those caused by infrequent system behaviors. These figures also show that the MAP-ms detector behaves differently on the windspeed data than on the temperature data. Following the large change in windspeed, the MAP-ms detector quickly returns to classifying the majority of the windspeed data as normal; however, it continues to

classify data as anomalous for approximately 12 hours following the initial large increase in windspeed, at a rate higher than the expected false positive rate. Following the initial windspeed increase, the windspeed variability is higher than usual (as should be expected during a storm), and the data that are classified as anomalous appear to represent more extreme deviations from the general windspeed pattern than the data that are classified as non-anomalous. Following the large drop in temperature, however, the MAP-ms detector does not return to classifying the temperature measurements as non-anomalous for approximately five hours. This difference in behavior is related to the state transition model used by the MAP-ms detector, as well as the assumption in the MAP-ms method that the anomalies are independent in time. Since, in general, the historical data indicate that the windspeed changes more rapidly than the temperature, the MAP-ms detector requires more evidence (in the form of measurements) in order to change its belief state about the air temperature than it requires to change its belief state about the windspeed. Once the belief state of the MAP-ms detector has changed to reflect the decrease in temperature caused by the storm, it ceases to classify new measurements as anomalous because, following the initial decrease in temperature, the observed air temperature does not exhibit higher variability than usual. On the other hand, after the initial increase in windspeed, the variability of the windspeed remains larger than usual; thus, even though the MAP-ms detector quickly reflects the increased windspeed, it continues to classify data that exhibit large deviations from the general wind pattern as anomalous. If the MAP-ms method had assumed that the anomalies were correlated in time, rather than time independent, the detector would have continued to classify the windspeed and temperature data as anomalous for a longer period of time following the initial

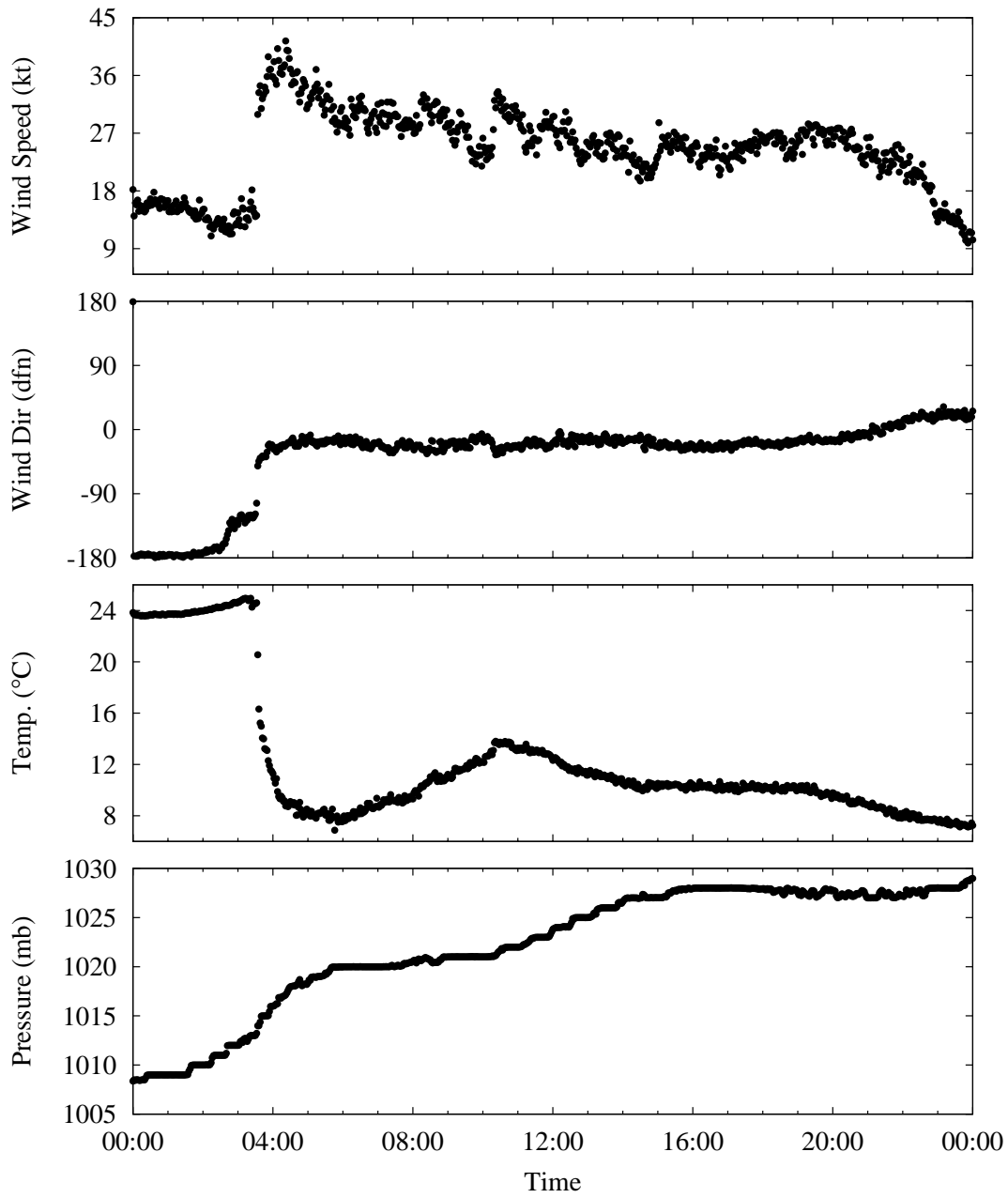


Figure 6.14: December 1, 2007 sensor measurements from platform CC003.

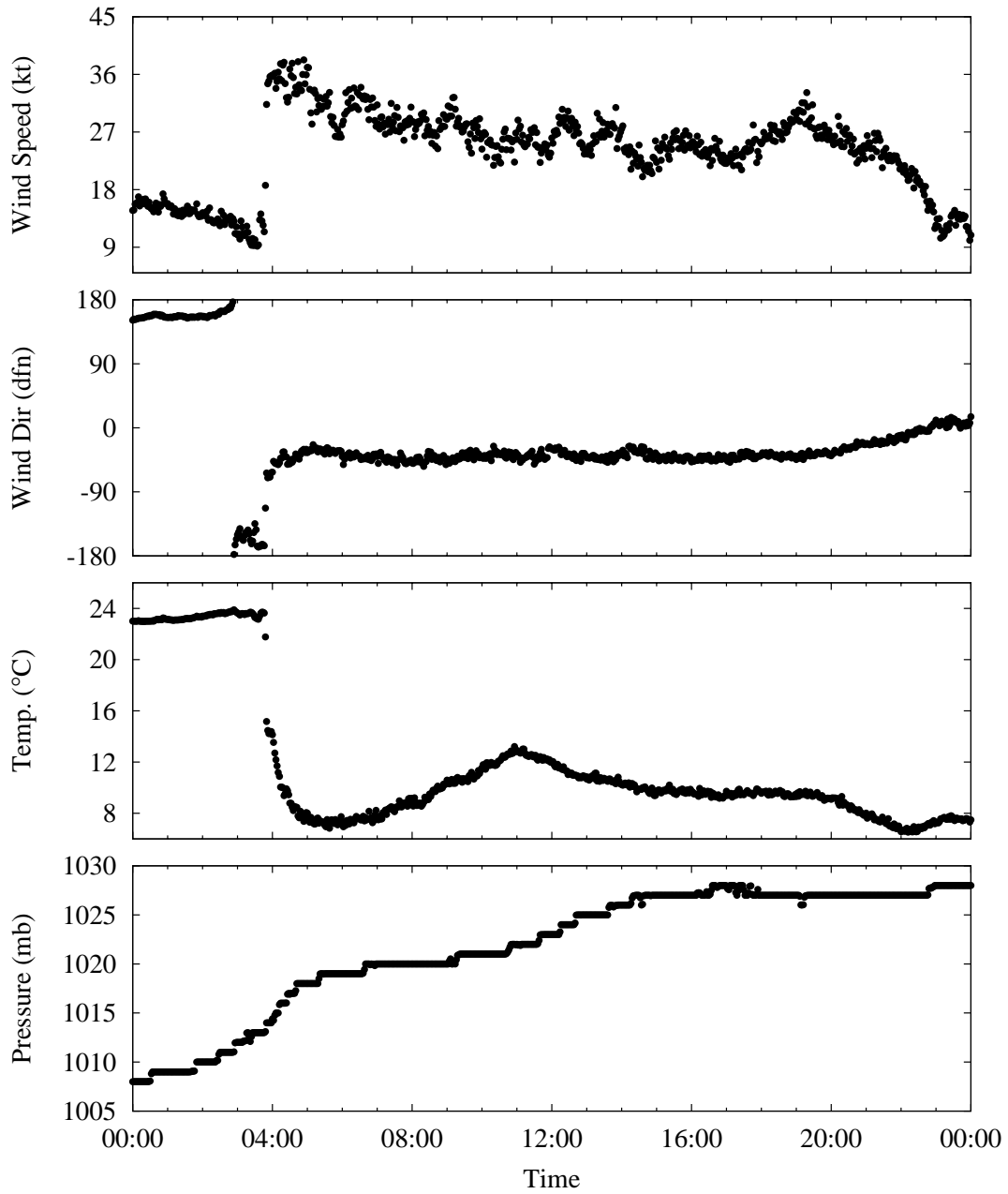


Figure 6.15: December 1, 2007 sensor measurements from platform CC009.

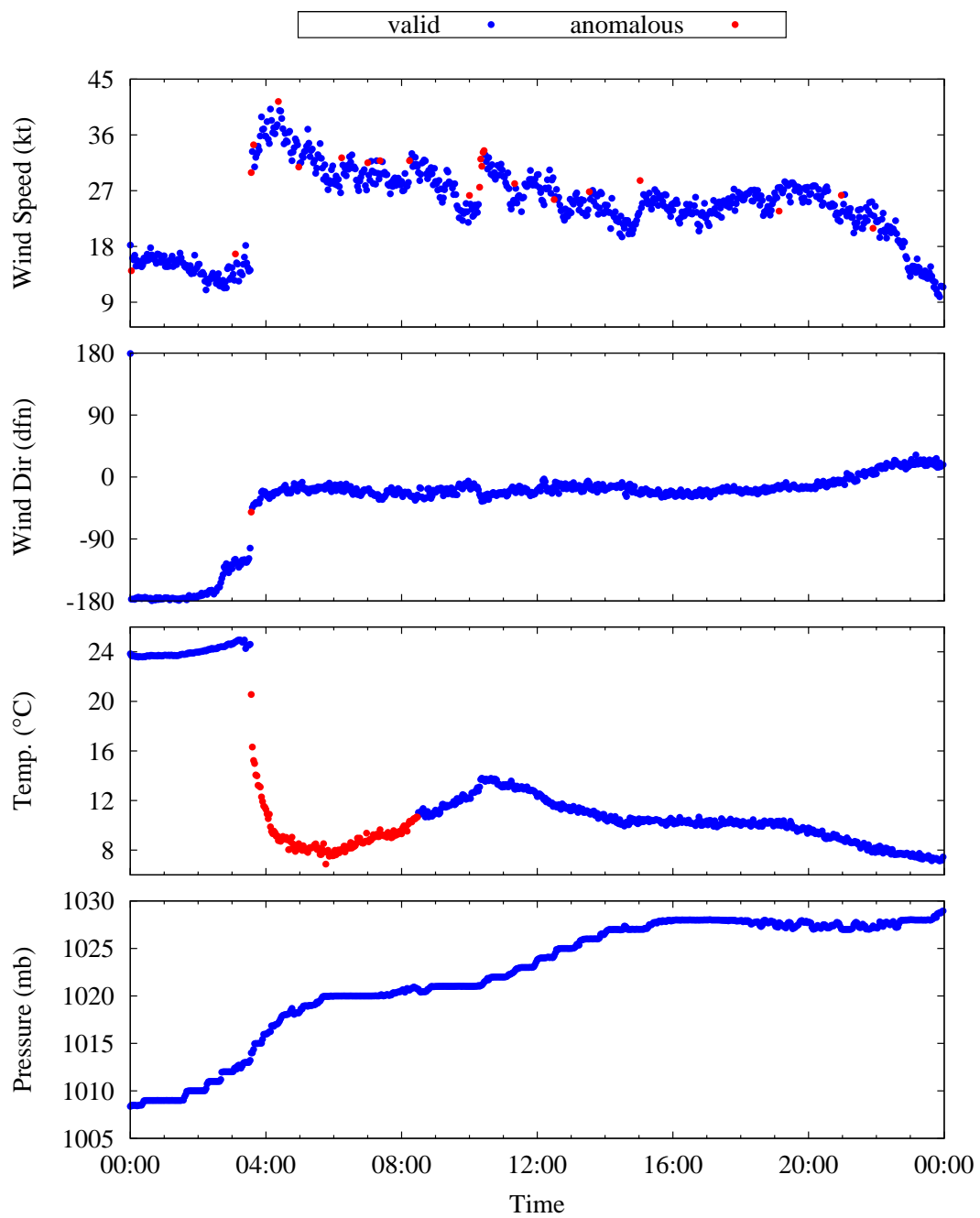


Figure 6.16: Classification of the December 1, 2007 sensor measurements from platform CC003 by the MAP-ms-50k detector.

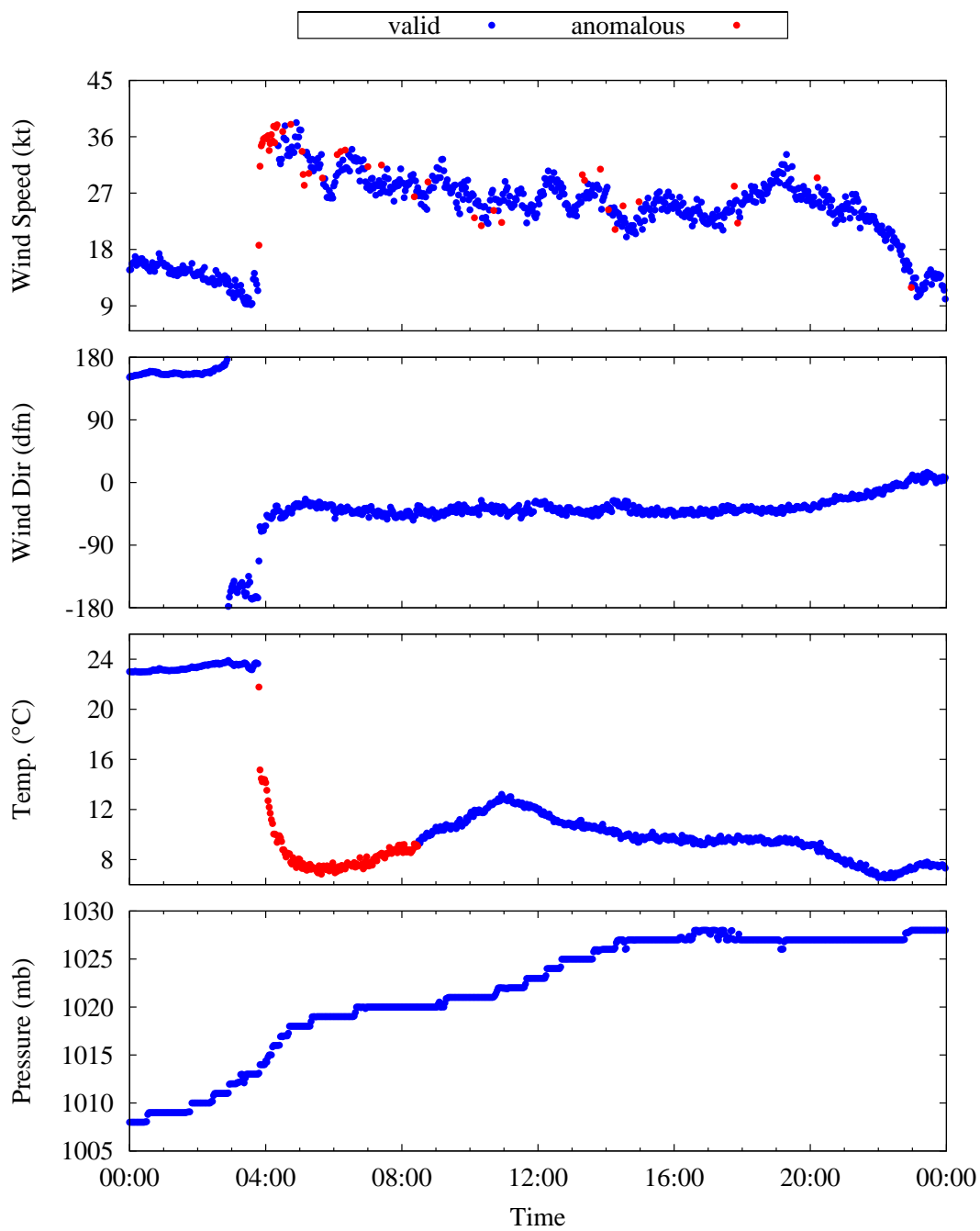


Figure 6.17: Classification of the December 15-16, 2007 sensor measurements from platform CC009 by the MAP-ms-50k detector.

increase/decrease than was observed in Figure 6.16 and 6.17. Though the results are not shown, the MAP-ms detector with 1k, 10k, and 100k particles also exhibited behavior similar to that shown in Figures 6.16 and 6.17, as did the BCI-rfk detector.

6.3 Discussion

The previous section demonstrated that the BCI-rkf and MAP-ms anomaly detection methods developed in this study can reliably identify anomalies in the SERF meteorological data (windspeed, wind direction, air temperature, and barometric pressure) collected at two spatial locations within Corpus Christi Bay, despite the employment of DBNs with linear system transition models. Although DBNs exist that can address non-linear dynamics directly, simple data transformations can be sufficient to allow the use of filtering algorithms like robust Kalman filtering or Rao-Blackwellized particle filtering, which achieve significant computational economy through the use of linear transition models. For example, recall that the wind direction variable was transformed into two variables that varied smoothly between -1 and 1 by calculating the cosine and sine of the wind direction. Because of the narrow range over which these transformed variables vary, and because of the relationship between them, these variables are themselves non-linear. However, the case study demonstrates that DBNs using a linear transition model were successful at performing anomaly detection on the transformed wind direction. This result suggests that these methods can address sensors measuring smoothly varying data, regardless of whether the process is linear or non-linear. For highly non-linear processes, however, transformations that project the data into a higher dimensional space, in which it behaves linearly, should be considered. For

example, if the process behaves quadratically, then a new state variable and corresponding measurement, which represents the square of the process state variable, could be added.

For processes whose measurements do not vary smoothly, more complex DBNs should be explored. One important example of a process whose measurements do not vary smoothly is rainfall, since rain often starts suddenly. Because of the binary nature of rainfall modes (i.e. raining/not raining), a DBN that explicitly models these modes should be used. Such a model would also require indicators of when the mode would change (e.g. Doppler radar measurements).

For this case study, the BCI-rkf and MAP-ms anomaly detectors were able to evaluate new measurements more quickly than the measurement frequency of the sensors, so they can be used to detect anomalies in real time. Because other sensor arrays on which the anomaly detectors developed in this study could be deployed may contain substantially more sensors than the Corpus Christi Bay meteorological sensor array, and because additional sensors could always be added to any existing array, it is valuable to consider how the time complexity of the anomaly detection methods would scale as a function of the number of sensors. Since robust Kalman filtering and Rao-Blackwellized particle filtering have commonalities with Kalman filtering, the analysis will begin with Kalman filtering. Assuming that there are n sensors, each measuring one process variable, and that each DBN state variable corresponds to one of the n sensor measurements, the system state covariance matrix (matrix Σ in Equation 2.3) and the measurement

covariance matrix (matrix R in Equation 2.3) will have dimensions $n \times n$. Propagating the state distribution forward to the next measurement time and updating for the new measurements via Equation 2.3 requires a finite number of matrix multiplications and inversions of $n \times n$ matrices. Using the Coppersmith-Winograd algorithm, each multiplication or inversion is an $O(n^{2.376})$ process (Coppersmith & Winograd 1990); thus, the time complexity of the BCI-kf detector will scale approximately with the square of the number of sensors. As discussed in Section 2.5.2, robust Kalman filtering is basically a weighted average of k Kalman filters, where k is the number of Gaussian components. Since the number of Gaussian components of the BCI-rkf detector is related to the number of sensors as 2^n , the BCI-rkf detector will scale as $O(2^n n^{2.376})$. As discussed in Section 2.5.3, the Rao-Blackwellized particle filter is essentially a population of p Kalman filters, where p is the number of particles; thus, the MAP-ms detector will scale as $O(p n^{2.376})$. As discussed in the previous section, the upper limit for p is inversely proportional to the frequency of the least likely measurement status, though to achieve a performance similar to that of the BCI-rkf method, the number of particles, p , needs to be much larger than 2^n . These results suggest that p increases faster than an exponential function of the number of sensors.

From this analysis, it appears that the time complexity of both the BCI-rkf and MAP-ms detectors scales faster than an exponential function of the number of sensors, thus suggesting that these methods would be intractable for a very large number of sensors. However, this analysis assumes that all of the sensor data streams modeled by the DBNs are highly correlated, thus requiring a fully-coupled model of the processes being

measured. For processes that are marginally correlated, however, a fully-coupled model would not be necessary. Decoupling weakly correlated processes within the DBN framework does not significantly affect the quality of the DBN model (Boyen and Koller 1998) and would result in significant computational economy.

Another issue to consider is the effect of different sensor sampling frequencies on the anomaly detection algorithms. All of the data considered in the case study represent two-minute averages of the processes being measured. Compared to the one-second averaged data considered in Chapter 5, these data have higher variability between chronologically sequential measurements. Thus, as the averaging interval or temporal support increases, it is expected that the data variability will also increase. Furthermore, some sensor measurements, like those considered in this study, are time averages, while other sensors (e.g. radar) make point measurements. Thus, care must be exercised when modeling sensors operating at different measurement frequencies or sensors measuring both time-averaged and point data.

One approach to incorporating sensors with different measurement support is to ignore the support of the measurements. In this method, the greatest common factor of the different measurement frequencies is used as the measurement frequency of the DBN, and observations for the measurements are only available when the sensors actually report measurements. For example, if there are two sensors, one with a two- and one with a four-minute measurement frequency, then the DBN would be set up such that its time slices occur every two minutes. For the first sensor, a measurement would be

available at every DBN time slice, while for the second sensor, a measurement would be available at every other DBN time slice. This method would be most effective at merging sensors with similar measurement support such as data streams containing only point measurements or a combination of point measurements and measurements averaged over a small interval. Since this method does not account for the measurement support, it may be undesirable for merging data from sensors with widely varying averaging intervals. In this case hierarchal DBNs (Murphy 2002) may be helpful. Hierarchal DBNs have a graphical structure that represents different time granularities. In the example of two sensors operating at frequencies of two and four minutes, respectively, the DBN would expect two measurements of the two-minute sensor for every measurement of the four-minute sensor. The limitation of using a hierarchal DBN for real-time anomaly detection is that classification of recent measurements would occur at the frequency of the least frequent sensors. Using the example above, classification of the first of the two-minute measurements would have to wait until the four-minute measurement was taken. Detection could still take place in near-real-time if the least frequent measurement interval was reasonably short (e.g. on the order of hours).

The Bayesian framework of the anomaly detection methods presented in this chapter also renders them well-suited for concurrently processing multiple non-stationary data streams that may contain many missing values. Since these methods consider multiple correlated data streams at the same time (unlike the anomaly detection methods suggested in Chapter 5), they can process data immediately following one or more missing values in a particular data stream, thus giving them an advantage over the detection methods

presented in Chapter 5. Furthermore, information from other sensors improves the classification accuracy of the detectors and may be instrumental in identifying certain types of anomalies, such as those caused by the slow drift of one sensor. Coupling the detection process, however, increases the computational burden by increasing the dimensionality of the state distribution that must be propagated from one measurement interval to the next. For this reason, it may be desirable to decouple marginally correlated processes to decrease the computational demand, especially when the number of sensors becomes large.

The performance of the anomaly detectors can also be used to guide expansion of the existing sensor array or even to design new sensor arrays. Since the detectors appear to be sensitive to the inclusion of highly correlated data (e.g. data from the same type of sensor at a different spatial location), data streams in which anomalies are more difficult to detect likely contain information that is not well described by information in the other sensor data streams, and the sensor array would benefit from the addition of a new sensor in the same spatial region that measures a similar process. For example, consider a sensor array composed of a single windspeed sensor on platform CC003. Figure 6.7 shows that the false negative rate of the detector operating on this array is high (5%), indicating that this array provides insufficient information to describe the windspeed at platform CC003. This result could prompt sensor managers to deploy a sensor that collects data correlated with the CC003 windspeed—such as the windspeed sensor on platform CC009. As demonstrated in Figure 6.7, the anomaly detector operating on the resulting array composed of two windspeed sensors at CC003 and CC009 has significantly fewer false

negatives, indicating that the new array more robustly measures the windspeed at CC003. A quick comparison with the detector operating on the CC009 windspeed data alone indicates that the array containing both sensors more robustly measures the windspeed at platform CC009 as well.

The anomaly detectors may also be useful for helping researchers determine where to locate new sensors in the environment. To accomplish this goal, the DBN employed by the anomaly detector would first be used to create forecasts of the system state at a future time. Then an interpolation of these forecasts and uncertainty estimate (e.g., using kriging or cross-validation) over the spatial area of interest (in this case Corpus Christi Bay) would be performed. This process would provide a measure of the uncertainty in the spatial estimates that could then be used to identify locations within the system where new sensor platforms should be considered.

Because DBNs do not require that the processes they model be stationary, but only that the process dynamics be stationary, the DBNs employed by the BCI and MAP-ms detectors would only have to be retrained if the dynamics of the system were to change. While it is unlikely that system dynamics will change quickly, periodic re-parameterization of the DBN may be desirable, so that new data streams or new information about the types of anomalies that may be encountered can be included in the DBN. Parameter learning is somewhat time-consuming, requiring several minutes on a RedHat Linux workstation equipped with an Intel Xeon 2.4GHz processor and 1 GB of memory. However, a dual model approach (also suggested in Chapter 5) can be used, in

which a new model is trained while the previous model is being used for anomaly detection, such that data to be processed do not back up while re-parameterization of the DBN is occurring.

The results of the case study indicate that the more complex representation of the system used by the MAP-ms method resulted in better performance than the less complex representation used by the BCI-rkf method, though this improvement was marginal and required a significant increase in computational time. This result indicates that for the meteorological sensors considered in this study, the BCI-rkf method would likely be preferable to the MAP-ms method. However, the DBN framework employed by the anomaly detectors can address significantly more complex models of the system dynamics than those used here. For example, a description of system anomalies (i.e. infrequent system responses such as a severe storm) can be added to the DBN. Like the measurement anomalies addressed in this study, system anomalies can be represented either through a Gaussian mixture model for the system state distribution or through a discrete variable indicating whether or not a system anomaly has occurred. The former strategy permits the use of robust Kalman filtering, while the latter requires the use of particle filtering. This modification would aid the detectors in distinguishing between anomalies caused by infrequent system responses (like the storm front described in the previous section) and those caused by faulty sensors or data transmission errors—a valuable distinction for real-time QA/QC. Additionally, seasonal or diurnal behaviors or other behaviors of non-linear dynamics could be incorporated within the DBN through the use of discrete state variables indicating the season, time of day, or other “switch”

indicating a particular dynamic mode. For these more complex DBNs, particle filtering (e.g. Rao-Blackwellized particle filtering) would be required. Thus, it may be necessary to investigate methods to improve the efficiency of particle filtering, such as variational methods, which use a state approximate distribution with a more compact representation than conventional particle filters, (Murphy 1999).

6.4 Conclusion

This chapter presents three Bayesian anomaly detection methods employing DBNs and compares the well-known Kalman filter to the robust Kalman filter and the recently developed Rao-Blackwellized particle filter, which have not yet found wide application in environmental research. These DBNs were implemented such that they are robust to missing values in the sensor data streams by adaptively modifying the filtering method to use only the available measurements. The Bayesian anomaly detection methods perform fast, incremental evaluation of data as they become available; can scale up to large quantities of data; and require no *a priori* information, regarding process variables or the types of anomalies that may be encountered. Furthermore, these methods can process data from multiple sensors at the same time, and thus, as demonstrated, can be applied to a network of heterogeneous sensors.

The value and efficacy of the BCI-kf, BCI-rkf, and MAP-ms anomaly detection methods are illustrated using a case study involving eight data streams, including windspeed, wind direction, air temperature, and barometric pressure, at two spatial locations within Corpus Christi Bay. In this case study, the performance of these detectors was evaluated using a

suite of synthetic and actual data anomalies. Synthetic anomalies permitted direct comparison of the three methods with each other, as well as with the autoregressive data-driven anomaly detector (AR_ADET) presented in Chapter 5. This comparison indicated that all the methods require less time to evaluate a new measurement than the frequency at which the measurements are collected; thus, all the methods are suitable for real-time anomaly detection. Additionally, comparison indicated that the BCI-rkf and MAP-ms methods misclassified significantly fewer data points than did the BCI-kf or AR_ADET methods. The BCI-rkf and MAP-ms methods also performed well at identifying anomalous data caused by two real anomalous events. In the first event, one of the sensors failed, resulting in corrupted data that manual QA/QC had failed to detect. The second event was caused by the passage of a particularly severe storm front. The detection of both system anomalies, such as the storm front, and measurement anomalies, caused by the failing sensors, indicates that even with no *a priori* information about the types of anomalies that could be encountered, the Bayesian anomaly detectors were effective at identifying real anomalies in the data.

If anomaly detection were to be incorporated into an adaptive sampling method, then the non-specific identification of both types of anomalies would be acceptable (since it is often desirable to know under what conditions the sensors fail). However, if anomaly detection were to be incorporated into a real-time QA/QC or into a data cleaning system, then identifying both system and measurement anomalies might be undesirable. Since both the Rao-Blackwellized particle filter and the robust Kalman filter could be modified to address these types of anomalies, both the BCI-rkf and MAP-ms methods are suitable

for this application. However, given that the particle filter employed in the MAP-ms method is capable of filtering more complex DBNs than the robust Kalman filtering algorithm used in the BCI-rkf method, the MAP-ms method would be able to address more complex systems than would the BCI-rkf method. This suggests that the MAP-ms method may be preferable to the BCI-rkf method for complex environmental systems.

Because both process and measurement anomalies are of interest to researchers, and distinguishing the two types of anomaly can be useful in many instances, development of detectors capable of making this distinction would clearly be beneficial. While there are existing models (e.g. Koushanfar *et al.* 2003) that explain how sensor failure affects data, the parameters of these models are specific to each deployed sensor, and sufficient instances of sensor failure have not been observed to parameterize these models, especially for new sensor deployments (such as the Corpus Christi meteorological array). Deployment of one of the DBN-based anomaly detectors presented here on a sensor array would make possible the accumulation of a labeled set of data corrupted by sensor failure that could be used to create better models of sensor failures. These models could then be incorporated into the DBN-based anomaly detector, such that the detector could recognize known types of sensor failures. This feature would be beneficial for suggesting remedial action on the sensor, as well as for aiding the detector in discriminating between measurement and process anomalies.

Chapter 7: Concluding Remarks

Understanding and predicting the behavior of large-scale environmental systems is necessary for addressing many challenging problems of environmental interest, such as (1) the design of groundwater remediation strategies, (2) the development of early warning systems for natural disasters, like hurricanes or tsunamis, and (3) the understanding of conditions that cause natural events of concern, like hypoxia in the Gulf of Mexico. However, due to issues relating to the scalability of predictive models and the unavailability of parameters for these models, it is difficult to apply predictive models to large-scale systems. This research investigates the use of data mining for addressing challenging problems of environmental interest related to these two issues.

Data mining employs computational techniques from statistics, machine learning, pattern recognition, and other disciplines to extract knowledge from data. Traditionally, data mining has been used by businesses and financial institutions. Recently, however, it has increasingly come to be used in the sciences, to extract information from the large quantities of data being generated from experimentation and observation. In this research, data mining was used to address two complex problems of environmental interest: (1) upscaling models of solute transport in porous media, and (2) anomaly detection in environmental sensor data streams.

Chapter 4 investigated the development of upscaled solute transport models using genetic programming (GP), a domain-independent modeling tool that searches the space of mathematical equations for one or more equations that describe a set of training data. An

upscaling methodology was developed that facilitated both the GP search and the implementation of the resulting models. A case study demonstrated this methodology by developing vertically-averaged equations of solute transport in perfectly-stratified aquifers. The solute flux models developed for the case study were analyzed for parsimony and physical meaning, resulting in an upscaled model of the enhanced spreading of the solute plume, due to aquifer heterogeneity, as a process that changes from predominantly advective to Fickian. This case study not only demonstrates the use and efficacy of GP as a tool for developing upscaled solute transport models, but it also provides insight into how to approach more realistic multi-dimensional problems with this methodology.

Chapter 5 developed a real-time anomaly detection method for environmental data streams, which can be used to identify data that deviate from historical patterns. The method is based on an autoregressive data-driven model of the data stream and its corresponding prediction interval. It performs fast, incremental evaluation of data as it becomes available, scales to large quantities of data, and requires no *a priori* information regarding process variables or the types of anomalies that may be encountered. Furthermore, this method can be easily deployed on a large heterogeneous sensor network. Sixteen instantiations of this method were compared based on their ability to identify measurement errors in a windspeed data stream from Corpus Christi, Texas. The results indicated that a neural network model of the data stream, coupled with replacement of anomalous data points, performs well at identifying erroneous data in that data stream.

Chapter 6 developed three automated anomaly detection methods that employ dynamic Bayesian networks (DBNs). Unlike the method presented in Chapter 5, these methods considered several data streams at once using all of the streams concurrently to perform coupled anomaly detection. Dynamic Bayesian networks are Bayesian networks with network topology that evolves over time, adding new state variables to represent the system state at the current time. Filtering (e.g. Kalman filtering or Rao-Blackwellized particle filtering) can then be used to infer the expected values of unknown system states, as well as the likelihood that a particular sensor measurement is anomalous.

Measurements with a high likelihood of being anomalous are classified as such. Like the method in Chapter 5, these methods perform fast, incremental evaluation of data as it becomes available; scale to large quantities of data; and require no *a priori* information regarding process variables or the types of anomalies that may be encountered.

Furthermore, these methods can be easily deployed on a large network of heterogeneous sensors. This study investigated these methods' abilities to identify anomalies in eight meteorological data streams from Corpus Christi, Texas, and compared them to the best-performing method from Chapter 5. The results indicated that DBN-based detectors, using either robust Kalman filtering or Rao-Blackwellized particle filtering, outperform a DBN-based detector using Kalman filtering and the autoregressive data-driven anomaly detection method of Chapter 5 at identifying synthetic anomalies. These methods were also successful at identifying data anomalies caused by two types of real-world events: sensor failure and large storms.

The results of this research thus indicate that data mining can be used to improve predictive modeling of large-scale systems. In the future, extension of this work should proceed in several directions. First, although the GP upscaling method presented in Chapter 4 is not limited to one-dimensional upscaled models, the case study illustrating this method does not address the creation of multi-dimensional upscaled models. The creation of multi-dimensional upscaled models of solute transport using GP will require learning a model for a multi-dimensional vector quantity. Therefore, ensuring that mass is conserved will be more difficult than in the one-dimensional case presented here, and thus, new objectives may be necessary to guide the GP search. Furthermore, a method should be developed to reduce the observed bias of the correlation coefficient (r^2) metric for capturing early time behavior.

Second, the quality of DBN model predictions for filling data gaps should be evaluated further explored. Because the DBN model framework allows for the coupling of correlated processes, the Bayesian anomaly detectors developed in this study show particular potential for the effective prediction of missing measurements, but more extensive testing is needed to determine their performance under a suite of sensor operating scenarios (e.g. one off-line sensor, multiple off-line sensors, or one or more malfunctioning sensors). With further testing, this feature of the anomaly detectors could be very valuable for a real-time QA/QC system that prepares data for real-time forecasting. In this case, when a sensor goes off-line, the anomaly detector would provide predictions of the missing measurements in real time to the forecasting system.

Third, it was demonstrated that Bayesian anomaly detection methods (i.e. BCI-rkf and MAP-ms method) presented in Chapter 6 are successful at identifying both transient and persistent data anomalies resulting from intentional corruption of data, sensor failure, and storm events. These types of anomalies were identified using a simple anomaly description that required no *a priori* information regarding the types of anomalies that would be encountered in the data. It may be desirable, however, to use more complex DBN models that better represent the system being monitored. Not only would more complex models permit higher quality state predictions, but they would also allow the anomaly detectors to distinguish between process anomalies (e.g. data anomalies resulting from storms) and observation anomalies (e.g. data anomalies resulting from sensor or data transmission faults). For example, Kitagawa (1987) demonstrated that using a heavy-tailed distribution (rather than a Gaussian distribution) in the state transition model of a DBN improved the accuracy of filtered results when unusually abrupt changes in time-series data occurred. This result suggests that incorporation of a non-Gaussian system dynamics model may aid the Bayesian anomaly detectors in distinguishing between process and observation anomalies. Parameterizing these models, however, will require some *a priori* information regarding both the anomalous behavior of the system being modeled and the types of measurement anomalies that could occur. This information would be necessary to permit the anomaly detectors to distinguish between system anomalies and measurement anomalies. Such information could be gathered through the deployment of an anomaly detector on an operational sensor network with operator feedback on the types of anomalies detected.

Another improvement in DBN modeling that should be investigated is the incorporation of variables to address behavioral modes, such as seasonal or diurnal variations. These variables could be represented as discrete or continuous. For example, seasonal behavior could be represented (1) discretely, by a variable with four values indicating spring, summer, fall, winter, or (2) continuously, by four variables that vary continuously from 0 to 1, indicating the influence of the seasonal behaviors on the process at a particular measurement interval. Incorporating non-Gaussian process models and behavioral modes would improve the performance of the Bayesian fault detectors presented in Chapter 6, although it would also increase their computational complexity.

Fourth, the process dynamics modeled by the DBNs in Chapter 6 are data-driven (i.e. learned from historical patterns), rather than being derived from an understanding of the underlying physics of the process. For many environmental systems, such physics-based models exist. Thus, the combination and could be incorporated into the DBN framework should be explored. One possible approach would be to create additional observation nodes in the DBN for each physics-based model output. The physics-based model could be run external to the DBN, and at every DBN time slice the physics-based model output could be considered as a sensor measurement. This is similar to how sequential data assimilation is performed (Madsen & Cañizares 1999), except that the goal would be a more accurate filtered state estimate, not calculating error statistics for the physics-based model. Another more complex method of incorporating the process physics would be to use the output of a stochastic, physics-based model as a transition model in a particle filter.

Fifth, the Rao-Blackwellized particle filter used by the MAP-ms anomaly detection method achieves significant computational economy over more traditional particle filters by decomposing the state variables into two subgroups: (1) variables that can be represented as linear Gaussian and (2) variables that cannot. This decomposition allows the Rao-Blackwellized particle filter to exploit the properties of linear Gaussian distributions that lead to the well-known Kalman filtering equations. Since many environmental processes are non-Gaussian and exhibit heavy-tailed behavior, the linear Gaussian assumption may not be the most suitable for modeling these processes. In order to exploit the Rao-Blackwellization decomposition, as well as accommodate processes with heavy-tailed distributions, a modification of the Rao-Blackwellized particle filter used in Chapter 6 should be explored. This modification would replace the Gaussian distribution of the state variables with a mixture-of-Gaussians distribution; thus, the robust Kalman filtering equations can be used to track the continuous state variables. This modified Rao-Blackwellized particle filter would still be significantly more efficient than traditional particle filtering (because there is a closed form representation of the distribution of the continuous state variables), and it would address the heavy-tailed behavior exhibited by environmental processes.

Sixth, since particle filtering will likely be necessary for investigations addressing more complex DBNs, methods to improve the efficiency of particle filtering, such as hypothesis collapsing, factorial models, and variational methods, should be explored. In hypothesis collapsing (Lerner *et al.* 2000), similar particles are merged into an aggregate

particle, reducing the number of particles. The set of aggregate particles that results from hypothesis collapsing retains particles that are sampling low probability events (e.g. multiple sensor failures), while reducing the overall number of particles needed to sufficiently characterize a DBN's belief state. Factorial models (Boyen & Koller 1998, Ghahramani & Jordan 1997) reduce the dimensionality of the belief state of a DBN by decoupling weakly correlated processes, thus improving the efficiency of filtering. Errors associated with decoupling weakly correlated processes have been shown by Boyen and Koller (1998) to diminish over time; thus, the resulting approximation is stable. Boyen and Koller (1998) have also suggested that the strong assumption of complete independence can be relaxed to an assumption of conditional independence with little loss of computational economy. Variational methods (Jordan *et al.* 1999, Murphy 1999) use additional parameters, referred to as variational parameters, to represent the belief state of a DBN more compactly. Thus, not only can variational methods be used to decouple correlated processes into independent sub-processes (similar to factorial models), but they can also be used to improve model predictions (based on the Gaussian assumption) for non-Gaussian processes. Increasing the efficiency of particle filters through the use of these methods would permit the MAP-ms method to address data from a large number of sensors, as well as measurements of processes that are both non-linear and non-Gaussian in real time.

Finally, the BCI-rkf and MAP-ms anomaly detectors should be deployed on an operational environmental sensor network. This task would not only help to thoroughly evaluate the detectors' performance over a long test period, but it would also aid

administrators by improving the quality of the collected data , since these detectors identified anomalous data caused by sensor failures that were not detected through manual QA/QC. Finally, only through long-term deployment of the method can information necessary to improve the detectors' performance be gathered. Such information would include descriptions of how different sensors behave under normal and abnormal operating conditions, which could be used to help parameterize DBNs such that they could distinguish between system and process anomalies or recognize known sensor failures, which would allow corrective action to be suggested.

This research has demonstrated the promise of emerging data mining methods for addressing challenging environmental systems problems. These methods are capable of finding patterns in data regarding complex systems that can be used to create models that provide quick and accurate forecasts of the system. Some of these models are expressed in a syntax that researchers can understand, and thus, can be analyzed for information regarding the underlying processes that generated the data. This information can lead to new understandings of the physical processes being modeled. While data mining requires large quantities of high-quality data, these data are becoming increasingly available through the deployment of sensors into the environment. Furthermore, because the computational expense of data mining increases with the expressivity of the model being generated and with the number of features in the data set (a phenomenon referred to as the “curse of dimensionality”), it is often necessary to select only salient features and to limit the model's expressivity. However, as demonstrated in this dissertation, data transformations often can be used to reduce a

problem's dimensionality and facilitate the application of data mining methods to problems of environmental interest.

References

- Amenu, G., Markus, M., Kumar, P., & Demissie, M. (2007). Hydrologic applications of Minimal Resource Allocations Network (MRAN). *Journal of Hydrologic Engineering*, 12(1), 124-229.
- Ananthanarayanan, V., & Holbert, K. E. (2004). Power System Sensor Failure Detection and Characterization using Fuzzy Logic. In *Proceedings of the 7th IASTED International Conference on Power and Energy Systems*, 291-296.
- Anderson, B. D. O., & Moore, J. B. (1979). *Optimal filtering*. Englewood Cliffs, NJ: Prentice-Hall.
- Aris, R. (1956). On the dispersion of a solute in a fluid flowing through a tube. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 235(1200), 67-77.
- Arulampalam, M. S, Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing* 50(2), 174-188.
- Babovic, V. & Abbott, M. B. (1997b). The evolution of equations from hydraulic data Part II: Applications. *Journal of Hydraulic Research*, 35(3), 411-430.
- Babovic, V. , Keijzer, M., Aquilera, D., & Harrington, J. (2001). An evolutionary approach to knowledge induction: Genetic programming in hydraulic engineering. In *Proceedings of the World Water and Environmental Resources Congress*.
- Babovic, V., & Abbott, M. B. (1997a). The evolution of equations from hydraulic data Part I: Theory. *Journal of Hydraulic Research*, 35(3), 397-410.

- Babovic, V., & Bojkov, V. H. (2001). *Runoff modelling with genetic programming and artificial neural networks*. D2K (Tech. Rep. No. D2K TR 0401-1). Hørsholm, Denmark, Danish Hydraulic Institute-Water and Environment.
- Balas, C. E. Koç, L., & Balas, L. (2004). Predictions of missing wave data by recurrent neuronets. *Journal of Waterway, Port, Coastal and Ocean Engineering*, 130(5), 256-265.
- Banzhaf, W., & Langdon, W. B. (2002). Some considerations on the reason for bloat. *Genetic Programming and Evolvable Machines*, 3, 81–91.
- Banzhaf, W., Nordin, P., Keller, R. E., & Francone, F. D. (1998). *Genetic Programming- An Introduction; On the Automatic Evolution of Computer Programs and its Applications*. San Francisco: Morgan Kaufmann.
- Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41(1), 164-171.
- Beckie, R. (1998). Analysis of scale effects in large-scale solute-transport models. In G. Sposito (Ed.) *Scale dependence and scale invariance in hydrology*, (pp. 314-334), New York: Cambridge University Press.
- Beckie, R. Aldama, A. A., & Wood, E. F. (1996). Modeling the large scale dynamics of saturated groundwater flow using spatial-filtering theory: 1. Theoretical development. *Water Resources Research*, 32(5), 169-1280.
- Belle, R., Upadhyaya B., & Skorska, M. (1983). Sensor fault analysis using decision theory and data-driven modeling of pressure water reactor subsystems. *Nuclear Technology* 64, 70-77.

- Bertino, L., Evensen, G., & Wackernagel, H. (2002). Combining geostatistics and Kalman filtering for data assimilation in an estuarine system. *Inverse Problems*, 18, 1-23.
- Bishop, C. (1995). *Neural Networks for Pattern Recognition*, New York: Oxford University Press.
- Blum, R. S., Zhang, Y., Sadler, B., and Kozick, R. (1999). On the approximation of correlated non-Gaussian noise pdfs using Gaussian mixture models. In *The 1st Conference on the Applications of Heavy Tailed Distributions in Economics, Engineering and Statistics*.
- Bolton, R. J., & Hand, D. J. (2001). Unsupervised profiling methods for fraud detection. In *Proceedings of Credit Scoring and Credit Control VII*, 5-7.
- Bonner, J. S., Kelly, F. J., Michaud, P. R., Page, C. A., Perez, J., Fuller, C., Ojo, T., & Sterling, M. (2002). Sensing the Coastal Environment. In *Proceedings of the 3rd International Conference on EuroGOOS; Building the European Capacity in Operational Oceanography*, 167-173.
- Box, G., & Jenkins, C. (1970). *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day Inc.
- Boyen, X., & Koller, D. (1998). Tractable inference for complex stochastic processes. In *Proceedings of the Conference on Uncertainty in AI*.
- Boyen, X., & Koller, D. (1999). Exploiting the architecture of dynamic systems. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*.
- Broemling, L. (1985). *Bayesian analysis of linear models*. New York: Marcel Dekker.

- Bulut, A., Singh, A. K., Shin, P., Fountain, T., Jasso, H., Yan, L., & Elgamal, A. (2005). Real-time nondestructive structural health monitoring using support vector machines and wavelets. In N. Meyendorf, G. Y. Baakline, and B. Michel (Eds.), *Proceedings of the SPIE Advanced Sensor Technologies for Nondestructive Evaluation and Structural Health Monitoring*, Vol. 5770, 180-189.
- Casella, G., & Robert, C. P. (1996). Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1), 81-94.
- Coppersmith, D., & Winograd, S. (1990). Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9, 251–280.
- Dagan, G. (1984). Solute transport in heterogeneous porous formations. *Journal of Fluid Mechanics*, 145, 151–177.
- Dawid, A. P. (1973). Posterior expectation for large observations. *Biometrika*, 60, 664-667.
- DePold, H., Volponi, A., Siegel, J., & Hull, J. (2003). Validation of diagnostic data with statistical analysis and embedded knowledge. In *Proceedings of the American Society of Mechanical Engineers, International Gas Turbine Institute, Turbo Expo IGTI*, Vol. 1 , 573-579.
- Devore, J. L. (1995). *Probability and Statistics for Engineering and the Sciences* (4th Ed.). Pacific Grove, CA: ITP/Duxbury.
- Digalakis, V., Rohlicek, J. R., & Ostendorf, M. (1993). ML Estimation of a stochastic linear system with the EM algorithm and its application to speech recognition, *IEEE Transactions on Speech and Audio Processing*, 1(4), 431-442.

- Doucet, A., de Freitas, N., Murphy, K., & Russell, S. (2000a). Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI2000)*, 176-183.
- Doucet, A., Godsill, S., & Andrieu, C. (2000b). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3), 197-208.
- Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification*. New York; Wiley-Interscience.
- Efendiev, Y., Durlafsky, L. J., & Lee, S. H. (2000). Modeling of subgrid effects in coarse-scale simulations of transport in heterogeneous porous media. *Water Resources Research*, 36(8), 2031–2041.
- Efron, B., Olshen, R. A. (1978). How broad is the class of normal scale mixtures? *The Annals of Statistics*, 6(5), 1159—1164.
- Elliott, A. J., & Jones, B. (2000). The need for operational forecasting during oil spill response. *Marine Pollution Bulletin*, 40(2), 110-121.
- Elman, J. (1990). Finding structure in time. *Cognitive Science*, 14, 179-212.
- Fantoni, P. F., & Mazzola, A. (1996). Multiple failure signal validation in nuclear power plants using artificial neural networks. *Nuclear Technology*, 113, 368-374.
- Fischer, H. B., List, E. J., Kob, R. C. Y., & Brooks, N. H. (1979). *Mixing in Inland and Coast Waters*. New York: Academic Press.
- Frankel, I., & Brenner, H. (1989). On the foundations of generalized Taylor Dispersion theory. *Journal of Fluid Mechanics* 20, 97-119.
- Frühwirth, R. (1995). Track fitting with long-tailed noise: a Bayesian approach. *Computer Physics Communications*, 85, 189-199.

- Gelhar, L. W., & Axness, C. L. (1983). Three-dimensional stochastic analysis of macrodispersion in aquifers. *Water Resources Research*, 19(1), 161-180.
- Gelhar, L. W., Gutjahr, A. L., & Naff, R. L. (1979) Stochastic analysis of macrodispersion in a stratified aquifer. *Water Resources Research*, 15(6), 1387-1397.
- Ghahramani, Z. and Hinton, G.E. (1996). *Parameter estimation for linear dynamical systems* (Tech. Rep. No. CRG-TR-96-2). University of Toronto, Department of Computer Science.
- Ghahramani, Z., & Jordan, M. I. (1997). Factorial hidden Markov models. *Machine Learning*, 29, 245-275.
- Ghil, M., & Malanotte-Rizzoli, P. (1991). Data assimilation in meteorology and oceanography. *Advances in Geophysics*, 33, 141-266.
- Giannella, C., Han, J., Pei, J., Yan, X., & Yu, P. S. (2002). Mining frequent patterns in data streams at multiple time granularities. In H. Kargupta, A. Joshi, K. Sivajumar, & Y. Yesha, (Eds.), *Proceedings of the NSF Workshop on Next Generation Data Mining*. AAAI/MIT.
- Goel, P., Dedeoglu, G., Roumeliotis, S. I., & Sukhatme, G. S. (2000). Fault detection and identification in a mobile robot using multiple model estimation and neural network. In *Proceedings of the IEEE International Conference on Robotics and Automation*.
- Goovaerts, P. (1997). *Geostatistics for Natural Resource Evaluation*. New York: Oxford University Press.

- Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *Proceedings of the Institute of Electrical Engineering, F*, Vol. 140, 107-113.
- Gross, A. M. (1973). A robust confidence interval for location for symmetric, long-tailed distributions. In *Proceedings of the National Academy of Science* 70(7), 1995-1997.
- Guinaso, N. L., Jr., Yip, J., Reid, R. O., Bender III, L. C., Howard, M., Lee III, L. L., Walpert, J. N., Brooks, D. A., Hetland, R. D., & Martin, R. D. (2001). Observing and Forecasting Coastal Currents: Texas Automated Buoy System (TABS), In *Proceedings of OCEANS 2001 MTS/IEEE*, 1318-1322.
- Güven, O., Molz, F. J., & Melville, J. G. (1984). An analysis of dispersion in a stratified aquifer. *Water Resources Research*, 20(10), 1337-1354.
- Gwo, J., D'Azevedao, E., Frenzel, H., Mayes, G. Y., Jardine, P., Salvage, K., & Hoffman, F. (2001). HBGC123D: A high-performance computer model of coupled hydrogeological and biogeochemical processes. *Computers & Geosciences*, 27, 1231-1242.
- Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques* (2nd Ed.). New York: Morgan Kaufmann.
- Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning*. New York: Springer-Verlag.
- Hecht-Nielsen, R. (1990). *Neurocomputing*. New York: Addison-Wesley.
- Hodge, V. J., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review* 22,85-126.

- Huang, W., Murray, C., Kraus, N., & Rosati, J. (2003). Development of a regional neural network for coastal water level predictions. *Ocean Engineering*, 30, 2275-2295.
- Jeffreys H. (1961). *Theory of Probability* (3rd Edition). Oxford, UK: Clarendon Press.
- John, G. H. (1995). Robust decision trees: Removing outliers from databases. In *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*,. 174-179.
- Johnson, L. D., & Sakoulis, G. (2003). *Maximizing Equity Market Sector Predictability in a Bayesian Time Varying Parameter Model*. Available at SSRN: <http://ssrn.com/abstract=396642> or DOI: 10.2139/ssrn.396642.
- Jordan, M. I, Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning* 37, 183-233.
- Jordan, M. I. (2002). *An Introduction to Probabilistic Graphical Models*. Unpublished manuscript.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D), 35-45.
- Keijzer, M. (2002). *Scientific Discovery using Genetic Programming*. Unpublished doctoral dissertation. Danish Technical University, Lyngby, Denmark.
- Keijzer, M., & Babovic, V. (1999). Dimensionally aware genetic programming. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith (Eds.), *Proceedings of the Genetic and Evolutionary computation Conference*, 42-49. San Francisco: Morgan Kaufmann.

- Keijzer, M., & Cattolico, M. (2002). An example of the use of context-sensitive constraints in the ALP system. In A. M. Barry (Ed.) *GECCO 2002: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference*, 128-132.
- Keijzer, M., Babovic, V., Ryan, C., O'Neill, M & Cattolico, M. 2001. Adaptive logic programming. In L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke (Eds.) *Proceedings of the Genetic and Evolutionary computation Conference*, 42-49. San Francisco: Morgan Kaufmann.
- Kitagawa, G. (1987). Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82(400), 1032-1041.
- Kitanidis, P. K. (1992). Analysis of macrodispersion through volume-averaging moment equations. *Stochastic Hydrology & Hydraulics*, 6, 5-25..
- Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence* 97, 273-324.
- Koller, D., & Lerner, U. (2000). Sampling in Factored Dynamic Systems. In A. Doucet, J. F. G. de Freitas, & N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag.
- Koushanfar, F., Potkonjak, M., & Sangiovanni-Vincentelli, A. (2003). On-line fault detection of sensor measurements. In *Proceedings of IEEE Sensors*, Vol. 2, 974-979.
- Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

- Koza, J. R., Bennett, F. H. Andre, D., & Keane, M. A. (1999). *Genetic Programming III: Darwinian Invention and Problem Solving*. San Francisco: Morgan Kaufmann.
- Kozuma, R., Kitamura, M., Sakuma, M., & Yokoyama, Y. (1994). Anomaly detection by neural network models and statistical time series analysis. *Neural Networks 1994, IEEE World Congress on Computer Intelligence*.
- Krajewski, W. F., & Krajewski, K. L. (1989). Real-time quality control of streamflow data—A simulation study. *Water Resources Bulletin*, 25(2), 391-399.
- Langdon, W. B., & Poli, R. (2002). *Foundations of Genetic Programming*. Heidelberg, Germany: Springer-Verlag.
- LeBlank, D. R., Garabedial, S. P., Hess, K. M., Gelhar, L. W., Wuadri, R. D. Stollenwerk, K. G., & Wood, W. W. (1991). Large-scale natural gradient tracer test in sand and gravel, Cape Cod, Massachusetts 1: Experimental design and observed tracer movement. *Water Resources Research*, 27(5), 895–910.
- Leonard, B. P. (1988) The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection. *Computer Methods in Applied Mechanics & Engineering* 88, 17-74.
- Lerner, U., Parr, R., Koller, D., & Biswas, G. (2000). Bayesian fault detection and diagnosis in dynamic systems. In *Proceedings of the AAAI*.
- Liu, X., & Goldsmith, A. (2004). Kalman filtering with partial observation losses. In *43rd IEEE Conference on Decision and Control: Vol. 4*. (pp. 4180- 4186).
- Mackay, D. M., Freyberg, D. L., Robberts, P. V. & Cherry, J. A. (1986). Natural gradient experiment on solute transport in a sand aquifer I: Approach and overview of plume movement. *Water Resources Research*, 22(13), 2017–2029.

- Madsen, H., & Cañizares, R. (1999). Comparison of extended and ensemble Kalman filters for data assimilation in coastal area modelling. *International Journal for Numerical Methods in Fluids*, 31, 961-981.
- Markus, M., (2005). Issues in designing automated minimal resource allocation neural networks. In *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks IJCNN '05*, 2671- 2673.
- Maybeck, P. S. (1979). *Stochastic Models, Estimation, and Control* (2nd Ed.). New York: Academic Press.
- Mehranbod, N., Soroush, M., Piovoso, M., & Ogunnaike, B. (2003). Probabilistic model for sensor fault detection and identification. *AIChE Journal* 49(7),1787-1802.
- Mei, C. C. (1992). Method of homogenization applied to dispersion in porous media. *Transport in Porous Media*, 9, 261-274.
- Meinhold, R. J., & Singpurwalla, N. D. (1989). Robustification of Kalman filter models. *Journal of the American Statistical Association*, 84(406), 479-486.
- Minsky, M., & Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA: MIT Press.
- More, A., & Deo, M. C. (2003). Forecasting wind with neural networks. *Marine Structures* 16(1), 35-49.
- Mourad, M., & Bertrand-Krajewski, J.-L. (2002). A method for automatic validation of long time series of data in urban hydrology. *Water Science & Technology* 45(4-5), 263-270

- Murphy, K. P. (1999). A variational approximation for Bayesian networks with discrete and continuous latent variables. In *Proceedings of the Conference on Uncertainty in AI*.
- Murphy, K. P. (2002). Dynamic Bayesian networks. In M.I. Jordan (Ed.), *An Introduction to Probabilistic Graphical Models*. Unpublished Manuscript
- Nairac, A., Townsend, N., Carr, R., King, S., Cowley, P. & Tarassenko, L. (1999). A system for the analysis of jet engine vibration data. *Integrated Computer-Aided Engineering* 6(1), 53-65.
- Nicholson, A. E., & Brady, J. M. (1994). Dynamic belief networks for discrete monitoring. *IEEE Transactions on Systems, Man, and Cybernetics* 24(11), 1593-1610.
- Nitsche, L. C., & Brenner, H. (1989). Eulerian kinematics of flow through spatially periodic models of porous media. *Archive for Rational Mechanics & Analysis*, 107(3), 225-292.
- NRC (National Research Council) (2006). *CLEANER and NSF's Environmental Observatories*. Washington, D.C.: National Academy Press.
- NSF (National Science Foundation) (2005). *Sensors for Environmental Observatories Report: of the NSF Sponsored Workshop December 2004*. Arlington, VA: NSF.
- Peña, D. & Guttman, I. (1988). Bayesian approach to robustifying the Kalman filter. In J. C. Spall (Ed.), *Bayesian Analysis of Time Series and Dynamic Models*. New York: Marcel Dekker, Inc.

- Peña, D., & Guttman, I. (1989). Optimal collapsing of mixture distributions in robust recursive estimation. *Communications in Statistics. A. Theory and Methods* 18, 817-883
- Ramanathan, N., Balzano, L., Burt, M., Estrin, D., Kohler, E., Harmon, T., Harvey, C., Jay, J., Rothberg, S., & Srivastava, M. (2006). *Monitoring a Toxin in a Rural Rice Field with a Wireless Sensor Network* (Tech. Rep. No. 62). Los Angeles: University of California Los Angeles, Center for Embedded Network Systems (CENS).
- Ramaswamy, S., Rastogi, R., & Shim, K. (2000). Efficient algorithms of mining outliers from large data sets. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 427-438.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65:386-408.
- Rousseeuw, P., & Leroy, A. (1996). *Robust Regression and Outlier Detection* (3rd Ed.). New York: John Wiley & Sons.
- Rubin, Y., Bellin, A., & Lawrence, A. E. (2003). On the use of block-effective macrodispersion for numerical simulations of transport in heterogeneous formations. *Water Resources Research*, 39(9), 1242-1252.
- Rubin, Y. 2003. *Applied Stochastic Hydrogeology*. New York: Oxford University Press.
- Rumelhart, D., McClelland, J., & The PDP Research Group (1987). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press.

- Rummelhart, D. E., Hinton, G., & Williams, R. (1986). Learning internal representations by error propagation. In D. E. Rummelhart, J. L. McClelland, and the PPD Research Group (Eds.). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, 318-62. Cambridge, MA: MIT Press, Cambridge, MA.
- Russell, S., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach* (2nd Ed.). Saddle River, NJ: Prentice Hall.
- Sastry, K., O'Reilly, U.-M., Goldberg, D. E. & Hill, D. (2003). Building-block supply in genetic programming. In R. Riolo, & B. Worzel (Eds.) *Genetic Programming in Theory and Practice*, 137–152. Boston: Kluwer Academic Publishers.
- Savic, D., Walters, G. A., & Davidson, J. W. (1999). A genetic programming approach to rainfall-runoff modelling. *Water Resources Management*, 13, 219–231.
- Schick, I. C., & Mitter, S. K. (1994). Robust recursive estimation in the presence of heavy-tailed noise. *The Annals of Statistics*. 22(2): 1045-1080.
- Schmidt, S. F. (1981). The Kalman filter: Its recognition and development for aerospace applications. *Journal of Guidance and Control*, 4, 4-7.
- Schütze, H., & Silverstein, C., (1997). Projections for efficient document clustering. In *Proceedings of SIGIR '97*, 74-81.
- Shah, K., Bonner, J., Trujillo, D., Page, C., & Kelly, F. (2005). Development of real-time data monitoring system for coastal margin research. In *Proceedings of the 2005 International Oil Spill Conference*.

- Shumway, R. H., & Stoffer, D. S. (1982). An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3(4), 253-264.
- Silvestri, G., Verona, F., Innocenti, M., & Napolitano, M. (1994). Fault detection using neural networks. *In Proceedings of the IEEE International Conference on Neural Networks*, Vol. 6, 3796-799.
- Sorenson, H. W. (1966). Kalman filtering techniques. In C. T. Leondes (Ed.), *Advances in Control Systems*, Vol 3. New York: Academic Press.
- Sorenson, H. W. (1988). Recursive estimation for nonlinear dynamic systems. In J. C. Spall (Ed.), *Bayesian Analysis of Time Series and Dynamic Models* (pp. 127-166). New York: Marcel, Dekker, Inc.
- Spall, J. C. (1988). An overview of key developments in dynamic modeling and estimation. In J. C. Spall (Ed.), *Bayesian Analysis of Time Series and Dynamic Models*. New York: Marcel Dekker, Inc.
- Sposito, G. (1997). Ergodicity and the 'scale effect'. *Advances in Water Resources*, 20(5-6), 309-316.
- Steeffel, C. I. (2001). *GIMRT, Version 1.2: Software for modeling multicomponent, multidimensional reactive transport. Users guide* (Tech. Rep. No. UCRL-MA-143182). Livermore, CA, Lawrence Livermore National Laboratory.
- Sudicky, E. A. (1986). Natural gradient tracer experiment on solute transport in a sand aquifer: Spatial variability of hydraulic conductivity and its role in the dispersion process. *Water Resources Research*, 22(13), 2069-2082.

- Tang, J., Chen, Z., Fu, A., & Cheung, D. (2002). A robust outlier detection scheme in large data sets. In *Proceedings of the 6th Pacific-Asia Conference on Knowledge Discovery and Data Mining*.
- Taylor, G. (1953). Dispersion of soluble matter in solvent flowing slowly through a tube. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 219(1137), 186–203.
- Upadhyaya, B., Glockler, O., & Eklund, J. (1990). Multivariate statistical signal processing for fault detection and diagnostics. *ISA Transactions* 29(4),79-95.
- Upadhyaya, B., Mathai, G., & Green, J. (1988). Data clustering and prediction for fault detection and diagnostics. In *Proceedings of the American Control Conference*, Vol. 1, 650-651.
- van der Merwe, R., de Freitas, J. F. G., Doucet, A., & Wan, E. A. (2000). *The Unscented Particle Filter* (Tech. Rep.). University of Cambridge, Department of Engineering,.
- Vasquez, D., & Fraichard, T. (2004). Motion prediction of moving objects: a statistical approach. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 4, 3931-3936.
- Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K., & Colucci, D. (2001). High-Performance Wide-Area Optical Tracking: The HiBall Tracking System, *Presence: Teleoperators and Virtual Environments* 10(1), 1–21.
- Whitaker, S. (1999). *The Method of Volume Averaging*. Norwell, MA: Kluwer Academic Publishers.

- Wood, B. D., Cherblanc, F., Quintard, M., & Whitaker, S. (2003). Volume averaging for determining the effective dispersion tensor: Closure using periodic unit cells and comparison with ensemble averaging. *Water Resources Research*, 39(8), 1210.
- Yonekawa, K., & Kawahara, M. (2003). Application of Kalman Filter Finite Element Method and AIC. *International Journal of Computational Fluid Dynamics*, 17(4), 307-317.
- Zhang G. P., & Qi, M. (2005). Neural network forecasting for seasonal and trend time series. *European Journal of Operational Research* 160, 501-514.

Author's Biography

David J. Hill received his Bachelor of Science degree from Cornell University, Ithaca, New York, in Civil and Environmental Engineering in 1999. During the completion of this degree, he was fortunate to be mentored by Prof. T. D. O'Rourke who sparked his interest in groundwater systems, and Prof. Edwin (Todd) Cowan who sparked his interest in numerical modeling. While at Cornell, David also participated in the Engineering Co-Operative Program, working for Cytec Industries as an Environmental Engineering Technician. After completion of his B.S. at Cornell, David decided to pursue his graduate studies at the University of Illinois at Urbana-Champaign under the guidance of Prof. Albert Valocchi. His master's thesis entitled "Modeling Nitrogen Transport and Transformation in a Heterogeneous, Three-Dimensional, Tile-Drained Aquifer" discussed the development and application of a three-dimensional physically-based model of nitrogen transport through tile-drained agricultural fields. During the completion of his master's research, David became interested in data mining and its application to environmental systems problems. In May 2002, he was admitted to the PhD program in Civil and Environmental Engineering at the University of Illinois at Urbana-Champaign, where he studied under the direction of Prof. Barbara Minsker. His research interests include the use of data mining methods (particularly statistical, machine learning, and pattern recognition tools) to create innovative and efficient solutions to complex environmental systems problems; the development of technologies that facilitate the collection of high-quality data with environmental sensors; and the use of sensor data for modeling large-scale environmental systems, modeling large and complex water resources systems, and creating environmental information technology.