USE OF DATA-DRIVEN MODELS TO IMPROVE PREDICTION OF PHYSICALLY BASED GROUNDWATER MODELS

BY

TIANFANG XU

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Civil Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2012

Urbana, Illinois

Adviser:

Professor Albert J. Valocchi

Abstract

Current analyses of groundwater flow and transport typically rely on a physically-based model (PBM), which is inherently subject to error and uncertainty from multiple sources including model structural error, parameter error and data error. The model uncertainty can be difficult to quantify, and is propagated to the prediction. In this study, complementary data-driven models (DDMs) are used to improve prediction of groundwater flow models. The DDMs, trained with the historical residual of the PBM, have the capability to compensate for the defects of PBM. Five machine learning techniques, instance-based weighting (IBW), locally weighted regression (*loess*), decision trees (DT), artificial neural networks (ANN) and support vector regression (SVR) are employed to construct the DDMs, and their performance of enhancing the prediction of the PBM is compared. Before the DDMs updating, cluster analysis is implemented on the dataset to improve the robustness and efficiency of the framework. The framework is tested in two real-world case studies based on the Republic River Compact Association (RRCA) model and the Spokane Valley Rathdrum Prairie (SVRP) model. The DDMs reduce the root-mean-square errors (RMSE) of the temporal, spatial and temporal plus spatial head prediction of the RRCA model by 82%, 60% and 48% respectively. In the SVRP case study, the DDMs reduces the temporal head forecast of the PBM by 79%. Localized DDMs that are conditioned on each cluster outperform global DDMs without clustering. It is also demonstrated that clustering significantly reduces the computational cost of training and cross validation of the DDMs. After clustering, the run-time of DDMs is negligible comparing with the PBM, which makes the framework very computationally efficient.

Acknowledgments

I am grateful to the National Science Foundation Hydrologic Science Program for financial support under Grant No. 0943627. I would like to express my gratitude to my advisor, Professor Albert J. Valocchi, for providing his insight, guidance and encouragement during this research. I also thank Dr. Yonas K. Demissie, Jaesik Choi, Professor Eyal Amir and Professor Feng Liang for helpful discussions. Finally, I would like to acknowledge Dr. Paul A. Hsieh for sharing the calibration dataset of the Spokane Valley Rathdrum Prairie model.

Table of Contents

Chapter 1

INTRODUCTION

1.1 Background

Physically-based models (PBM) of groundwater flow and solute transport are the principle quantitative tools used in subsurface water analysis and management. Along with the ever increasing availability of computational power, measurements and improved understanding of the dynamics of hydrogeologic systems, there are increasing requirements for accuracy for these models. On the other hand, the inherent uncertainty in groundwater modeling has been widely recognized in the literature [34, 35, 27, 13, 38, 45, 18].

The uncertainty comes from three primary sources: structural error, parameter error and data error. As assemblies of assumptions and simplifications, groundwater models are inevitably imperfect approximations to the true system. The structural error can arise from omission and/or misrepresentation of site characteristics and hydrogeologic processes during conceptualization [35, 38, 13, 34], as well as from the mathematical implementation, e.g. spatial and temporal discretization [30]. Parameter error comes from the difficulty to capture, on a wide range of scales, the heterogeneity of hydrogeologic environments that exhibit both systematic and random spatial variations. It is generally infeasible to obtain enough measurements to capture such hydrologic complexity [35, 34], thus the use of "effective" parameter values, as conceptual aggregate representations of heterogeneous hydrogeologic properties, is essential [30]. Often parameter values are not measurable and need to be estimated by indirect means, which may introduce uncertainty. Finally, data error refers to the measurement error and the uncertainty in forcing terms, which in most cases are specified or estimated from in situ observations.

Structural, parameter and data errors collectively lead to reducible (epistemic) and irreducible (aleatoric) error when the groundwater models are used in prediction. The lumped error is typically represented as the misfit between the observed quantity of interest and its simulated counterpart. A large portion of such misfit cannot be ascribed to measurement error, and may be mitigated in a variety of ways.

Calibration (also known as the inverse problem) is the most common practice to achieve better prediction by reducing parameter uncertainty [27]. The traditional approach of calibration is to first postulate a deterministic model structure, then tune the values of parameters until the simulation results of the model match corresponding observations to a satisfactory degree. In the last decade, sophisticated automatic calibration software based on nonlinear regression, like PEST [17], have been developed and popularized. Techniques like regularization and dimension reduction have been proposed to deal with the non-uniqueness problem and to allow additional parameterization complexity [16, 42, 34]. In contrast with PEST, which seeks the least-square estimate of parameter values, the Shuffled Complex Evolution algorithm (SCE) was proposed in [20] as a global optimization strategy applicable to a broad class of single-criterion calibration problems. The algorithm was further extended to multiobjective complex optimization that enables the use of multiple complementary measures [50]. The SCE algorithm and its multiobjective version were later adapted to fit into a stochastic framework. [46] presented an efficient Markov Chain Monte Carlo sampler called the Multiobjective Shuffled Complex Evolution Metropolis (MOSCEM) algorithm, which converges to an ensemble of parameter sets instead of a single estimation. This algorithm utilizes the concept of Pareto optimality, and uses many points along the Pareto front to parameterize the model when making predictions.

The above approaches capture at best the larger-scale variations of subsurface hydrogeologic properties, failing to resolve heterogeneity exhibited at smaller scales [35]. Attempting to infer the complexity of the reality from limited field measurements, the inverse problem is inherently ill-posed and has infinite number of solutions. A unique solution can be obtained via zonation, using pilot points and/or Tikhonov regularization, however with the price of a loss of detail in the calibrated field. In [34], Moore and Doherty noted that the estimated parameter value at any point is a weighted average of the true hydrogeologic property over a much larger area. They further argued that, as a result of the calibrated model being unable to replicate the detail of the true system, the model-to-measurement misfit can show a high degree of spatial correlation. MOSCEM, on the other hand, is able to account for uncertainties associated with model parameters [46]. In the groundwater modeling community, high-resolution Monte Carlo methods have been developed that describe the spatial variability and scaling of hydrogeologic medium properties geostatistically. This family of methods generates multiple random realizations of model parameters on a fine grid, then analyzes the predictive results of all realizations statistically. The generated model parameter fields should be conditioned on field measurements of the hydrogeologic properties as well as observations of the state variables (e.g. water levels). Generating and conditioning a sufficient number of realizations makes this methods very time consuming [37]. The computational cost associated with these Monte-Carlo based methods makes them infeasible for many groundwater models especially when there are many parameters.

Beven and Freer [4] pointed out that for a complex environmental system, satisfactory agreement with observations can be achieved by many different model structures and parameter sets. Although the stochastic methods stated above account for parameter error by allowing for exploration in the parameter space, they are still based on a single model conceptualization or structure, like the traditional calibration approaches. It has been widely recognized in the literature that an inadequate model structure or conceptualization is far more detrimental to its predictive accuracy than suboptimal parameter values [35, 22, 38, 30, 25]. In addition, as suggested by Doherty [18], the model parameters might be over-adjusted during calibration to compensate for the model structure defects, which would deteriorate the prediction. As a result, predictions based on a single conceptual model are subject to statistical bias and/or underestimation of uncertainty.

Multimodel approaches based on a suit of conceptual models have been developed to explicitly handle structural error. Hoeting et al. [28] suggested that averaging over all models can improve upon the systematic bias and general limitations of a single model. In the hydrologic literature, the generalized likelihood uncertainty estimation methodology (GLUE) was proposed and has been successfully applied to a variety of hydrologic problems [3]. GLUE identifies several competing model structures and postulates parameters' prior joint probability distribution for each structure. It then implement Monte Carlo simulations and run each model with realizations drawn from the prior distribution. The simulation results are compared with corresponding observations to determine the likelihood of each combination of model structure and parameter realization based on a subjective likelihood function. Those combinations that yield acceptable replication of observation are then run in prediction mode, and their outputs are weighted by the likelihood to generate the final prediction. Unlike GLUE which uses a subjective likelihood function to weight the combinations of model structure and parameter set, the Bayesian model averaging (BMA) appraoch determines the model weights by calculating their posterior probabilities integrated over parameters with Bayes' theorem. Both GLUE and BMA are computationally demanding. To render BMA computationally feasible for groundwater modeling, Neuman [7] proposed the maximum likelihood version of BMA (MLBMA), which avoids the need for Monte Carlo simulation and integration by calibrating each model against observations. For GLUE, BMA and MLBMA, it is critical to select a set of mutually exclusive models that adequately spans various aspects of the real system. However, there exists no well-accepted guidelines in the literature about how to achieve this goal [30]. In addition, MLBMA involves calibration of each model in the ensemble, making it still computationally intensive.

This section has provided a brief overview of approaches to reduce parameter and/or structural uncertainties. On the other hand, as noted by Gupta [24], it is often difficult to disaggregate errors into their source components to understand, manage and reduce the uncertainty. On the other hand, for the decision-making process in water resource management, it is important to know the total model uncertainty lumped from all possible sources. This suggests to use a "model residual" approach that skips any distinction among uncertainty sources and directly analyzes the model residuals to build a model of predictive uncertainty [36]. Here the term "residual" refers to the misfit between measurement of system state (e.g. water level) and corresponding model output. It has been shown that a dominant portion of the misfit cannot be ascribed to measurement error, and is not random but systematic (see for example [19, 23]). Furthermore, it is demonstrated that the residual is likely to show in many groundwater studies a high degree of spatial and temporal correlation [43, 49, 15, 19, 34]. These arguments justify the approach of characterizing the model residual as being dependent on a set of predictors, which may include time and spatial locations.

Traditional statistical approaches to model the residual in hydrologic modeling often assume the residual be an independent identically distributed process, usually Gaussian with zero mean, assumptions rarely satisfied in practice. Many methods have been proposed either to manipulate the residuals so that they satisfy such assumptions, or to relax these assumptions [36]. Another trend has been to develop data-driven models based on machine learning techniques to model the residuals. Abebe and Price [1] approached the residual modeling of a rainfall-runoff model by applying a parallel artificial neural network (ANN) model to forecast the errors of the conceptual model. Solomatine and Shrestha [41] utilized the M5 model tree to predict the quantiles of predictive error in rainfall-runoff modeling. It was suggested that "the historical model residuals are the best available quantitative indicator" of the defects of modeling process [41], and hence data-driven models can capture the defects by learning from the residuals. Later, Demissie et al. [14, **?**, 43, 49] introduced this idea into groundwater modeling by proposing a complementary framework where separately-developed data-driven models (DDMs) were used to enhance the prediction error of the physically based MODFLOW models. This framework constructs complementary DDMs that are capable of recovering the unknown information about the real hydrogeologic system that is not represented correctly by the numerical model. This framework is not restricted to any particular type of model error and hence is advantageous if the sources of prediction error are multiple and not easily identifiable. In addition, it does not invoke any statistical assumption about the error distribution. Furthermore, unlike calibration and Monte-Carlo based approaches, the framework only runs the PBM once, thus making it suitable for PBMs requiring long running time. On the other hand, the effectiveness and efficiency of this framework could be compromised when applied to large datasets.

1.2 Scope of Thesis

In this thesis, the complementary modeling framework proposed in [15] will be applied to real-world case studies to test the ability of machine learning techniques for improving prediction accuracy of physically based groundwater flow models (PBM). We improve this framework by introducing clustering to make it more robust, flexible and computationally efficient when applied to large datasets. It will be demonstrated that implementation of cluster analysis before applying DDMs enables the latter to be better adapted to local patterns in the PBM residuals. In addition, clustering also reduces the computational cost of implementing some DDMs so that it becomes feasible to develop and tune them with the limited computation power and memory provided by desktop computers. Besides cluster analysis, five machine learning methods are employed to build DDMs: instance based weighting (IBW), locally weighted regression (*loess*), decision tree (DT), artificial neural network (ANN) and support vector regression (SVR). The DDMs are used to improve the head prediction of the MODFLOW model in two real-world case studies that are calibrated in different manners.

The complementary framework is expected to yield more accurate predictions as long as systematic patterns are presented in the PBM residuals. This assumption is tested using techniques borrowed from statistics and information theory. The resulting information is then used to determine whether it is appropriate to apply DDMs in a specific prediction scenario. In addition, the dependence between the residuals and other available data is also analyzed and helps to select input data for DDMs. Furthermore, the techniques are also used to analyze the remaining residuals after the DDMs are used to correct the PBM's error, as one performance measure.

1.3 Thesis Outline

Chapter 1 has presented a brief overview of attempts to improve physically-based groundwater model predictions. It also summarizes the scope and arrangement of the thesis. Chapter 2 presents the complementary modeling framework as well as a brief introduction to the machine learning techniques used to build the DDMs. The chapter also introduces techniques to analyze residuals and measure the performance of PBM and DDMs. Chapters 3 and 4 present two real-world case studies. Detailed implementation of the framework and the development of DDMs are described. Performance of DDMs are presented and discussed there. Finally, Chapter 5 summarizes the conclusions and suggests topics for futrue work.

Chapter 2

COMPLEMENTARY MODELING FRAMEWORK

2.1 Overview of Framework

This thesis employs the complementary modeling framework proposed in [15] and [14]. This framework adopts the perspective of optimality instead of equifinality, and works with a single calibrated model, since calibration is the most common practice in groundwater flow and solute transport analysis. The framework is based on the observation that a normal distributed error term with zero mean and small variance is generally not achievable via regression-based calibration in groundwater modeling. This is due to the limitations of calibration described in Chapter 1. In fact, systematic error (bias) can be found in the simulation results of even well-calibrated models, as noted by Demissie et al. [15]. In addition, the predictions made by the calibrated model are prone to error larger than the uncertainty inferred from the calibration process, as the latter is artificially made small by potentially over-tuning parameter values to compensate for the model structural defects. The bias of the optimal (calibrated) model can be viewed as resulting from the lumped uncertainty that includes errors associated with model structure, parameters, input stress and measurements. This suggests the potential of data-driven models (DDMs) to compensate (at least partly) for the discrepancy between the physically based model (PBM) and the real-world system [41].

In this thesis, the framework proposed by Demissie et al. [14, 15, 43, 49] is improved and applied to enhance the head prediction accuracy of groundwater flow models. The task of the complementary modeling framework (shown in Figure 1) is to use DDMs to learn, from the historical errors, the dependency of the error (ϵ) on selected input variables conditioned on the state variables of the groundwater flow model. When forecasting, the head prediction of the PBM ($\hat{h}$) is adjusted with the bias predicted by the DDMs ($\hat{\epsilon}$) to get a more accurate prediction $\hat{h}^{new}$. Since MODFLOW is the most commonly used groundwater modeling tool, MODFLOW will be used interchangeably with groundwater flow models and PBM in the remainder of the thesis.

Compared with the framework proposed in [14] and [15], the presented work incorporates a greater number and variety of machine learning techniques to build DDMs. Their performance is compared and analyzed. In addition, clustering is introduced to localize DDMs and improve computational efficiency in the revisited case study presented in Chapter 3. Furthermore, the improved framework is applied to a new case study as presented in Chapter 4.

The error of a well-calibrated PBM should approximately follow a normal distribution with zero mean and reasonably small variance. The data-driven models cannot be applied in this case, since the residuals are considered to be random. As discussed previously, normally distributed error is usually not achievable in practice via calibration. Nevertheless, it is still vital to conduct residual analysis before attempting to implement the complementary modeling framework, as the existence of bias and structure in the error of PBM is a key premise of the effectiveness of DDMs. In the case studies presented in Chapters 3 and 4, historical residuals (ϵ) are evaluated to check the presence of spatial and temporal structure, and to identify the extent and predictability of these systematic patterns.

The Normal probability plot is employed to test the hypothesis that the residuals of the PBM is normally distributed. It is a quantile-quantile (QQ) plot of the standardized data against the standard normal distribution. Departure from a straight line indicates departures from normality.

The temporal correlation is examined using the Durbin-Watson (DW) statistic [21], which is defined as

$$DW = \sum_{i=2}^{N_t} (\epsilon_t - \epsilon_{t-1})^2 / \sum_{i=2}^{N_t} \epsilon_t^2,$$

where ϵ_t and ϵ_{t-1} are the residuals at time step t and $t-1$ respectively, and N_t denotes the number of time steps. If DW substantially deviates from two, then there is evidence of positive (<2) or negative (>2) correlation. The spatial correlation of the residuals is checked via the empirical semivariogram of the residuals.

The above-mentioned approaches to detect temporal and spatial structure are limited to linear dependence, while the Mutual Information (MI) is able to detect and quantify nonlinear relations among data [9]. Given two random variables $\mathbf{X}, \mathbf{Y}$, the MI is defined as

$$I(\mathbf{X}, \mathbf{Y}) = \int \int p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dxdy, \tag{2.1}$$

where $p(x, y)$ is the joint probability distribution function (*pdf*) of $\mathbf{X}, \mathbf{Y}$, and $p(x)$, $p(y)$ are the marginal *pdfs* of X, Y respectively. If $\mathbf{X}, \mathbf{Y}$ are independent, then $I(\mathbf{X}, \mathbf{Y}) = 0$. On the other hand, a high value of the MI score indicates a strong dependence between the two random variables. In practice, the *pdfs* in Eq. (2.1) are typically unknown, hence they are estimated from the data or realizations $(x_1, y_1), ..., (x_N, y_N)$. The MI score is then computed using

$$I(\mathbf{X}, \mathbf{Y}) = \frac{1}{N} \sum_{i=1}^{N} \log \frac{f(x_i, y_i)}{f(x_i)f(y_i)}, \tag{2.2}$$

where N is the size of the dataset, and $f(x_i)$, $f(y_i)$ and $f(x_i, y_i)$ are the marginal and joint *pdfs* estimated at (x_i, y_i). The base 2 is used in this work.

2.2 Overview of Machine Learning Methods

This section provides a brief overview of the machine learning techniques used to develop DDMs. In contrast with the physical-process-based groundwater flow models, machine learning techniques learn directly from the data. From a set of training data, a machine learning algorithm learns a mapping from the inputs (features) to the outputs (target) that can be generalized to predict on unseen (testing) data.

2.2.1 Clustering

Cluster analysis is an unsupervised data mining technique that partitions data into groups with the goal of maximizing the similarity of data within the same group and minimizing the similarity of data among groups. Central to cluster analysis is the similarity (or dissimilarity) measure, based on which the clustering method partitions individual objects [26]. The similarity measure is chosen in accordance with the demands of specific problem.

Clustering is beneficial in two cases. First, if the residual analysis shows local patterns within the dataset, it is then reasonable to develop "localized" DDMs instead of developing a global DDM. DDMs that are conditioned on an individual cluster allow for additional flexibility. Second, in cases of dealing with large datasets, dividing the data into smaller subsets improves the computation efficiency and makes model selection by cross validation more feasible.

The k-means clustering and agglomerative hierarchical clustering algorithms are used in this thesis, as described in the following paragraphs. Both of the algorithms belong to "crisp" clustering, where an object is assigned to one and only one cluster.

K-means Clustering

The K-means is one of the most popular clustering algorithms. It is a top-down method that iteratively minimizes the *within-cluster point scatter* [26]. It chooses the squared Euclidean distance to measure the dissimilarity between two objects (data points)

$$d(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j||^2. \tag{2.3}$$

The *within-cluster point scatter* is a natural loss (or "energy") function based on Eqn. (2.3). Let $C(i)$ denote an assignment that maps an object $\mathbf{x}_i$ to the kth cluster, C denote the assignments of all data points and K denote the number of clusters, the loss function is defined as

$$W(C) = \frac{1}{2}\sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} d(\mathbf{x}_i, \mathbf{x}_j). \tag{2.4}$$

It has been proved that minimizing $W(C)$ can be achieved by solving the enlarged optimization problem

$$\min_{C,\{\mathbf{m}_k\}_1^K} \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||\mathbf{x}_i - \mathbf{m}_k||^2, \tag{2.5}$$

where $\mathbf{m}_k$ denotes the mean vector associated with the kth cluster. The k-means clustering starts with a initial guess of assignment C, then iterately repeats the following two steps until convergence [26]:

1. Given current assignment C, minimize Eqn. (2.5) with respect to $\{\mathbf{m}_1, ..., \mathbf{m}_k\}$ yielding updated cluster centers;

2. Given current $\{\mathbf{m}_1, ..., \mathbf{m}_k\}$, minimize Eqn. (2.5) by assigning each data point to the closest cluster center, i.e.

$$C(i) = \underset{1\leq k\leq K}{argmin} ||\mathbf{x}_i - \mathbf{m}_k||^2. \tag{2.6}$$

MATLAB® Statistic Toolbox™ subroutine `kmeans` is used to implement k-means clustering in this work.

Agglomerative Hierarchical Clustering

As the name suggests, hierarchical clustering arranges the clusters into a natural hierarchy. At the bottom of the hierarchical structure, each cluster containes one single data point. The bottom-up *agglomerative hierarchical clustering* strategy starts at the bottom and at each level recursively merges a pair of clusters into a single cluster [26].

In this thesis, the Mahalanobis distance

$$d_M(\mathbf{x} - \mu) = \sqrt{(\mathbf{x} - \mu)^T S^{-1} (\mathbf{x} - \mu)} \tag{2.7}$$

is used to specify the pairwise dissimilarities between data points. μ and S denote respectively the mean and covariance of the data set $\mathbf{x}$'s. In this work, Ward's linkage [47] is used as criterion of joining clusters, that is to minimize the sum of squares of any two clusters that can be merged at each step. MATLAB® Statistic Toolbox™ subroutines `linkage` and `cluster` are used to implement agglomerative hierarchical clustering.

2.2.2 Locally Weighted Learning

Locally weighted learning falls in the category of instance based learning, or lazy learning [2], which defers processing of training data until a query (prediction request) needs to be answered. One appealing trait of this class of learning algorithms is that they are straightforward to interpret, compared to the regression methods introduced later. Instead of using a single global model to fit all of the training data, as for most learning methods, lazy learning enables query-specific local models by fitting the training data only in a region most relevant to the query point. In particular, locally weighted learning methods measure the relevance with weighting functions. Two types of locally weighted learning are tested in this work.

Instance Based Weighting

Instance-based weighting (IBW) extends the widely-used lazy learning k-Neareat Neighbor method (kNN) by introducing a weighting function [2]. For a query $\mathbf{x}'$, IBW first finds its n nearest neighbors, denoted as $\{\mathbf{x}_j^*\}, j = 1, 2, ...n$, in the training set that contains N data points ($\{\mathbf{x_i}\}, i = 1, 2, ..., N$), then estimates the residual at $\mathbf{x}'$ by

$$\hat{\epsilon}(\mathbf{x}') = \sum_{j=1}^{n} w_{\mathbf{x}'|\mathbf{x_j^*}} \epsilon(\mathbf{x}_j^*), \tag{2.8}$$

where $w_{\mathbf{x}'|\mathbf{x}_j^n}$ denotes the weight of j-th neighbor. From now on we use a hat ($\hat{\epsilon}$) to differentiate the residuals estimated by DDMs from the real value (ϵ).

In this thesis, two weight functions are used:

$$w_{\mathbf{x}'|\mathbf{x}_j^*} = \frac{\alpha}{||\mathbf{x}' - \mathbf{x}_j^*||^p} \tag{2.9a}$$

$$w_{\mathbf{x}'|\mathbf{x}_j^*} = \beta exp(-||\mathbf{x}' - \mathbf{x}_j^*||^2/q^2). \tag{2.9b}$$

α and β are scaling factors to ensure $\sum_{j=1}^{n} w_{\mathbf{x}'|\mathbf{x}_j^*} = 1$, and p, q are parameters to be tuned. One characteristic of (2.9a) is that, the weight assigned to $\mathbf{x}_j^*$ very close to the query $\mathbf{x}'$ can become unboundedly large as the distance approaches to zero (as shown in Figure 2a), so that other neighbors might be shadowed. In contrast, the weights allocated by (2.9b) are bounded by one (Figure 2b).

Despite its simplicity, IBW has been very popular and successful [26, 40]. The level of complexity of the estimation suface of IBW can be tuned by varying the parameters, making this group of methods very flexible. For example, decreasing the size of neighborhood (n), (and/or) increasing p (and/or) decreasing q, will decrease the mean error (bias), yet increase the variance of the fit. In some cases, using the same parameters globally may fail to provide satisfactory estimation. Typically, for kNN, Wettschereck and Dietterich [48] suggested to vary the value n locally within different parts of the input space to account for varying characteristics of the data. In this thesis, the values of the above parameters are tuned via cross-validation for each subset, as described in section 2.2.1 and 3.4.

Locally Weighted Regression

Locally weighted regression, or *loess*, fits a regression surface of the predicators (input features) using a multivariate smoothing process [11]. It can estimate a much wider class of regression surfaces than with the usual classes of parametric functions, such as polynomials. To estimate the target (the residual in this work) at a query point $\mathbf{x}'$, *loess* first finds a neighborhood of $\mathbf{x}'$ in the training dataset ($\{\mathbf{x}_i\}, i = 1, 2, ..., N$). Unlike IBW which computes $\hat{\epsilon}(\mathbf{x}')$ by weighted averaging, *loess* fits a polynomial over the neighborhood, denoted as $\{\mathbf{x}_j^*\}, j = 1, 2, ...n$. The number of neighbors, n, is typically user-specified, as discussed later. In contrast with traditional regression which treats every data point equally, *loess* weights each point in the neighborhood according to its distance to the query point $\mathbf{x}'$. The popular weighting function used in *loess* takes the tri-cubic form,

$$K(\mathbf{x}', \mathbf{x}_j^*) = \begin{cases} \left\{1 - \left[\frac{||\mathbf{x}' - \mathbf{x}_j^*||}{d(\mathbf{x})}\right]^3\right\}^3 & ||\mathbf{x} - \mathbf{x}_j^*|| < d(\mathbf{x}) \\ 0 & ||\mathbf{x} - \mathbf{x}_j^*|| \geq d(\mathbf{x}), \end{cases} \tag{2.10}$$

and

$$d(\mathbf{x}) = \max_j ||\mathbf{x} - \mathbf{x}_j^*||. \tag{2.11}$$

Then a polynomial of degree d is fitted on the neighbors $\{\mathbf{x}_j^*\}$ by minimizing the weighted least squares loss function:

$$L = \underset{\alpha(\mathbf{x}'), \beta(\mathbf{x}'),\ j=1,\ldots,d}{min} \sum_{j=1}^{n} K(\mathbf{x}', \mathbf{x}_j^*) \left[\epsilon(\mathbf{x}_j^*) - \alpha(\mathbf{x}') - \sum_{k=1}^{d} \beta(\mathbf{x}')(\mathbf{x}_j^*)^k\right]^2. \tag{2.12}$$

The estimated residual at query $\mathbf{x}'$ is therefore given by

$$\hat{\epsilon}(\mathbf{x}') = \hat{\alpha}(\mathbf{x}') + \sum_{k=1}^{d} \hat{\beta}_k(\mathbf{x}')(\mathbf{x}_j^*)^k. \tag{2.13}$$

To achieve satisfactory performance, the user-specified bandwidth of neighborhood (n) needs to be optimized. This is done by tuning the ratio of n to N, or the *span* parameter (δ), via cross validation. Similarly to IBW, there is also a tradeoff between variance and bias in choosing δ. Larger *span* indicates that more data points are used to build the local regression model; therefore the fitted polynomial surface tends to be smoother. As a result, the variance of the *loess* model is low while the bias may be large. Smaller value of δ, on the other hand, yields a more "wiggling" regression surface that fits the data better at the price of high variance.

Another point to note is the degree of polynomial (d). From Eqn. (2.13), the input variables themselves are directly used as fitting variables if we fit linear polynomials ($d = 1$). On the other hand, if a quadratic polynomial is used, the fitting variables include the input features, their squares and their cross-products. Usually d = 1 or 2 suffices to yield satisfactory estimates, hence higher orders are rarely used. Quadratic polynomials tend to outperform linear polynomial when the regression surface has substantial curvature.

The `loess` class in the R software environment is used to implement *loess*.

2.2.3 Decision Trees

Decision tree (DT) is a conceptually simple yet powerful nonparametric tool for classification and regression [26]. A tree-based classification and regression algorithm CART is briefly described here. CART recursively partitions the feature space, in a binary fashion, into rectangular regions, and fits a constant value in each one. For the purpose of clarity, some of the notations used in this section are different from other part of the thesis.

The bold symbol $\mathbf{x}$ is still used to denote the input data as vectors, while the capital letters $X_1, X_2, ..., X_p$ denote the input features, which are considered as random variables. Within the framework described in section 2.1, for example, X_1, X_2 could be the $\langle x, y \rangle$ location of the monitoring wells. A binary split at $X_j = s$ partitions the space into two regions R_1 and R_2 such that

$$R_1 = \{\mathbf{x} | X_j \leq s\} \; and \; R_2 = \{\mathbf{x} | X_j > s\}. \tag{2.14}$$

Constant c_1 and c_2 are then assigned to the data in R_1 and R_2 respectively. Figure 3a is an illustration of a tree with two input features X_1 and X_2. Figure 3b shows the corresponding partition of the two-dimensional space. The CART algorithm adopts a greedy strategy to seek the splitting variable j and split point s at every non-terminal node. Starting with all the data, the algorithm solves the following minimization problem:

$$\min_{j,s} \left[\min_{c_1} \sum_{\mathbf{X} \in R_1} (\epsilon_i - c_1)^2 + \min_{c_2} \sum_{\mathbf{X} \in R_2} (\epsilon_i - c_2)^2 \right]. \tag{2.15}$$

The splitting process is repeated on the resulting regions, and a large tree T_0 is grown until some minimum node size is reached at leaves, or terminal nodes. T_0 is then pruned with a pruning level selected based on the tree cost which is usually estimated by cross validation on the training dataset. The cost of the tree is the sum over all terminal nodes of the estimated probability of a node times the cost of a node. The probability of a node is computed as the proportion of data points that satisfy the condition for the node. For regression trees, the cost of a node is the mean squared error at that node, i.e. $\sum_{\mathbf{X} \in R_i} (\epsilon_i - c_i)^2$ for node i [5]. In this work, tree regression is implemented with the MATLAB® `classregtree` class of routines, which adopts the above stated algorithm.

The key advantage of regression tree is its straightforward interpretability. On the other hand, the disadvantages include: (1) Regression trees are unstable and have high variance, and are sensitive to data noise [33]; (2) The regression surface lacks smoothness; (3) The representational power of binary splitting is restricted [26]. It is also worth noting that DT is similar with IBW in that both of them partition the feature space into regions and fit a simple model in each region.

2.2.4 Artificial Neural Network

Artificial Neural Network (ANN) has been widely applied to water resources prediction and forecasting [31]. The representational power of ANN makes it particularly suitable for learning functions whose general form is unknown in advance. It has been proved that a feedforward network with three layers can approximate any function to arbitrary accuracy, given sufficient nodes in each layer. In practice, networks of feasible size are able to represent a rich space of highly nonlinear functions [33].

Inspired by biological learning processes, ANNs are built out of a densely interconnected set of units (nodes). Each unit takes a number of real-value d inputs (could be outputs of other units) and produces a single real-valued output, which may be the input to other units [33]. In this thesis, we focus on the multilayer perceptron network (MLP) that frequently appears in hydrological forecasting applications. A typical MLP network is comprised of an input layer, one or more hidden layers and an output layer. An MLP with one hidden layer and output layer of one node is represented by the network diagram shown in Figure 4. For feed forward networks, like MLP, information flows from left to right through the connections between units. Each unit, or neuron, computes a single real-valued output based on a weighted sum of its real-valued inputs (possibly the output of other neurons) plus a bias term, and a transfer function. The sigmoid function is typically used for hidden layer(s) while the linear function is usually used for the output layer. The weights and bias of all units can be learned by the Backpropagation algorithm, which attempts to minimize the loss function

$$E(\mathbf{w}, b) = \frac{1}{2}\sum_{i}(\hat{\epsilon}_i - \epsilon_i)^2 \tag{2.16}$$

where the subscript i denotes the index of the training sample, and $\hat{\epsilon}_i, \epsilon_i$ denote respectively the estimated and real residuals of training data $\mathbf{x}_i$. A typical setting of the Backpropagation algorithm is to minimize the loss function by Levenberg-Marquardt method and update the weights and bias of all units by the gradient decent with moment approach.

Since there is no regularization term in the loss function $E(\mathbf{w}, b)$, the backpropagation MLP is prone to overfitting the training data. One approach to prevent overfitting is the *early stopping* technique that takes advantage an independent validation dataset. For every iteration step during the training process, the network that is learned from the training set is tested on the validation set, and the generalization error is monitored. In the early stage of training, the generalization error normally decreases with the training error, until the network begins to overfit the data. The training is terminated when the generalization error continuously increases, and the coefficients at the minimum of the generalization error are chosen.

A major shortcoming of MLP is that the Backpropagation algorithm is only guaranteed to converge to some local minima instead of the global minimum. In addition, as a *black box*, ANN is difficult to interpret.

In this work, ANN is implemented with the MATLAB® `Neural Network Toobox`™.

2.2.5 Support Vector Regression

Support vector machines (SVM) comprise a relatively new class of learning algorithm. The popularity of SVM applied to regression problems can be attributed to: 1) good generalization performance, because SVM seeks to minimize an upper bound of the generalization error rather than minimize the training error; 2) the solution of SVM is always globally optimal, while many other machine learning tools (eg. ANNs) are subjected to local minima;

3) the solution is represented sparsely by *Support Vectors*, which are a typically small subset of all training examples [6]. A brief overview of ε-SVR is provided here. For more details, readers are referred to [44] and [39].

Given a set of training data $(\mathbf{x}_1, \epsilon_1), ..., (\mathbf{x}_N, \epsilon_N)$, in SVM regression (hereby abbreviated as SVR), the input $\mathbf{x_i}$ is first projected to a higher dimensional *feature* space by the map $\Phi : \mathcal{X} \rightarrow \mathcal{F}$. Please note that here the *feature* is different from the *input*, while in the other parts of the thesis the two words are used interchangeably. Linear regression to approximate the unknown function $\epsilon(\mathbf{x})$ is then performed in the feature space $\Phi(\mathbf{x})$ instead of the input space $\mathbf{x}$:

$$f(\mathbf{x}) = w \cdot \Phi(\mathbf{x}) + b. \tag{2.17}$$

The coefficients w and b are estimated by solving the optimization formulation of SVR

$$minimize \;\; \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{2.18}$$

subject to

$$(w^T\phi(\mathbf{x}_i) + b) - \epsilon_i \leq \varepsilon + \xi_i, \tag{2.19a}$$

$$\epsilon_i - (w^T\phi(\mathbf{x}_i) + b) \leq \varepsilon + \xi_i^*, \tag{2.19b}$$

$$\xi_i, \xi_i^* \geq 0, \;\; i = 1, ..., N. \tag{2.19c}$$

Regularization by minimizing $||w||^2$ ensures the flatness of the solution. The second term in Eqn. (2.18) is derived from the ε-insensitive loss function

$$|\epsilon_i - f(\mathbf{x}_i)|_\varepsilon = max\{0, |\epsilon_i - f(\mathbf{x}_i)| - \varepsilon\}. \tag{2.20}$$

The constant C in Eqn. (2.18) determines the trade-off between the flatness of f and the tolerance of deviations larger than ε.

Usually the map $\Phi : \mathcal{X} \rightarrow \mathcal{F}$ is implemented implicitly via kernels such that

$$\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = K(\mathbf{x}_i, \mathbf{x}_j)$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product in $\mathcal{F}$. In this work, the popular *radial basis function* (RBF) is used as the kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma ||\mathbf{x}_i - \mathbf{x}_j||^2). \tag{2.21}$$

The kernal width parameter γ is optimized via cross validation.

Chang and Lin [8] recommends $\varepsilon = 0.1$ and to check a grid of values of C and γ by cross validation and select the hyperparameter setting that yields lowest cross validation error. On the other hand, Cherkassky and Ma [10] suggests to estimate the best regularization parameter C and error insensitive parameter ϵ by

$$C = max(|\mu + 3\sigma|, |\mu - 3\sigma|), \tag{2.22a}$$

$$\varepsilon = \tau \sigma_0 \sqrt{\frac{ln\ N}{N}}, \tag{2.22b}$$

where μ and σ are the mean and standard deviation of the training outputs ϵ_i's , τ is a coefficient usually equals to 3, σ_0 denotes the noise level, and N is the size of training dataset. The reason that cross validation is not used to optimize all hyperparameters is due to the prohibitively long computation time required to check a number of combinations of hyperparameter values in a grid-search fashion. In addition, the hyperparameters chosen analytically outperformed the values chosen by a preliminary cross validation attempt with a relatively coarse grid search. The LIBSVM codes developed by Chang and Lin [8] were used to implement the SVR. The codes can be downloaded from the LIBSVM website (http://www.csie.ntu.edu.tw/ cjlin/libsvm/).

2.3 Performance Evaluation

The performance of the PBM and DDMs is evaluated graphically and by statistical measures of goodness of fit. RMSE is adopted as the primary measure of the model accuracy. Following the notations in section 2.1 and letting N denote the number of predictions, $h_i, \hat{h}_i, \hat{h}_i^{new}$ denote the ith observed, MODFLOW simulated and DDMs updated head respectively, the root-mean-square error (RMSE) of the PBM predictions is defined as

$$RMSE(PBM) = \sqrt{\frac{\sum_{i=1}^{N}(h_i - \hat{h}_i)^2}{N}},$$

and RMSE for the DDMs updated predictions is

$$RMSE(PBM + DDM) = \sqrt{\frac{\sum_{i=1}^{N}(h_i - \hat{h}_i^{new})^2}{N}}.$$

The mean error (ME) is used to measure the overall bias of the models. The performance of models can also be evaluated graphically via residual plots and cumulative distribution function (*cdf*) plots of mean absolute error (MAE). Their use is demonstrated later in the case studies.

In addition, the residual analysis is carried out on both the PBM and DDMs updated predictions. The presence and extent of spatial and temporal structures are compared before and after DDMs updating, in order to examine the effectiveness of the DDMs to correct the bias of PBM predictions.

Chapter 3

CASE STUDY OF THE REPUBLICAN RIVER COMPACT ASSOCIATION MODEL

3.1 The RRCA Model

The framework described in Chapter 2 was applied to a regional groundwater flow model of the Republic River Basin covering portions of eastern Colorado, northwest Kansas and southwest Nebraska (Figure 5). In the last century, growing water demand for irrigation and other uses has led to dramatically increased the groundwater pumping, leading to water conflicts and litigation among the states. In 2002, the Republican River Compact Association (RRCA) model was constructed by experts representing Colorado, Kansas and Nebraska to determine the streamflow depletions to the Republican River due to well pumping in each state and streamflow accretions to the Republican River from recharge of water from the Platte River Basin to the north. The model, its documentation, input and output data are available via the RRCA website (http://www.republicanrivercompact.org/).

The RRCA model uses a modified version of the MODFLOW 2000. The model has a single layer, a uniform grid size of 1 square mile, and stress period of a month. The main stresses in the model include recharge, groundwater pumping, evapotranspiration and reservior stage. Recharge is comprised of natural recharge from precipitation and human-induced recharge from irrigation of cropland and seepage from irrigation ditches or canals. Parameters that were calibrated include hydraulic conductivity, precipitation recharge, the minimum saturated thickness, potential evapotranspiration rate spatial multipliers, multipliers to adjust the average recharge from 1918-1940 for the mound area in the northeastern portion where there are water transfers into the Republican River Basin from the the Platte River Basin to the north. . The model was calibrated based on head measurements at over 10,000 wells and baseflow at 65 gages from Jan.1918 to Dec.2000. The model was calibrated by “trial and error” and automated calibration techniques [32], but more detailed information concerning the calibration process is not available. Since 2000, the model has been run each year using new input data sets of pumping, canal losses and irrigation return.

3.2 Data Preparation

The database used in this thesis includes simulated and observed waterlevel data from 1918 to 2007. Cell-by-cell head output simulated by the RRCA model until 2007 is available on the RRCA website. This file follows the format of MODFLOW head output files, therefore it can be processed with common post-processing programs developed for MODFLOW. The major part of the waterlevel measurements from 1918 to 2000 were downloaded from the RRCA website as a group of database (DBF) files. The DBF files contain the data of the wells where the waterlevel measurement was taken, including the site identity number, x, y coordinates in the MODFLOW model, latitude and longitude, coordinate system, land surface elevation and other information. The DBF files also include waterlevel measurements and the measurement time. Waterlevel observations from 2001 to 2007 are downloaded from the USGS Water Data for the Nation (NWIS) online database (http://waterdata.usgs.gov/nwis/gw). The information of the wells included in the above mentioned DBF files are submitted for searching the database for related groundwater data. The waterlevel observations at these wells are then identified and retrieved in the form of depth to water. The waterlevel is then computed by subtracting depth to water from the land surface elevation at the well. Waterlevel measurements at additional wells that entered the NWIS database after the construction of the RRCA model are also retrieved. The geographical coordinates of these wells are projected to the RRCA model coordinate systems, so that this part of data can be merged with the others. The simulated waterlevel corresponding to measurements were linearly interpolated from the head output file of the RRCA model using `mod2obs`, a groundwater data utility code developed by the Watermark Numerical Computing. Instructions to use this utility code are available in [?]. The waterlevel records of wells range from a single measurement to continuous measurements at a few locations. Typically, the waterlevel is measured once each year usually during the non-irrigation season. Some wells that have few observations or have unreasonably high residual are excluded from the database. Overall, the database consists of over 300,000 waterlevel data from 1918 to 2007 at 3,078 wells within the model boundary that have no fewer than 10 observations and absolute mean residual less than 100 ft.

3.3 Residual Analysis

Considering its complexity and size, the RRCA model was calibrated to represent the hydrogeological characteristics of the Republican River Basin to a reasonable degree. However, normally distributed residuals with zero mean, an indicator of a well-calibrated model (section **??**, were not achieved. Overall, the mean error (ME) of the head simulated by the model during the calibration period (1918-2000) is -7.74ft. (negative value indicates simulated head is greater than the observed head), and the root-mean-square error (RMSE) is 33.38ft. The normal probability plot of the residual (Figure 6) deviates from the straight line. Further residual analysis indicated the presence of notable bias and relatively large magnitude of residuals in the MODFLOW model simulation results.

Local patterns can be found in the plot of mean residual at each well during the calibration period, as shown in Figure 7. For example, the MODFLOW model overpredicted the waterlevel in northeastern and western part of the model domain, while the model underpredicted the waterlevel in many places throughout Nebraska and Kansas. The spatial dependence of the historical residuals was further analyzed via semi-variograms, as presented in Figure 8.

In each plot, the variance of the difference between the residuals at two wells increases as the distance increases. This indicates the existence of spatial dependence.

The DW statistic described in section **??** was used to test the existence of temporal correlation. Since DW statistic works with time series data with uniform time intervals, only wells that have relatively uniformly spaced waterlevel measurements were included. Figure 9a shows the DW statistics of 51 wells with time step of one month, and Figure 9b shows the DW statistics of 71 wells with time step of one year. The DW statistics of the majority wells are substantially smaller than two, indicating that the residuals are correlated in time.

Finally, the correlation coefficient and Mutual Information score were used to quantify the overall spatial and temporal dependence of historical residuals. Table 1 lists the correlation coefficients and MI scores between normalized selected variables and the residual at all wells from 1918 to 2000. Please note that MI scores are not invariant to scaling and hence should only be compared on a relative scalewith each other. The MI scores show spatial dependency of the residual, while temporal dependency appears to be weaker. MI scores show that the x location is more relevant than y location to the residual, which is not reflected in the correlation coefficients that can only quantify linear correlation.

3.4 Framework Implementation

DDMs were built to forecast the prediction error (ϵ) of the MODFLOW model's head simulation ($\hat{h}$). The updated head $\hat{h}^{new} = \hat{h} + \hat{\epsilon}$ was then the final output of the framework. In accordance with the practical need of predicting using the MODFLOW model, the complementary modeling framework was implemented in three prediction scenarios: temporal, spatial, and temporal plus spatial prediction. In each case, the DDMs were trained first on one training dataset, and then validated on an independent testing dataset.

In temporal prediction (TP) case, the DDMs were trained with historical data (1918-2000), and validated on data during the prediction period (2001-2007) at the same wells. For spatial prediction (SP), wells were randomly split into group A and B. The DDMs were trained with the data at wells in group A from 1918 to 2007, and verified on the data of group B during the same time period. In the scenario of temporal plus spatial prediction (TSP), we trained the DDMs with 1918-2000 data of group A, then tested them on 2001-2007 data of group B. The sizes of datasets used in the three scenarios are summarized in Table 2. Please note that for SP and TSP, the wells are splitted randomly. The size of training and testing sets may change if the wells are splitted differently.

Table 1 shows that the historical residuals are related to the well location and the water-level simulated by the MODFLOW model, while the relation to the time of measurement is weaker. The selection of features were based on this observation, the cross validation results, and the fact that in the TP and TSP scenario DDMs extrapolate in terms of t. For TP and TSP the DDMs took as inputs the location of the wells $(x_w,\ y_w)$ where the head measurements were taken as well as the head computed by the MODFLOW model $(\hat{h})$. For SP, the observation time (t) was also included in the input features. Mathematically, the DDMs can be formulated as

$$\hat{\epsilon} = \begin{cases} f(x_w,\ y_w, \hat{h}) & TP, TSP \\ f(x_w,\ y_w, \hat{h}, t) & SP. \end{cases} \tag{3.1}$$

Local patterns were found in the residuals, as stated in section **??**. In addition, the datasets are large, especially for TP and SP (Table 2). Therefore cluster analysis was implemented as the first step. For temporal prediction (TP), the 3,078 wells in the database were clustered into 10 subsets using the agglomerative hierarchical clustering algorithm according to their spatial locations, and first and second moments of the head error. The simulation-measurement pairs in the database are then partitioned into ten subsets according to which cluster the well (where the measurement is taken) belongs to. Each subset was comprised of a training dataset containing data during the calibration period and a validation dataset during the prediction period. DDMs were developed for each subset respectively. In contrast, for SP and TSP, similar clustering cannot be implemented because training and validation datasets contain different wells. Instead, the 4 or 3 dimensional (depending on whether t is included) input data $\mathbf{x}$ is clustered by k-means algorithm. Again, each cluster consists of training and validation/testing dataset, and DDMs were developed for each cluster. The number of clusters is set as 10 for TP and SP. It was not tuned due to the prohibitively high computational expense. For the TSP scenario, the dataset include the three clusters out of the ten clusters used for SP that contain residuals in the prediction period. The result of k-means clustering is subject to randomness due to the initial guess of cluster center and the possibility to converge to local minima. Although the hierarchical clustering algorithm is not subject to this shortcoming, it involves computing the distance between every pair of data points, which is too computationally expensive in TP and TSP cases.

IBW, *loess*, DT, SVR and ANN models were built in TSP scenario. The comparison of performance of different DDMs was examined only in TSP case, because: 1) The computational expense makes it infeasible to conduct such comparison in all cases; the training and testing dataset are smaller for TSP than for TP and SP, thus the computation time is smaller. 2) In TSP scenario, the abilities of DDMs to implement both temporal and spatial inferences are tested, while only one type of inferences is tested in the TP or SP scenario. As will be shown in the next section, IBW and SVR outperformed other machine learning techniques in the TSP case, hence only these two DDMs were further tested for TP and SP scenarios. The parameters of the DDMs were tuned as described in Chapter 2.

Cross validation was used as the primary approach to select parameters for DDMs. Cross validation (CV) is a technique for assessing how DDMs will generalize to an independent data set. K-fold CV first partitions the training dataset into K subsets. For the kth subset, CV fits the DDM to the other K-1 subsets, and calculates the prediction error of the fitted model when predicting the kth subset. This process is repeated for $k = 1, 2, ..., K$, and the K testing results are averaged to compute the cross-validation estimate of prediction error [26], herein referred as CV error. Here cross validation is used to choose the optimal values of weighting function parameters and number of neighbors of IBW, span of *loess*, tree size of DT, the number of hidden nodes of ANN, as well as the kernel width of SVR. After cross validation, the DDMs with chosen parameters will be trained with the whole training dataset, and then used for forecasting.

The method to partition data during the cross validation (CV) process was adapted for the specific prediction scenario. For SP and TSP, the wells in the training set were divided into five groups. For five-fold CV of SP, the observations at the wells of one group were retained as testing data, and the DDMs were trained on the remaining data. This process was repeated five times so that every observation had been included in the testing data once. For TSP, in each one out of five runs of CV, the testing dataset is comprised of the observations at the wells of one group from 1995 to 2000, and the training data includes older residuals (1918-1994) at wells from other groups, unless stated otherwise. On the other hand, there is no straightforward way to implement a similar CV process for TP, therefore the data was simply randomly partitioned.

The process of parameter selection for DDMs is reported in detail for the TSP scenario. The parameters of DDMs in TP and SP scenarios were chosen similarly. Here we present the cross validation results for one of the three clusters. For IBW, the weighting function Eq. (2.9a) outperformed Eq. (2.9b) in the TSP scenario. Five-fold cross validation was implemented with various values of the power p of the weighting function and the number of neighbors n to find the combination that would yield the lowest CV error. The cross validation results are shown in Figure 10. The combination $p = 5, n = 128$ yields the smallest CV error, and hence is selected for the final IBW model used for forecasting that is based on the whole training dataset . For *loess*, second degree polynomial ($d = 2$) was used for locally weighted regression following the recommendation in [11, 12]. The span δ which controls the size of neighborhood was chosen by five-fold cross validation, as shown in Figure 11. As will be shown later in section 3.5.1, the generalization error of the *loess* model on one of the five subsets with any value of δ within the considered range is exceptionally larger than the generalization error on other subsets. This subset was therefore excluded when computing the CV error. $\delta = 0.35$ yields smallest CV error; hence the loess model with this span value was then fitted to the whole training set. For DT, CV determines the optimal size of the tree by tuning the pruning level. Unlike for other DDMs, cross validation for DT partitions the training data points into five subsets with roughly equal size. This is because partitioning the wells into five folds and then splitting the data points according to which fold the well belongs to would result in subsets that are much smaller than the training dataset, and that have sizes dramatically different from one another. Since tree levels are adapted to the sample size, it is more reasonable to determine the tree size based on cross validation subsets whose sizes are similar to one another, and are more representative of the size of the training dataset (each subset includes 80% of the training data points in five-fold CV). For each subset, a tree is fitted to the remaining data and used to predict the subset. The tree is pruned to subtrees with varying pruning levels, and the corresponding costs for each subset is pooled to compute the cost over the whole training dataset, as plotted in Figure 12. The CV error monotonically increases as the pruning level increases without any "turning point" as in the cross validation results of other DDMs. Although large, complex trees with more than 200 levels is favored by the CV results, they are risky in terms of generalization capability. From the perspective of parsimony, the best pruning level is chosen as the one that produces the smallest tree that is within one standard error of the minimum-cost subtree. Figure 12 shows that the best pruning level is 199, yielding a subtree that has 73 terminal nodes ($|T| = 73$). In the case of ANN, a single layer MLP is used as recommended in [31, 33]. CV chooses the number of nodes in the hidden layer to be 18, as shown in Figure 13. For SVR, the values of the regularization parameter C and the error insensitive parameter ε were selected analytically with Eq. (2.22a) and (2.22b), respectively. Five-fold cross validation was implemented to tune the kernel width parameter γ (Eq. (2.21)), as shown in Figure 14. $\gamma = 11$ was chosen as it corresponds to the lowest CV error. The selected values of parameters are summarized in Table 3.

3.5 Results and Discussion

The global performance averaged among cluster/subsets is reported. The TSP results are presented first, and the effectiveness of different DDMs is compared. The performance of IBW and SVR in all scenarios are reported in more detail later.

3.5.1 Comparison of DDMs Performance for TSP

IBW, *loess*, DT, ANN and SVR models were built to forecast the predictive error of the MODFLOW model in the temporal plus spatial prediction scenario. The parameters were tuned as stated in section 3.4, and their optimized values are listed in Table 3. Table 4 summarizes the ME and RMSE before and after DDMs updating. In the case of ANN, to account for the randomness of initial weights, the model is re-trained five times using different initial weights. The result shown is the average of the testing error over the five rounds. All DDMs effectively reduce the predictive error of the MODFLOW model, but to a varying degree. Among them, IBW and SVR yielded smaller RMSE than other DDMs.

The DDMs were also compared with each other in terms of computation cost. Table 4 lists the time that each DDM took to predict the residuals of the testing data. The comparison is based on run-time on a desktop with Intel®Core$^{\text{TM}}$2 Duo 3.16GHz×2 CPU and 4GB RAM.

The codes used to implement *loess*, DT, ANN and SVR are described in Chapter 2. Due to the random initialization of weights, the run-time of ANN is uncertain, yet it is usually longer than the running time of other DDMs. The computation times of all DDMs are negligible compared with the PBM. On the other hand, the memory required to process cross validation of the DDMs generally exceeds the memory of the above desktop computer, and would take much longer running time, if CV is conducted with the whole training dataset without clustering.

Improved comparison of the performance of DDMs calls for analyzing the effects of randomly splitting the dataset into training and testing sets. This can be done by repeating the above-stated implementation of DDMs many times. Each time we may re-split the training and testing wells, tune the DDMs via cross validation on the training dataset, and test the DDMs on the testing dataset. This process is cumbersome. A simpler approach was adopted that took advantage of the five-fold cross validation results. The lowest generalization error of DDMs on each subset was taken as their testing error, which is shown in Figure 15 and 16. The performance of DT and especially *loess* is subject to high variance. This is primarily because DT is not as robust as the other three DDMs in dealing with noisy data 2.2.3, and *loess* is not good at predicting near or beyond the boundary of the training dataset domain [26]. On the other hand, IBW, ANN and SVR are less sensitive to the splitting of the data and perform well on all subsets. It is interesting to note that the variation of the performance of DT and IBW among different training and testing datasets resemble one another, but the generalization error of DT is no lower than that of IBW in all five cases. IBW and DT are similar to one another in that they both partition the input space into small subregions, then fit simple functions within each subregion. However, DT partitions the input space into non-overlapping rectangular subregions while IBW divides the space into overlapping neighborhoods of irregular shape based on the number of neighbors. In addition, DT fits constant values within each rectangular support, thus its regression surface is characterized by non-continuous “platforms”. It is therefore not surprising that the performance of IBW is better than DT.

While Figure 15 and 16 suggest that ANN performs well with different training and testing dataset, its performance is subject to instability due to the random initial weights assigned to the network before being trained, and the algorithm converges to local minima instead of the global minimum. As mentioned earlier, when used to forecast the PBM's error after the year 2000, the ANN model is re-trained five times using different initial weights. The prediction error of the five trials varies from 18.3 to 22.9 ft. It has been recognized in the literature [6, 39, 26, 10] that SVR is superior to many traditional regression techniques, including ANN, because it is designed to minimize an upper bound of the generalization error rather than the training error, and is not prone to local minima (please refer to section 2.2.5 for details). Another disadvantage of ANN is its relatively long training time. Consequently IBW and SVR are considered to be the better DDMs, and are therefore employed in TP and SP scenarios.

It is interesting to note that the global bias (ME) resulting from different DDMs may have different signs. For example, ME of DT updated waterlevel predictions is -1.84ft., while the ME after SVR updating is 1.59 ft. The similar magnitude and opposite signs of ME indicate that better results can be achieved by combining the results of DT and SVR.

3.5.2 Performance of IBW and SVR

To compare DDMs performance in different prediction scenarios, the mean error (ME) and root mean squared error (RMSE) of the head predicted by the MODFLOW model ($\hat{h}$) and IBW/SVR corrected model ($\hat{h}^{new}$) are summarized in Table 5. In all three prediction scenarios, both IBW and SVR effectively improved the accuracy of head prediction of the MODFLOW model, reducing the RMSE by over 82% (TP), 60%(SP) and 48% (TSP). The DDMs removed most of the global bias, reducing the ME to around zero. Figure 17 shows the estimated cumulative distribution function of the absolute residual before and after DDMs updating. Point (x, y) in this plot means that y portion of the waterlevel observations have absolute residual that is no larger than x ft. In all three prediction scenarios, the lines of DDM-corrected results lie to the left of the thick grey line representing the raw MODFLOW model results, indicating the reduction in magnitude of residuals.

Best overall performance was achieved in the temporal prediction scenario, mainly due to the strong temporal correlation in the residuals, and because the prediction lead time (seven years) is relatively short compared with the time range of the training data (83 years). On the other hand, the performance of DDMs in SP and TSP cases is not as good as in TP, due to the varying residual patterns among wells and the relatively sparse data coverage in space. The improvement of prediction accuracy in TSP is less than that in SP, as the former involves temporal extrapolation in addition to spatial interpolation.

Figures 19, 20 and 21 present the hydrographs of representative wells at the locations shown in Figure 18. In general, the DDMs significantly improved the prediction accuracy. For those wells where the MODFLOW model predicted the trend of water level correctly but with bias, the DDMs "shifted" the MODFLOW prediction to correct the bias (Figure 19a, Figure 21a and b). In cases where MODFLOW made inaccurate prediction of the trend, the DDMs can still compensate, however the effectiveness of DDMs for trend correction varied for each well. For instance, in Figure 20a, both IBW and SVR corrected the shape of hydrograph simulated by the MODFLOW model. In Figure 19d, SVR did better than IBW in adjusting the hydrograph predicted by PBM. In Figure 19b and 20b, SVR corrected the trend of the MODFLOW model's prediction and yielded a good fit with the observations, while IBW maintained the trend of the MODFLOW model prediction. Figure 20c, 21c and 21d show cases in which neither IBW nor SVR effectively corrected the shape of PBM predicted hydrographs. The performance of DDMs on these wells was not satisfactory.

In general, both IBW and SVR yielded relatively smooth prediction compared with the fluctuating measurements, because the DDMs do not account for measurement error, and the regularization scheme discourages a complex estimation surface. Specifically, the DDMs had difficulty in recovering the interannual waterlevel fluctuation of the measurements (e.g. Figure 19c and 20b) at some wells. This problem may be alleviated by adding pumping rate, evaporation and precipitation data into the input features of DDMs. In some cases, wiggling occurred in DDMs estimates that was not found in observation or PBM prediction, because DDMs may choose as neighbors or support vectors those data points that were actually irrelevant to the query. A likely solution to this problem is to increase the number of clusters, so that the DDMs can be tuned to better adapt to the local patterns of the residuals.

3.5.3 Discussion on TP Results

The results of temporal prediction case are analyzed in more detail here. The mean error of the waterlevel simulation at each well during prediction period (2001-2007) is plotted in Figure 22. The magnitude of residuals was significantly reduced by the DDMs. Residuals greater than 50 ft or less than -50 ft were eliminated, and the local patterns were largely removed except in the northeastern part. In that mounding area, the DDMs overly adjusted the waterlevel simulation of the RRCA model, in that the overpredicting was inversed to underpredicting. The reason is still unclear.

Figure 23 shows the empirical semi-variograms of the mean error at each well before and after DDMs updating. It can easily be seen from 23a that some spatial structure exists with *sill* around 1100 sft. and *range* of approximately 40 miles. The spatial dependence is reduced as shown by Figure 23b. On the other hand, Figure 24 shows that, while the positive correlation of residuals with time was reduced for most wells, the DDMs introduced negative temporal correlation (DW¿2) at some wells.

From the perspective of water resource management, it is important to know how the waterlevel would change in the future, because such trend serves as the major indicator of the change in groundwater storage, and is related to the efficiency of groundwater development. In addition to the hydrographs at representative wells shown in Figure 19 to 21, the global trend of waterlevel is plotted in Figure 25. Only the wells that have waterlevel measurements from 2001 to 2007 are included in the plots. The complementary framework with SVR generated better prediction of the change in waterlevel than the MODFLOW model in the western part of the basin. However it worked no better than the PBM in the northeastern part (the mound area).

3.5.4 MI Scores Before and After DDMs Updating

The DW test and semi-variogram test results of SP and TSP are not presented here to avoid redundancy. Instead, the change in MI scores between input features and the residuals before and after applying the complementary framework are summarized in Figure 26. To compute the MI scores, the input features $(x_w, y_w, t, \hat{h})$ were scaled so that the values of each feature had zero mean and unit variance. The residuals were not scaled because scaling $\epsilon, \hat{\epsilon}$ respectively would make the MI scores incomparable. Comparison cannot be made across prediction scenarios, because the input features are scaled differently. Generally, the DDMs reduced the dependence of the residuals on space and time, as well as computed head, with the only exception a slight increase of MI scores associated with t in TP case. Hence the temporal and spatial dependency of the residual reported in section 3.3 have been alleviated.

3.6 Summary

The complementary framework described in Chapter 2 was tested on a case study based on the RRCA model. The results presented in this chapter show that given structure existing in the error of the PBM, DDMs can learn the temporal and spatial patterns of the residual from historical data. The trained DDMs effectively predicted the error of PBM in different testing scenarios (TP, SP and TSP). It is shown that the head prediction corrected by DDMs more closely fits the observations than the simulation of the MODFLOW model. In addition, the DDMs reduced the degree of temporal and spatial dependence of the residual. Furthermore, the performance of several types of machine learning techniques is compared in TSP scenario. It is found that IBW, SVR and ANN yield more accurate head prediction than *loess* and DT. ANN requires longer training time than other DDMs, and its performance is not stable. Consequently IBW and SVR are considered as superior and are hence further tested in TP and SP cases.

While the DDMs prove to be effective, their performance at some wells is still unsatisfactory. This suggests that a part of the residual cannot be modeled based on the input features included. By incorporating more inputs, for example, pumping rate, precipitation and evaporation, the DDMs could be improved to better compensate for the structural defects and parametric uncertainty of the PBM.

Chapter 4

CASE STUDY OF THE SPOKANE VALLEY RATHDRUM PRAIRIE MODEL

4.1 The SVRP Model

As the second real-world case study, the complimentary modeling framework was applied to improve the head prediction of the Spokane Valley-Rathdrum Prairie (SVRP) model. The SVRP aquifer covers approximately 326 square miles across the states of Idaho and Washington, and supplies drinking water to more than 500,000 residents. A MODFLOW-2000 model was jointly developed by the USGS, Idaho Department of Water Resources, the University of Idaho, and Washington State University. The model has a uniform cell size of 1,320 by 1,320 ft., and stress period of 1 month from September 1990 through September 2005. The SVRP aquifer is represented by one active layer except in Hillyard Trough and Little Spokane River Arm. In those areas, the model has three active layers (shown in Figure 27). The model was calibrated by PEST (version 10), a model independent parameter estimation tool, using as calibration targets over 1,500 groundwater level measurements and 313 measurements of streamflow gains and losses along segments of the Spokane and Little Spokane Rivers during October 1995 to September 2005. The first five years of the simulation are considered as the warm-up period, thus observations prior to October 1995 are excluded from the calibration data. Model parameters calibrated include hydraulic conductivity, specific yield, vertical hydraulic conductivity of riverbed sediments, and hydraulic conductance of riverbed and lakebed sediments. For more details about the model, readers are referred to the documentation [29] that is available on the project website (http://wa.water.usgs.gov/projects/svrp/summary.htm). The cell-by-cell head output file of the model and calibration targets were obtained from the model developer.

The database used in this case study consists of the waterlevel calibration dataset, plus additional observations from October 1995 to September 2005 that became available via the USGS Water Data for the Nation online database (http://waterdata.usgs.gov/nwis/gw) after construction of the model. The data preparation process is similar to the process described in section 3.2. In contrast with the RRCA case study, no data beyond the calibration period is included because of the lack of publicly available input or output data after September 2005. As in the RRCA case study, simulated waterlevel corresponding to a measurement is linearly interpolated from the cell-by-cell head output file of the MODFLOW model using `mod2obs`. In total, the database contains 2196 simulated and measured head pairs.

Water-level measurements at several wells in the calibration dataset are excluded from the calibration data. A well is excluded if it is in bedrock, or meets the groundwater mound beneath the losing segment of the Spokane River, or is located along the model boundary. The latter two cases are considered as model defects, and thus are expected to be modeled by the DDMs. Because of this, and also due to the lack of information to identify wells that are in the bedrock, these wells are included in the database used in this thesis.

4.2 Residual Analysis

The residuals from Oct. 1995 to Sep. 2004 were analyzed. As will be explained in the next section, data during October 2004 to September 2005 were retained as testing data, and thus were considered unknown at this stage.

Calibration of the SVRP model was implemented as nonlinear least square regression that sought to minimize the discrepancy between observation and simulation. The model was well calibrated, and a satisfactory match to calibration targets was achieved. The mean error at calibration wells is as low as -0.23 ft., and the RMSE is 3.65 ft. On the other hand, the ME and RMSE of simulated head at non-calibrated wells have larger magnitudes, being 15.29 ft. and 29.97 ft., respectively. Overall, for all the observations in the database, the ME is 4.48 ft., and the RMSE is 15.73 ft. The residuals are plotted against the standard normal distribution in Figure 28. While approximately 60% of the residuals fit the normal distributions well, their distributions are tailed and skewed (asymmetric). In addition, the prediction of the MODFLOW model is overall biased, with the mean error deviating from zero.

Mean error at each well is shown in Figure 29. In contrast with the RRCA case study, the strong spatial structure or local patterns of residuals exhibited in Figure 7 is not found in this plot. The weaker degree of spatial dependence is further shown in Figure 30 a-c. The empirical semivariograms show a more prominent nugget effect and smaller range (if it exists at all) compared with Figure 8. Nugget refers to the height of the jump of the semivariogram at the discontinuity at the origin. It represents variations of the residuals at a much smaller scale than any of the measured pairwise distances, for example, due to measurement error. It can be seen that the ratio of the nugget to the sill (limit of the variogram tending to infinite distance) in Figure 30 a-c is larger than that in Figure 8. The range refers to the distance at which the difference of the semivariogram from the sill becomes negligible. It indicates the spatial scale of the structure of the residual, and a short range suggests weaker spatial dependency.

While the spatial dependence is weak, strong temporal dependence is found in the residuals. Figure 30d plots the DW statistics calculated at 38 wells selected from the 351 wells included in the database. These wells are monitored with relatively uniform time interval, and their locations are scattered throughout the model domain. The DW statistics of most wells are significantly smaller than 2, an indication of the presence of temporal correlation stated in section **??**.

4.3 Framework Implementation

The residual analysis results reveal strong temporal yet weak spatial dependence, therefore DDMs were applied only in the temporal prediction (TP) scenario. Unlike in the RRCA case study, the SVRP model is not run by the model developers beyond the calibration period to forecast the groundwater flow condition in the future with new input data. To run the model in temporal prediction mode, a variety of input stress data including well pumping, recharge, and inflow from tributary basins and lakes to the SVRP aquifer needs to be updated. Calculating new recharge input requires data on precipitation, and return flow infiltration rate, amount and distribution of (im)permeable surface, and transmission time of downward infiltration to the groundwater table. Calculating the pumping rate requires pumping records of water utilities to compute domestic water usage, and irrigation acreage and crop water demand to estimate agricultural widthdrawal. Much of this required information is not available in public domain, and consequently it is not possible to prepare new input data to run the SVRP model in the future. As a result, no waterlevel prediction is available beyond the calibration period (October 1995 to September 2005). It is therefore not possible to use DDMs to predict the forecast error of the calibrated model after September 2005. Instead, the data from October 1995 to September 2004 was used as the training set, and the data from October 2004 to September 2005 is used to validate the DDMs. The validation set is considered as unknown during the stages of residual analysis and development of the DDMs. The training set is comprised of 1,556 data values at 346 wells, and the validation set includes 635 values at 53 wells out of the 346 wells in the training set. Five wells which only have observations since October 2004 are excluded.

IBW and SVR models were built to forecast the error (ϵ) of the SVRP model simulated head ($\hat{h}$), and the updated head $\hat{h}^{new}$ was then computed by adding the error $\hat{\epsilon}$ to $\hat{h}$. The DDMs took as input features the well location (x_w, y_w) and MODFLOW computed head $\hat{h}$. The measurement time t was excluded because: 1) in the TP scenario, DDMs were used for extrapolation in terms of t; 2) cross validation error with t as an input was larger than the error without t; 3) the MI scores between the residual in the training set with $x, y, t, \hat{h}$ is 0.185,0.193,0.092,0.1394 respectively, thus t is less related to the residual. Input features were scaled to the interval $(-1, 1)$ to ensure that they had the same magnitude and to improve numerical stability and efficiency. Local patterns were not found in the residuals and the dataset is small, hence it is not necessary to implement clustering analysis before implementing DDMs updating. Thus the IBW and SVR models were trained with the whole training set.

Preliminary experiments showed that for IBW, the inverse distance weighting function (Eq. (2.9a)) outperformed the exponential weighting function (Eq. (2.9b)), therefore the former was adopted. The parameters of IBW, the number of neighbors (n) and the power p in Eq. (2.9a) were tuned via five-fold cross validation with the training set. For details of implementing cross validation, please refer to section 3.4.The cross validation results are shown in Figure 31. In Figure 31b the CV curves of higher powers are rather flat beyond 100. The largest number of neighbors plotted is 1,244, corresponding to the largest value possible in cross validation. From Figure 31b it is appealing to choose $n = 1244, p = 2$. Figure 31b, however, shows that this set of parameters yields relatively large bias (-0.61 ft.) that exceeds a standard error above the lowest bias. A compromise between minimizing the magnitudes of RMSE and ME is to choose $n = 1244, p = 3$, which yields ME of -0.27 ft. and RMSE of 8.38 ft.

The parameters of SVR were chosen analytically and via cross validation, as described in section 2.2.5. The parameter C that determines the trade-off between goodness-of-fit and model complexity was computed by Eq. (2.22a), and the error insensitive parameter ε was given by Eq. (2.22b). The kernel width parameter γ was tuned by five-fold cross validation implemented on the training set, as shown in Figure 32. The averaged CV error appears to favor larger γ which means smaller kernel width. However the CV results of IBW suggest to use large number of neighbors. Here we follow the one-standard error rule recommended by [26], that the most parsimonious model whose error is no more than one standard error above the error of the best model should be chosen.Small value of γ indicates larger kernel width (Eq. (2.21)), and tends to yield a smoother regression surface. Consequently $\gamma = 450$ is selected. It corresponds to mean error of -3.95 ft., slightly lower than the best ME (-3.75 ft.). The selected parameters for IBW and SVR are summarized in Table 6.

4.4 Results and Discussion

The DDMs with the selected parameter values were tested with the validation set. The performance of DDMs is shown in Figure 33. Both IBW and SVR effectively reduced the residuals. IBW slightly outperformed SVR, reducing ME and RMSE by 69%, 78% respectively. The effectiveness of DDMs is further demonstrated in the residual plots in Figure 34. The bias shown in Figure 34a is largely removed in Figure 34b and 34c. For example, the SVRP model tends to underpredict the head in the range of 1950 2000 ft. This is corrected by IBW and SVR. The mean error at each well of the testing set (October 2004 - September 2005) before and after DDMs updating is shown in Figure 35. IBW reduced the mean errors of all wells to the range of -10 to 10ft.

Figure 37 shows the hydrographs of representative wells. The locations of the selected wells are shown in Figure 36. IBW improved the head prediction accuracy at all four wells. It yields a rather wiggling regression surface despite of the large number of neighbors used. This is because the weighting function with power $p = 3$ assigns small weights except to a few nearby neighbors (Figure 2a). In Figure 37d, IBW adjusted the hydrograph so that its shape fits the observations better, but did not improve the shape of hydrograph in b and c. On the other hand, SVR yielded smoother estimation surface and could not replicate the water level fluctuation. It reduced the error of MODFLOW model in Figure 37a-b, but did not perform well in c-d.

Figure 38 shows the semi-variograms of the residuals in the validation set before and after correction by IBW and SVR. The semivariograms in Figure 38b have a larger nugget than the semivariogram in Figure 38a. It can be seen that the ratio of the nugget to the sill in Figure 38b is larger than that in Figure 38a. This suggests that after the DDMs updating, the random error (for example the measurement error) becomes the principle component in the uncertainty while systematic residuals have been largely removed. On the other hand, the DDMs reduced but did not eliminate the temporal dependence of the residuals. As shown in Figure 39, the DW statistics of residuals increase after the correction by the DDMs, consistent with a reduction in temporal correlation.

It is interesting to note that the DDMs parameters are adapted to the structure of residual. For example, residual semivariograms in the RRCA case study (Figure 8) have larger range compared with the semivariograms in this case. As a result, the SVR model developed in this chapter adopted a smaller kernel width. Taking into account the scaling factors of input features, the kernel width parameter $\gamma = 450$ corresponds to a distance of 1.18×10^4ft., which is of the same order of magnitude with the range of the semivariograms (approximately 2×10^4 ft.). Rigorous inference of the DDMs parameter, of course, calls for semivariogram analysis in the three-dimensional input space.

4.5 Summary

In this chapter, the complementary modeling framework described in Chapter 2 was tested on a second case study based on the SVRP model. Unlike the RRCA model, the SVRP model is calibrated to a better degree using nonlinear regression techniques. Strong temporal dependence but weak spatial dependence is found, hence the framework is only implemented in the temporal prediction scenario. It has been shown that the DDM parameters are related to the residual structure, which can be revealed by residual analysis techniques. Both DDMs significantly reduced the head prediction error of the MODFLOW model, and IBW slightly outperformed SVR. The effectiveness of DDMs of correcting the trend of waterlevel trend, however, needs to be improved.

It should be noted that while the SVRP case study serves as a good example where the PBM is more sophisticatedly calibrated than the RRCA model, it is not as rigorous a validation test case as the RRCA model, because no PBM predictions beyond the calibration period are available, as explained in section 4.3. Because of this, the DDMs were trained on residuals from the first nine years of the calibration period.

Chapter 5

CONCLUSION

This work presents an extension of the complementary data-driven framework developed and applied in [15, 14, 43, 49]. Although a well-constructed and calibrated physically based model can provide useful information about the system behavior, the unaccounted uncertainties involved in conceptualization, parameterization and calibration can lead to predictions contaminated by systematic bias. The framework constructs data-driven models based on machine learning techniques and trains them using historical residuals of the physically based groundwater flow models. The trained DDMs are expected to have the capability to compensate for the defects of the PBM. These DDMs are used to correct the forecast of the PBM, leading to reduction in bias of the forecast.

In this thesis, cluster analysis was introduced to make the framework more robust and efficient for complex real-world case studies. Localized DDMs conditioned on the data within each cluster can better adapt to the local patterns of the residual, and are thus more flexible than global DDMs without clustering. It is also found that the use of cluster analysis before implementing DDMs updating sufficiently reduces the computational cost required by training and calibration of the DDMs. Five machine learning techniques (IBW, *loess*, DT, ANN and SVR) were employed to build DDMs, and their performance to enhance the predictive ability of the PBM was compared in the TSP scenario of the RRCA case study. It was shown that IBW, SVR and ANN yielded more accurate waterlevel prediction than *loess* and DT. Among the DDMs, ANN requires significantly longer training time than other models. Consequently IBW and SVR were selected and further tested in the TP and SP scenario, and the SVRP case study.

This framework is shown to successfully improve the head prediction accuracy of two regional groundwater flow models in various prediction scenarios. In the RRCA case study, the DDMs reduced the RMSE by 82% (TP), 60% (SP) and 48% (TSP). In the SVRP case study, the DDMs reduced the RMSE for the TP scenario from 17.47ft. to as low as 3.74 ft. in one-year forecasting. It should be noted that the RRCA model is a more valid test case, because PBM predictions beyond the calibration period are available unlike in the SVRP case study. Best overall performance is found in the temporal prediction scenario, because of the relatively short lead time of forecast, denser data coverage and stronger temporal correlation.

Compared with the conventional approaches for improving model predictions reviewed in Chapter 1, the strength of the presented approach lies in:

- It can handle error from multiple sources. Compared with calibration and multimodel approaches that only account for parameter and/or structural error, the presented framework is advantageous if the sources of prediction error are multiple and not easily identifiable.
- The framework does not invoke any assumption on the error distribution, in contrast to conventional regression-based calibration and the statistical residual modeling approaches.
- The approach is easy to implement. It is computationally efficient compared with calibration and Monte-Carlo simulation based methods that run the PBM numerous times. In addition, the framework does not call for the establishment of a set of competing conceptual models, which is required by multimodel methods.
- The framework shares the advantage of data-assimilation techniques in that newly available data can be easily incorporated to update the DDMs.

One limitation of the method is that the effectiveness of the DDMs depends on the presence and degree of spatial and temporal patterns in the residuals. A possible solution to enhance the robustness of DDMs is to incorporate more input features and to use more sophisticated feature selection and extraction techniques. Another limitation lies in the well-known problem of extrapolation of machine learning techniques, that generalization of DDMs would be risky if the situation represented by a query is beyond the scope of the training data. Although not examined in the two case studies presented here, temporal forecasting with long lead time might be challenging for the DDMs, particularly if the structural patterns in the residuals change over time. In addition to the problem of extrapolation, DDMs are often criticized for being difficult to interpret. Indeed, except for DT that can be straightforwardly expressed as a tree structure, other machine learning techniques used to construct the DDMs are considered as *black boxes*. As another limitation, using DDMs in the presented complementary fashion cannot preserve mass balance, which is the basis for the PBM.

The research presented in this thesis can be further extended in the following directions. The effectiveness of the DDMs can be enhanced with more sophisticated techniques of feature selection. The parameters of DDMs can be selected with the aid of residual analysis which reveals the data patterns. Furthermore, the potential of DDMs to facilitate prediction uncertainty quantification needs to be explored. In addition, the complementary framework can be extended to other types of predictions of interest and for solute transport problems. Applied to aggregate predictions like baseflow, the DDMs may be faced with the challenge that much less data are available, unlike in the presented two case studies with sufficient waterlevel measurements

Chapter 6

FIGURES

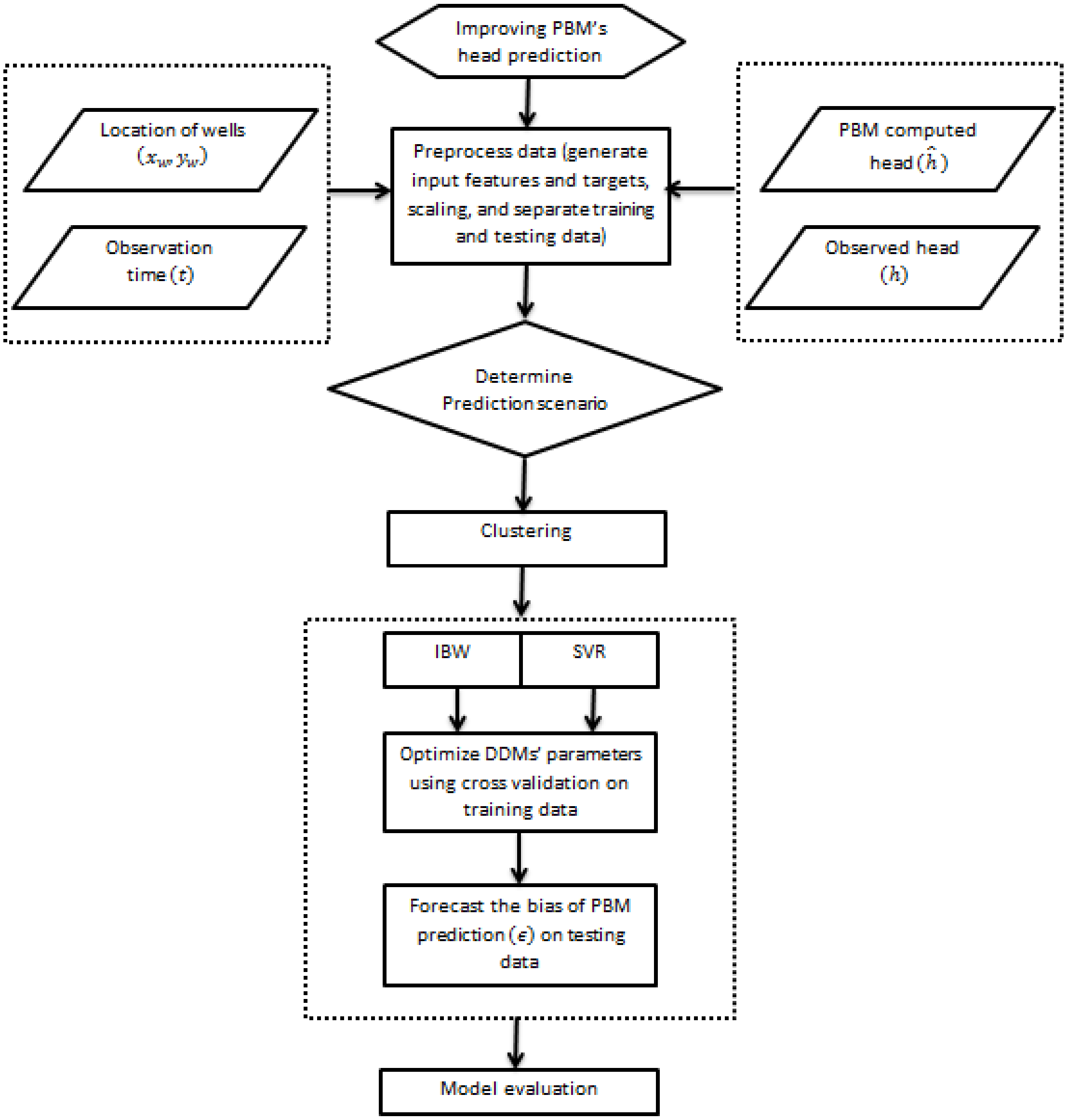
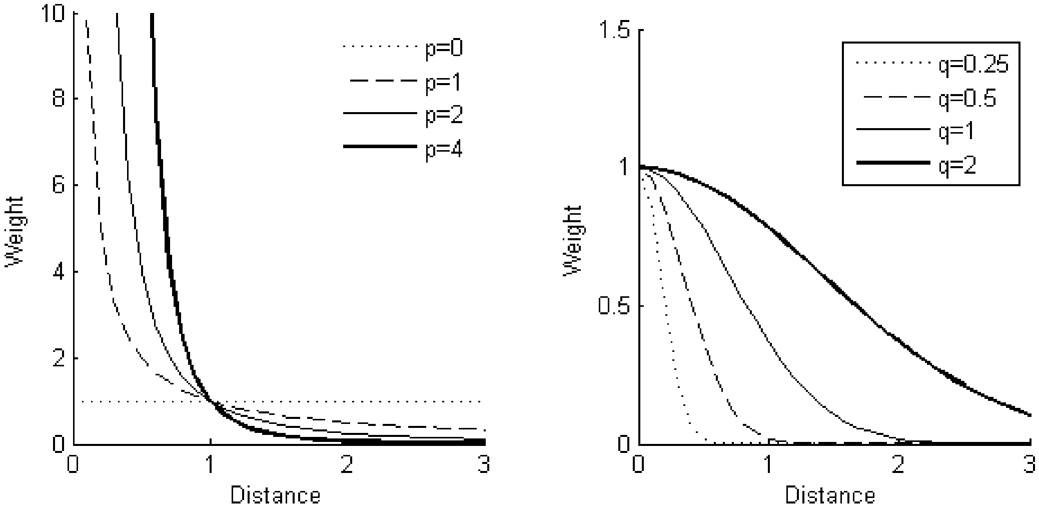

Improving PBM's head prediction
Location of wells (x_w, y_w)
Observation time (t)
Preprocess data (generate input features and targets, scaling, and separate training and testing data)
PBM computed head $(\hat{h})$
Observed head (h)
Determine Prediction scenario
Clustering
IBW
SVR
Optimize DDMs' parameters using cross validation on training data
Forecast the bias of PBM prediction (ϵ) on testing data
Model evaluation

Figure 1: The framework of using complementary DDMs to improve head prediction of PBM.



(a) Weight function (2.9a)

(b) Weight function (2.9b)

Figure 2: The two weight functions defined in Eqn. (2.9) with varying p, q. The horizontal axes denote the Euclidean distance $||\mathbf{x}' - \mathbf{x}_j^*||$, and the vertical axes show the corresponding weight $w_{\mathbf{x}'|\mathbf{x}_j^*}$. The scaling factors α, β are set to one.

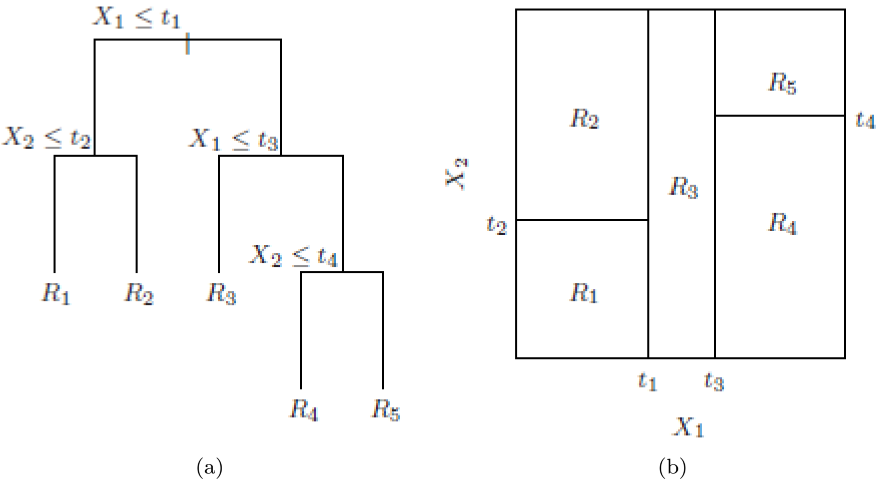


Figure 3: A decision tree with two input features (a) and its partition (b). Taken from [26].

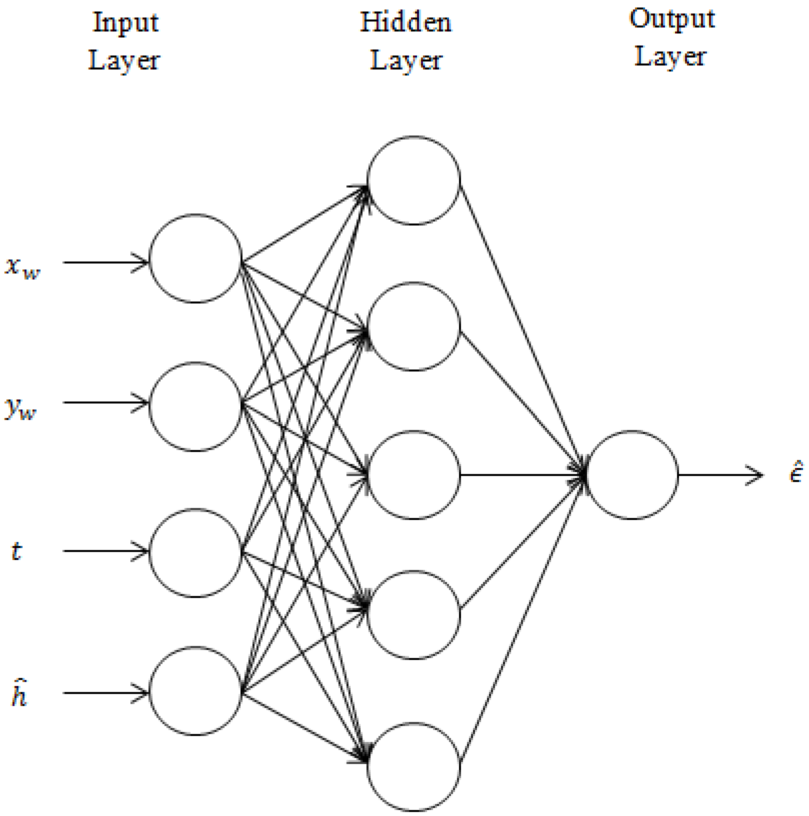


Figure 4: ANN with one hidden layer that estimates the residual based on input features of well location, observation time and MODFLOW computed head.

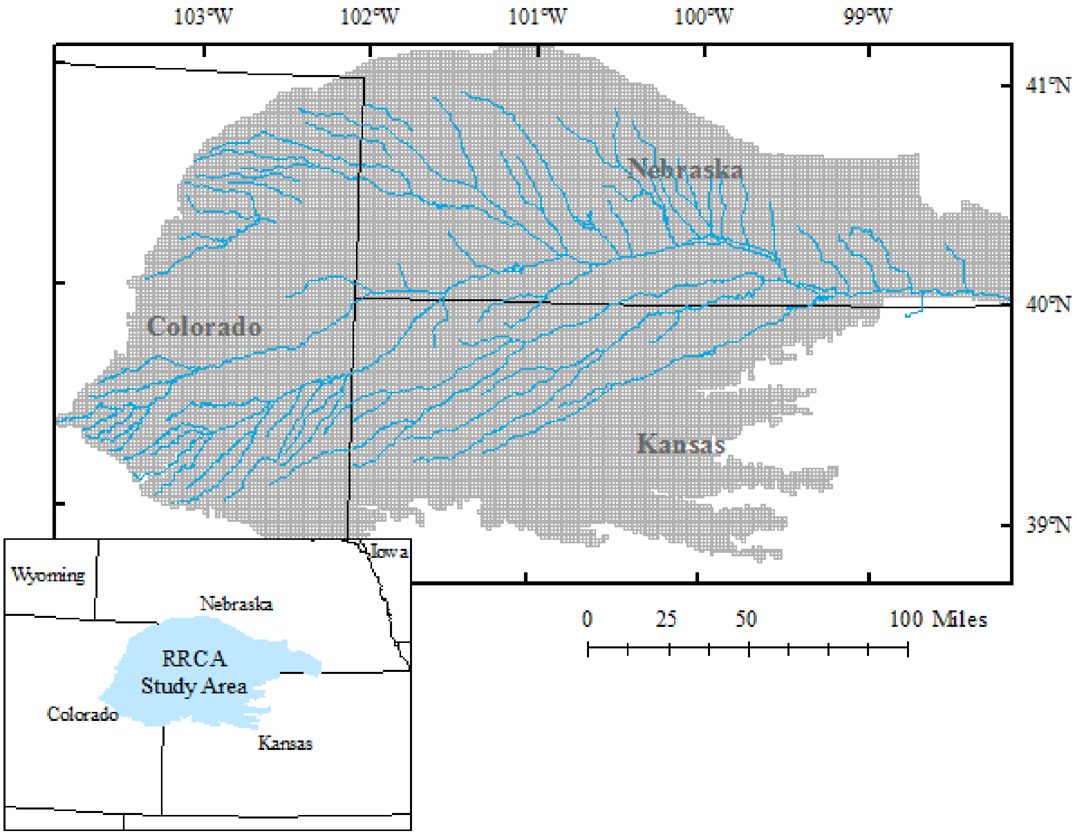


Figure 5: Republican River Basin covering portions of eastern Colorado, northwest Kansas and southwest Nebraska.

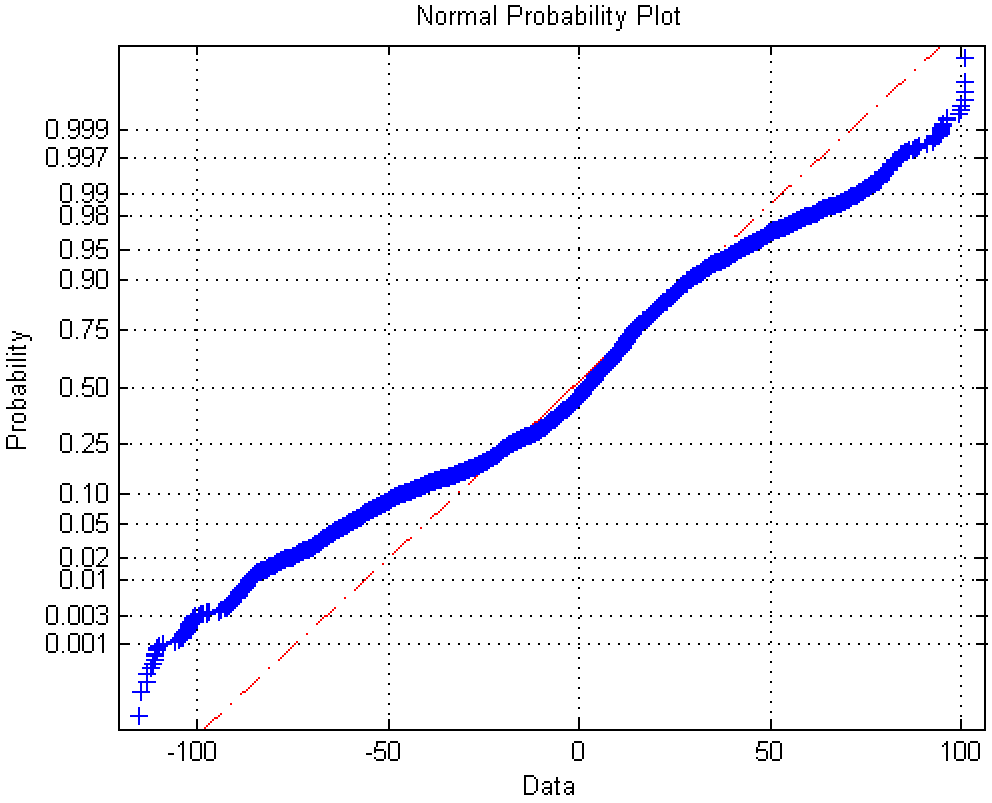


Figure 6: Normal probability plot of the residuals of the MODFLOW model.

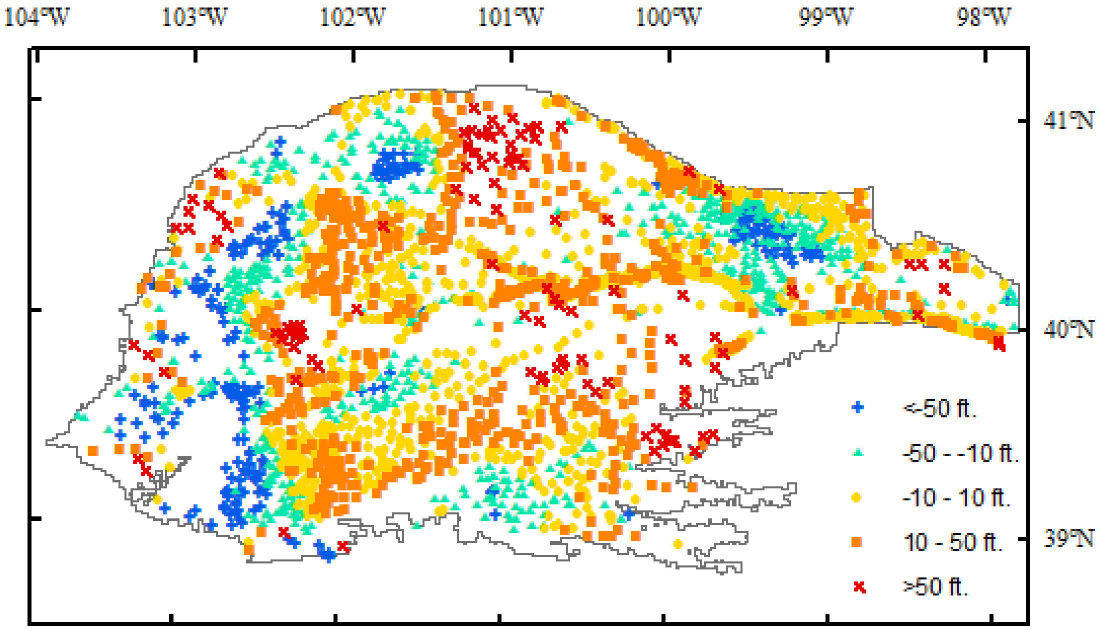
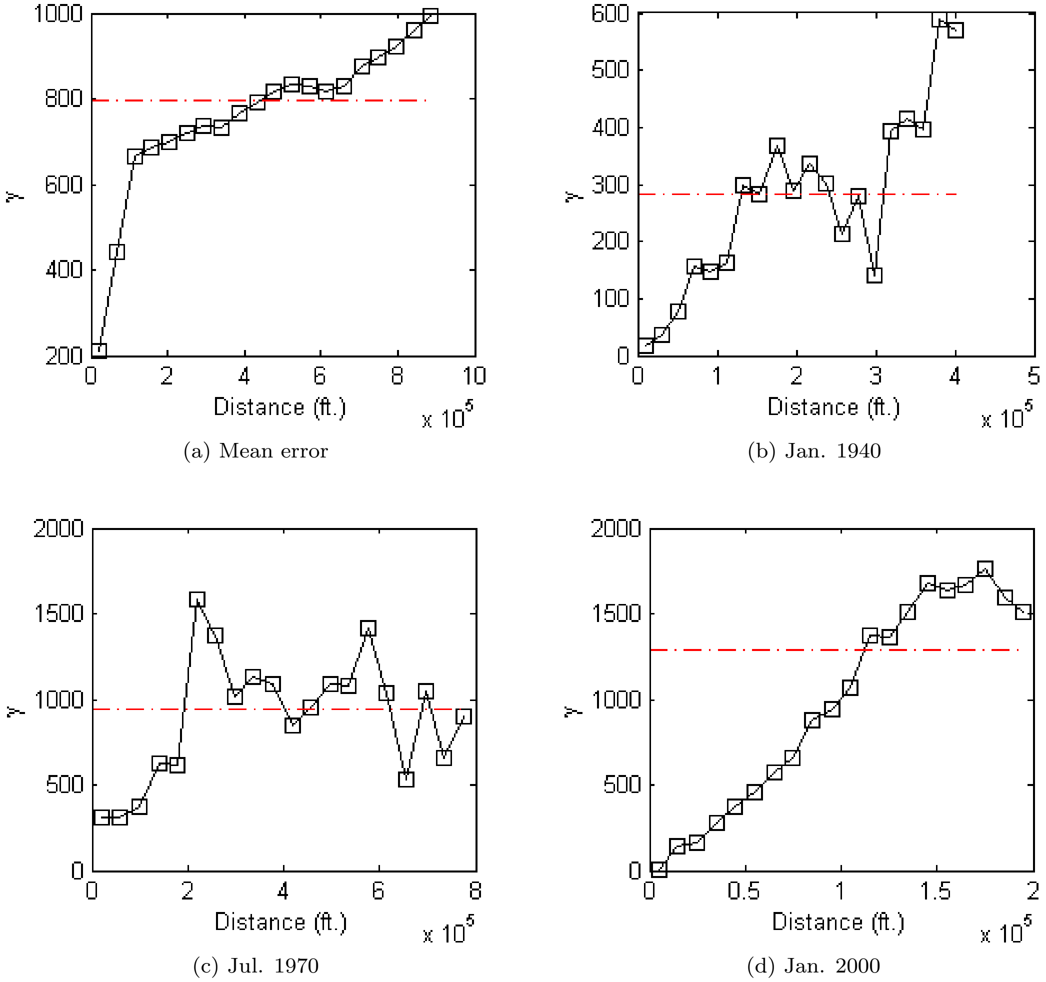


Figure 7: Mean error at individual well during calibration period (1918-2000).



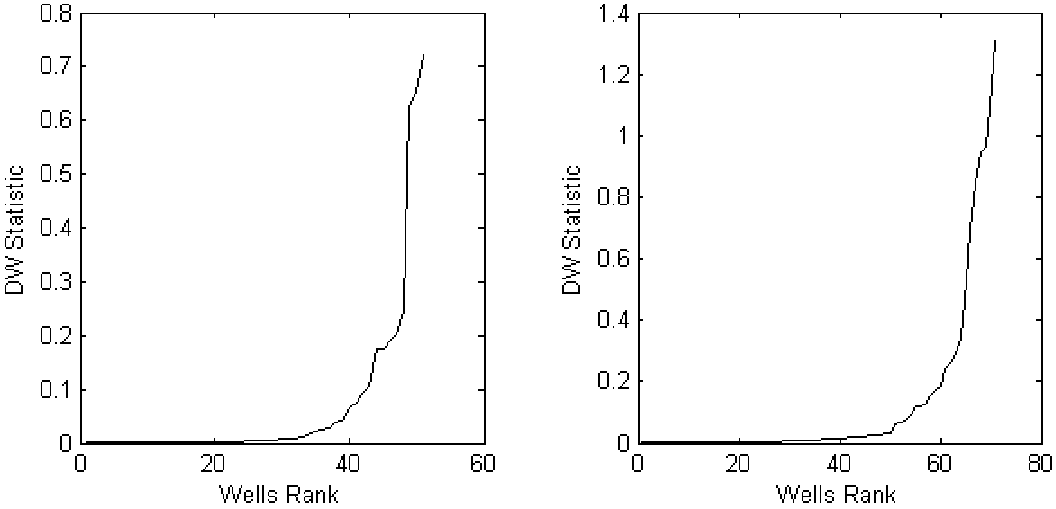
(a) Mean error

(b) Jan. 1940

(c) Jul. 1970

(d) Jan. 2000

Figure 8: The empirical semi-variograms of the residuals. The red dashed lines indicate the variance of the residuals. (a) is computed based on the mean error at each well, (b) is based on the data observed in Jan. 1940, (c) is based on data in Jul. 1970 and (d) is based on data in Jan. 2000.



(a)

(b)

Figure 9: DW statistics of selected wells of one-month (a) and one-year (b) lag. The wells are sorted in terms of DW statistic. The horizontal axes show the index of wells.

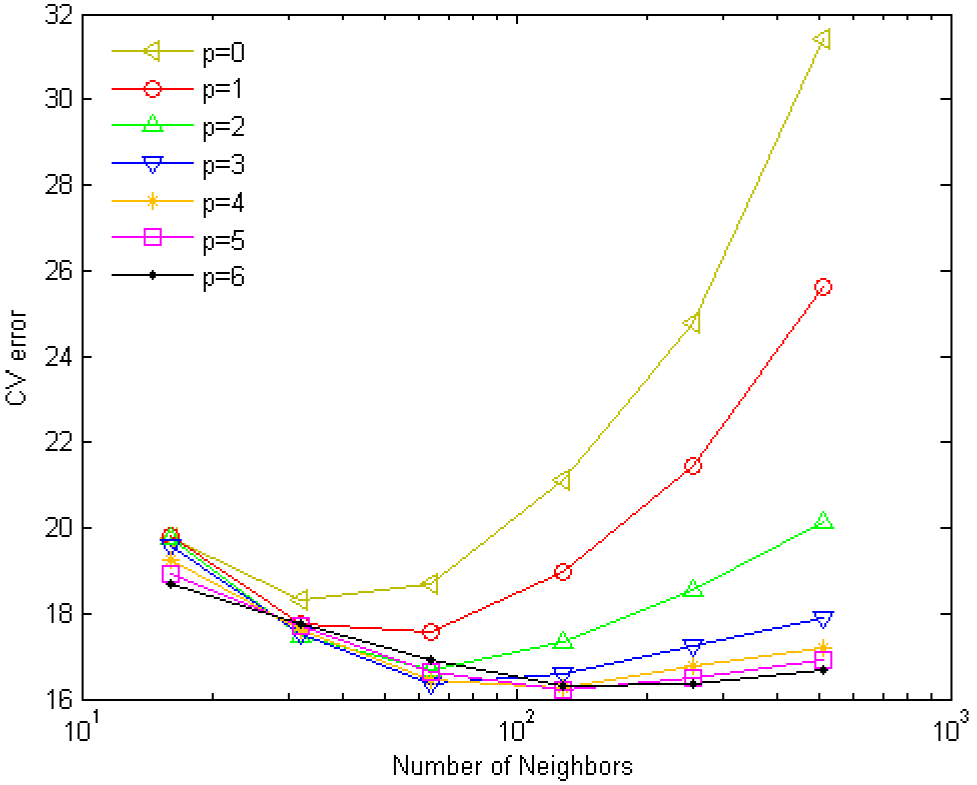


Figure 10: Cross validation results of IBW for selecting number of nearest neighbors n and the power p in weighting function. The horizontal axis denotes the values of n, and the lines represent different values of p.

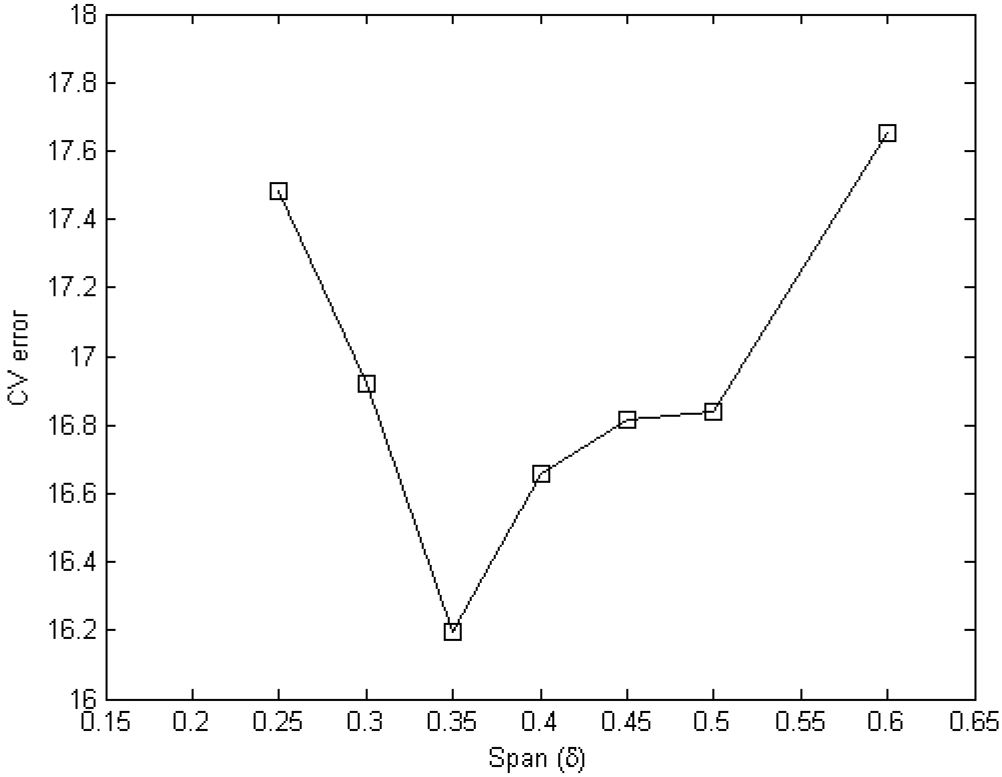


Figure 11: Cross validation results of *loess* for selecting the span parameter δ.

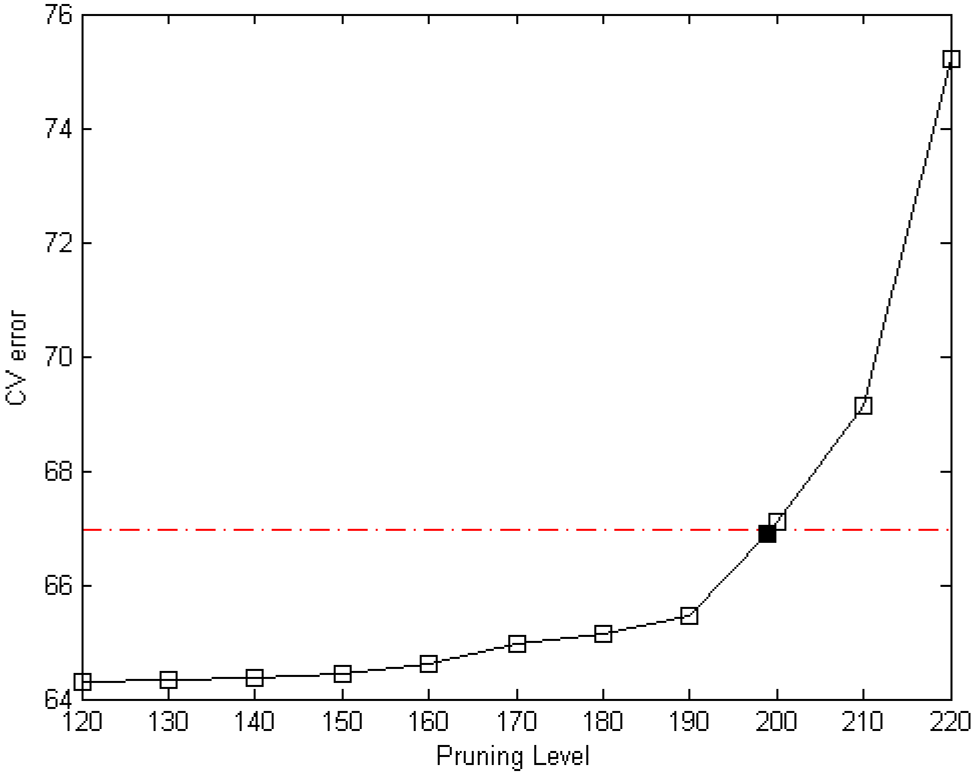


Figure 12: Cross validation results of DT for selecting the pruning level δ. The red dashed line is one standard error above the lowest CV error.

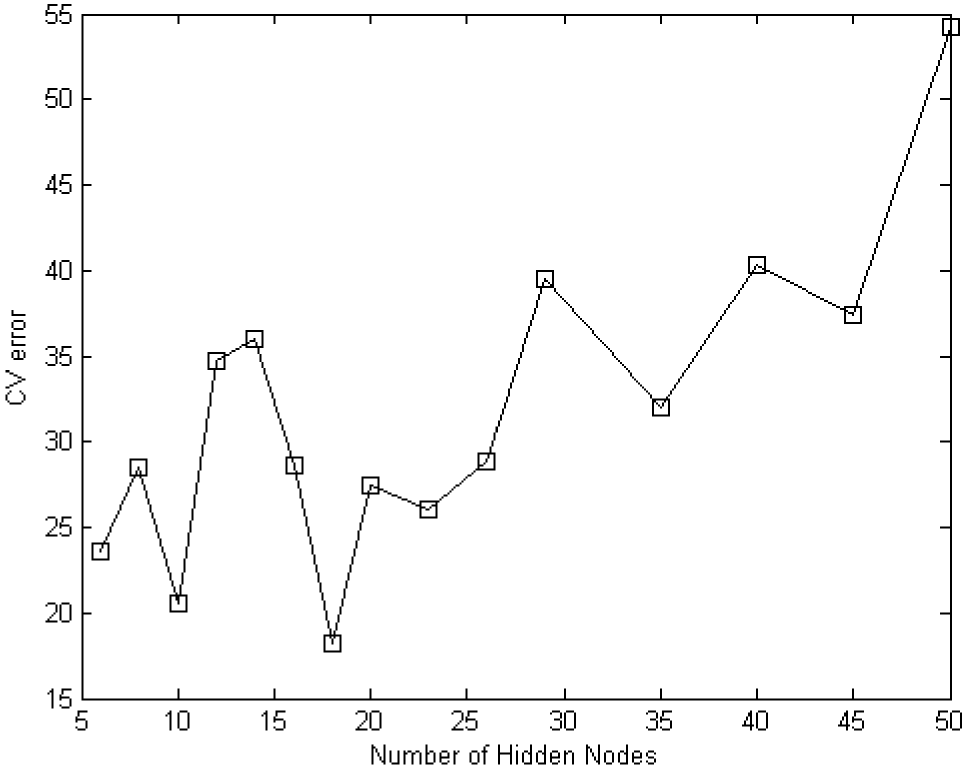


Figure 13: Cross validation results of ANN for selecting the number of hidden nodes.

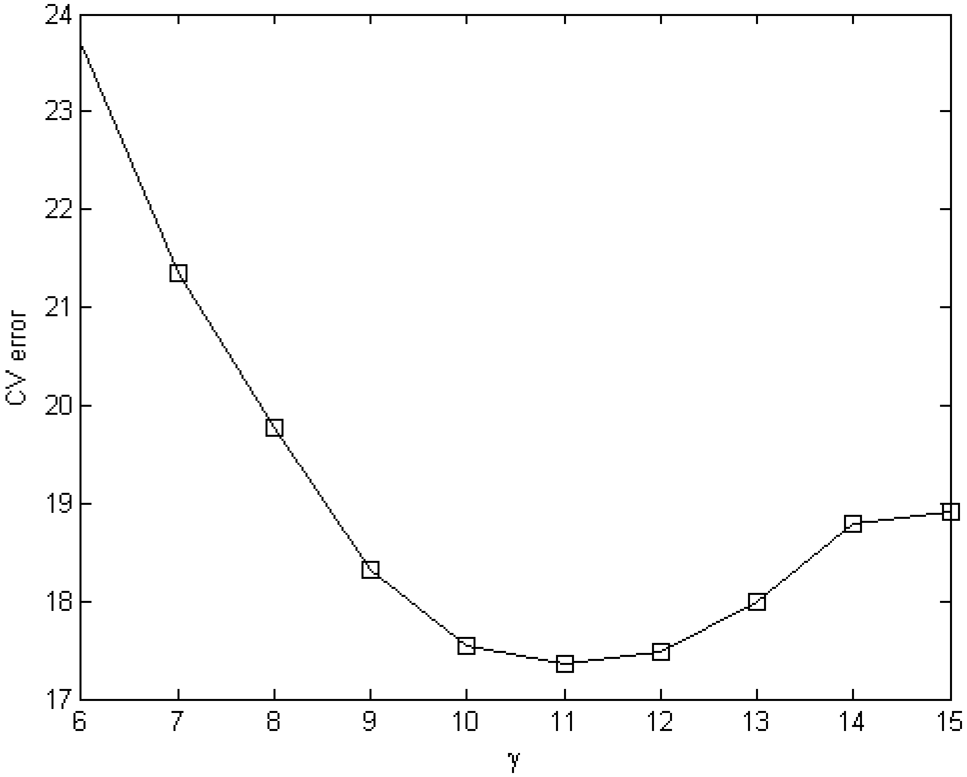


Figure 14: Cross validation results of SVR for selecting the kernel width γ of the RBF kernel defined in Eq. (2.21).

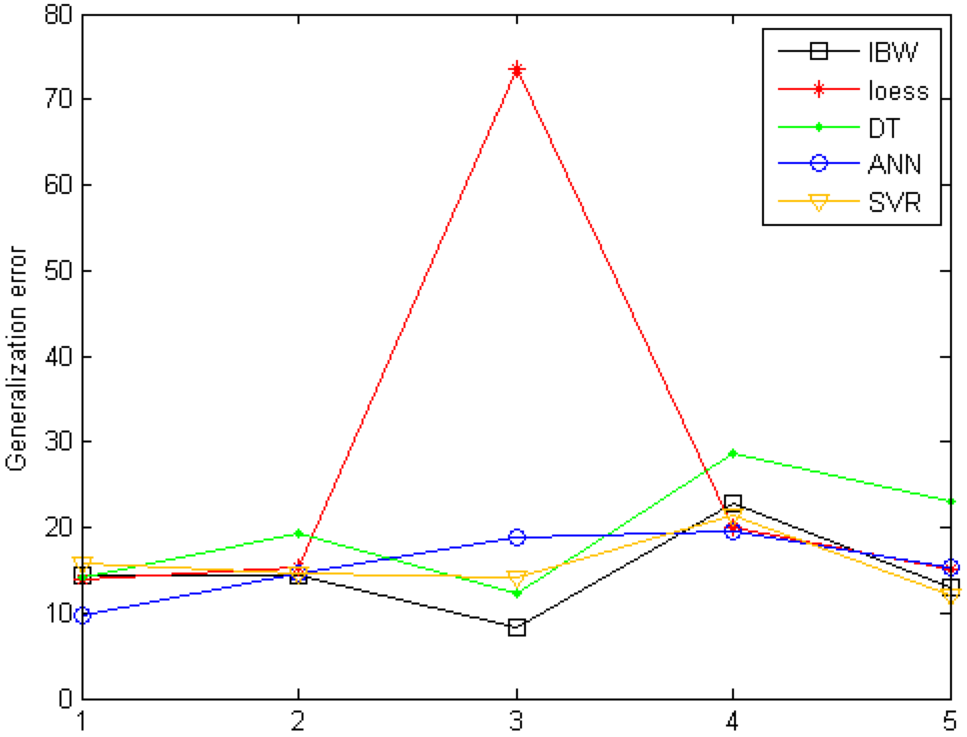


Figure 15: DDMs generalization error on five subsets. The numbers on the horizontal axis denote the individual cross validation subsets.

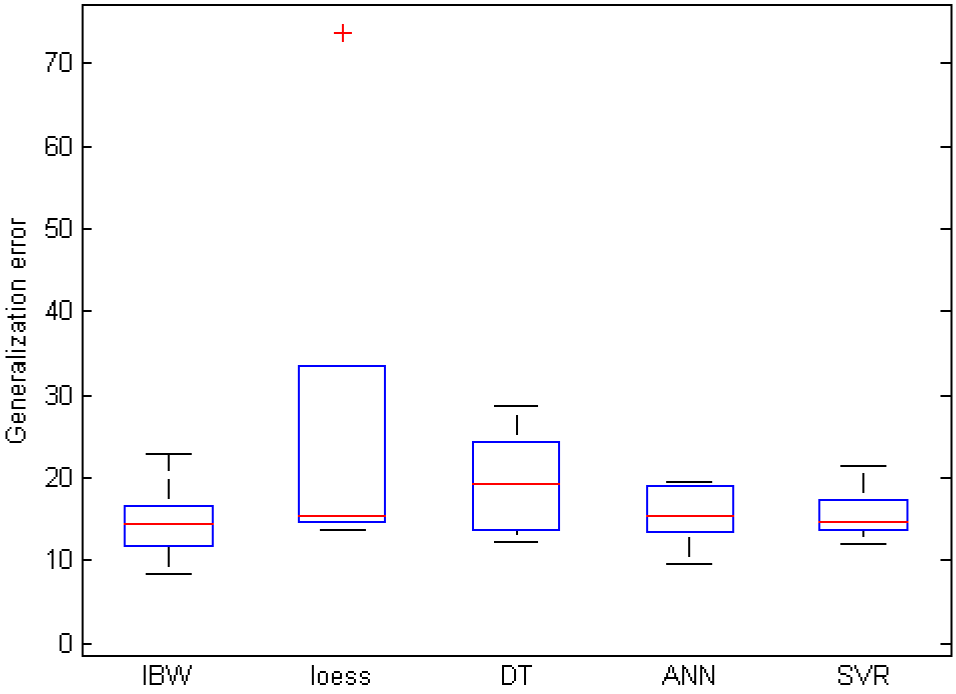


Figure 16: Box plot of the performance of various DDMs on five cross validation subsets. The central red line in each box is the median, the edges of the box are 25th and 75th quantiles, the whiskers extend to the most extreme data points not considered outlier, and outliere is marked as red plus sign.

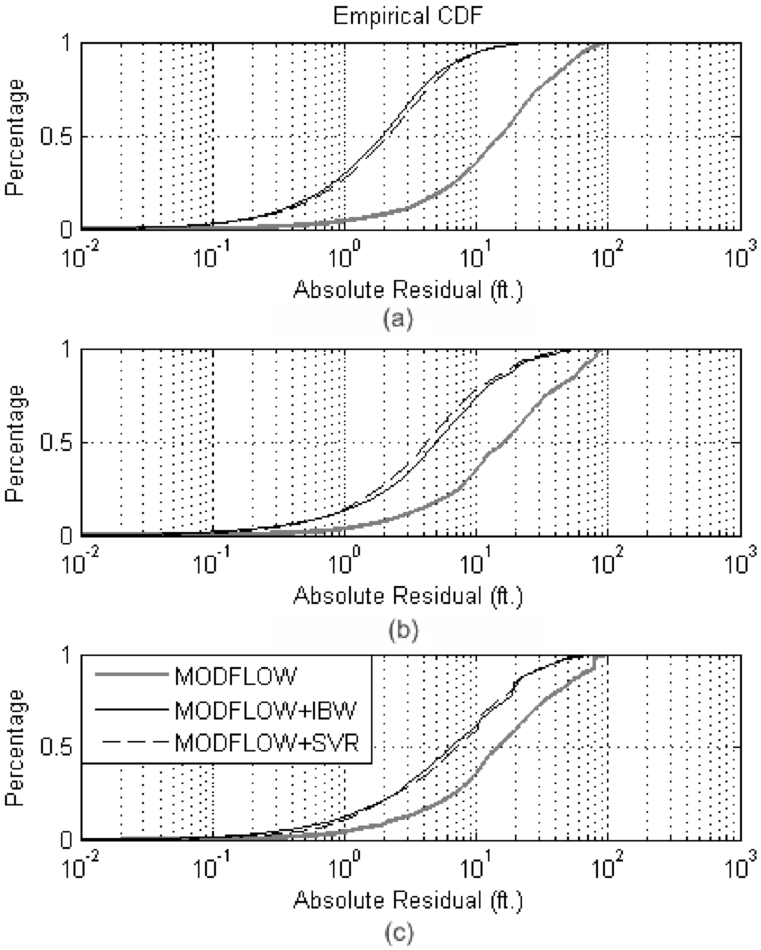


Figure 17: The Empirical CDF plot of absolute residuals before and after correcting using IBW and SVR of TP(a), SP(b) and TSP(c).

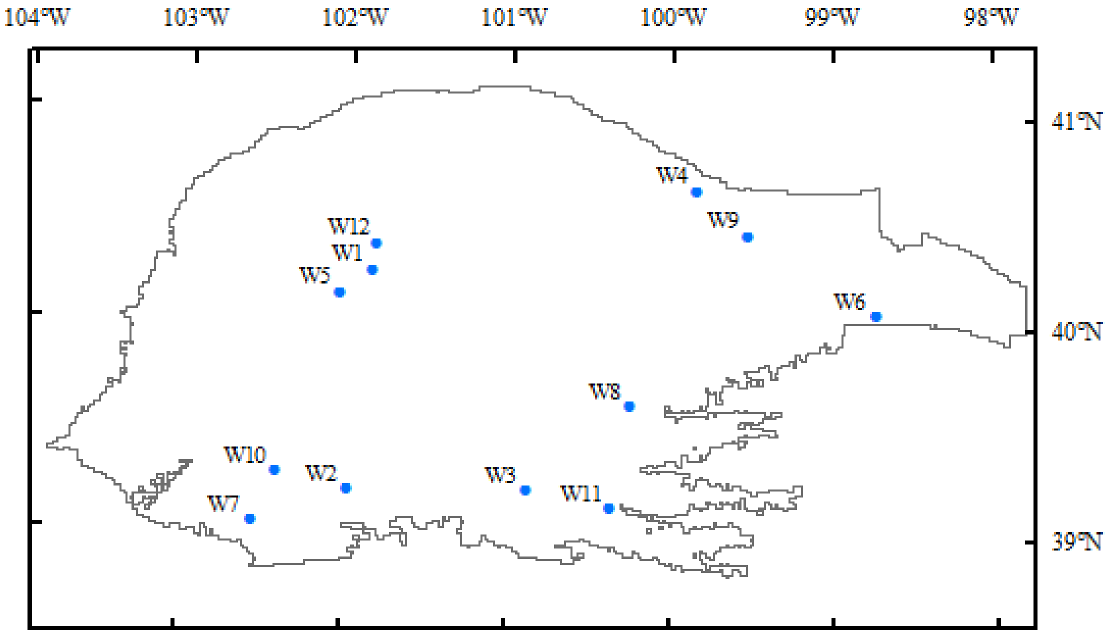


Figure 18: Locations of the selected wells in Figure 19, 20 and 21.

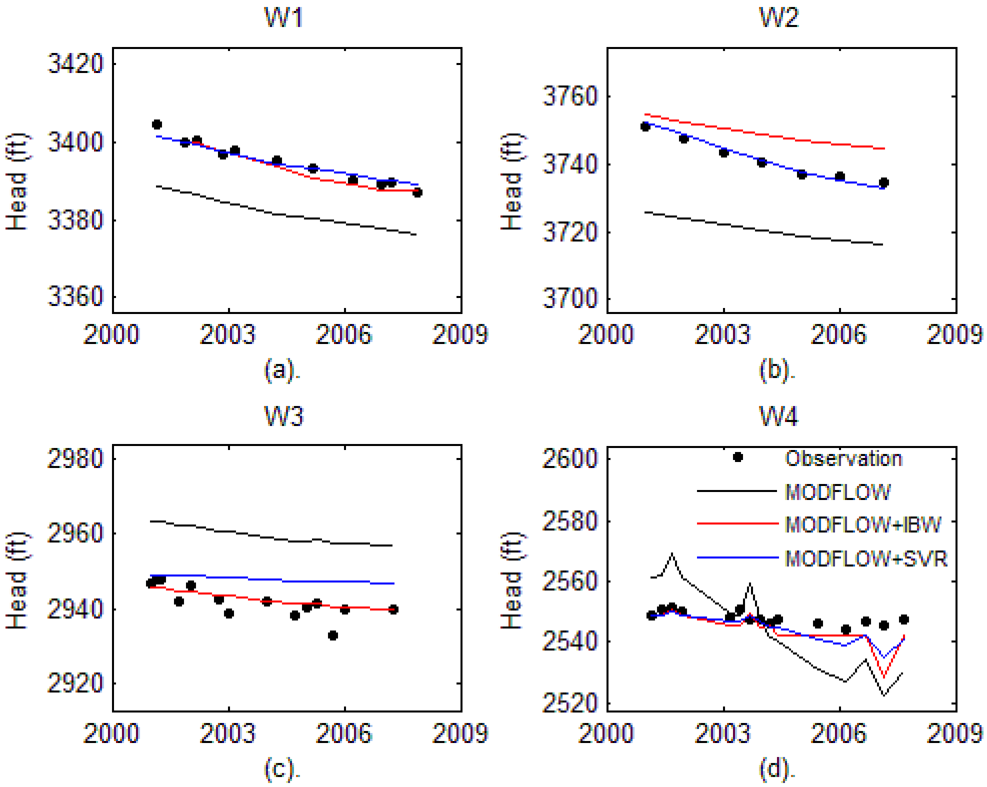


Figure 19: Measurements, MODFLOW predicted and DDMs-updated hydrographs at representative well locations of temporal prediction.

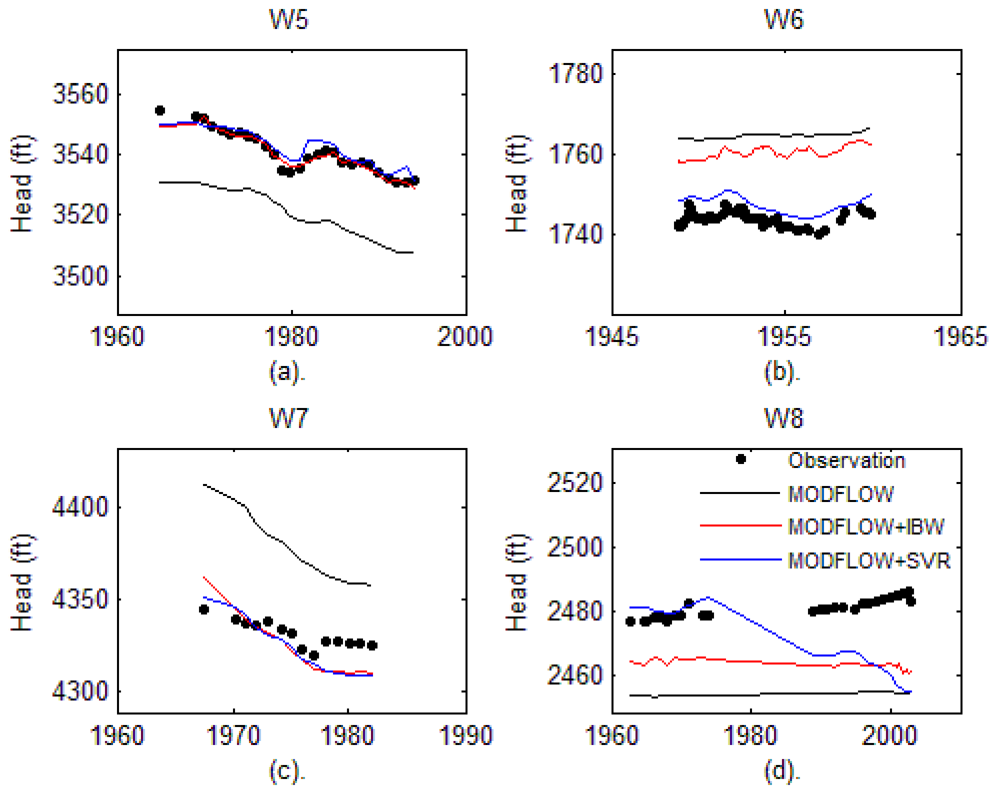


Figure 20: Measurements, MODFLOW predicted and DDMs-updated hydrographs at representative well locations of spatial prediction.

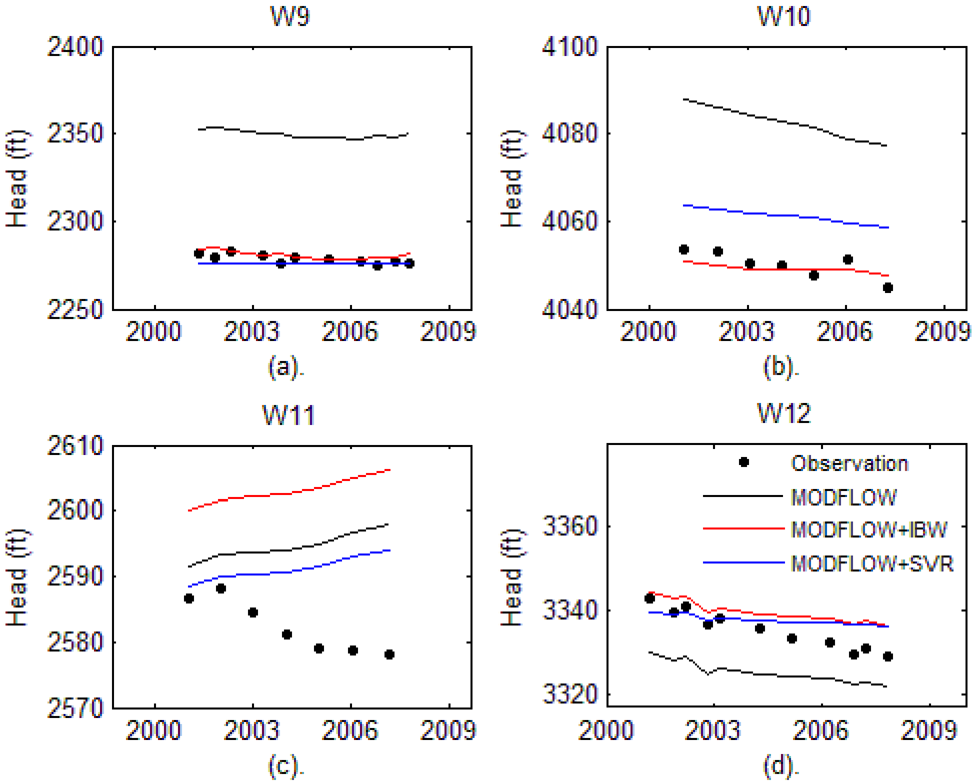


Figure 21: Measurements, MODFLOW predicted and DDMs-updated hydrographs at representative well locations of temporal+spatial prediction.

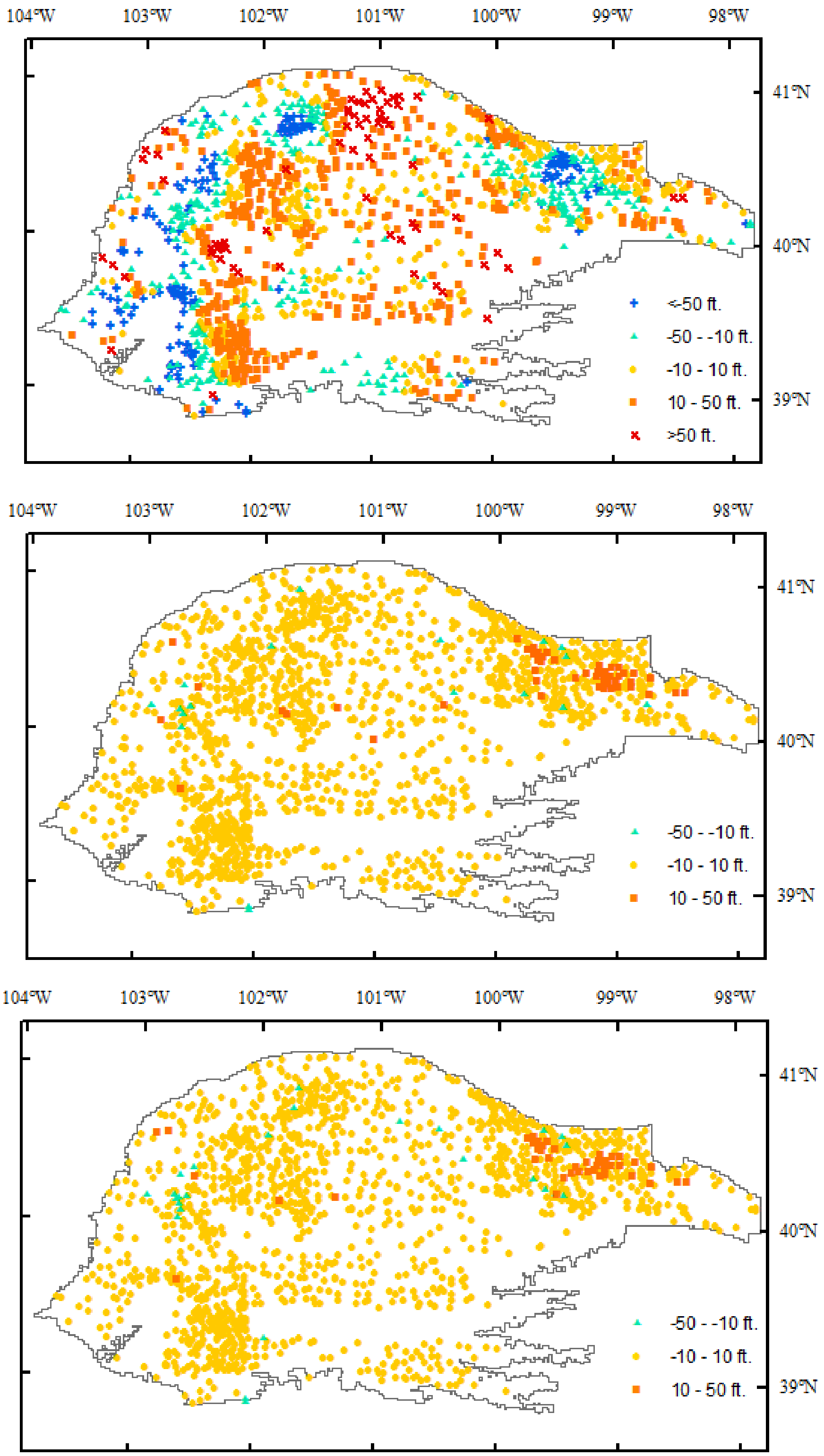
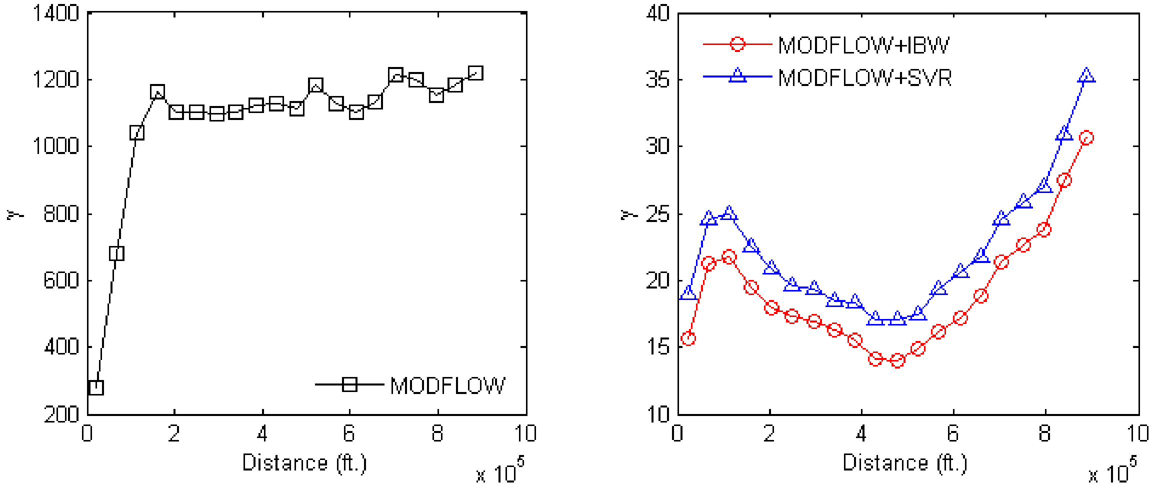


Figure 22: Mean error at each well (2001-2007) of the MODFLOW model (top), IBW corrected MODFLOW model (middle) and SVR corrected MODFLOW model (bottom).



(a)

(b)

Figure 23: The empirical semi-variograms of the mean error at each well before (a) and after (b) DDMs updating.

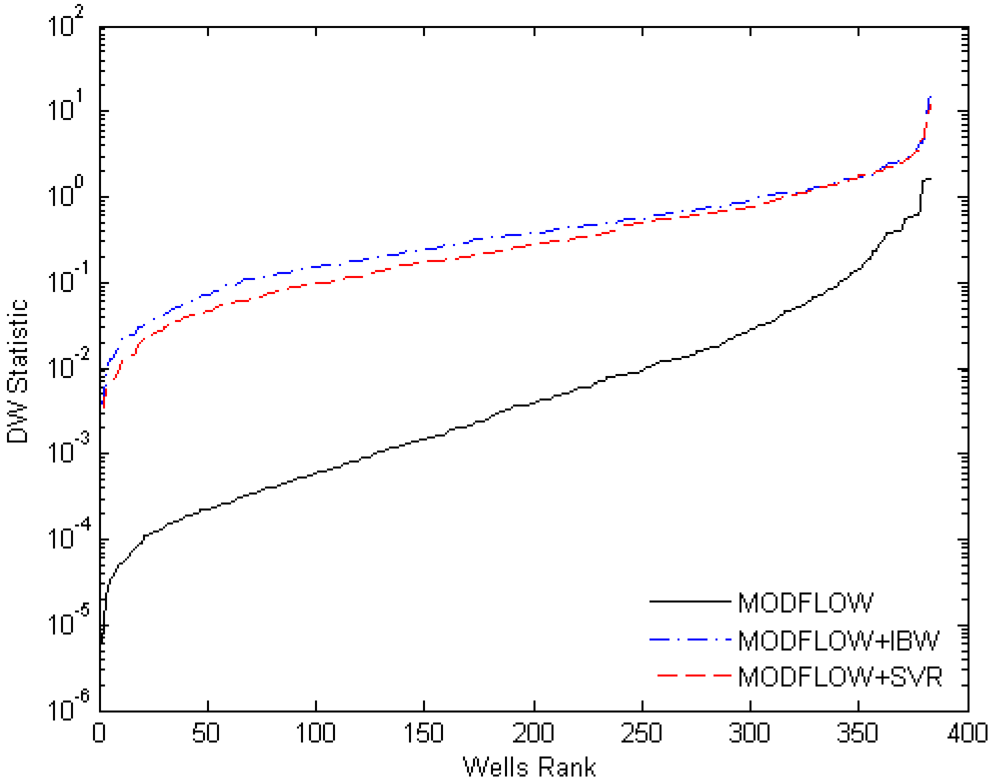


Figure 24: The DW statistic of residuals at selected wells (2001-2007) of one-year lag before and after DDMs updating.

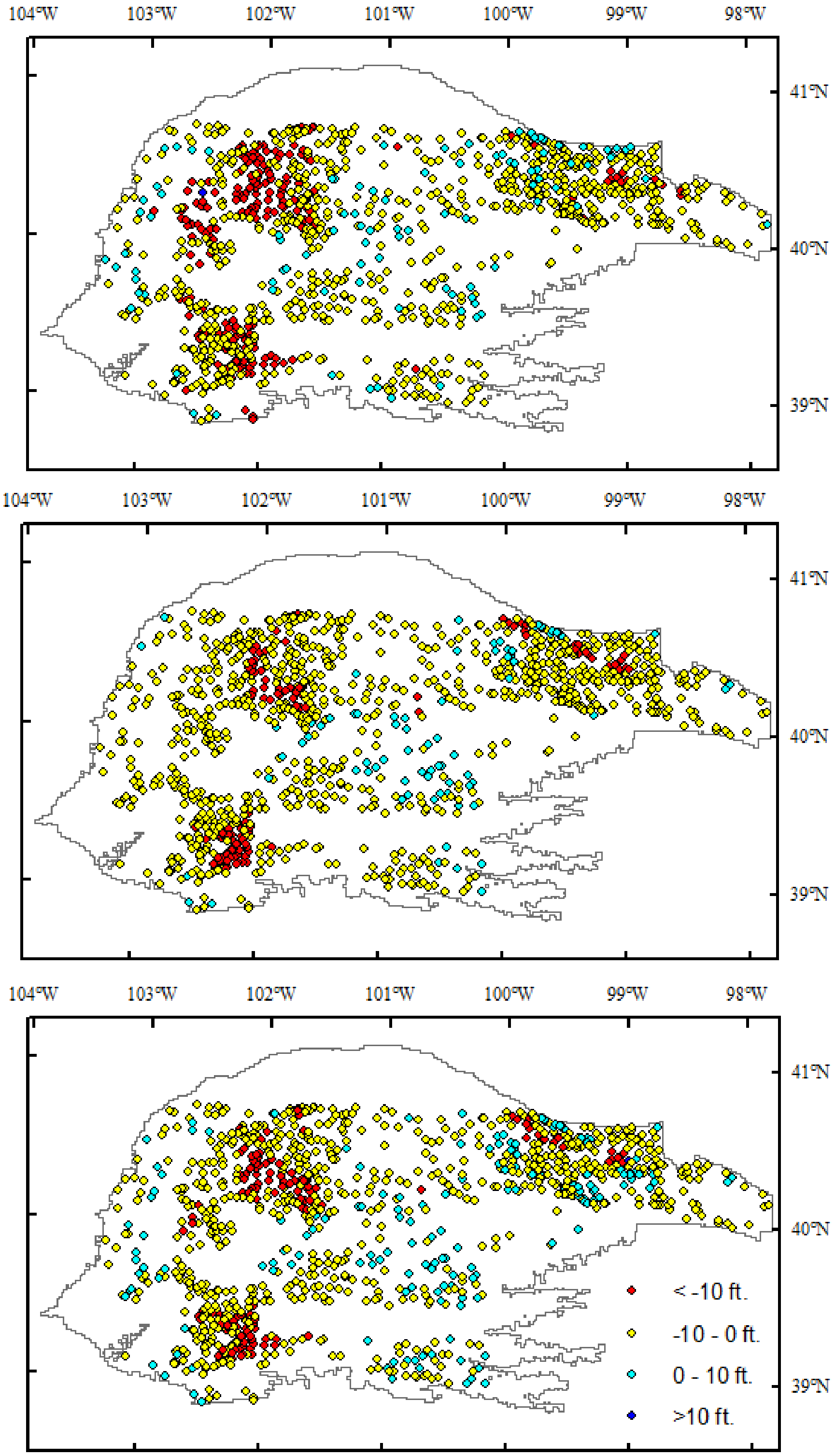
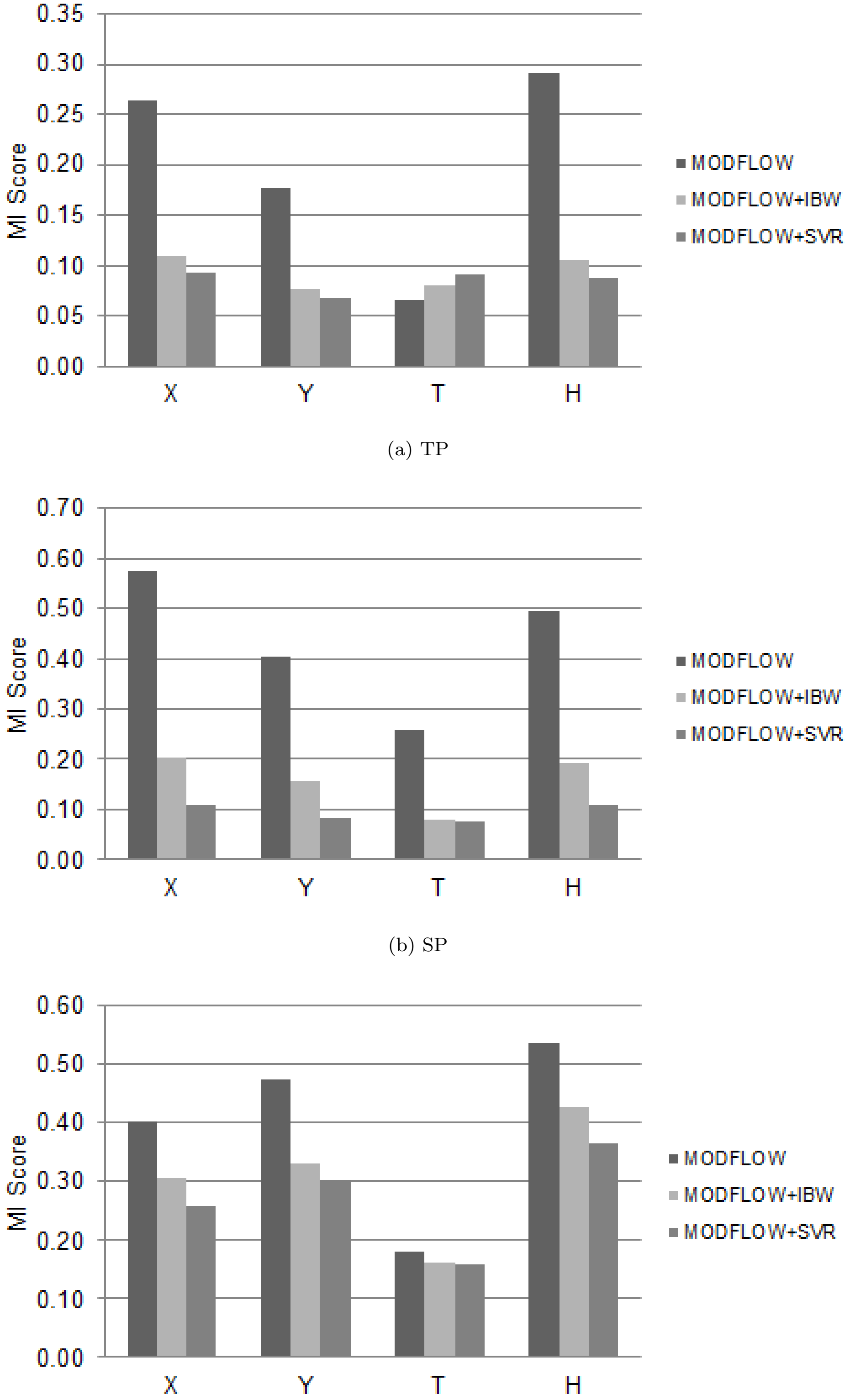


Figure 25: Change in waterlevel at monitoring wells according to the measurements (top), the MODFLOW model (middle) and SVR updated results (bottom). Negative values indicate declining groundwater table, while positive values indicate rising groundwater table.



(a) TP

(b) SP

(c) TSP

Figure 26: MI scores between the residuals and well location (x_w, y_w), time of measurement (t), and MODFLOW model computed waterlevel $(\hat{h})$.

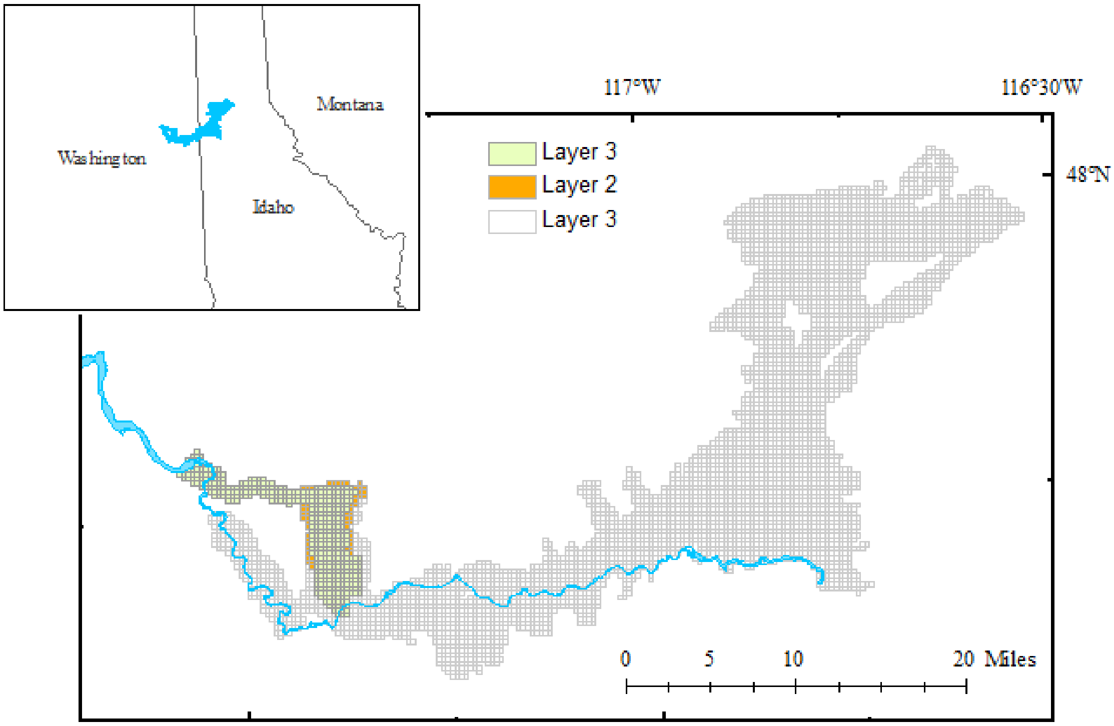


Figure 27: The Spokane Valley-Rathdrum Prairie aquifer on the border of Washington and Idaho. The Spokane River is shown in blue. The three layers are shown in different colors. The grids represent the spatial discretization of the MODFLOW model.

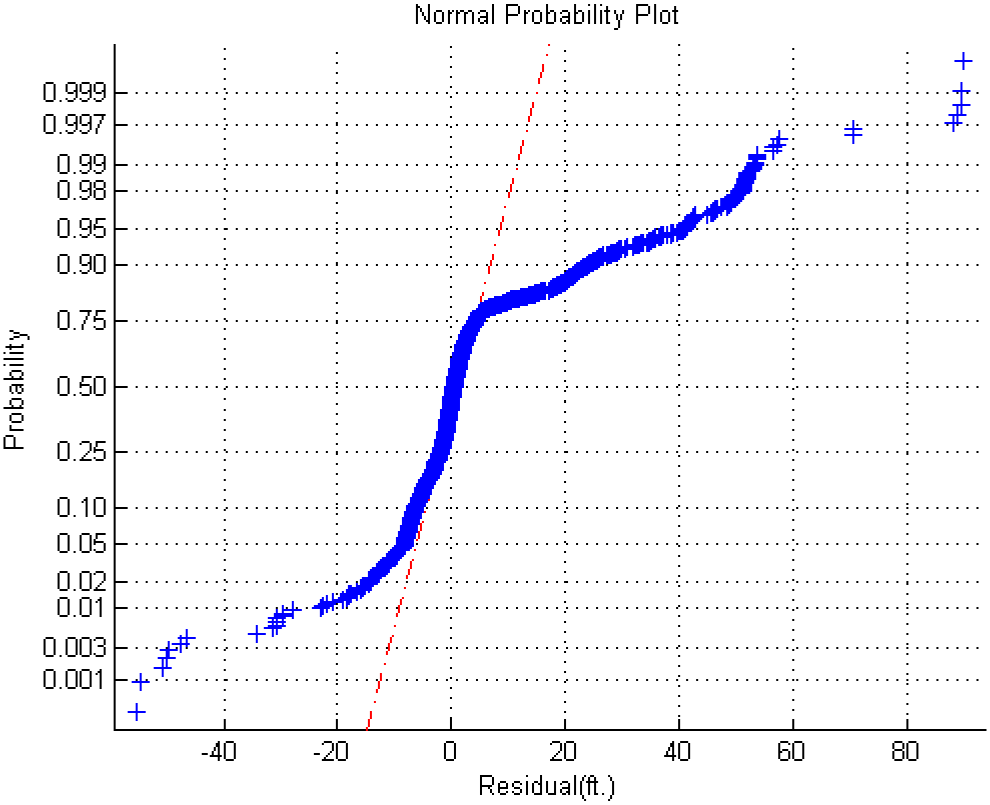


Figure 28: Normal probability plot of the residuals from Oct. 1995 to Sep. 2004.

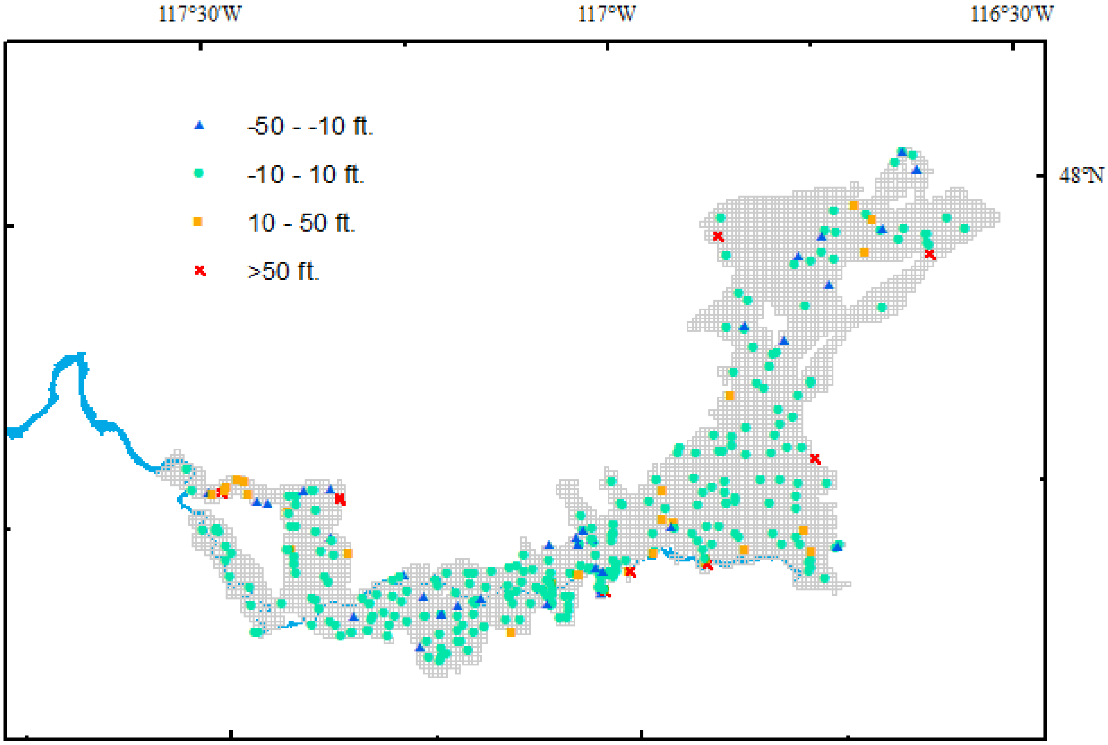
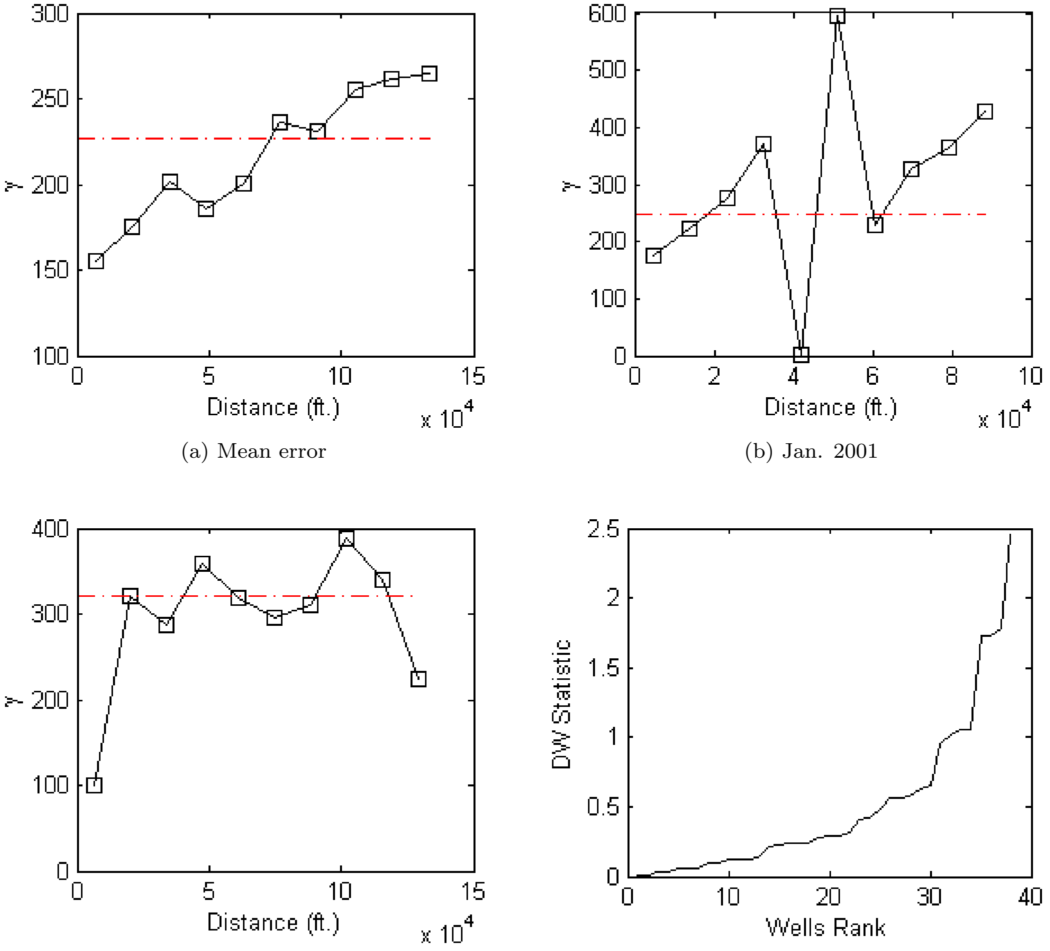


Figure 29: Mean error at each well from Oct. 1995 to Sep. 2004.



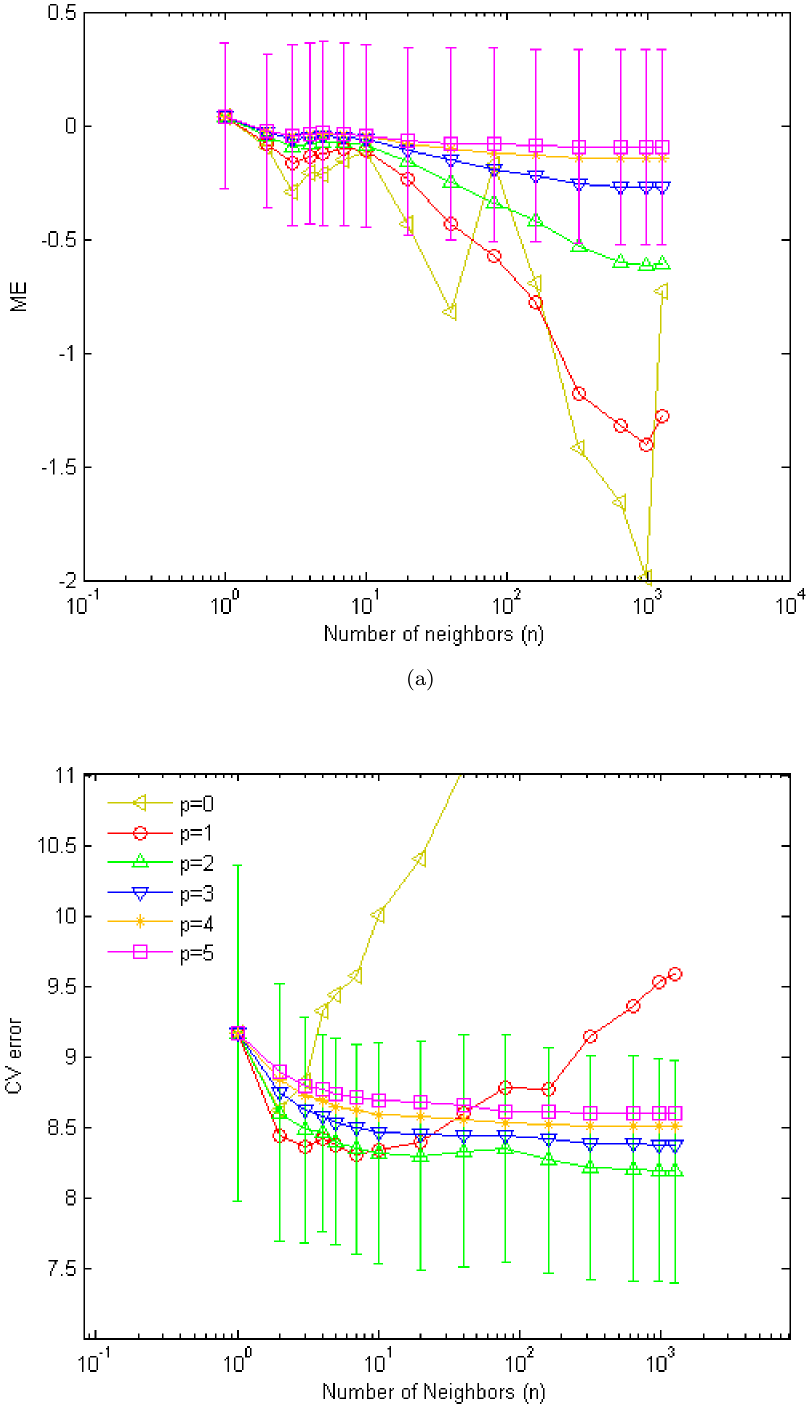
(a) Mean error

(b) Jan. 2001

(c) Jul. 2004

(d) DW Statistic

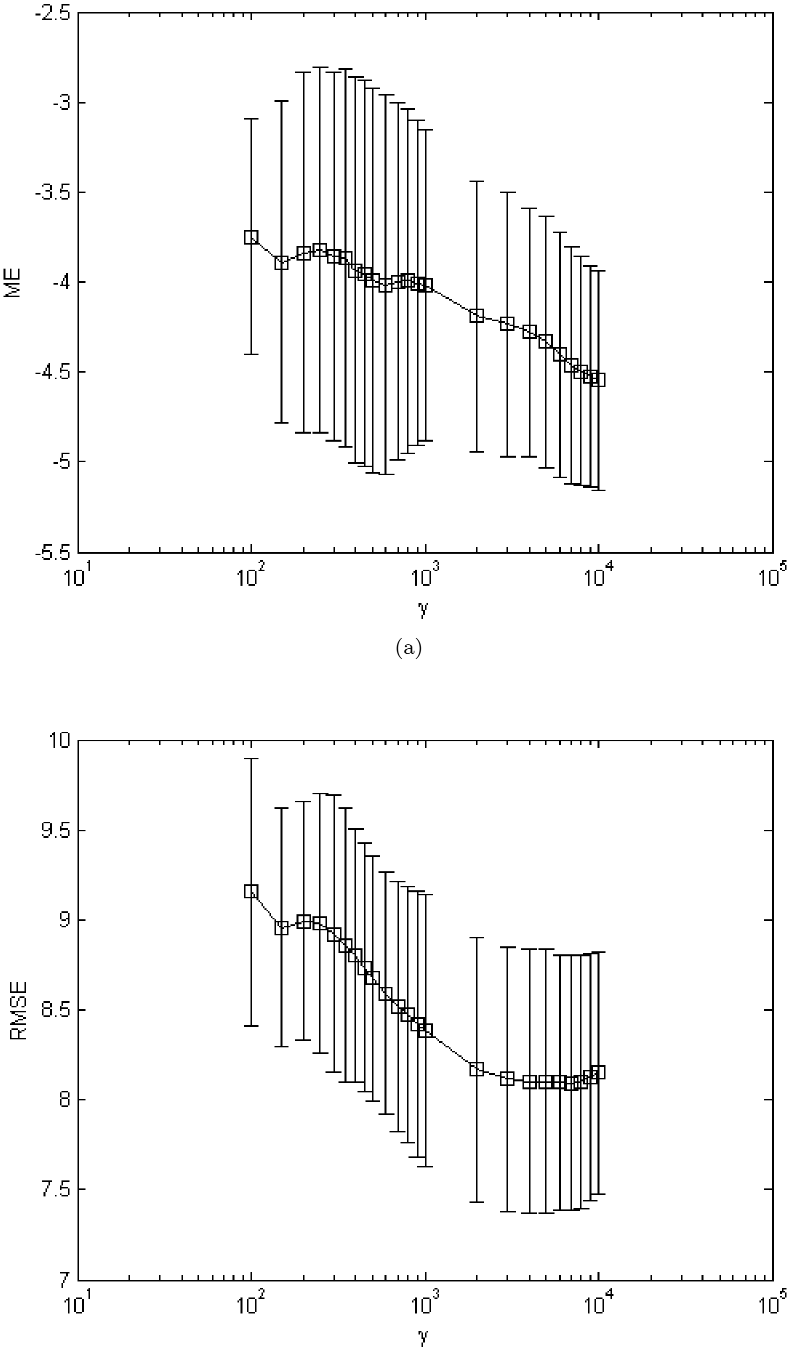
Figure 30: (a-c): The empirical semi-variograms and DW statistic of the residuals. The red dashed lines indicate the variance of the residuals. (a) is computed based on the mean error at each well, (b) is based on the data observed in Jan. 2001, (c) is based on data in Jul. 2004, (d): The DW statistics of the residuals at selected wells.



(a)

(b)

Figure 31: Cross validation results of IBW for selecting the number of neighbors and the power of weighting function. The square curves show the mean testing ME (a) or RMSE (b) of the five folds, and the error bars indicate one standard deviation of ME or RMSE of the five folds.



(a)

(b)

Figure 32: Cross validation results of SVR for selecting kernel width parameter γ. The square curve show the mean testing ME (a) or RMSE (b) of the five folds, and the error bars indicate one standard deviation of ME or RMSE of the five folds.

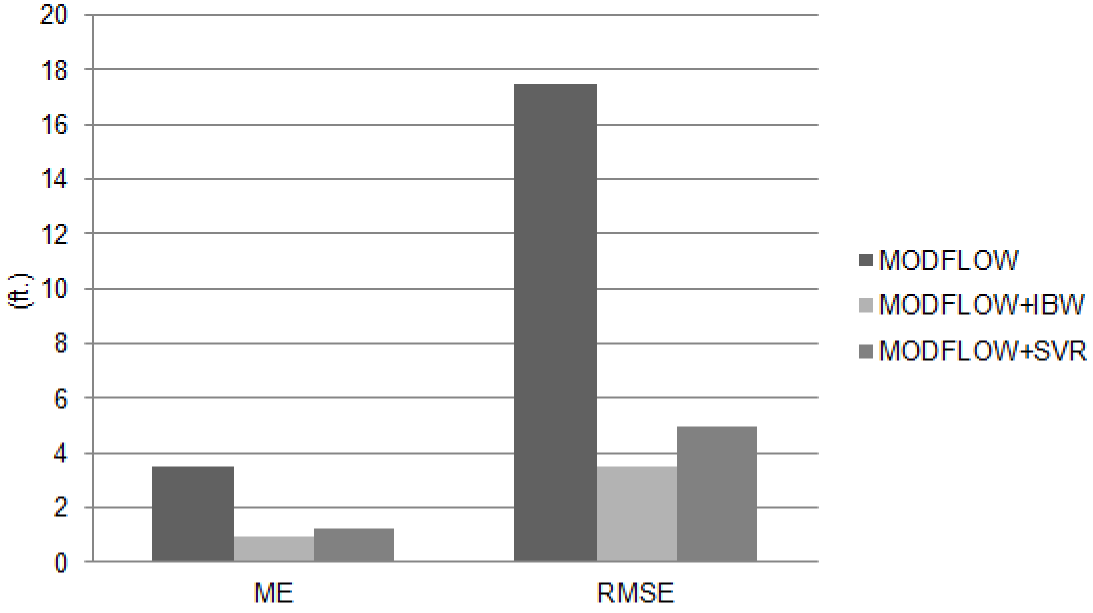
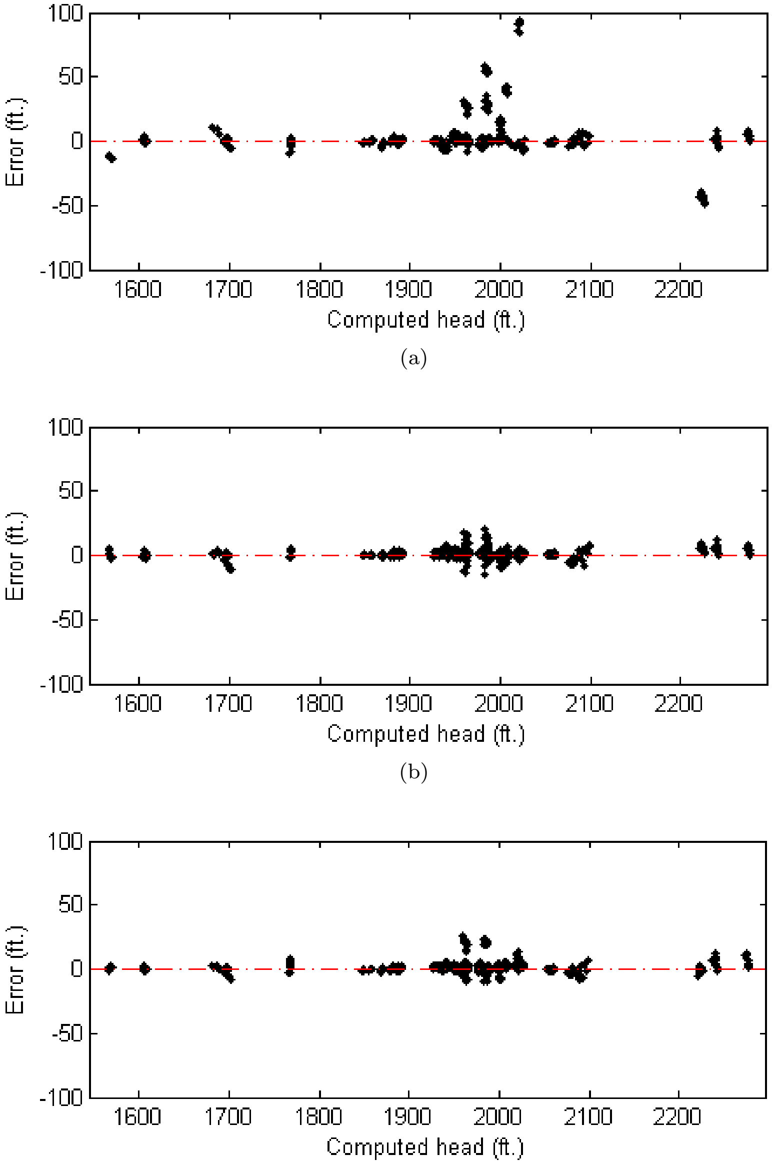


Figure 33: ME and RMSE before and after DDMs updating.



(a)

(b)

(c)

Figure 34: Plot of residuals versus the head computed by the MODFLOW model ($\hat{h}$). (a). The residuals of the MODFLOW model; (b). Residuals after corrected by IBW; (c). Residuals after corrected by SVR.

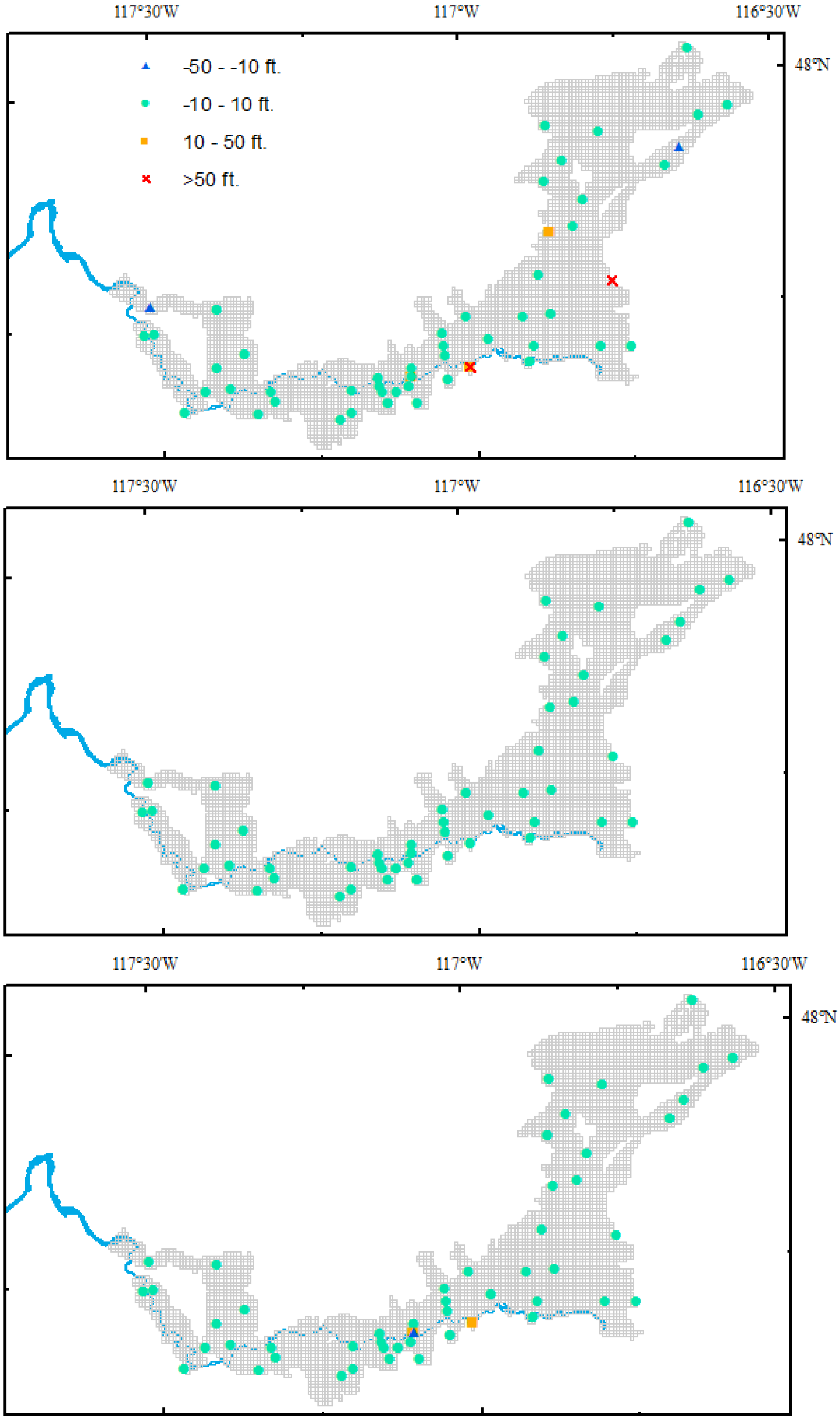


Figure 35: Mean error at each well during Oct. 2004 to Sep. 2005 of the MODFLOW model (top), IBW corrected MODFLOW model (middle) and SVR corrected MODFLOW model (bottom).

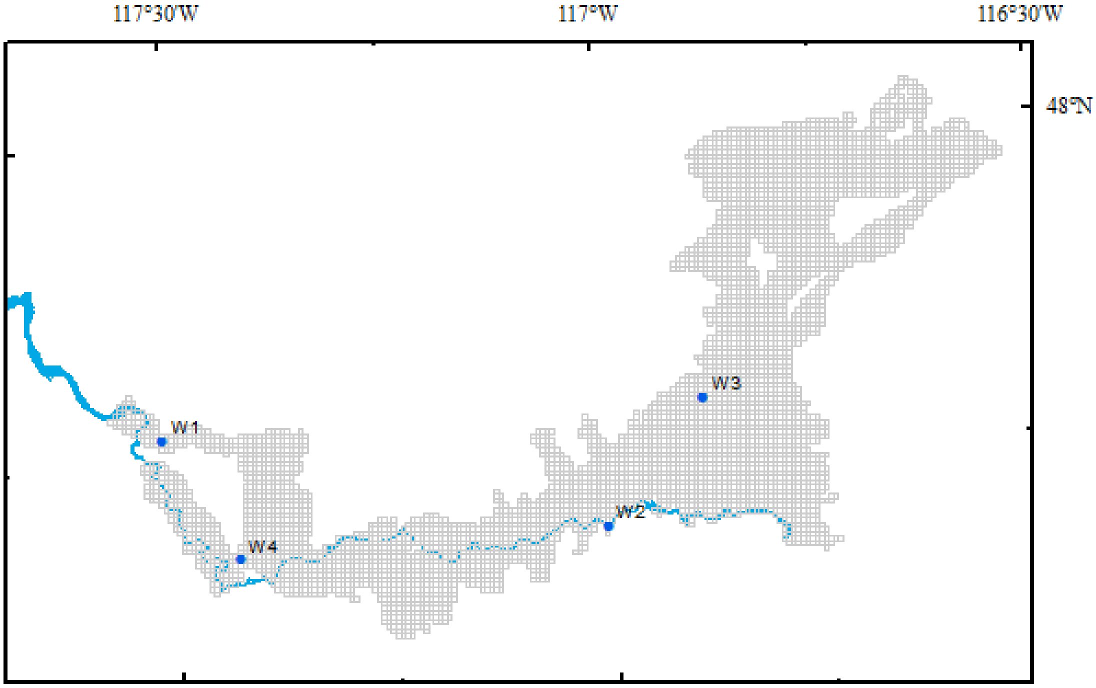


Figure 36: Locations of the selected wells in Figure 37.

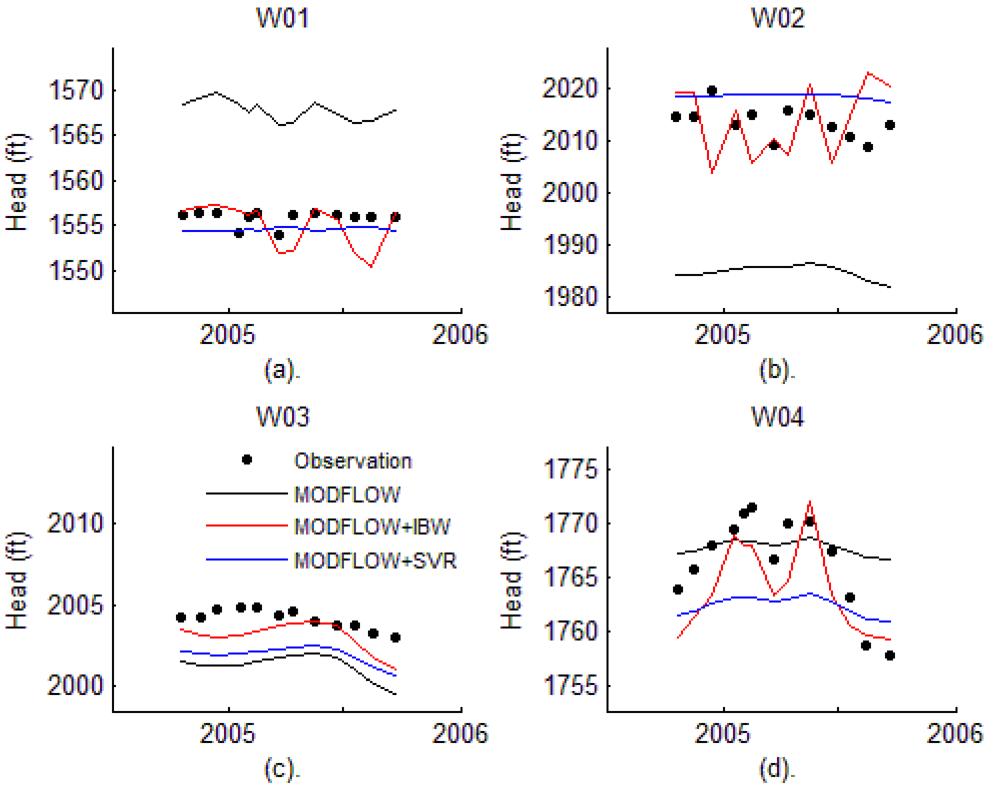
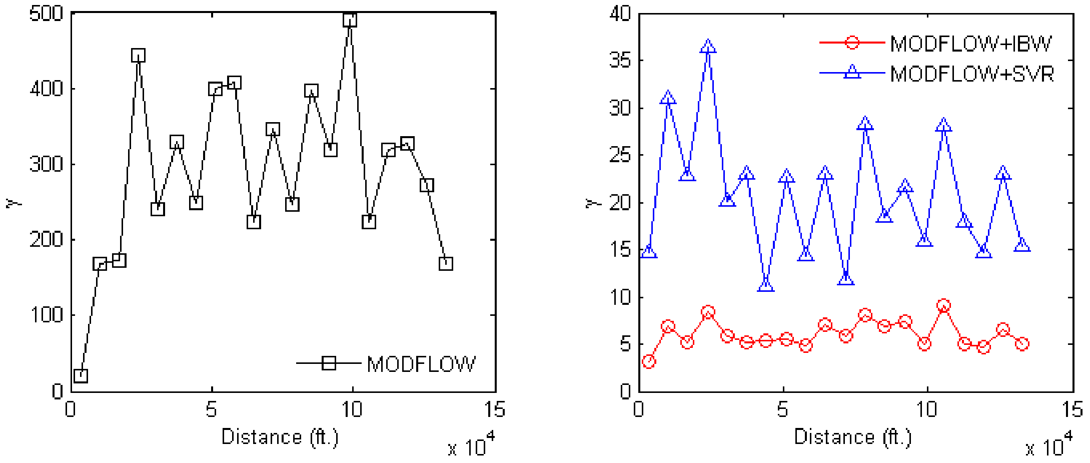


Figure 37: Observed, MODFLOW predicted and DDMs-updated hydrographs at representative wells.



(a)

(b)

Figure 38: The empirical semi-variograms of the mean error at validation wells during Oct. 2004 - Sep. 2005 before (a) and after (b) DDMs updating.

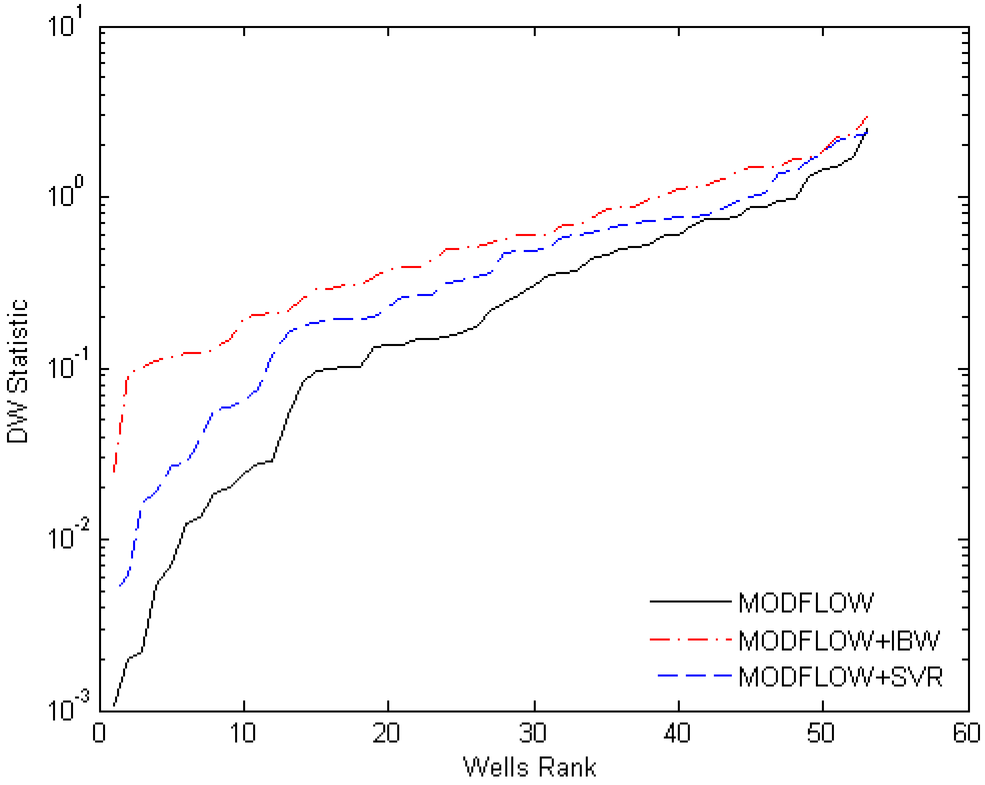


Figure 39: The DW statistic of residuals at selected wells (Oct. 2004 - Sep. 2005) of one month lag before and after DDMs updating.

Chapter 7

TABLES

Table 1: The correlation coefficients and MI scores between the residual (from 1918 to 2000) and other variables.

	x_w	y_w	t	$\hat{h}$
corr. coef.	-0.0283	0.1197	-0.0858	-0.1064
MI	0.3057	0.1323	0.0672	0.1879

Table 2: Number of monitoring wells and waterlevel measurements of the datasets used in TP, SP and TSP scenario.

	Training Data		Testing Data	
	# Wells	# Samples	# Wells	# Samples
TP	3357	310130	1476	12226
SP	2846	198370	1597	81346
TSP	1213	57652	627	6332

Table 3: The selected values of DDMs parameters for TSP for each cluster that contains residuals in the prediction period.

	IBW		*loess*		DT	ANN	SVR		
Cluster	p	n	d	δ	$\|T\|$	H	ϵ	C	γ
1	5	128	2	0.35	73	18	1.07	106.86	11
2	2	64	2	0.2	116	18	0.79	126.61	1.5
3	1	32	2	0.5	47	12	2.15	181.91	0.8

Table 4: Summary of DDMs performance and run-time (second) in the TSP scenario on Inter(R) Core(TM)2 Duo CPU E8500 3.16GHz×2.

	MODFLOW	IBW	*loess*	DT	ANN	SVR
ME	-6.94	-0.10	-0.99	-1.84	0.29	1.59
RMSE	33.95	**17.50**	19.28	19.85	21.38	**17.60**
Time (s)	>1800	67.5	199.7	18.9	(approx.) 462.7	117.1

Table 5: The error of head prediction before and after corrected by DDMs, averaged over all clusters, for TP, SP and TSP.

	ME (ft.)			RMSE (ft.)		
	TP	SP	TSP	TP	SP	TSP
MODFLOW	-2.29	-9.71	-6.94	30.23	34.46	33.95
MODFLOW+IBW	0.81	-0.72	-0.10	5.32	13.84	17.5
MODFLOW+SVR	0.03	0.22	1.59	5.16	13.05	17.6

Table 6: The selected values of DDMs parameter values.

Model	IBW		SVR		
Parameter	p	n	ϵ	C	γ
Value	3	1244	0.72	49.73	450

REFERENCES

[1] AJ Abebe and RK Price. Managing uncertainty in hydrological models using complementary models. *Hydrological sciences journal*, 48(5):679–692, 2003.

[2] C.G. Atkeson, A.W. Moore, and S. Schaal. Locally weighted learning. *Artificial intelligence review*, 11(1):11–73, 1997.

[3] K. Beven and A. Binley. The future of distributed models: model calibration and uncertainty prediction. *Hydrological processes*, 6(3):279–298, 1992.

[4] K. Beven and J. Freer. Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the glue methodology. *Journal of Hydrology*, 249(1):11–29, 2001.

[5] L. Breiman. *Classification and regression trees.* Chapman & Hall/CRC, 1984.

[6] LJ Cao, KS Chua, WK Chong, HP Lee, and QM Gu. A comparison of pca, kpca and ica for dimensionality reduction in support vector machine. *Neurocomputing*, 55(1-2):321–336, 2003.

[7] J. Carrera and S.P. Neuman. Maximum likelihood method incorporating prior information. *Water Resources Research*, 22(2):199–210, 1986.

[8] C.C. Chang and C.J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[9] K. Chau and C. Wu. *Hydrological predictions using data-driven models coupled with data preprocessing techniques.* LAP Lambert Academic Publishing.

[10] V. Cherkassky and Y. Ma. Practical selection of *svm* parameters and noise estimation for *svm* regression. *Neural Networks*, 17(1):113–126, 2004.

[11] W.S. Cleveland and S.J. Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American Statistical Association*, pages 596–610, 1988.

[12] W.S. Cleveland and E. Grosse. Computational methods for local regression. *Statistics and Computing*, 1(1):47–62, 1991.

[13] R.L. Cooley. *A theory for modeling ground-water flow in heterogeneous media.* Number 1679. US Dept. of the Interior, US Geological Survey, 2004.

[14] Y.K. Demissie. *Data-driven models to enhance physically-based groundwater model predictions*. PhD thesis, University of Illinois at Urbana-Champaign, 2008.

[15] Yonas K. Demissie, Albert J. Valocchi, Barbara S. Minsker, and Barbara A. Bailey. Integrating a calibrated groundwater flow model with error-correcting data-driven models to improve predictions. *Journal of Hydrology*, 364(3-4):257–271, 2009.

[16] J. Doherty. Ground water model calibration using pilot points and regularization. *Ground Water*, 41(2):170–177, 2003.

[17] J. Doherty, L. Brebber, and P. Whyte. Pest: Model-independent parameter estimation. *Watermark Computing, Corinda, Australia*, 122, 1994.

[18] J. Doherty and S. Christensen. Use of paired simple and complex models to reduce predictive bias and quantify uncertainty. *Water Resources Research*, 47, 2011.

[19] J. Doherty and D. Welter. A short exploration of structural noise. *Water Resources Research*, 46(5), 2010.

[20] QY Duan, V.K. Gupta, and S. Sorooshian. Shuffled complex evolution approach for effective and efficient global minimization. *Journal of optimization theory and applications*, 76(3):501–521, 1993.

[21] J. Durbin and GS Watson. Testing for serial correlation in least squares regression. iii. *Biometrika*, 58(1):1–19, 1971.

[22] P. Gaganis and L. Smith. A bayesian approach to the quantification of the effect of model error on the predictions of groundwater models. *Water Resources Research*, 37(9):2309–2322, 2001.

[23] P. Gaganis and L. Smith. A bayesian approach to the quantification of the effect of model error on the predictions of groundwater models. *Water Resources Research*, 37(9):2309–2322, 2001.

[24] H.V. Gupta, K.J. Beven, and T. Wagener. Model calibration and uncertainty estimation. *Encyclopedia of hydrological sciences*, 2005.

[25] D.R. Harp and V.V. Vesselinov. Analysis of hydrogeological structure uncertainty by estimation of hydrogeological acceptance probability of geostatistical models. *Advances in Water Resources*, 2011.

[26] Trevor. Hastie, R. Tibshirani, and JH Friedman. *The elements of statistical learning*. Springer, 2001.

[27] M.C. Hill and C.R. Tiedeman. *Effective groundwater model calibration: With analysis of data, sensitivities, predictions, and uncertainty*. Wiley-Interscience, 2007.

[28] J.A. Hoeting, D. Madigan, A.E. Raftery, and C.T. Volinsky. Bayesian model averaging: a tutorial. *Statistical science*, pages 382–401, 1999.

[29] Barber M.E. Contor B.A. Hossain A. Johnson G.S. Jones J.L. Wylie A.H. Hsieh, P.A. Ground-water flow model for the spokane valley-rathdrum prairie aquifer, spokane county, washington, and bonner and kootenai counties, idaho. Technical report, 2007.

[30] Y. Liu and H.V. Gupta. Uncertainty in hydrologic modeling: Toward an integrated data assimilation framework. *Water Resour. Res*, 43(7):1–18, 2007.

[31] H.R. Maier and G.C. Dandy. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental Modelling & Software*, 15(1):101–124, 2000.

[32] V. McKusick. Final report for the special master with certificate of adoption of rrca groundwater model. Technical report, 2003.

[33] T.M. Mitchell. Machine learning. wcb. *Mac Graw Hill*, page 368, 1997.

[34] Catherine Moore and John Doherty. The cost of uniqueness in groundwater model calibration. *Advances in Water Resources*, 29(4):605 – 623, 2006.

[35] SP Neuman and PJ Wierenga. A comprehensive strategy of hydrogeologic modeling and uncertainty analysis for nuclear facilities and sites. university of arizona. Technical report, Report NUREG/CR-6805, 2003.

[36] F. Pianosi, L. Raso, and L. Raso. Dynamic modelling of predictive uncertainty by regression on absolute errors. *Water Resources Research*, 48, 2012.

[37] J.C. Refsgaard, S. Christensen, T.O. Sonnenborg, D. Seifert, A.L. Hojberg, and L. Troldborg. Review of strategies for handling geological uncertainty in groundwater flow and transport modelling. *Advances in Water Resources*, 2011.

[38] J.C. Refsgaard, J.P. Van der Sluijs, J. Brown, and P. Van der Keur. A framework for dealing with uncertainty due to model structure error. *Advances in Water Resources*, 29(11), 2006.

[39] A.J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[40] D.P. Solomatine, M. Maskey, and D.L. Shrestha. Instance-based learning compared to other data-driven methods in hydrological forecasting. *Hydrological Processes*, 22(2):275–287, 2008.

[41] D.P. Solomatine and D.L. Shrestha. A novel method to estimate model uncertainty using machine learning techniques. *Water Resour. Res*, 45, 2009.

[42] M.J. Tonkin and J. Doherty. A hybrid regularized inversion methodology for highly parameterized environmental models. *Water Resources Research*, 41(10):W10412, 2005.

[43] A.J. Valocchi, Y.K. Demissie, T. Xu. Improving prediction of regional-scale groundwater flow models through exploratory data analysis and complementary modeling. In *Proceedings of MODFLOW and MORE 2011: Integraded Hydrologic Modeling*, 2011.

[44] V. Vapnik. *Statistical learning theory.* Wiley, New York, 1998.

[45] J.A. Vrugt, C.G.H. Diks, H.V. Gupta, W. Bouten, and J.M. Verstraten. Improved treatment of uncertainty in hydrologic modeling: Combining the strengths of global optimization and data assimilation. *Water Resour. Res*, 41(1):W01017, 2005.

[46] J.A. Vrugt, H.V. Gupta, L.A. Bastidas, W. Bouten, and S. Sorooshian. Effective and efficient algorithm for multiobjective optimization of hydrologic models. *Water Resour. Res*, 39(8):1214, 2003.

[47] J.H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, pages 236–244, 1963.

[48] D. Wettschereck and T.G. Dietterich. Locally adaptive nearest neighbor algorithms. *Advances in Neural Information Processing Systems*, pages 184–184, 1994.

[49] T. Xu, A.J. Valocchi, J. Choi, E. Amir. Improving groundwater flow model prediction using complementary data-driven models. In *Proceedings of XIX International Conference on Water Resouces CMWR 2012*, 2012.

[50] P.O. Yapo, H.V. Gupta, and S. Sorooshian. Multi-objective global optimization for hydrologic models. *Journal of Hydrology*, 204(1):83–97, 1998.