Spring 4-25-2014

A New Spatio-Temporal Data Mining Method and its Application to Reservoir System Operation

Abhinaya Mohan
University of Nebraska-Lincoln, abbhey@gmail.com

A NEW SPATIO-TEMPORAL DATA MINING METHOD AND ITS APPLICATION TO RESERVOIR SYSTEM OPERATION

by

Abhinaya Mohan

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Peter Z. Revesz

Lincoln, Nebraska

May, 2014

A NEW SPATIO-TEMPORAL DATA MINING METHOD AND ITS APPLICATION TO RESERVOIR SYSTEM OPERATION

Abhinaya Mohan, M.S.

University of Nebraska, 2014

Adviser: Peter Z. Revesz

This thesis develops a spatio-temporal data mining method for uncertain water reservoir data. The goal of the data mining method is to learn from a history human reservoir operations in order to derive an automated controller for a reservoir system. Spatio-temporal data mining is a challenging task due to the reasons: (1) spatio-temporal datasets are usually much larger than spatial data sets, (2) many common spatial techniques are unable to deal with objects that change location, size or shape, and (3) complex and often non-linear spatio-temporal relationships cannot be separated into pure spatial and pure temporal relationships.

Support Vector Machines (SVMs) have been extensively and successfully applied in feature selection for many real-time applications. In this thesis, we use SVM feature selection to reduce redundant and non-discriminative features in order to improve the computational time of SVM-based data mining. We also propose combining Principal Component Analysis (PCA) with multi-class SVMs. We show that SVMs are invariant under PCA transformations and that PCA is a desirable dimension-reduction method for SVMs. We propose also an extension of the SVM Regression approach to be able to perform spatio-temporal data mining.

As a case study, we apply our spatio-temporal data mining method to derive an automated controller for the North Platte River Reservoir system. This reservoir system has multiple reservoirs, whose spatial location and the variables in each reservoir have

been incorporated in our reservoir operations model. Further, each reservoir stage or status changes over time by the opening and closing of a dam to control the water levels. We show that by inputting the selected features from a spatio-temporal dataset by the PCA has achieved excellent results and could speed up the evaluation of data mining by SVM by an order of 10 while maintaining comparable accuracy. The North Platte River Reservoir system case study shows that the SVM Regression approach combined with PCA is an efficient tool for spatio-temporal data mining.

DEDICATION

Dedicated to my inspiring parents for being pillars of support.

ACKNOWLEDGMENTS

I would like to take this opportunity to thank my adviser Dr. Peter Revesz for going out of the way to guide me in my research and academics with his encouragement and support. I am indeed very grateful to his timely advice and motivation to pursue this area of research.

Also, I would like to thank Dr. Jitender Deogun and Dr. Ashok Samal for agreeing to be on my exam committee. I would like to thank my family and friends for always being there when I needed them most.

Contents

Chapter 1

INTRODUCTION

With the introduction of various data monitoring and data generation technologies, there has been a tremendous burst of data collected or archived. Digitization methods such as sensors, barcodes, quick response (QR) codes have made huge amounts of operational data readily available and the sizes of these warehouses of data only seem to be increasing due to the availability of powerful and cost effective database systems. The continuous growth data requires effective data processing and transformation methods by which the information and knowledge can be retrieved or obtained. This called for data mining techniques by which this embedded information can be put to good use. A large number of data mining tools and techniques are currently available for indentifying data characteristics, patterns and operational rules. The interest in data mining has inspired many new research fields that emphasize both the theoretical and the practical development of effective technological tools. However, actual implementations carried out on applications other than the mainstream ones have been very limited. This limitation can be attributed to numerous factors such as data availability, size, effectiveness and the consequences of implemention.

With regards to the applications, there are two scenarios; either large dataset (greater

than 100,000 records) or small dataset(less than 100,000 records). Large data are those which contain millions or billions of records. Existing research and statistical techniques have proved that working with larger data is relatively easy as there is large amount of information available for a thorough analysis. Moreover, calculating the determining values such as regression coefficients, mean, medians etc. make statistically more sense with regards to the performance of the actual model. With the increase in the sample size, the probability of the errors also diminishes. In other words, it is easier to identify any *faint pattern* which outlines the behavioral characteristics of the given data, in a large dataset.

One of the primary data mining methods is the *outlier detection paradigm.* Likewise, clustering and pattern matching are yet other important techniques. Both of the above, work on identifying the above mentioned faint pattern which is fairly easy when the study data set is considerably large. Calculating the global optima gives a general idea about the decisions for the pattern or clustering to be followed for the successful development of the model. On the other hand, small data is much harder to work with because determining how the data break out into clusters and identifying a definite structure are not straightforward.

In traditional applications of computer science such as *remote sensing, railways reservations and banking,* the amount and the complexity of the data gathered by current enterprises is inherently large and is constantly growing at an exponential rate. Consequently, extensive analysis of the data processes and transactions have been carried out through data mining. Over the years numerous data mining models such as artificial neural networks, genetic Algorithm, decision trees, association Rule induction, data visualization etc have been proposed have been applied for various practical applications. Careful analysis of this large amount of data is first carried out to find the data mining model that is best for the given problem. This is then followed by extensive testing to check the consistency of the

performance. Then end result is an effective data mining model that encompasses all the operational rules, working and even aid in predicting the future working of the system. For large data applications a number of data mining models and methodologies have been tried, tested and successfully deployed for everyday operations.

However, not all applications are data rich by default. In traditional applications, the data is archived solely for the purpose of data mining and analysis. Unfortunately, this assumption of availability of huge databases or warehouses containing data exactly in the desired fashion for analysis is not always true. Even if the repository has a decent amount of stored data, the final data that can be effectively used for the above mentioned operation might be considerably less. Certain applications like flood forecasting, geriatrics, internal medicine, geological studies will have data that is collected over a relatively short period of time say ten years. In such instances, even small data must still be studied to develop operational rules.

It has been seen by many researchers that system operations are not just restricted to the static behavior of data but rather is the interactive behavior of data spread over a wide time period [20],[7] and [22] . Data processing and querying techniques are increasingly used in water resource management in the recent times. The amount of data available for any hydrological management task is generally extensive because hydrology is a data intensive field. The technological advancements specifically the in data collection and the geographic information systems area has enabled users to capture various aspects of the water systems ranging from the widespread physical to the high precision chemical features. Accurate water quality assessment would help with better understanding for developing models or understanding patterns. Water resource management is expected to be completely automated in the near future, that is the decisions and operations will no longer be manually suggested by technicians and experts but instead through automated software systems with the aid of input models, required queries etc.

1.1 TEMPORAL DATA MINING

Many data mining techniques were originally targeted towards simple structured datasets such as relational databases or structured data warehouses. More importantly these algorithms were used to classify data that occur in the same time period or in other words, those datasets that are not expected to change much over time. However, technological improvements and the internet have led to more sophisticated data collection methods thus resulting in more complex data systems such as multimedia, spatial and temporal databases which unstructured or semi-structured. In the field of engineering, sensor based monitoring applications such as tele-communication control or time-stamped system monitoring. In finance, applications such as analyzing the transaction logs to analyze product sales or inventory trends prove important for business planning [29]. In the field of healthcare, most patient data are temporal sequences. These patient data encompass the complex diagnostic systems such as ECG, EEG and patient chart details that record vitals or the effectiveness of the treatments. They usually arise with either sensor-based monitoring, such as telecommunications control or log-based systems monitoring. In finance, applications on the analysis of product sales or inventory consumptions are of great importance to business planning [5]. In healthcare, temporal sequences originate by complex data acquisition systems ECGs or even with simple ones like measuring the patient's temperature or treatment's effectiveness [10].

In the last years development of medical informatics, the amount of data has increased considerably and more than ever, the need to react in real-time to any change in the patient behavior is crucial. Therefore, there is an increased demand to carry out data mining for these complex temporal data. In addition to these datasets being voluminous, the traditional machine learning and data mining algorithms do not work well on time series data due to their high dimensionality, feature correlation, and large presence of

noisy data. One major problem that arises during the mining process is treating data with temporal features or attributes.

1.1.1 Challenges in Temporal Data Mining

Temporal databases [3] usually do not accept the same types of updates and queries as traditional snapshot databases. Temporal databases contain explicit information about the start and end time of transaction. Hence temporal data only allow updates as corrections or new versions rather than modifying the entire record. If it is required to modify the whole temporal data, then it is inserted into the database as a new record. Temporal queries may make use of fairly complicated temporal selection criteria. Chapter 3 discusses both supervised and unsupervised pattern analysis methods based on the training method in detail. However, when working with temporal data; the ideal requirements for employing the supervised classification methods are hardly met. In other words, the data is hardly independent and identically distributed. Instead, there is frequent overlapping or limited amount of data that can be effectively used for training purposes etc. Some of the common problems when working with temporal data are as follows:

- **Heterogenous Data Sources :** Temporal data framework requires analyzing data instances there are recorded n units back in time. Based on the period of this unit (monthly, yearly, or every decade) the data collection method might have undergone significant changes in methodology and format during these periods of time. The sources for these databases might also have changed completely. Hence the data fusion method has to put together information from these multiple sources into a single composite picture of the relevant processes.
- **Small Datasets :** Most classifiers are based on properties such as Euclidian distance,

Gaussian distribution, and regression equations, which are valid only when applied on large training data. Since for temporal data, the input data sets are combined from multiple sources and might be included over large periods, there is a possibility that much of the data collected from these sources are not needed for the given purpose of the study. For example, the rainfall data which is used for a reservoir operation optimization might have originally come from a groundwater study data. Hence the number of groundwater data attributes that can be used for data mining the reservoir operation function may be limited.

It is necessary to choose carefully an appropriate data mining algorithm when working with N-dimensional transaction databases. Supervised learning methods that allow for event analysis in a multi-dimensional space are usually good choices.

1.2 SPATIAL AND SPATIO-TEMPORAL DATA MINING

1.2.1 Spatial Data Mining

As described earlier, spatial and time-varying information is inherently present in most geographical or hydrological applications. Therefore, it is useful to develop techniques that efficiently summarize and discover trends in spatial and spatio-temporal data and help in decision making. Temporal data mining carried out in Phase 1 focused on identifying the operational rules of a single reservoir under consideration. However, in real-time system, reservoirs are standalone systems. In location dependent data applications such as those use that geographical or hydrological data, including the time-varying data analysis with the spatial data analysis would help in developing real-time solutions.

1.2.2 Spatio-Temporal Data Mining

Spatial data records information regarding the location, shape and its effect on features (e.g. geographical features). When such a data is time variant, it is called spatio-temporal data. Spatio-temporal data mining is an upcoming research topic which focuses on studying and implementing novel computational techniques for large scale spatio-temporal data. Progress in hardware technologies such as portable display devices, wireless devices has enabled an increase in availability of location- based services. In addition, GPS data is becoming increasingly available and accurate. These developments pave way to a range of new spatio-temporal applications such as distributed systems, location based advertising, disease monitoring, real estate process etc. Having explored temporal data mining, the next step is to extend data mining techniques to spatio-temporal data. This is because in most cases, the spatial and temporal information is implicitly present in the databases; it might be either metric-based (e.g. size) or non metric based (e.g. terrain, storm path) or both. It is therefore necessary to acknowledge it before carrying out data mining processes that target developing real time models. The spatial and temporal dependencies are inherently present in any spatio-temporal databases. Spatio-temporal data mining can thus be defined as identifying the interesting and non-trivial knowledge from large amounts of spatio-temporal data. Spatio-temporal data mining applications range from transportation, remote sensing, satellite telemetry, monitoring environmental resources and geographic information systems (GISs) (Roddick [53]). Mining information from such datasets is an important problem. In most cases, these databases changes with time, it is therefore important to capture the evolutionary behavior of the spatial data points with respect to time as these give insights for predicting future occurrences of events such as hurricanes, drought and groundwater contamination.

1.2.3 Challenges in Spatio-Temporal Data Mining

In addition to the challenges mentioned for working with temporal data, spatio-temporal data presents the following challenges:

- **Size :** Spatio-temporal objects are typically large and do not have well-defined shape or boundaries thus making them more complex to describe and record (Miller and Han [38]). Recent technological advances in computational sciences have resulted in huge amounts of data. Therefore, the data mining approaches must scale well to large data sets.
- **Geometric Properties :** The geometric properties associated such as shape and size of the objects are important when modeling real time operational systems like designing a water shed which requires information about the size of the area, the pressure and temperature conditions etc.
- **Data Aggregation :**
  - The geographical data collected might have different topological and geometric frameworks. In some cases the data attributes about the geographical location might be unique. Generalization of such datasets would be counter-productive. Research techniques are also more focused on modeling spatial relationships among features in an attempt to better understand the underlying processes.
  - **Class Aggregation :** When working with large-scale framework of databases the samples collected might not contain the designated set of class labels. For example, in a remote sensing application, when the geographical location of the type of the region of the specified geographical location is being recorded, class labels are often used in thematic maps. This labeling in some cases might be erroneous. There might also be a difference in the class aggregation depending on the data source.

- **Skewed Data :** In some cases the spatio-temporal data may skew more towards one objective. For instance, the high-resolution satellite imagery usually has abundant spatial information but might not record the time aspect like image change according to the time of the day. In another case, sensors stationed for monitoring events give precise and detailed information with respect to time of events but offer little information about the spatial relationship of the distributed sensors.
- **High Correlation :**
  - Temporal data contain an intrinsic dependency between instances, and thus display a high correlation between the data values. If the reservoir operation is considered, the amount of rainfall that is received in a particular location in the current month affects the water levels in the nearby reservoir which consequently affects the water available for the subsequent time periods.
  - Identifying the temporal relationships from the Spatio-temporal data requires implicit modeling of the spatio-temporal constraints and auto correlation so that an inference can be made about important events such as the durability of the reservoir over time and its ability to withstand shocks.
- **Spatial Auto Correlation :**This is the situation in which the value of a variable at a location is related to the values of the same variable at the locations nearby. Geographical data has spatial dependency i.e. attributes of nearby location influence each other and tend to possess a great degree of similarity. For example, the type of land cover tends to be more similar at a distance of one meter than a few miles. Ignoring spatial autocorrelation might result in errors that increase multifold in a high dimensional data.

1.3 PROBLEM STATEMENT

Many researchers studied multi reservoir systems, but their models and rules were based on the assumption that the reservoir network or the state variables is a linear network operating on linear constraints and costs. In this thesis, we propose a model for deriving the operational rules for composite reservoir operations using data mining that addresses the aforementioned challenges.

We propose a new spatio-temporal data mining method RF-SVR (Reduced Feature-Support Vector Regression) which combines Support Vector Regression Analysis with PCA (Principal Component Analysis) to effectively model the non-linear spatial relationship between the dependent variable (outcome) and a set of predictors in a spatial framework. We also apply the new spatio-temporal data mining method to the North Platte River reservoir system, which is a challenging application in our region. Water reservoir operators need to balance releasing water for current uses and storing water for future uses. They also need to avoid high water levels which might cause an overflow and flooding of the surrounding lands. Skillful reservoir operators have an accumulated knowledge that is not easy to analyze using existing models. Our spatio-temporal data mining was able to learn from past reservoir operational data and derive an automated controller that can be cheap and as effective as human controllers.

The main research question for this work is how efficient data mining models can be designed and developed for the prediction of reservoir releases from both single reservoir and multi-reservoir context. The solution to this can be found out by answering the following sub questions.

1. How to carry out data mining on high-dimensional multi-variate Spatio-Temporal Data framework?
2. How can feature reduction be carried out to further enhance the model developed?

3. What types of data mining algorithms can be used?
4. How should the performance of each of the data models be evaluated?

1.4 OBJECTIVES

With the above background, in order to address the problem statement, the objectives of the present research were set as follows:

1. To develop data models for multi-variate Spatio-Temporal framework in a data short environment.
2. To enhance the model by carrying out feature reduction with data mining techniques.
3. To carry out a sensitivity analysis of various kernel parameters on the performance of both Spatial and Spatio-Temporal data mining techniques.
4. To demonstrate the applicability of data mining with case studies of Reservoir System Operation.
5. To make an inter-comparison of the performance of Temporal, Spatial and Spatio-Temporal data mining.

1.5 OVERVIEW OF CONTRIBUTIONS

We have developed an efficient data mining method for a high dimensional spatio-temporal data which can be summarized as:

- We have developed a Support Vector Regression (SVR) based Spatio-temporal data model that captures Spatial and Temporal correlation of data variables.
- Rather than developing descriptive rules about the underlying operations, we have developed a predictive data mining model that employs regression to predict

numerical values thus making it very relevant and applicable for modeling dynamic real time events.

- The temporal dependence of the data mining model is found to be upto the maximum lag period of 3 for the spatio-temporal data model. Thus the temporal data mining with the data upto third lag would be more sufficient to take care of the dimension of factor space which is already efficient enough for many interesting applications, and this is true for the present case namely reservoir operation data mining. It is also possible to use the temporal data mining model as first level pre-processing model in the case of spatio-temporal models which takes care of both time scale as well as spatial scale.
- Since we are dealing with high dimensional (multivariate) data in a data-short environment (less number of instances), in order to further enhance the model, we developed a Reduced Feature Support Vector Regression (RF-SVR) that employed Principal Component Analysis (PCA) to carry out feature reduction.
    1. The original high dimensional spatio-temporal dataset whose input vector consisted of 42 feature resulted in a feature-reduced dataset of 5 feature set input vectors which is a factor of 8 reduction.
    2. The RF-SVR model also reported a considerable improvement in training time; the RF-SVR (11.3 secs) was 10 times more faster than the SVR model (132.9 secs).
- The RF-SVR model performed much better that the spatial data model in general.
- In comparison with the SVR spatio-temporal model, RF-SVR model reports slightly lower performance but still displays an above average accuracy. Thus, the size (attributes of the data sets) does not greatly inuence the performance of the data mining algorithms as high accuracy of prediction was observed in the data-short environment with high and low dimensional data.

The model was developed in a data short environment. But when extending the data model to a larger datasets the n fold decrease in attribute and training time would be very efficient, Thus establishing the scability of our data mining model.

The enhancements in efficiency reported with lower dimension data and lower training time outweigh the slight decrease in performance of the data model. Therefore the developed model can be used as a benchmark for spatio-temporal data models.

Part of this work; research on temporal data mining has been presented in the ACM-SIGSPATIAL '12 conference and subsequently published in the proceedings.[42]. The work on spatio-temporal data mining model has been submitted to the International Database Engineering and Applications Symposium (IDEAS) '14, Porto, Portugal [41].

1.6 OUTLINE OF THE THESIS

The outline of the thesis is as follows:

Chapter 1 gives an introduction to this research work giving the motivation, objectives, scope and the main contributions of this work. Chapter 2 presents a detailed literature survey on the various data mining models and algorithms popular in application areas with special reference to Spatial and Spatio-Temporal data. A brief overview of the earlier works on reservoir automation is also included. Chapter 3 discusses the background on data models and mining techniques. Chapter 4 describes the proposed spatio-temporal model. Chapter 5 describes the various steps carried out in developing the data model and its implementation in the North Platte River reservoir system case study. Chapter 6 presents the experimental results and the performance analysis for each of the temporal, spatial and spatio-temporal data models. Chapter 7 puts forth some discussions and conclusions.

Chapter 2

RELATED WORKS

Data mining first began as a sequence similarity search in sequence databases by Agrawal et al. [1]. Fayyad et al. [16] presented an overview of the Knowledge Discovery in Databases (KDD) and the relationship between data mining and KDD.

2.1 TYPES OF DATA MINING

The databases form the most essential ingredient of any data mining method. Different data mining algorithms work differently on the data and thus the patterns identified in a given dataset is based on the data mining technique employed. Fayyad [15] listed two types of data mining tasks: descriptive data mining tasks that describe the intrinsic characteristics of the existing data, and predictive data mining tasks that attempt to do predictions based on inference on available data. The end goal of data mining is to build a data model for the given dataset. The generative aspect of data mining consists of the building of a model from the data. The various data mining techniques can be broadly categorizes as follows:

- Association

- Classification
- Clustering
- Regression
- Sequence discovery

[2],[10], [45],[47] and [62] are some of the data mining works using the above techniques.

2.2 TEMPORAL DATA MINING

Temporal data mining is often carried out with one of the following intentions; predictions, classification, clustering, search and pattern identification (Laxmanan [30]). Early predictive models assumed a linear combination of the sample values (Box et al [7], Chatfield [9]). But later, neural networks (Mozer[43]) and AI modeling were employed to develop non linear temporal modeling. Clustering in time series data provides an opportunity to understand it at a higher level of abstraction by studying the characteristics of the grouped data. Temporal data clustering has numerous applications ranging from understanding protein structure to learning and characterizing financial transactions. The interest for pattern identification in large time series data is comparatively recent and was originated from data mining itself. Pal et al. [47] addressed pattern recognition from data mining perspective. The sequential pattern analysis was then used to identify the features after which they are input into classifiers (like Nave Bayes Classifier) for data processing [6]. The spatial dimension describes whether the objects considered are associated to a fixed location (e.g., the information collected by sensors fixed to the ground) or they can move, i.e., their location is dynamic and can change in time. Revesz and Triplet [52] considered the classification of temporal changes in flu patterns.

2.3 SPATIAL AND SPATIO-TEMPORAL DATA MINING

While there been a tremendous amount of research pursued in mining the relational and transactional database systems, much work is to be carried out in other non-traditional applicative systems like in spatial databases. Advancements in spatial databases such as spatial data structures (Samet and Aref[55]), spatial reasoning and image database system make it imperative to carry out research in spatial data mining. Güting [20] carried out a detailed analysis of the major technical concepts of spatial database systems. Koperski [28] presented an overview of the various methodologies for knowledge discovery in spatial database systems. Spatial data mining consists of techniques like generalization, clustering and mining association rules (Agrawal and Srikant [2]) wherein the ultimate goal is to integrate and devise methods for the analysis of large and complex databases. Association rule mining was carried out by extending the research in traditional large transactional database systems (Han [22]) taking into consideration the spatial attributes (Mennis and Liu [36]). Data mining carried out on such datasets generates association rules that are expressed as a set of characteristic and discriminant rules. An association rule takes the form A$\rightarrow$ B where A is the antecedent and B is the consequent; both of which are predicates. However the case with association rule mining is that it typically generates a large number of rules and usually expert intervention is required to filter out the essential findings. Various approaches such as nearest neighbor classification (Koperski [28]), decision tree using the C4.5 algorithm visualization (Quinlan [49])have been employed. It can bring to light interesting patterns and knowledge like dependencies on spatial and non-spatial attributes thus aiding in developing models that encompass all of the real time contributing factors especially for applications such as geo-marketing, disaster assessment, preventive planning and explorative geographic information systems

applications.

There is a rich bibliography when it comes to spatial data mining. But spatio-temporal data mining has not received little attention. Tsoukatos and Gunopulos [65] carried out Sequential Pattern mining for spatio-temporal data. It employed a depth-first-search approach for identifying long sequential patterns throughout the database. Mamoulis et al. [35] worked on discovering periodic events in the given spatio-temporal dimension; the pattern are defined as spatial regions rather than a sequence. Heuristics are then used to detect the frequently observed regions. Association rule mining techniques were also employed to study the relationship between spatial and temporal aspects in an antecedent-consequent form of the rules. Revesz and Triplet [51] provided a classification method for spatio-temporal weather patterns based on constraint databases that were introduced in [24]. Shekhar et al. [57] put forth that the rules should be targeted at identifying attributes embedded in the spatio-temporal framework of the data rather than the database itself. Preprocessing is a major contributor for extracting the required information from the spatial data systems. Indexes were proposed by Tao et al. [64] for better selection from the given set of attributes for data mining. Mennis and Liu [36] employed association rule mining by developing concept hierarchies for generating spatio-temporal association rules. Lazarevic et al. [31] used linear regression techniques to study complex geographical associations in spatio-temporal data. However, multivariate spatio-temporal data seldom exhibit linear dependency between variables. Non-linear combination data model using wavelet transformation was carried out by Dong [13] to develop efficient fits for non-linear model.

2.4 DATA MINING FOR RESERVOIR OPERATION

Applying data mining methods to hydrological databases has been in practice for quite some time now. Earlier works focused on identifying the operational rules of a single reservoir under consideration; however, reservoirs are rarely standalone. Solamatine and Torres. [60] first developed a hydrodynamic model for reservoir and ground water control using ANN based optimization. Sudha et al. [63] found that decision tree (DT) performs better than linear programming. Further, Wei and Hsu [67] compared the efficiency of DT algorithm with the optimal rules derived from regression based rules for flood control reservoir, and inferred that DT based reservoir rule curves performed better than the regression based rules. Stochastic models by Loucks et al. [34] were then employed so that the state variables can be used to estimate the probability of the outcomes. A study was carried out by Tejada et al.[19] that involved three stochastic dynamic models of a multi-reservoir system in California. The findings confirmed the need to include the state variables; the set of relevant hydrological information for best results. Owing to the increased complexity of the datasets, data mining techniques were then employed. Spate [61] presented the main concepts, when carrying out hydro-data mining. Coulibaly [12] presented a dynamic ANN model to optimize reservoir inflow values. Glezakos et al. [18] implemented a neural networks model on time series dataset created using evolutionary clustering. The heuristics learned during meta-learning are included in the training phase of the model to develop models for watershed management. Most of the works for hydrological optimization employ conventional or modified forms of ANN; however, ANN is not very effective at multivariate operation forecasting.

Chapter 3

DATA MINING

3.1 INTRODUCTION AND KDD IN DATABASES

Data mining, also called knowledge discovery in database, is defined as a nontrivial extraction of implicit, previously unknown, and potentially useful information from databases. Data mining has caused a tremendous impact in information management in the recent years due to a vast variety of data and data collection methods. While database management systems (DBMS) analyze observations in a micro view, the data mining systems examine the observations (records) in a macro view. In other words, information stored in a database is different from the knowledge it contains and this knowledge can be deduced or discovered by data mining. The tools are employed to discover previously unknown, valid patterns and relationships in large data sets. These tools can include statistical models, mathematical algorithms, and machine learning methods such as neural networks or decision trees, Hence data mining consists of more than collecting and managing data, it also includes analysis and prediction.

Historical Perspective: Statistical techniques were first employed to carry out computerized data analysis, but statistics does not characterize this dependency at a conceptual

level, and give a substantial explanation or description of this relationship. The details generated are usually not enough for detailed analysis and future predictions. There was thus a gap between the data explaining and data understanding. This prompted the development of techniques that were capable of discovering information concealed in data that offered a more intelligent explanation of the nature of relationships Data Mining. Every day, thousands of organizations are implementing data mining techniques to better understand purchase behavior, human demand, industrial needs, geographical events, financial market analysis etc. Besides the commercial perspectives, data mining also has significant impact from a scientific perspective. Data is collected and stored at enormous speed. The humungous amount of information is collected through the sensors in remote satellites. Telescopes scanning the skies store files which are inherently very large in size (ranging from Terabytes to Petabytes). When working with gene data on projects such as gene expression or genome data, the records are usually large. Traditional techniques are not feasible on such raw data. In such cases data mining aids the scientists in classification and segmenting of data and in hypothesis formation. Data mining employs a variety of tools to discover patterns and relationships that can be used to make valid predictions.

Knowledge discovery in databases (KDD)

Data mining has been popularly treated as a synonym of knowledge discovery in databases. The patterns identified by data mining can also be synonymously stated as discovering the hidden knowledge about the given dataset. KDD however is described as a systematic (algorithmic) approach for the data analysis, exploration and modeling, to discover knowledge wherein Data mining is a step in this process.

The steps involved in the KDD process as stipulated by [15] is illustrated in Figure 3.1. Each of the individual steps are explained as follows:

Step 1. Data Cleansing

In this step, measures to enhance the quality of the data are taken. This includes data

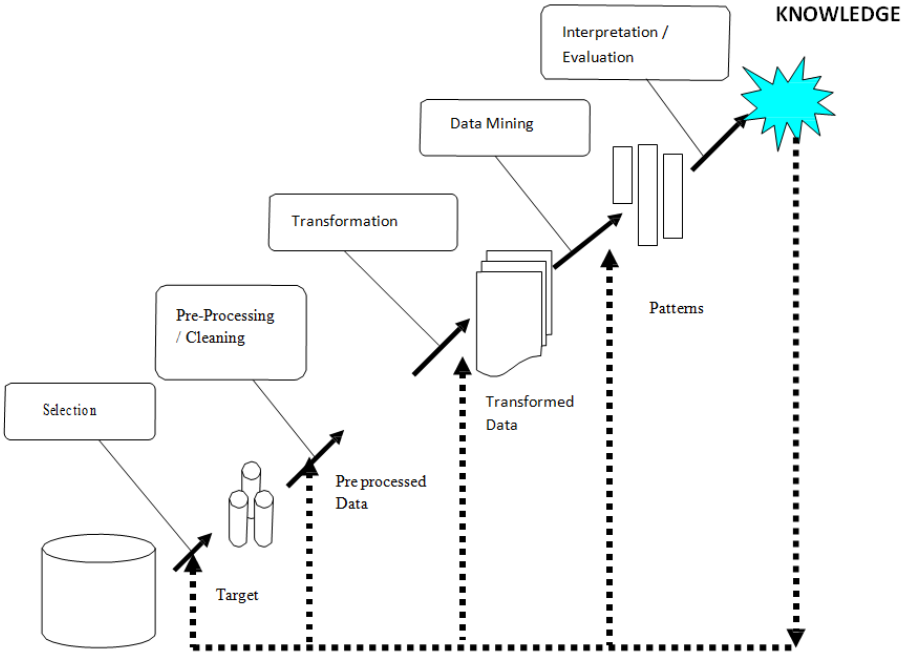


Figure 3.1: **Steps of KDD process**

cleansing steps such as handling missing data, noise correction and applying filters for outliers. With the goals of the Process defined, the dataset to carry out the KDD process should be prepared next. Data maybe collected from various different heterogeneous sources and thus requires an assessment of what data is available, what data is required and then integrating all of the available data to create the best dataset that contains all the attributes in it to carry out the data mining task on. In many situations the only possible solution is cleansing the data.

This process is very important because the Data Mining learns and discovers from the available data and in most cases form the base for devising models that extrapolate the given function. So it is important that the dataset encompasses as many attributes so that an effective model can be developed. Also, in this step noise data and irrelevant data are removed from the collection. A prediction model for this attribute will be developed,

and then missing data can be predicted. Although most mining routines have some procedures for dealing with incomplete or noisy data, they are not always robust. Instead, they may concentrate on avoiding over fitting the data to the function being modeled. Some of the commonly used data cleansing techniques are explained below.

1. **Binning:** In this technique, the data is grouped by consulting the values of the neighbor data. The data is sorted into a series of equi depth data buckets or data bins. The smoothed data is then sorted using the mean or median of the bin.
2. **Regression:** In this approach, grouping is done by fitting the data to a regression function. Linear regression is employed to find the line that best fits both the variable. This linear fitting enables to predict the value of one variable with another. As an extension, multiple regression is employed when there are more than two variable involved. Here as the name suggests, the data is fit onto a multidimensional surface. In both, the regression equation of the fit is used to smooth out the noisy data.
3. **Clustering:** This method treats data points which are farther compared to the other major clusters as noisy data (outliers). These clusters are chosen based on algorithm which is decided by the problem at hand.

Step 2. Data Integration

As the data is collected from heterogeneous sources, it sometimes might just not be available because it was not considered important for the given task at hand or at that particular instance of time. Since the data used for data mining is generally historical, the data source cannot be changed or improved. It is most likely that this step has been carried out in the earlier processes. The data usually comes from many sources, internal and external. The internal data might include transactional data along with other data recorded in other forms such as spreadsheets. External data includes all other data which usually add weight to the existing data and lend a hand during data mining operations

in case of demographic, geographic, industrial and financial applications. Data mining even though can be carried out in small scale; on a set of data files or a couple of data tables, is usually carried out in large scale. This involves an integration data of different size, format and sources into one repository which in turn become warehouses for data. Since in most cases data is combined from heterogeneous sources, there is a degree of coherence that has to be achieved. The input data may be from data tables, flat files or Data Warehouse. So it is necessary to choose a file format that puts together maximum amount of data present. Again, during the integration it is important to keep a check on the schema of the data (both the input datasets and the final dataset created). It is here that the metadata which is defined as the data about the data can be put to good use. The metadata if used properly would eliminate errors in schema integration and in some cases human intervention is also necessary to thoroughly understand the data and establish the relationship between various entities, thus establishing the Entity-Relationship. How can like real world entities from multiple data sources be 'matched up'. This is referred to as the entity identification problem. The manual mapping of entities and attributes; for example checking that the student_id in the examination table is properly matched with the student_id in admissions table is carried out in this step. Following this integrity check, a redundancy check should also be carried out so that there are not multiple instances of the same data stored. Redundancy may also be in the form of multiple tables in the database storing the same data. Missing data might turns out to be tricky when working with multiple data sources as there might be databases or tables which might not use null values to represent the unknown or not applicable data. It is therefore important to identify this attribute and develop a prediction model with this attribute in mind. The ways to handle missing values are analogous to those for handling wrong data. Thus in

this step, multiple data sources, often heterogeneous, are combined in a common source.

Step 3. Data Selection

The data relevant to the analysis is decided on and retrieved from the data collection. One of the reasons for non availability or the inconsistent data may be due to equipment malfunctions which might result in missing or misinterpreted readings of the data. Such duplicate tuples created also require cleaning before selection. Inconsistencies or duplicates might also arise while data being recorded as codes and this needs to be set alright in this step.

Step 4. Data Transformation

It is also known as data consolidation; in this phase the selected data is transformed into forms appropriate for the mining procedure. In this stage, the source data is worked on to generate the destination data. Having an integrated and clean data repository is not sufficient to perform data mining on them. Sometime the data mining tools might not be able to handle the entire data repository as it might be diverse or might not be very productive. So it is important to select the relevant data transform it, de-normalize it if need arises and then set up to a minable format. This data aggregation can be done either horizontally by choosing long data instances or vertically by choosing along feature sets. Data transformation includes a more varied collection of operations: creation of derived attributes: by aggregation, combination or discretization. Techniques such as smoothing, data cube construction might also be a step during this transformation process.

Step 4. Data Mining

It is the crucial step in which intelligent techniques are applied to extract potentially useful patterns. Also the nature of the dataset determines the attributes that are given as the goal of the supervised learning method. For instance, the statistical or data mining method to be applied on the model might be decided on the attribute with missing data. The objective of the data mining identified earlier in the KDD process determines the

task.

Exploratory Data analysis methods can range from the basic exploratory techniques to the multivariate exploratory techniques. Methods such as examining the statistical distributions of variables (like mean, mode, skewness or calculating the coefficient matrices) and assessing them for threshold values all come under the basic statistical exploratory methods. Multivariate exploratory techniques on the other hand work with cluster analysis, Stepwise regression, Classification or Time series analysis to identify patterns in multivariate data.

It is also useful to describe data entries as classes or concepts in a summarized, concise yet precise format. These descriptions can be carried out using data characterization and data discrimination. *Data Characterization* is a summarization of the characteristic features of a given target data class. The data objects pertaining to the given class is collected from the queries. The output of data characterization can be presented in many forms such as Piecharts, Bargraphs, multi-dimensional tables. Data Discrimination is a comparison of the general features of targeting class data objects with the general features of objects from one or a set of contrasting classes. User can specify target and contrasting classes and data objects be collected from the queries. *Data discrimination* usually employed to generate report assessment about improvement or change in performance. The methods employed for data discrimination is same as characterization but with an emphasis on comparative measures to help distinguish between the target and the contrasting classes. Depending on whether the task demands-predictive or descriptive, appropriate measures are taken. If the main task interests in describing or devising relation between data objects, then description methods are employed. In other words, these techniques aim at cognition about the final model. On the other hand, for the tasks oriented towards explaining or predicting or re-cognition of the data variables based on the already studied data, the values of future variables, then predictive data mining techniques are to be used.

The latter is under the assumption that this training with the data with be applicable for the future models to a good degree. Depending on the end goal of the data mining task, there are different algorithms that are employed. A broad classification of the algorithm types is listed below:

- ***Classification:*** These algorithms are targeted at predicting discrete variables; based on the characteristics of the attributes in the dataset the data are assigned to a particular set of labels. The output is usually as nominal data. Methods such as categorization, probability estimation are forms of clustering.
- ***Association:*** These algorithms aim at deciphering relationships between the existing attributes/ variables in the given dataset. The associations might be directional or sequential.
- ***Regression:*** This technique is commonly employed when working with continuous data systems such as trend analysis which is based on careful assessment of the datasets.

The existence of many techniques for solving a few tasks is due to the fact that no technique can be better than the rest for all possible problems. The final implementation of the data mining algorithm might include several runs of the algorithm wherein the parameters are fine-tuned and the results recorded are satisfactory. For instance, deciding on the appropriate attribute selection method when employing regression techniques.

Step 6. Pattern Evaluation

This is where the data mining algorithm is evaluated and assessed for the patterns generated. The evaluation begins by comparing the mined results with that of the KDD goals originally stated. It might even involve re-evaluating the dataset and carrying out data transformations again to achieve better results. This step focuses on the comprehensibility and feasibility of the developed model. Once the results are evaluated to satisfaction, then the discovered knowledge is documented for further usage making sure that the

experiments and the results are replicable. In this step, interesting patterns representing knowledge are identified based on given measures.

Step 7. Knowledge Representation

This is the final step in KDD process, in which the discovered knowledge is visually presented to the user. This essential step uses visualization techniques to help users understand and interpret the data mining results. The knowledge discovered is presented in a concise manner so that it is simple and easy to understand and interpret for the user. Typically visualization techniques are used for this purpose. And finally the knowledge so gained from the above information process is to be incorporated. In order to translate the positive results of knowledge discovery into positive results for the system concerned, it is necessary to use a holistic evaluation and an ambitious presentation of the models to the thoroughly know-how and daily operation of the system.

3.2 PREDICTIVE DATA MINING

Data mining has been used in a wide range of applications. Temporal data mining tasks fall into one of the following categories [3] (i) prediction,(ii) classification, (iii) clustering, (iv) search & retrieval and (v) pattern discovery. However, temporal data forecasting works with predicting the future behavior and values based on the past behavior and sample values; so predictive and classification data mining techniques are typically employed.

3.2.1 Supervised Learning

A simple definition for predictive data mining is given as (Mitchell [40]):

Given a sample input dataset containing the input-output pairs, supervised learning aims at finding the function that learns from the given input-output pairs, and predicts the output for any

a new unseen test input so that the predicted output is as close as possible to the actual output.
Each parameter is represented by a variable which can take a numerical value. Each variable is called a feature and the set of variables is called a feature space. The number of features is the dimension of the feature space. The actual characteristic of the item we want to predict is called the label or class of the item. The name supervised comes from the fact that the input-output example pairs are given before hand by the expert and learnt by the system. Examples of the supervised learning include thematic map generation (classification) from satellite images, medical image processing for identifying tumors or other (ab)-normalities, recognition of hand written characters from the scanned documents, stock market trend analysis, and speech recognition. Predictive data mining is major research area. Concerning temporal data mining, it is one of the most extensively applied and studied task.

The input-output pairs, also called training dataset, is denoted by (x_i, y_i), where x_i's are often vectors of measurements over the d-dimensional attribute space and y_i's are the class labels. The label y for a test example is unknown. The output of the classifier is a conjecture about y, i.e. a predicted y value. Often each label value y is a real number. In this case, supervised learning is called "regression" and the classifier is called a "regression model." The word "classifier" is usually reserved for the case where label values are discrete. For example, in remote sensing image classification for a wildlife reserve, the input attribute space consists of various geological terrain (e.g., flat, mountainous, water-cover etc.), and the input vectors (x_i's) are the values of distance from the water body i^{th} region, and the outputs (y_i's) are the class labels that decide if that location is suitable for a watch tower in the reserve area. The classifiers are found by various methods using a set of training examples, which are items where both the set of features and the set of labels are known. A linear classifier maps a feature space X to a set of labels Y by a linear function.

Definition 3.1 A general definition of a linear classifier $f(\vec{x})$ be given as:

$$f(\vec{x}) = \langle \vec{w}.\vec{x} \rangle + b = \sum_i w_i x_i + b \tag{3.1}$$

where $w_i \epsilon$ R are the weights of the classifiers and b ϵ R is a constant.

The value of $f(\vec{x})$ for any item $\vec{x}$ directly determines the predicted label, usually by a simple rule. For example, in binary classifications if $f(\vec{x}) \geq 0$, then the label is +1 else the label is -1.

3.3 PREDICTIVE CLASSIFIER

Classification is one of the most common data mining technique employed in supervised learning, but has not been fully explored in temporal data mining. Since traditional classification algorithms are difficult to apply to sequential examples, because there are a vast number of potentially useful features for describing each example, an interesting improvement consists on applying to identify patterns that helps to understand the behavior better. Classification is relatively straightforward if generative models are employed to model the temporal data. Deterministic and probabilistic data models can be easily applied for classification as they inherently answer the question of whether or not a sequence matches the given model.

Classification can be formally defined as finding a function *f(x)* which maps the input patterns x_i onto output classes y_i (sometimes y_i's are also denoted as ω_i). The main objective is to assign a label to each tuple in the classified dataset, given corresponding feature vector x_j in the input dataset. The classification task targets at assigning labels to the dataset. Given a set of data objects whose classes are already known, and each of which has a known vector of variables, our aim is to construct a rule which will allow us to assign future objects to a class, given only the vectors of variables describing the future

objects.

The primary objective is to learn what the distinctive features describe each of the classes. So that when an unlabeled dataset is entered into the system, it can automatically determine the class to which the particular data instance belongs. However as mentioned above, actual applications employing classification is limited. Granted, several authors have presented work that aims specifically at optimizing time series data, there are very few instances of the classification techniques that actually make use of the temporal information. We now briefly discuss the commonly used schemes namely the Naïve Bayes classifier, Decision tree classifier and Artificial Neural Networks.

3.3.1 Naïve Bayes Classifier

A popular supervised learning technique is the Bayesian statistical methods which allow taking into account prior knowledge when analyzing data (Hand [23]). Its popularity can be attributed to several reasons. It is fairly easy to understand and design the model; it does not employ complicated iterative parameter estimation schemes. It simplicity makes it easier to extend it to large scale datasets. Another reason is that it is also easy to interpret the results. The end users do not require prior expert knowledge in the field which is how it derived the name Naïve Bayes classifier. In the Bayesian approach, the objective is to find the most probable set of class labels given the data (feature) vector and *a priori* or prior probabilities for each class. It essentially reduces an n-dimensional multivariate problem to n-dimensional univariate estimation. A simple working of the Naïve Bayes classifier is as follows:

For the sake of simplicity, the class label y is either 1 or 0. The classifier aims at using the initial set of data objects with known class membership to learn and calculate scores such that scores larger than a particular threshold t are assigned the class label 1 and a

class label of 0 for values less than the threshold. This score calculated is the basis for the classifier.

Let $P(y|x)$ be the probability that an object with measurement vector $x = (x_1, \ldots, x_p)$ belongs to class y. The score is calculated from the ratio

$$P(1|x)/P(0|x) \tag{3.2}$$

However, $P(y|x)$ can be further decomposed as $f(x|y)P(y)$, where $f(x|y)$ is the conditional distribution of x for class y objects, and $P(y)$ is the probability that an object will belong to class y if we know nothing further about it (the 'prior' probability of class y). eq. (3.2) now becomes

$$f(x|1)P(1)/f(x|0)P(0) \tag{3.3}$$

It is evident from above that $f(x/y)$ and the $P(y)$ are the functions that determine the classifier. $P(y)$ *is* the proportion of the class y objects in the training set; assuming that the training set was a randomly selected sample from the original dataset. And $f(x|y)$ is given by $\prod_{j=1}^{p} f(x_j|y)$, which is the product of all of the univariate distributions $f(x_j|y)$, j=1,2,....,p for each of the labels y (0 and 1).

Bayes formula for a dataset of set of l classes can thus be generalized as:

$$p\left(\frac{y_i}{x}\right) = \frac{p\left(\frac{x}{y_i}\right) p(y_i)}{p(x)} \tag{3.4}$$

Where $P(y_i|x)$ is the posterior probability i.e. probability of instance x being in class y_i, $P(x|y_i)$is the conditional probability density i.e. probability of generating instance x given class y_i, and $P(y_i)$ is the a priori probability distribution i.e. probability of occurrence of class y_i. $P(x)$ is the probability of finding a feature vector x from the set of l classes. $P(x)$ is given by:

$$\sum_{j=1}^{l} P(x|y_i)P(y_i) \tag{3.5}$$

Effectively, the Naive Bayes reduces a high-dimensional density estimation task to an uni-dimensional kernel density estimation. This makes the boundary separation for the classification task easier.

3.3.2 Decision Tree Classifier

A decision tree is a classifier expressed as a recursive partition of the instance space. Classification trees are frequently used in applied fields such as finance, marketing, engineering and medicine. The classification tree is useful as an exploratory technique. A decision tree may incorporate nominal or numeric or even both of attributes types. The decision tree consists of nodes that form a rooted tree, meaning it is a directed tree with a node called "root" that has no incoming edges. All other nodes have exactly one incoming edge. Test nodes are those which have outgoing edges and the remaining nodes are the leaf nodes which are also referred to as the decision nodes. The leaf nodes represent the class labels y_i. For each new sample (i.e., feature vector x), the classification algorithm will search for the region along a path of nodes of the tree to which the feature vector x will be assigned. Each of the internal nodes splits the instance space into two or more sub-divisions. The split is based on a certain discrete function used as input. In the simplest and most frequent case, each test considers a single attribute, such that the instance space is partitioned according to the attribute's value. There are two possible types of divisions or partitions: Nominal partitions: a nominal attribute may lead to a split with as many branches as values there are for the attribute. Numerical partitions: typically, they allow partitions in ranges like 'greater than' or 'less than' or 'in between'. Each leaf is assigned to one class representing the most appropriate target value. In some

case, the leaf nodes may also the probability vector that represents the probability of the target attribute having a certain value. Typically, internal nodes are represented as circles and the leaf node as triangles. Each node is labeled with the attribute it tests, and its branches are labeled with its corresponding values. So depending on the values, the final response can be predicted by iteratively traversing through the appropriate node down the tree and understand the behavioral characteristics of the entire dataset as a whole.The termination/stop criteria or the pruning method determine the complexity of the tree. Complexity is given as a measure of any one of the following:

- Total number of nodes in the tree.
- Total number of leaves in the tree.
- Height of the tree and the number of attributes used.

Ideally, each non-leaf node should correspond to the most applicable input attribute from the set of all attributes already traversed in the path from the root node to that node i.e. it should be the most descriptive node so that when it comes to prediction, the number divisions made or nodes traversed from the root is kept to a minimum. Decision trees were frequently used in the 90s by artificial intelligence experts because they can be easily implemented and they provide an explanation of the result. A decision tree is a tree with the following properties:

- Each internal node tests an attribute.
- Each branch corresponds to the value of the attribute.
- Each leaf assigns a classification.

ID3 (Quinlan [49]) was the first devised decision tree algorithm. It is a greedy algorithm that uses information gain as splitting criteria. The tree is designed to stop when all the instances belong to a single value of a target feature. The tree is first initialized with the original set S as the root node. The algorithm iteratively computes the entropy of each of

the attribute in the set S and selects the one with minimum entropy.

Entropy is defined as the degree of disorderliness. When pertaining to the definition of information, the higher the entropy of the data (model), the more is the amount of information required to better describe the data. So when building the decision tree, it is ideally desired to minimize the entropy while reaching the leaf nodes. As upon reaching the leaf nodes there in no more information required (therefore a zero entropy) and all instances have the values assigned to the target label.

Entropy of a dataset X with respect to the assigned label is given as

$$Entropy(X) = \sum_{i=1}^{n} P_i log_2 P_i \tag{3.6}$$

Where $P_i$is the proportion of the instances in the dataset that take the ith value of the target attribute labels from the set of n different labels. The entropy value so calculated is used to measure the Information Gain. Information gain by definition is the reduction in entropy.

Definition 3.2 Let X be the set of input attributes and they are partitioned into subsets ranging S_1,S_2,.....S_n. The weighted average of the information needed to classify of each of the subset's elements give the information needed to identify the class of a data instance of X.

$$H(X,S) = \sum_{i=1}^{n} \frac{|S_i|}{S}\ H(S_i) \tag{3.7}$$

Where $|S_i|$ is the subset of instances of S where X takes the value i and |S| is the number of instances. The fact that the entropy typically decreases when the training instances are partitioned on an attribute is used for Information Gain. It is defined as the difference between the information needed to classify an element of S before knowing the value of X, H(S) and the information needed after partitioning the dataset on the basis of knowing the value of X, H(X,S). Thus information gain due to an attribute X for a set S is given as:

$$Gain(X, S) = H(S) - H(X, S) \tag{3.8}$$

In order to decide which attribute to split upon, the ID3 algorithm computes the information gain for each attribute, and selects the one with the highest gain. Quinlan suggested an evolution of ID3, the C4.5 algorithm. Rather than gain that tends to favor attributes with large number of values, it uses the gain ratio as the splitting criteria which use the following ratio:

$$Gain\ ratio(X, S) = Gain(X, S)/Split\ Info(X, S) \tag{3.9}$$

where split info is the information due to the split of S on the basis of the value of the categorical attribute X. The working of the C4.5 can be summarized as follows:

- If all cases belong to the same class, then a leaf node is created and the node is returned along with the class label.
- Each of the attribute is tested for potential information gain which is calculated based on the probability of each case with a particular attribute value belonging to the particular class. If there is no information gain, then a decision node is created higher up in the tree with the expected value of the class.
- If a new class is encountered, a deciding node is created higher up the tree using the expected value.

These three base cases are used to identify the selection criteria and then the best attribute to branch on.The C4.5 is designed to overcome the drawbacks of the ID3. C4.5 can handle continuous attributes in addition to the discrete attributes buy setting up a threshold and splitting the attributes values along it (less than, greater than or equal to). If the data is incomplete i.e. missing attributes in the training dataset, then most implementation have options to make those as missing and those attributes are not considered when calculating

Gain values. Most importantly, C4.5 also employs pruning. Most data sets contain a fraction of data instances that are not very well defined compared to the neighboring ones. Pruning techniques are required to reduce classification errors so that the resultant model is realistic. C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes.

3.3.3 Artificial Neural Networks

Neural network is a mathematical model developed in an effort to mimic the human brain (Browne and Yang [8]). The neural network consists of the layered interconnected set of nodes/processors which are analogous to the neurons in the human brain. Each of these nodes has weighted connection to the nodes in the adjacent layers where the information is received from one node, and weighted functions are used to compute the output values. As the model learns, the weights changes while the set of inputs is repeatedly passed through the network. Once the model has completed the learning phase (with the training dataset), the test data is passed through the network and classified according to the values in the output layer. Once trained, an unknown instance passing through the network is classified according to the values seen at the output layer. Artificial neural networks (ANN), being a non-parametric model is very well suited for applications like large databases, remote sensing, weather forecasting etc. Various studies have been carried out to demonstrate the performance improvement over traditional classifier model. Also since it does not rely on any assumptions concerning the underlying density function, it is very well capable of handling multi source data. The perceptron model proposed by Rosenblatt [54] is a single layer neural network whose weights and biases are trained to produce a correct target vector when presented with the corresponding input vector.

The training technique used is called the perceptron learning rule. The perceptron

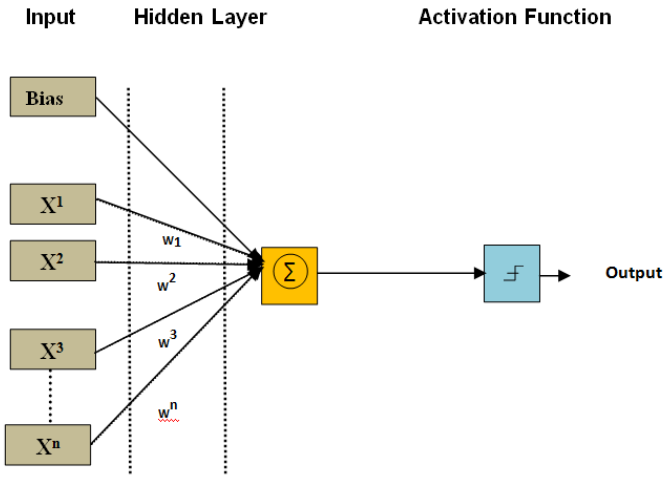


Figure 3.2: **Schematic representation of the neural network model**

was new in its ability to learn with the training data and randomly distributed data. The perceptrons were very well suited for the classification problems where data can be completely separated along a hyperplane. The perceptron works on the linearly separable data by employing the following learning rule

$$Bias = Bias + [Expected\,output(E) - Actual\,output(A)] \tag{3.10}$$

And the weights are calculated as follows:

$$W^i = W^i + (E - A)X^i \tag{3.11}$$

Where X^i is the input vector, W^i is the vector of weight and E and A are the expected and actual values respectively. Neural networks, in which signal flows from input to output (forward direction) are called, feed forward neural networks. A single input can only solve the linearly separable problems. For more complex problems, Multi-layered feed forward neural networks can be used. Multi-layer perceptrons (MLP) are simple interconnections of neurons organized into multiple layers, typically consisting of input, output and one or more hidden layers. These are designed to send their signals forward,

and then propagate the errors backwards through the network. The back-propagation algorithm is the most common method of training multilayer feed-forward neural networks. The back propagation uses supervised leaning which means that the model is trained first a set of input and known outputs and the errors (difference between the actual and the expected results) at the output layer is propagated back to adjust the weights of network connections. Appropriate modification of the weights is then carried out at the output layer and then moved backward through the hidden layers . Figure 3.2 shows a basic schematic representation of the neural network model where (x^1, x^2, x^3 ...x^n) denotes the input vectors and (w^1, w^2, w^3...w^n) is the weights vector which are usually real values. If the input causes the perceptron to fire, then a positive weight is added and if the input causes an inhibition, a negative weight is added. During the training, if the output is as expected then no changes are made, else the weights and bias are updated (Freund [17]). Once an entire pass through all of the input vectors is done and it no longer produces errors, training is complete. In order to model strong and mild nonlinear mappings, a correct choice of activation function and bias is necessary.

Activation function: The output of that node given an input or the net-input to the hidden unit is defined as the activation function. The perceptron modeled Rossenblatt used a Heaviside step function as the activation function. A step function is a basic on/off type function, if 0>x then 0, else if x>=0 then 1. Hence depending on the type of input, output and problem domain, suited functions are adopted at respective layers. However, with Multi-layer Perceptron the logistic sigmoid or the hyperbolic tangent functions are used.

Sigmoid function: The stronger the input is, the faster the neuron fires. The sigmoid offers increased stability in multi layered networks, as the sigmoid curve allows for differentiation (which is effective for the Back Propagation during the training phase). Sigmoid functions are characterized by their S-Shaped curve and are expressed mathematically as

$$f(x) = \frac{1}{(1 + e^{-x})} \tag{3.12}$$

Hyperbolic Tangent function: It is defined as the ratio between the hyperbolic sine and the cosine functions or expanded as the ratio of the half-difference and half-sum of two exponential functions in the points x and -x (Karlik [25])

$$Tanh(x) = Sinh(x)/Cosh(x) = \frac{e^{x} + e^{-x}}{e^{x} - e^{-x}} \tag{3.13}$$

The hyperbolic tangent function will produce positive numbers between -1 and 1. Since this activation function also has a derivative, it is effective for gradient descent (back propagation) training methods. MLP uses these functions because their derivatives are easy to compute and is easily expressed as the function of the inputs. These functions curves also contain a linear region that helps in regularizing the network using weight decay. During the training phases of the ANN, the weights and the value in the activation functions are both adjusted. The bias is used to keep a check on the threshold. The bias neuron is usually considered as just another input, i.e. it is connected to next layer but does not have any incoming connection from the previous layers and it always takes the value 1. The bias is the one responsible for shifting the activation function (curve) to the left or to the right. Mathematically, the bias can thus be expressed as

$$Weighted\,sum(v^{j}) = \sum_{i=1}^{m} w^{ij} x^{i} + bias \tag{3.14}$$

The algorithm works as follows:

1. The network is first initialized with uniformly distributed random weights.
2. The input vectors $X^{i}$and the desired outputs $o^{j}$are read.
3. Training: The network is then supplied with training data set.

4. For each of the data till the network converges i.e. a minimum error value is assigned.

   a) Forward-Pass: The input vectors X^i are transformed into output vectors Y^i using the equation:

$$y^j = \frac{1}{1+e^{v^j}} where v^j is the weighted\ sum. \tag{3.15}$$

   b) The difference between the desired output and the actual output is computed and recorded as :

$$Error = o^j - y^j \tag{3.16}$$

   c) Backward-Pass: The error signal at the output units is propagated backwards starting from the output units throughout the entire network. Thus both back- propagation through the output layer and back-propagation through the hidden layer have to be considered. The error is calculated as:

   For the Hidden layer

$$\Delta^j = y^j(1-y^j) \sum_{k=1}^{m} \Delta^k w^{jk} \tag{3.17}$$

   Each layer j in the hidden layer is connected to each layer k in the output layer with an edge of weight w^{jk}, for k = 1, . . . , m. All the possible backward paths should be included when calculating the back-propagated error up to layer j in the hidden layer.

   For the Output layer which considers the backpropagation path from the output of the hidden network up to the output layer j:

$$\Delta^j = y^j(1-y^j)(d^j - o^j) \tag{3.18}$$

The weights are updated using the following equation:

$$w^{ij}(t+1) = w^{ij}(t) + \eta \Delta^j y^j \tag{3.19}$$

where $w^{ij}(t)$ is the weight from the input node i to the hidden node j at a given instant t, η is learning rate of the network and Δ^j is the error calculated as above.

ANN is very good at handling classification when the dataset is categorical and has missing attributes. However with increase in the data size, the training time and complexity of the network increases but it is compensated by a drastic improvement in accuracy in most cases.

This Chapter discusses the various data mining concepts, methods and techniques.The disciplines of statistics and data mining have also been discussed to prove that these areas are highly interrelated and share a symbiotic relationship. This chapter also helps to gain a major understanding of the various data mining algorithms and the way these can be utilized in various real-life applications and the way these algorithms can be used in the descriptive and predictive data mining modeling. Models like decision tree, neural network and Naive Bayes classifier are described in detail. Important research works carried out using these models are reviewed in this chapter. Decision trees are very simple to interpret and work faster. Neural networks manage missing values and categorical values very efficiently. Naive Bayes insist that their numeric data should be normally distributed. The chapter ends with the discussion on the selection of appropriate data mining techniques in a particular scenario. From the discussions, it can be concluded that for data mining one cannot find a single classifier that outperforms every other but rather one can find a classifier that performs well for a particular domain.

Chapter 4

SPATIO-TEMPORAL DATA MINING METHOD

The purpose of this research is to demonstrate an effective data mining technique for a spatio-temporal dataset. Most spatio-temporal mining techniques discussed earlier employ linear classifier or a modified version of the linear classifiers. But in Spatio-temporal data, the occurrence of a data event is strongly constrained or dependent on the nature of the event in the neighboring unit and the previous instance of the same event. When linear classifiers are applied, the data models results in high residuals because the spatial relationships between the attributes is not appropriately captured. Additionally, the dependency between attributes in spatio-temporal data is usually not summed up by a linear function. In order to develop a model with good fit, the non-linear relationship must be effectively captured. In traditional applications, the data is archived solely for analysis purposes thus usually having large number of instance to work with. However, most spatio-temporal application aggregate data from different sources which results in relatively smaller dataset to carry out data mining operations. The data model developed should be able to capture the non-linear relationship between the various variables. These

include the relationships between the input variables, with the variable itself at different points in time and the interaction between the spatially dispersed variables.

4.1 SUPPORT VECTOR REGRESSION AND SMOREG

Traditional data mining algorithms operate under the assumption that the data is independently and identically distributed. However as discussed earlier, these assumptions are often not true for spatio-temporal datasets which is why the classical data mining methods perform ineffectively on them. Nearby data points have a greater degree of similarity and influence the spatial value strongly. It is important to relax this assumption to quantify the spatial dependence and factor it into the techniques for estimation of missing values. Thus there was need for a classifier that operates on high dimensional data and is flexible enough to model data from diverse sources without compromising accuracy for which SVM classifier suited best. The Support Vector Machine (SVM) is a novel machine learning method based on Statistical Learning Theory. SVM gained popularity due to its ability to handle non-linear decision boundaries using methods originally intended for linear classifiers thereby providing high generalization abilities (estimation accuracy) (Cortez [11]). SVM employs kernels functions which enables it to work with multi-dimensional data (to generate the optimal hyperplane).

A support vector machine (SVM) consists of a set of related supervised learning techniques that analyze data and recognize patterns used for classification and regression analysis. Recently, SVM has been extended beyond the optimization model to time-series modeling. Mukerjee [44] employed SVM to carry out modeling of non-linear chaotic time data. Kim [27] developed a model for financial time series forecasting using SVM. Babovic et al. [4] demonstrated that SVM produced better results over 12 time periods in comparison to ANN for water level predictions. The training of the SVM (formulated as a

Quadratic Programming optimization) is the determining factor. For data of large dimension, the kernel matrix would become a bottleneck. Smola and Scholkopf [59] proposed an iterative algorithm called Sequential Minimal Optimization (simply SMO), for solving the regression problem using SVM. This algorithm is an extension of the SMO algorithm proposed by Platt [48] for SVM classifier design. It was further improved by Shevade et al. [26]. The remarkable feature of the SMO algorithms is that they are fast as well as very easy to implement. They also improved the SMO algorithm to solve regression problems thus introducing the SMOReg algorithm. The SMOReg algorithm drastically reduced the training time and space complexity (O (1)). A regression prediction is then obtained as input from SMOReg taking the attribute value of the test instance as input. Typically, the SVM is a non-probabilistic binary linear classifier that takes in a set of input data and predicts the possible classes for the output. The data points are considered as vectors and the ideology is to separate these points with n-dimensional hyperplane. Out of the numerous hyperplanes generated, the one wherein the greatest margin exists is chosen. Considering a data set $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$ with $\mathbf{x}_i \epsilon \mathbf{R}^d$ being the input data vector and $\mathbf{y}_i$is the corresponding binary label wherein $\mathbf{y}_i \epsilon \{-1, +1\}$, SMOReg tries to find the hyperplane which divides the input data points (say, the initial storage or the inflow) and identify the function relating these features to the output value(through regression). In formula this reads as:

$$f(x) = \langle w, x \rangle + b \qquad (w, x \in R^d) \tag{4.1}$$

where $\mathbf{w}$ and $\mathbf{b}$ are the slope and offset of the regression function and $\mathbf{x}_i$ represents the data points. The optimization problem here can be solved by minimizing the following cost function. Firstly the slope minimization, $||\mathbf{w}||$:

$$Minimize \left(\frac{1}{2} ||w||^2 \right) \tag{4.2}$$

Where $||\mathbf{w}||$ is the slope and $\left(\frac{1}{2}\right)\ ||\mathrm{w}||^2$ is the term characterizing the model complexity. And subjected to

$$f\left(x\right).y_i \geq 1 \qquad \forall i = 1, 2, ..., n \tag{4.3}$$

The final quadratic dual problem to be solved in the SMOReg algorithm (Smola [59], Cortez[11]) is as follows:

$$Maximize_{\alpha_i,\beta_i} \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_i \beta_j (\varphi\left(x_i\right).\varphi\left(x_j\right)) \alpha_i \alpha_j \right\} \tag{4.4}$$

subjected to

$$0 \leq \alpha_i \beta_i \leq C \qquad \forall i = 1, 2.n \tag{4.5}$$

$$\sum_{i=1}^{n} y_i \alpha_i = 0 \tag{4.6}$$

where $\alpha_i,\ \beta_i$ are Lagrange variables.

It does so by converting the problem into smaller set of quadratic polynomial SVM sub-problems and each of these sub–problems involve two Lagrange's multipliers. Now the constraints (from (4.5) and (4.6)) are reduced to the following:

$$0 \leq \alpha_1, \alpha_2 \leq C y_1 \alpha_1 + y_2 \alpha_2 = k \tag{4.7}$$

The working of the algorithm is as follows:

1. A Lagrange's multiplier $\alpha_1$that does not satisfy the Karush–Kuhn–Tucker conditions [66] (KKT) is chosen.

2. Another Lagrange Multiplier α_2 is chosen and optimization is done. The objective is to choose a value for α_2 such that the distance is maximized.

The above two steps are carried out for all the sub-problems till convergence and the solution for above problem is obtained. In (4.4), the $(\varphi(x_i).\varphi(x_j))$ is the Kernel function ***K(x)*** which is the function that transforms the non-linear input space into a high dimensional feature space in which the solution of the problem can be represented as being a straight linear classification or regression problem. Here x_i,x_j are the data points and φ is the vector representing the mapping function.

Since the nature of the data is usually unknown, it is very difficult to make, on beforehand, a proper choice out of the kernels. For this reason, during the model building process, usually more than one kernel is applied to select the one which gives the best prediction performance. For a chosen kernel to be valid, it must satisfy Mercer's theorem [37]. There are four types of Kernel functions that will be used for this experimentation. Table 4.1 gives the mathematical kernel functions; x and y are input vectors and C is the trade-off value determining the degree of the polynomial. The parameters σ and w (are from the Pearson function) and control the half-width and the tailing factor of the peak. The degree of the polynomial is given by d; when $d=1$ it reduces to the linear kernel and $d=2$ it becomes the quadratic kernel and so on.

Polynomial kernel fits many polynomials, each within specified overlapping neighborhoods. Polynomial kernel produces surfaces that account for more local variation and fits the specified order (e.g., zero, first, second, third) polynomial using all points only within the defined neighborhood. The neighborhoods overlap and the value used for each prediction is the value of the fitted polynomial at the center of the neighborhood.

However, for large sample sizes, the polynomial function is not sufficient, and normalization is carried out. By normalization, the maximum value of the kernel is bounded to 1. Radial basis function based kernels are used for calculating smooth surfaces from large number of data points. This method is not adequate for sample points with large changes

Table 4.1: **Various Kernel functions used in the study**

Kernel Type	Function
Polynomial Kernel (PK)	$K\left(x^i, y^j\right) = \left(x_T^i . y^j + 1\right)^d$
Normalized Polynomial Kernel (NPK)	$K_r\left(x^i, y^j\right) = \frac{\left(x_T^i . y^j + 1\right)^d}{\sqrt{\left(x_{T+1}^i + y_{T+1}^j\right)}}$
Pearson VII Universal Kernel (PUK)	$K_r\left(x^i, y^j\right) = \frac{1}{\left[1 + \left(\frac{2(x^i + y^j)\sqrt{2^{\frac{1}{w}} - 1}}{\sigma}\right)^2\right]^w}$
Radial Basis Function Kernel (RBF)	$K_r\left(x^i, y^j\right) = exp(\gamma \left\| x^i - y^j \right\|^2)$

within small distances. RBF kernels are exact interpolators and the basic concept in this kernel is to fit a surface through the data points while minimizing the total curvature of the surface (Tveito [46]).

Pearson Universal Kernel is modeled based on the Pearson function by Karl Pearson which is effective in fitting a variety of line widths and shapes with peaks. The Pearson VII function has the possibility to change easily, by adapting its two parameters, from a Gaussian into a Lorentzian peak shape and more. This unique property makes it possible to use the Pearson VII function as a generic kernel.

Different kernels provide different generalization for the same problem at hand. Performance of each of the kernel is problem specific, so one cannot be trumped as better compared to another. The question is which kernel functions provide good generalization for a particular problem. We could not say that one kernel outperforms the others. Therefore, one has to use more than one kernel functions for a particular problem. Some validation techniques such as bootstrapping and cross-validation can be used to determine a good kernel (Smola [59]). For instance, RBF has a parameter and one has to

decide the value of before the experiment. Therefore, selection of this parameter is very important in order to achieve the expected accuracy.Therefore, a nonlinear regression function $\phi(f_t,f_s)$ is constructed by performing the SVR on the temporal forecasts f_t and the spatial forecasts f_s to find out the best spatiotemporal forecasting values.
Support vector regression is employed to find nonlinear combination function $\phi(f_t,f_s)$ (to generate the overall forecasting, the selection of the kernel function and corresponding parameters plays a significant role in obtaining good forecasting.

4.2 PRINCIPAL COMPONENT ANALYSIS

PCA is designed to combine the variation in a correlated multi-attribute dataset to a set of uncorrelated components which is a linear weighted combination of the original variables. PCA is particularly useful in high dimension (large number of attributes) data wherein patterns are not easily perceived. The other main advantage of PCA is that once the patterns are identified , they are compressed without losing much information. The steps in calculating Principal Components [58] can be outlined as thus.

1. To begin with, the data is to be centralized. The mean value is subtracted from the data values so that the data is centralized.
2. The covariance matrix is populated. Covariance is a measure of how much the data objects mean with respect to each other.
3. The Eigen vectors are next calculated. The eigenvector with the highest value is the Principal Component of the data set.
4. The Eigen vectors are ordered by Eigen value starting from the largest. Order the Eigen vectors by Eigen value, highest to lowest. This gives us the components in order of significance
5. Next, the components of lesser significance (those that result in over-fitting) are

selected to be not included in the final set of components. The result is the final set of components.

4.3 SUPPORT VECTOR REGRESSION (SVR) MODEL

In traditional applications, the data is archived solely for analysis purposes thus usually having large number of instance to work with. However, most spatio-temporal application aggregate data from different sources which results in relatively smaller dataset to carry out data mining operations. The data model developed should be able to capture the non-linear relationship between the various variables. These include the relationships between the input variables, with the variable itself at different points in time and the interaction between the spatially dispersed variables. Also since the end goal is to develop a predictive data model to estimate numerical values, the traditional linear classifiers that typically assign class labels cannot be used to develop ideal fits. Additionally, data models developed should work well with small training samples.

Reservoir operation forecasting is inherently a complex task. Even for a standalone system, numerous interactions between the input and the output parameters must be considered to determine the release operational values. However more common than usual, the framework usually consists of multiple reservoirs thus resulting in increased operational complexity. For example the storage value dictated of a downstream reservoir is very much dependent on release levels of the first reservoir in the river system. But the release value of a given reservoir is more strongly dependent on the inflow and release values of immediate reservoirs than the others in the framework. So at the outset, the multi-reservoir framework would seem as a multivariate spatial data application. But in order to develop real time models, it is necessary to include the temporal information about the spatial distribution of events at various instances in time. Recording snapshot

of the spatial data at different instances is required to understand the temporal evolution of the spatially distributed data units and changes of the same data unit with time. Thus the Reservoir operation modeling is an excellent application of a spatio-temporal data model.

We therefore propose a SVM based multivariate non-linear regression model for spatio-temporal data. The training of the SVM (formulated as a Quadratic Programming optimization) is the determining factor. The Support Vector Regression (SVR) model that is known to provide competitive results in data short environment handles nonlinear problems by transforming the multivariate problem to multiple binary problems (section 2) and SMOReg . For the spatial model, the SVM classifier takes as input the current (at a given time instant t) values reservoir variable of each of the reservoirs in the framework. But in case of the spatio-temporal data, the values of the reservoir variables at previous time instances (t-1, t-2 and t-3) are also considered in addition to the current values. The kernel function is the major bottleneck for performance. Our implementation of the SMOReg employs a quadratic kernel function which is given as

$$K(x, y) = (x^T y + C)^2 \tag{4.8}$$

Data is usually split into two datasets as a) training and b) testing. The parameters of the data mining algorithm are estimated through training phase. The estimated parameters or the hidden relationships efficiency in mapping the patterns are evaluated in testing phase. The testing dataset must not contain patterns from the training dataset. The parameters of the data mining algorithm are updated until testing phase maps the pattern more the minimum acceptable limit.

ALGORITHM 1: RF- SVR

INPUT:

Training set that consists of a spatio-temporal framework **D** ordered as an $m \times n$ matrix, where

m is the number of training sample and **n** is the number of input features (variables) and an **m** dimensional set **Y** of output variables.

An **m** dimensional set Y of output variables.

A training dataset **E** while has the same label index **n** as **D** and has **l** instances (i.e. has dimensions $l \times n$).

The explained variance value **v**.

OBJECTIVE: Implement the SVR function to develop an ideal fit.

1: Read D, Y, E, v;

2: P= Feature_Reduction (D, v);

3: Q=Feature _Reduction (E,v);

4: Trainfile=(P,Y);Testfile=(Q);

/**Read in training data file (the attributes are numerical). The file is a CSV file format and the first row contains the attribute names/***

5: DataSource train= new DataSource(TrainFile);

6: trainData=train.getData();

*/*Define class index*

trainData. setClassIndex(trainData.numAttributes()-1);

*/*Carrying out Support Vector Regression(SVR)*/* 7: SMOReg model=new weka.classifiers.function.SMOReg();

8: model.setC(1.0);

9: w=weka.classifier.functions.supportvector.RegSMOImproved();

10: Model.setRegOptimizer(w);

*/*kernel function (here for Quadratic Kernel function)*/*

11: kernel=weka.classifier.functions. supportVector.Polykernel();

12: kernel.setExponent(2.0);

13: model.setKernel(kernel);

14: model.buildClassifier(trainData); */*Developing training model*/*

15: DataSource test=new DataSource(Testfile);

16: testData=test.getDataset();

17: n=testData.numInstance();

18: **for** i=1:n

19: values (i)=model.classifyInstances(testdata.instance(i-1));

20: **end**

21: **for** i=1:n

22: inst=testData.instance(i-1);

23: vals=inst.toDoubleArray();

24: diff=vals(end)-value(i);

25: fprintf("%d Actual =%.1 f Predicted=%.1f Residue=%.1 f\n", vals(end),vals(i), diff);

26: **end**

ALGORITHM 2: Feature_Reduction (D,Y,v)

1: Read Data;

2: A=Transpose (Data);

3: [n m]=size(A);

4: Amean=mean of data;

5: AStd=standard deviation of data (A);

6: V=covariance of data (A);*/*Covariance matrix of input database*/*

7: B=calculate zscore(); */*PCA from covariance*/*

8: [coeff,score,latent]=princomp(B);

9: PC=coeff;*/*Principal components calculation*/*

*/*Explained variance*/*

10: Explained variance=cumsum(variance(score))/ sum(variance(score));

11: Input variance value k

12: J=Length (k)

13: **for** i=1 to J

14: **if** Explained variance (i) ¡=K

15: count=i;

16: **end**

17: **end**

18: Selected PC components=PC (:, 1:count);*/*PCA component selection*/*

19: Z1= (((B *selected_PC) * selected_PC')* repmat(AStd,[n 1])) + repmat(Amean,[n ,1]);

20: Data_set=Transpose (Z1);

21: **Return** Explained variance (count);

22: **Return** number of components;

4.4 REDUCED FEATURE SUPPORT VECTOR REGRESSION (RF-SVR) MODEL

Including temporal information into the spatial data drastically increases the dimensions of the aggregated spatio-temporal dataset. The resulting situation is that there is a relatively large number of attributes but only no more than hundreds of instances of the dataset. However, all of the input features do not necessarily have the same degree of descriptiveness. The target output feature might have a stronger degree of correlation with a definite set of the input features. So we propose a Reduced Feature-Support Vector Regression by employing PCA for dimension reduction. As described in section 2, PCA enables us to identify the minimal number of components that can contain the maximum amount of information. Algorithms 1 and 2 demonstrate the implementation of the RF-SVR Algorithm using a quadratic kernel function for Spatio-Temporal data systems. This reduction in the size of the effective dataset consequently decreases there the computation time thereby further enhancing the efficiency of the SVR model.

Chapter 5

METHODOLOGY

As mentioned in Chapter 1, the objectives of the present research were set as follows:

1. To develop data models for multi-variate Spatio-Temporal framework in a data short environment.
2. To enhance the model by carrying out feature reduction with data mining techniques.
3. To carry out a sensitivity analysis of various kernel parameters on the performance of both Spatial and Spatio-Temporal data mining techniques.
4. To demonstrate the applicability of data mining with case studies of Reservoir System Operation.
5. To make an inter-comparison of the performance of Temporal, Spatial and Spatio-Temporal data mining.

In the present research, the first level of classification methods namely, temporal data mining (using temporal data base), spatial data mining (using spatial data of the same variable at various points of interests spread over the space) and a combination of both spatial and temporal data mining method has been applied with the case study data of reservoir operation. These methods are briefly illustrated in the following sections. Subsequently, the performance measures used for evaluation of these methods with reference to data mining or knowledge extraction have been

explained in detail in this chapter.

The research was carried out in two phases. The first phase consists of temporal data mining and the second is spatial data mining including spatio-temporal data mining. Though KDD steps followed for both the models are the same (as discussed in Chapter 3), the methods are described separately for clarity of presentation. Detailed description of the 4-step KDD process involved in temporal data mining is presented first, followed by the same for the spatio-temporal data mining model. This chapter covers the extensive process of data mining carried out on both the data sets. The phase I and phase II of the research work carried out have been pictorially represented and depicted in Figures 5.1 and 5.2 respectively.

5.1 PHASE I: TEMPORAL DATA MINING

To begin with, the difference between the temporal data mining and time series analysis is discussed.

Definition 5.1.*Temporal Data Mining* is defined as a single step in the process of Knowledge Discovery in Temporal Databases that enumerates structures (temporal patterns or models) over the temporal data, and any algorithm that enumerates temporal patterns from, or fits models to, temporal data is a Temporal Data Mining Algorithm. (Lin et al. [33]).

Definition 5.2. A *time-series* T is an ordered sequence of n real-valued variables $T = (t_1, \ldots, t_n)$, $t_i \epsilon$ R. A time series is obtained from the observations of an underlying process in the course of which values are collected from measurements made at evenly spaced *time intervals* and according to a distinct *sampling rate*. A (discrete) time series [a (discrete) stochastic process] is a sequence of random numbers. It is supposed that the observed value of the series at time t is a random sample of size one from a random variable X_t; for $t \epsilon \{1,\ldots,n\}$. A time series of length n is a random sample of a random vector like this $(X_1,\ldots,X_n)$. The random vector is considered as part of a discrete-time stochastic process, and observed values of the random variables are used for the time series analysis processes. A time series will thus contain subsequent attributes or numerical

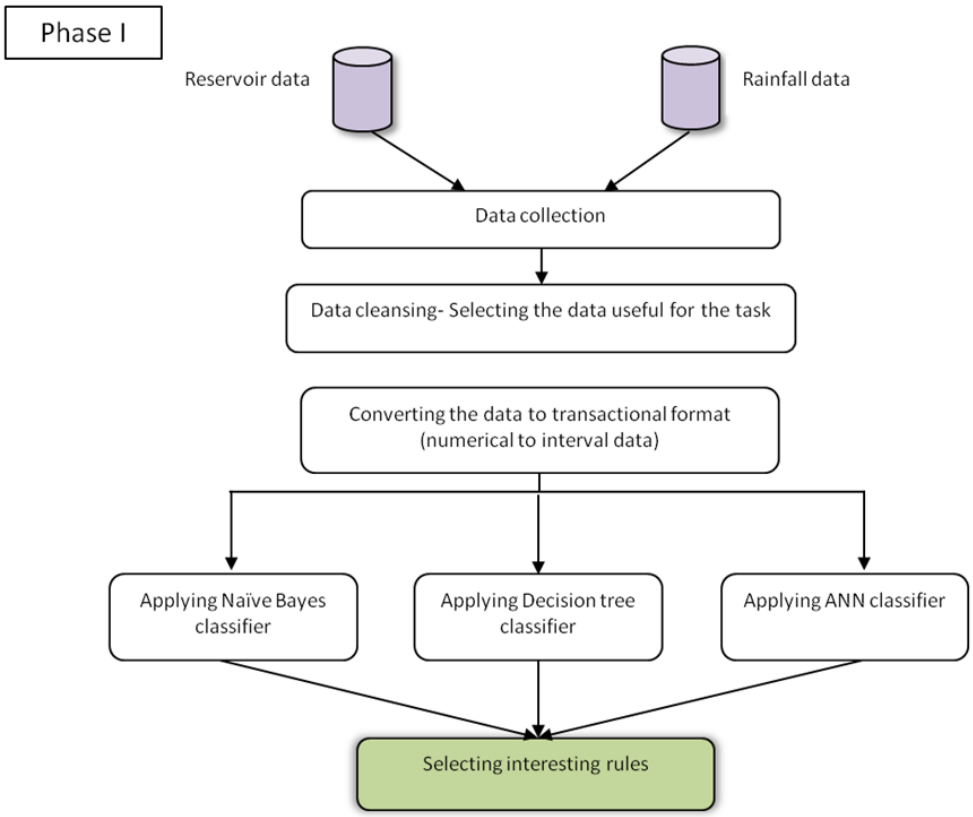


Figure 5.1: **Steps carried out in Temporal data mining**

values over contiguous time instants.

Some of the fields which carried out time series analysis are weather forecasting applications, financial trend analysis, business transactions and process automation. Temporal data mining on the other hand is fairly recent and has different constraints. There is a difference in the size and nature of the data collection and data organization methods. The data mining techniques employed should be capable of efficiently modeling and analyzing prohibitively large datasets. The objective of temporal data mining is also different from that of time series mining in terms of the knowledge to be derived. More commonly than usual, one does not even know which variables in the data are expected to exhibit any correlations or causal relationships. Furthermore, the exact model parameters may be of little interest in the data mining context. For example, the time-stamped list of the books checked out from the library could use data mining techniques to

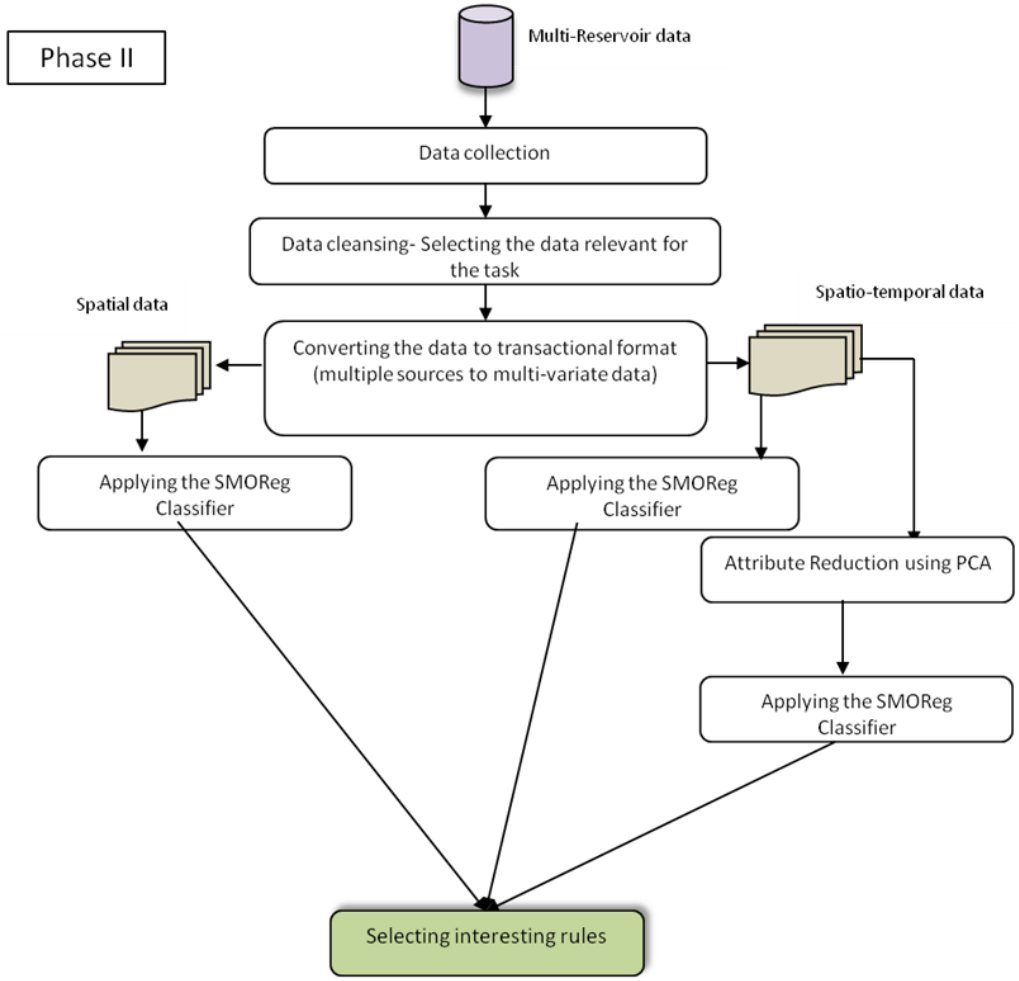


Figure 5.2: **Steps carried out in Spatial and Spatio-Temporal Data Mining**

gain knowledge about how the inventory should be stocked at various months of the year.

When working with temporal data, care must be taken to ensure that along with the other variables the time is correctly recorded. There may be many different interpretations for time. The particular time at which the information is recorded is referred to as valid time (This consists of a start time and an end time which is usually the time between the two consequent data records). There is also the transaction time that is the timestamp value recorded in along with the data.

The temporal data is usually of the following types (Roddick et al. [53]); *Static, sequential or time stamped*. *Static data* does not contain any temporal information but inferences can be made through transaction-time references. (E.g. Audit trail data, transaction log etc).*Sequential* data as per the

definition is a ordered sequence of the event. The nature of sequential data is such that even if there does not exist a reference to time, there is an inherent temporal relationship between the datasets. The qualitative relationship is essentially captured based on the time between the events. For instances, if working with transactional data then the order of occurrence of each of the transaction like before, after, at the same time etc would make up the sequence order. And finally, when explicit information about the time of occurrence is present it is categorized as *time stamped data*. However, the pattern analysis inferences need not necessarily be temporal in nature. Thus, the choice of a data model for temporal data mining is a very crucial and complex task as the model should be able to represent the dataset well so that appropriate data mining technique can be employed. And since the scope of data mining extends beyond standard data prediction, as when working with time series data more often than usual, one does not even know which variables in the data are expected to display and correlation or causal relationships. For instance a time stamped data containing the list of purchases made could reveal which combinations of items tend to be frequently bought and even perform a trend analysis to compare last year's transaction with that of this year. Thus it is necessary to choose an appropriate data mining algorithm when working with N-Dimensional transaction databases, as the goal here is to discover the underlying relationships in multivariate time sequences.

Supervised learning methods that allow for event analysis in a multi-dimensional space should be chosen. Data integration and reclassification methods discussed in chapter 3 are usually for data that measured at the same time. We propose to extend the existing linear classifiers to deal with temporal data instead of the traditional probabilistic induction methods. Typically, the standard classifiers take as input the data values in a current time (at time t) and space and do not keep track of past or future events. The temporal data mining however takes as input not only the current data values but also the history, that is past data values of upto m units (from t-m to t-1 time units) with the assumption that they influence the decision at time t so as to predict future database instances.

5.2 PHASE II: SPATIAL AND SPATIO-TEMPORAL DATA MINING

The purpose of this research is to demonstrate an effective data mining technique for a spatio-temporal dataset. Most spatio-temporal mining techniques discussed earlier employ linear classifier or a modified version of the linear classifiers. However, in spatio-temporal data, the occurrence of a data event is strongly constrained or dependent on the nature of the event in the neighboring unit and the previous instance of the same event. When linear classifiers are applied, the data models results in high residuals because the spatial relationships between the attributes is not appropriately captured. Additionally, the dependency between attributes in spatio-temporal data is usually not summed up by a linear function. In order to develop a model with good fit, the non-linear relationship must be effectively captured. In traditional applications, the data is archived solely for analysis purposes thus usually having large number of instance to work with. However, most spatio-temporal application aggregate data from different sources which results in relatively smaller dataset to carry out data mining operations. The data model developed should be able to capture the non-linear relationship between the various variables. These include the relationships between the input variables, with the variable itself at different points in time and the interaction between the spatially dispersed variables. Since the end goal is to develop a predictive data model to estimate numerical values, the traditional linear classifiers that typically assign class labels cannot be used to develop ideal fits. Additionally, data models developed should work well with small training samples.

These data far exceeded the humans ability to analyze in order to derive some useful results. Recent research studies on data mining have naturally extended the scope and application of data mining techniques to temporal data, relational data, spatial data, and spatio-temporal data bases. Especially the explosive growth of geographical database creates the necessity for non-trivial knowledge/information discovery from these data using data mining. The various spatio-temporal model and applications ranging from statistical models to modelling data as geometric objects

are discussed in Chapter 2. However, it requires manipulation of large datasets to find ideal fits for complex models. Often, the required solutions are not available in closed mathematical form and computer Intensive methods are needed. Besides devising the data model aspects such as scalability, outlier identification and ability to handle continuous and discrete data etc should also be addressed.

Data mining of spatio-temporal data is a challenge for several reasons. First, the spatio-temporal datasets are usually much larger than spatial data sets because of the numerous snapshots that they contain, if we use snapshot spatio-temporal databases. That means that spatio-temporal data mining algorithms must scale well for the larger size data. However, multivariate spatio-temporal data seldom exhibit linear dependency between variables. Non-linear combination data model using wavelet transformation was carried out by Dong [13] to develop efficient fits for non-linear model. Even though the algorithms discussed earlier are very popular, directly applying them to very large spatio-temporal data would not be effective. One of the underlying reasons for this complexity is that with increase dimensionality, the data points become equi-distant from one another, also known as the curse of dimensionality explored by Aggarwal [2]. The clustering and classification techniques are essentially based on the Euclidian distance. Rather than carrying out operations on the spatio-temporal data as such, data mining operations are carried out aggregated or abstracted data. This abstraction process thus decreases the underlying degree of dimensionality of the data. Traditional data mining algorithms operate under the assumption that the data is independently and identically distributed. However as discussed earlier, these assumptions are often not true for spatio-temporal datasets which is why the classical data mining methods poor ineffectively on them. Nearby data points have a greater degree of similarity and influence the spatial value strongly Roddick [53]. It is important to relax this assumption to quantify the spatial dependence and factor it into the techniques for estimation of missing values. Thus, the existing methods are not substantial for complex multi-variate spatio-temporal data analysis as they are bound by limitations such as slow convergence rate, global and local optima.

5.3 DATA MINING MODEL FOR RESERVOIR OPERATION

Reservoir operation plays an important role in water resources planning and management. The operation of reservoirs usually involves a large number of stakeholders with different objectives, such as irrigational needs, hydropower generation, measures for flood control. The legal and the institutional obligation also have to be taken into consideration. Though research has been carried out to optimize it, numerous reservoir systems are still operated based on expert knowledge, generic rules of thumb established throughout the ages. Mathematical and statistical model were developed to optimize the operations. But the reservoir framework is a delicate balance of trying to balance the need to retain plenty of water for irrigation and other uses of water while preventing an overflow of the reservoir that could cause flooding of the surrounding area. As a result water reservoir operators accumulate a certain set of skills and knowledge that are not easy to express mathematically. These attempts however make numerous assumptions about the systems operations, the environmental variables, the control variables to be optimized and even the optimization objective. Despite the efforts there exists a gap between theory and practice in devising optimal reservoir policies, and this calls for a more systematic data driven model which might prove to be beneficial for many real-time water reservoir frameworks.

To begin with, the data set used for the study is explored and the data collection methods are briefly mentioned. The next step is data preprocessing. Since hydrological data formats are usually complex and incomprehensible, it requires a fair amount of preprocessing so that the classifiers can be applied. The mining methods can now be applied to the dataset to generate data models. Applying the data mining classifiers constitutes the third step. Finally the data models developed are comparatively assessed and evaluated for best fit.

In theory, a data mining algorithm could learn general policies of handling the water reservoirs, and the learned policies could be automated in the future, avoiding occasional errors in human judgment and saving costs in human operators. In practice, the data mining task for water

reservoirs is more complicated than for regular data mining tasks because of the uncertainty involved. For example, the recorded amounts of water released and retained in the water reservoirs are typically uncertain, i.e., they are bounded by some minimum and maximum values. Moreover, the time of release is also uncertain, i.e., typically only monthly or weekly amounts are recorded. Zhen-Li [32] expressed a generalized objective function for deterministic reservoir system optimization as

$$max(or\ min)\sum_{t=1}^{T} f(s_t, r_t) \tag{5.1}$$

where r_t is an n dimensional set of control or decision variables during the time period t, which can be hourly, daily or monthy depending on the target optimization of the reservoir), T is the length if the operational time horizon; $s_t$is an n-dimensional state vector of storage in each reservoir at the beginning of the time period $t : f\ (s_t, r_t)$is the target function that has to be maximized.

5.4 TEMPORAL DATA MINING (PHASE I) CASE STUDY: STANLEY RESERVOIR

The first step in bridging the gap between the theoretical and practical implementation of the reservoir operation is identifying the contributing premises that drive the decision making process in reservoir operators. However, in most previous efforts, the selection of which hydrologic variable to incorporate as a state variable was merely a modeler's choice triggered by either model performance, some expert knowledge, or by simply following the status quo of reservoir optimization models.

The following are the state variables that are taken into considerations [42]:

- **Storage**: The amount of water in the reservoir at a recorded at the beginning of the month.
- **Rainfall**: The amount of precipitation that actually takes place (measured in mm).

- **Inflow**: The actual amount of water that collectively reaches the reservoir after rainfall through any water source.
- **Release / Outflow**: The amount of water that is released from the reservoir.

An important factor that is to be considered for estimating the release is that it satisfies the mass balance equation, which is given below:

$$\textit{Storage in next month } (S_{t+1}) = \textit{Current Storage } (S_t) + \textit{Inflow } (I_t) - \textit{Release } (R_t) - \textit{Evaporation } (E_t) \tag{5.2}$$

In addition, the release is a function of storage, inflow and demand. Therefore,

$$\textit{Release=f (storage, inflow, demand)} \tag{5.3}$$

The above two equations are used sequentially to validate the release data, before using it for data-mining purpose.

5.4.1 Data Source, Data Collection and Preprocessing

For the demonstration of the application of temporal data mining approach for derivation of knowledge rules for reservoir operation, the case study of Stanley Reservoir system of the Cauvery River basin, located in the South part of India, has been considered. In this study, a hydrological reservoir data case study is used to analyze the data models developed by using the data classifiers. Numerous datasets are available but we have considered the case study of the Cauvery River basin where the Stanley Reservoir was constructed. The Cauvery Basin is an Inter-State Basin covering areas in Kerala, Karnataka, Tamil Nadu and Union Territory of Pondicherry. Mettur Dam which creates the Stanley reservoir is one of the largest in India. It is built in a gorge, where the river enters the plains and provides irrigational facility for 5 districts which is about 271,000 acres of farm land. The length of the dam portion in the reservoir is 1,700m (5,600 ft) and it creates the Stanley reservoir. This multipurpose dam was planned to store the high flows during the southwest monsoon and distribute them evenly throughout the irrigation period, thus firming up

the irrigation provided by the canals and also constructing a good hydroelectric power station. The maximum height and width of the Dam are 214 and 171 feet respectively. It is built to have a maximum storage height of 120 feet and a capacity of 93.4 tmc ft (thousand million cubic feet). The reservoir receives water from both Kabini Dam and Krishna Raja Sagara located in Karnataka. The upstream is from Hogenakal falls. The Cauvery River irrigates 1,310 km2 (510 sq mi) of land each year. The maximum percentage of water requirements for irrigation in Tamil Nadu depends on the Stanley Reservoir from the Mettur Dam. It has 2 hydroelectric power stations called Dam and Tunnel Power House each of which a capacity of 40MW and 200MW respectively. Mettur is one of the sources of electricity for Tamil Nadu. The Mettur Thermal Power Station acts as a base load power station for the TamilNadu Electricity Board.

Data Collection : The data pertaining to this reservoir system were collected from the Water Resources Organization (WRO) of the Public works department, Government of Tamilnadu (TN-PWD), which collects and maintains the storage flow record databases on a daily basis. The reservoir data collection was through automated water level recorders installed at 5 salient locations. From the daily data, the monthly storage, inflow, river and canal sluices data and evaporated levels were computed over a period of 12 years (144 months) and the data base was created for this reservoir system. Additionally, the daily rainfall data recorded from the automated rain gauges were also collected and the average rainfall was computed using the Thiessen Weighting method and from the daily rainfall values, monthly rainfall values were then computed and included in the data base for the same 12 years.

Data Preprocessing : The dataset for experimentation was prepared after collating the data observations from two different sources. Though the database recorded the events in the same time interval the start time of the databases were different. The monthly data on reservoir parameters was as 12 year period, from 1998-2010. But the Rainfall data collected from the repository was more recent compared to the reservoir operational data. The database had records beginning from the year 2000. Integrating the data from the two sources for a meaningful dataset resulted in a dataset collected over a period of 10 years; 2000-2010. This resulted in a final reservoir dataset of

Table 5.1: **Data representation depicting the sample instances in the Reservoir**

MONTH	YEAR	RAINFALL(F_t)	INFLOW(I_t)	STORAGE(S_t)	RELEASE(R_t)
Aug	81	251_500	20001_30000	60001_70000	20001_30000
Sep	81	501_more	160001_more	60001_70000	120001_140000

120 instances; wherein it is consistent with all the values and their respective timestamps. It was also investigated for the presence of any missing data. The mean values were used for substitution of missing values. The target variable for the given data mining task is the Release value. The numerical data type is converted to interval data so as to avoid over fitting. The interval data consisted of splitting the numerical data to equi-width bins so that data mining operations can be easily carried out on it.

5.4.2 Data Transformation

However in the data model implemented, the temporal data variation is taken into consideration or in other words, reservoir release modeling is better modeled when history or earlier patterns are taken into account. The standard classifiers mine the relation existing between the data attributes (monthly data in case of the reservoir dataset) attributes to the target attribute. So in this case, the data mining algorithm mines the relationship between reservoir variables –Inflow (I_t), Initial Storage (S_t), Rainfall (F_t) and the Release(R_t) is the target variable. Thus the functional relationship can be expressed in the form:

$$R_t = f(F_t, I_t, S_t) \tag{5.4}$$

A sample illustration of the data used for the data mining process using standard classifier is given in Table 5.1. Considering each month as separate instance would result in missing out all the details that can be harvested if the data pertaining to earlier months are also considered in predictions are included. Temporal data essentially aims to harness or identify future release values based previous values. Consider the following for an elaborate usage of the temporal data. In case a given month's release was as per the stated value, a sudden rainfall would cause an

increase in the current storage value which would now impact the release value of subsequent months.

If standard classifiers are used then the release values predicted might not take into account this event and the release quantity might pose a threat to the watershed area. So in case of the temporal data, the release function can now be expressed as a function of previous and current periods of reservoir variables and can be represented as

$$R_t = f\left\{\left(F_{t-n}, I_{t-n}, S_{t-n},, R_{t-n},.F_{t-1}, I_{t-1}, S_{t-1}, R_{t-1}\right),\ \left(F_t, I_t, S_t\right)\right\} \tag{5.5}$$

The final aggregated ARFF file has the Month, Rainfall (in mm), inflow (in Mcft) and Storage (in Mcft)as Input and Release (in Mcft) as output. The attribute year is not included in the temporal data as the value of the year does not by any means affect the release.

5.4.3 Experimentation

Now that transactional dataset is created experiments are carried out on it. Mining has been carried out in JAVA using the WEKA API. (WEKA 3.6.9 [21]) The input dataset is supplied as a ARFF or CSV file. The data is split into two parts; Training and Testing data. In the training phase, the classifier is learns the data operation with a training data set. The training data set is used to train the model (to determine the optimal parameters set). If the models performance on the testing dataset is satisfactory then it can be put to operation. The training can be repeated until accuracy of the classifier reaches the minimum acceptable limit. In this experimentation, we have used 70% of the data for training the model and 30% of the data to test the effectiveness of the model. The extent to which this the classifier learns about the model is put to test in the testing phase. Here the classifier is used to predict the value of the target variable. The predicted value is them compared with the expected value to testify the efficiency of the data model constructed. This training and testing is carried out using all of the classifiers chosen for the data and the resulting rules are compared for efficiency.

5.5 SPATIO-TEMPORAL DATA MINING (PHASE II) CASE STUDY: NORTH PLATTE RIVER SYSTEM

5.5.1 Data Source, Data Collection and Preprocessing

The spatial data mining and spatio-temporal data mining models developed in this study have been demonstrated for its successful application with a case study of Multi-Reservoir system namely, North Platte River System in the Colorado river basin of USA.

The **North Platte River** is a major tributary of the Platte River. The Platte is actually 310 miles (500 km) long, but measured from its source stream, Grizzly Creek in Colorado (via the North Platte River), the system has a length of 990 miles (1,590 km) including the curve paths that the river takes. **The Platte River,** river of Nebraska is formed at the city of North Platte by the confluence of the North Platte and South Platte rivers. It then flows southeast into a big bend at Kearney, curves northeast, and travels east, south, and finally east before emptying into the Missouri River at Plattsmouth which is about 20 miles (32 km) south of Omaha from where it finally joins the Mississippi River to flow to the Gulf of Mexico. Other cities covered by the river along its path are Lexington, Grand Island, Columbus, and Fremont. The river is extremely thus precluding navigation. The origin of the river is essentially all of Jackson County, Colorado which is bound by the continental divide, the mountain drainage peaks and the Wyoming state boundary. Here, the North Platte is joined by several small creeks which originate from the snow covered mountains. Some of the creeks include Arapaho Creek, Colorado Creek, East Branch Illinois River, Jack Creek, Jewell Lake Trib., Grizzly Creek, North Fork of North Platte River, and the Encampment River in Wyoming. The river is narrower and flows faster in Colorado and Wyoming than in Nebraska. This makes the upper reaches of the river particularly good for recreational activities like rafting, canoeing and fly fishing. The Northgate Canyon is one the popular rafting sites in the river before it enters Wyoming. The river has been extensively dammed along its course.

On the north end, joined by the medicine bow is the Seminoe Dam which creates the **Seminoe Reservoir.** Approximately 70 to 80 percent of the annual North Platte River streamflow above

Seminoe Dam occurs from snowmelt runoff during the April -July period. Primary water demand is irrigation, and the period of delivery of irrigation water normally extends from May through September. The System furnishes irrigation water to over 440,000 acres of land in Wyoming and Nebraska. The Seminoe Reservoir is followed by the **Kortes Reservoir.** Kortes Reservoir provides a total storage capacity of 4,739 AF at elevation 6142.0 feet which is the level of the spillway crest. Still further downstream in about 50 miles the Sweetwater River joins the North Platte and forms the **Pathfinder Reservoir.** Pathfinder Dam and Reservoir, a major storage facility of the North Platte Project, has a total capacity of 1,016,507 AF at elevation of 5850.10 feet. Operationally, this structure is a bottleneck in the System with its maximum non-spillway release capability of approximately 6,000 cfs. The rated capacity of the left abutment outlet works through the two 60-inch jet flow gates is 2,928 cfs at elevation 5850.10 feet. It then flows northeast through Alcova and the Gray Reef reservoir before reaching Casper and flowing east-southeast therein into the Great Plain. It then passes through the **Glendo reservoir** which is a multiple-purpose natural resource development. It consists of Glendo Dam, Reservoir, and Powerplant; Fremont Canyon Powerplant; and Gray Reef Dam and Reservoir which is a re-regulating reservoir immediately downstream of Alcova Dam. Glendo Dam and Reservoir is the only storage facility for the Glendo Unit. The reservoir has a storage capacity of 789,402 AF, including 271,917 AF allocated to flood control. The Powerplant consists of 2 electrical generating units, with a total installed capacity of 38 MW. With both generating units operating at capacity and the reservoir water surface at elevation 4635.0 feet, approximately 3,920 cfs can be released through Glendo Powerplant. Guernsey Dam located about 25 miles below Glendo Dam, again stores and reregulates the flow of the river prior to entering the **Guernsey Reservoirs** into the Laramie River and finally crossed into western Nebraska where it merges with the South Platte to form the Platte River. The Platte flows in a large arc, east-southeast to near Fort Kearny and then east-northeast, across Nebraska south of Grand Island and on to Columbus. It then goes through Fremont, around Omaha, Waterloo and then finally turns south to join the Missouri river.

Data Collection : Out of the reservoirs discussed, the Seminoe Reservoir, the Pathfinder Reservoir

and the Glendo reservoir are the major contributors and the case study employs these major reservoirs. We have considered the major contributing reservoirs for the case study namely; Seminoe, Pathfinder and Glendo. The water levels are recorded both in upstream and the reservoir catchments using telemetric recorders situated at strategic locations along the reservoirs. The North Platte River basin data for this study was collected from the U.S Department of Interior (DoI), Bureau of Reclamation. The data was collected individually for each of the reservoirs over a period of 11 years and 5 months. The Inflow, Storage, Release and Rainfall data were collected in monthly intervals from Jan 2000 to May 2011.

Data Preprocessing : The Data collected was imported into Comma Separated Values (CSV) file format and sorted based on the date on which the data was recorded. The following are the state variables that are taken into consideration:

- **Initial storage**: The amount of water in the reservoir at a recorded at the beginning of the month.
- **End Storage**: The amount of water in the reservoir at a recorded at the end of the month.
- **Inflow**: The actual amount of water that reaches the reservoir after rainfall through any water source.
- **Release / Outflow**: The amount of water that is released from the reservoir.

An important factor that is to be considered for estimating the release is that it satisfies the mass balance equation, which is given below:

$$\textit{Storage in next month } (S_{t+1}) = \textit{Current Storage } (S_t) + \textit{Inflow } (I_t) - \textit{Release } (R_t) \tag{5.6}$$

The standard classifiers mine the relation existing between the data attributes (monthly data in case of the reservoir dataset) attributes to the target attribute. So in this case, the data mining algorithm mines the relationship between reservoir variables –Inflow (F_t), Initial Storage (I_t), End Storage (S_t) and the Release(R_t) is the target variable. Thus the functional relationship can be expressed in the form:

$$R_t = f(F_t, I_t, S_t) \tag{5.7}$$

5.5.2 Data Transformation

All of the involving reservoirs collectively influenced each of the reservoir release or downstream values i.e. the storage value of the Glendo reservoir at a given time is very much influenced by the input and the output variables of the Seminoe reservoir upstream and the Pathfinder reservoir downstream but not necessarily to the same degree. So devising operational rules about a given object would require information about the other data objects and their value in the spatial framework. For example, the spatial data (for the Glendo Reservoir) is given by,
*Release value of Glendo (**Gl**_R_t)=*

$$f\{Year, Month, \mathbf{Sm}\_F_t, \mathbf{Sm}\_I_t, \mathbf{Sm}\_S_t, \mathbf{Sm}\_R_t, \mathbf{Pf}\_F_t, \mathbf{Pf}\_I_t, \mathbf{Pf}\_S_t, \mathbf{Pf}\_R_t, \mathbf{Gl}\_F_t, \mathbf{Gl}\_I_t, \mathbf{Gl}\_S_t\} \quad (5.8)$$

Considering each month as separate instance would result in missing out all the details that can be harvested if the data pertaining to earlier months are also considered in predictions are included. Thus the spatio-temporal data model wherein the previous months water levels: *(t-1),(t-2) and (t-3)* are supplied as input along with the current water levels *(t)*. Temporal data essentially aims to harness or identify future release values based previous values. Consider the following for an elaborate usage of the temporal data. In case a given month's release was as per the stated value, a sudden rainfall would cause an increase in the current storage value which would now impact the release value of subsequent months. If standard classifiers are used then the release values predicted might not take into account this event and the release quantity might pose a threat to the watershed area. So in case of the temporal data, the release function can now be expressed as a function of previous and current periods of reservoir variables and can be represented as:

$$R_t = f\left\{\left(F_{t-n}, I_{t-n}, S_{t-n}, , R_{t-n}, .F_{t-1}, I_{t-1}, S_{t-1}, R_{t-1}\right),\ \left(F_t, I_t, S_t\right)\right\} \quad (5.9)$$

So the final spatio-temporal dataset incorporates the temporal information (as given in eq.(5.8) about the reservoir value for each of the spatially distributed reservoirs eq. (5.9).Now that transactional dataset is created experiments are carried out on it.

5.5.3 Experimentation

Once the spatial and spatio-temporal dataset are created, Mining has been carried out in JAVA using the WEKA API. The SMOReg algorithm for Support Vector Regression (SVR) classifier is applied for both the spatial and spatio-temporal dataset. The dataset is split into two parts as discussed earlier; training and testing. This experimentation uses 70% of the data for training the model and 30% of the data to test the performance effectiveness of the model. The role of the classifier is to learn the data model by creating a mapping between the reservoir parameters and help in deriving the target output. The estimated parameters or the efficiency of pattern matching is evaluated in testing phase. The testing dataset must not contain patterns from the training dataset. The parameters of the data mining algorithm are updated until testing phase maps the pattern more the minimum acceptable limit. The performance of the quadratic kernel function is compared with the RBF kernel, the Pearson Universal kernel and the Normalized poly kernel. The aim of this is to identify which kernel function gives the best fit for the given reservoir model. The training and testing is carried out for both the spatial and spatio-temporal datasets and the reservoir release rules generated are evaluated for efficiency.

5.5.4 Reduced Feature Support Vector Regression (RF-SVR) Spatio-Temporal Data model

The Spatio-temporal data employed contains *(t)*, *(t-1)*,*(t-2)* and *(t-3)* values of all of the state variables for determining the release value of each of the reservoir. So the Spatio-temporal data essentially can easily amount to an n-dimensional dataset. Therefore, Principal Component Analysis has been carried out on the spatio-temporal data to reduce the dimensionality to a much lower number to enhance execution time during the training and testing phase. We have employed MATLAB to generate the PCA components for the spatio-temporal data. Depending on the value of the explained variance, the reduced dimension dataset is generated. This is consequently tested with the SVR model as earlier.

5.6 PERFORMANCE EVALUATION MEASURES

5.6.1 Temporal Data Model

For the temporal data mining model we have measured the performance efficiency of the classifier using the Root Mean Square (RMS) error value and the number of Correctly Classified Instances. The root mean squared error value is a measure of the difference of the values predicted by a model and the values actually observed from the environment. We have implemented the RMS error calculations as follows [42]:

Let $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$,....,$\mathbf{a}_n$ be the set of actual values measured as readings.Since it is an interval data, the mode of this interval value is taken as the predicted value **p.** The set of predicted values is now given by $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$,...., $\mathbf{p}_n$. There exists Total of N_c classes and the classifier is trained using **n** instances. Then RMSE is given by:

$$RMS\ error\ = \sqrt{\frac{\sum_{i=1}^{n} (a_i\ - p_i\ )^2}{N_c - 1}} \tag{5.10}$$

5.6.2 Spatio-Temporal Data Model

The performance efficiency of the classifier is measured using the Correlation Coefficient, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) value. **The Root Mean Square error** value is a measure of the difference of the values predicted by a model and the values actually observed by the environment. Let $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$,....,$\mathbf{a}_n$ be the set of actual values measured as readings. $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$,...., $\mathbf{p}_n$ is the set of predicted values. There exist a total of N_c classes and the classifier is trained using **n** instances. The RMS value is calculated as follows:

$$RMS\ error\ = \sqrt{\frac{\sum_{i=1}^{n} (a_i - p_i)^2}{N_c - 1}} \tag{5.11}$$

This aggregated RMS error value is the measure of the degree of performance, i.e. the degree to which the classifier correctly predicts the required value.

Correlation Coefficient is measure of the degree to which one variable is related to another can be quantified by means of correlation coefficient. For a given set of observations given by (x_1, y_1), (x_2,y_2),...(x_n,y_n), the correlation coefficient is calculated as follows:

$$r_{xy} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}} \tag{5.12}$$

Where $\bar{x}$ and $\bar{y}$ are the mean values of x_i and y_i. The correlation coefficient values range from -1 to 1 therefore might exhibit either a positive or a negative correlation value.

The Mean Absolute Error gives the absolute value of the difference between the original and the predicted values and can be given by the expression:

$$MAE = \frac{1}{n} |y_i - a_i| \tag{5.13}$$

Chapter 6

RESULTS AND DISCUSSION

For each of the methods, data preprocessing results are described first and which is then followed by the results of data mining. The results are covered in separate sections for temporal and spatio-temporal data. Individual findings are discussed in each section and a general discussion concludes this chapter.

6.1 PHASE I: TEMPORAL DATA

The collation of the reservoir and rainfall datasets resulted in a considerable decrease of data records that were applicable for the study. Integrating the data from the two sources resulted in a dataset collected over a period of 10 years. The resulting dataset had 120 instances which were consistent in all the values and the timestamps beginning from 1999 to 2009. A sample of the data set created is shown in Table 6.1. Considering each month as separate instances would mean missing out on all the details that can be harvested if the data pertaining to earlier months are

Table 6.1: **Data representation depicting the sample instances in the Reservoir**

MONTH	YEAR	RAINFALL(F_t)	INFLOW(I_t)	STORAGE(S_t)	RELEASE(R_t)
Aug	81	251_500	20001_30000	60001_70000	20001_30000
Sep	81	501_more	160001_more	60001_70000	120001_140000

also included for predictions. Temporal data mining aims to identify future release values based several previous values. For temporal classification, the original data was modified to include time base information, more specifically, the old values of the reservoir features up to some i time units back in time in addition to the current values. Consider the following motivational example for the use of temporal data. Suppose the current months storage, rainfall and inflow were usual, but the previous months release was low. Then the current release should be higher than average to provide enough water down the river. Temporal data mining can capture this scenario, but regular data mining, which only looks at the current values, could lead to less than optimal result. Therefore, the temporal data looked back two additional months beside the current month. The experimental results of regular and temporal data classification were compared. As already mentioned, the data was split into two sets; the training data and the testing data. The training data were used to obtain some classifier. The classifiers performance was evaluated on the testing data. For the experimentation, 70 % of the available data is used for training and 30% for testing. The following steps were carried out in the experiments to predict the Release values R.

1. A set of 36 records is chosen randomly as the training set.
2. The remaining 70% of the dataset is chosen as test set.
3. Data models using Nave- Bayes classifiers, MLP and Decision tree is built using the training data.
4. The accuracy of the classification is tested on the testing set.

The average results of repeating the above procedure 5 times for n equal to 10, 25, 40, 50, 55, .., 70 using the Nave Bayes classifier is reported in Figure 6.1 Here n is the training set size control parameter given in percentage.

Similarly, Figure 6.2 and Figure 6.3 report the average results with the Multilayer Perceptron and the Decision Tree classifier respectively. The experiments show that adding time based information has significantly improved the reservoir release predictions using all the three- Nave Bayes, Multilayer Perceptron and the Decision Tree classifier. All the three data models indicate a

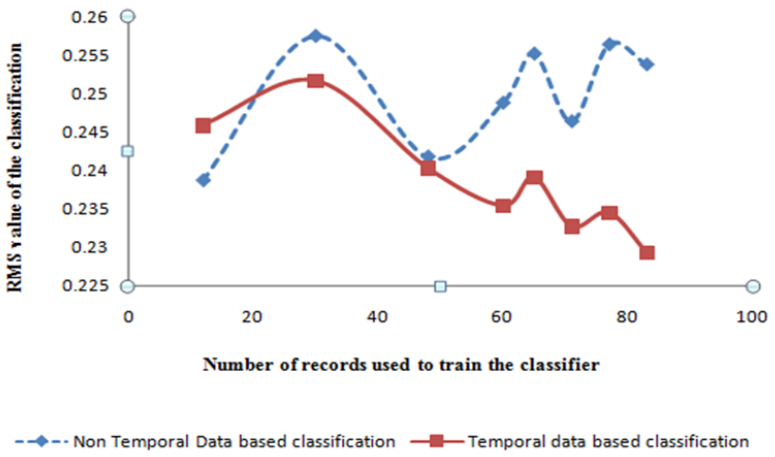


Figure 6.1: **Comparison of regular and temporal classification using the Nave Bayes classifier**

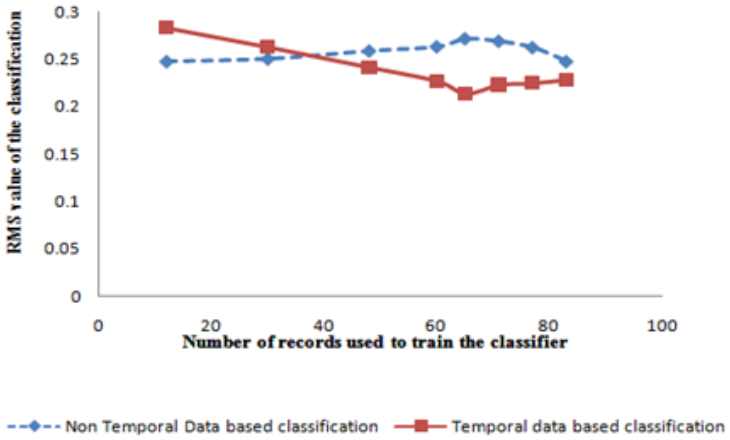


Figure 6.2: **Comparison of regular and temporal classification using the Multilayer Perceptron classifier**

visible decrease in the error values of temporal data compared to the regular data.

Though the regular data presents a comparatively lower RMS error to begin with, this cannot be taken as a valid state to measure the performance, because only very few percentages of data (10, 25) were considered. This is of significance; because the dataset consists of 120 instances and a model built using 10% of it i.e. 12 instances cannot be substantiated. However, upon further

Figure 6.3: **Comparison of regular and temporal classification using the Decision Tree classifier**

Table 6.2: **Performance Measures of Different Classifier Methods**

Classifier method	*Total No of Instances to be classified*	*Correctly Classified instances*	*Root mean square error*
Naïve bayes (Training)	84	70	0.121
Naïve bayes (Testing)	36	13	0.2330
Multilayer Perceptron (Training)	84	74	0.1138
Multilayer Perceptron (Testing)	36	15	0.2284
Decision trees (Training)	84	82	0.0088
Decision trees (Testing)	36	10	0.2548

training and modeling, the temporal data does display positive results in terms of decreased RMS error values.

Also interestingly all the classifiers, display an improvement, strongly backing up the theory that the history of the reservoir data does contain important information to help design a valid data mining model for the prediction of reservoir release. The performance of each of the classifier for the regular and the temporal data with the default 70-30 ratio is listed in Table 6.2. This default ratio is often used to state that 70 percent of the source data is used for training the data model and 30 percent is used for testing it.

6.2 PHASE II: SPATIO-TEMPORAL DATA MINING

Numerical experiments have been designed to generate operational rules that combine storage and release targets for multi-reservoir frameworks by assessing the reservoirs and the predictors for configuring operational rules. When working with a framework of several reservoirs in which there are no loops, the water reaching a reservoir downstream referred to as the feeder reservoir is strongly dependent on the release values of the set of all reservoirs upstream thus iterating the strong degree of auto correlation when working with spatial data.

6.2.1 Time Lag Selection

Temporal data mining predicts the behavior of a given event based on the patterns and observation in the past. When employing data mining algorithms to the temporal environment, the amount of information is critical. A small lag frame would contain too little data whereas considering a very large lag would results in inclusion of unwanted input variables which would consequently result in overfitting. Performance evaluation is thus carried out so as to choose the effective time lag so that the data model developed is succinct and efficient. The various lag instances; lag1*(t-1, t)*, lag2 *(t-2, t-1, t)* and lag3*(t-3, t-2, t-1, t)* and lag4*(t-4, t-3,..,t)* are evaluated for each of the reservoir datasets. For all reservoirs,best results were reported for lag3 wherein data upto (t-3) units back in time were considered. We have therefore considered data instances upto t-3 units back in time for our subsequent spatio-temporal model. The performance evaluation of a sample reservoir (Glendo) is reported in Table 6.3. The graphical illustration of the effectiveness of the lag3 fit for the glengo reservoir is shown in Figure 6.4.

Table 6.3: **Data Mining model results with various lags for the Glendo Reservoir**

Description	Correlation Coefficient	Mean Absolute Error	Root Mean Squared Error
Lag1	0.8843	24427.6	30695.8
Lag2	0.9663	11635.8	16760.9
Lag3	0.9696	11096.4	17836.6
Lag4	0.9085	22589.4	27577.6
Lag1 with PCA	0.9341	20160.8	23481.9
Lag2 with PCA	0.916	20591.2	28402.0
Lag3 with PCA	0.962	14554.6	21069.9
Lag4 with PCA	0.9221	19433.4	24385.3

Table 6.4: **Performance comparison of the Spatial and Spatio-temporal data for the SVR data model**

RESERVOIR	SMOREG KERNEL	SPATIAL			SPATIO-TEMPORAL		
		Correlation Coefficient	Mean Absolute Error	Root Mean Squared Error	Correlation Coefficient	Mean Absolute Error	Root Mean Squared Error
Seminoe	**Quadratic Kernel**	0.916	10696.3	24034.3	0.9879	3416.1	4179.7
	Normalized Polynomial Kernel	0.816	23399.1	44548.9	0.9125	13379.4	33731
	PUK	0.486	29385.8	54942.3	0.519	25541.1	53012.6
	RBF Kernel	0.777	24588.4	46694.3	0.871	19588.4	33694.3
Pathfinder	**Quadratic Kernel**	0.9088	12691.9	13250.7	0.9669	6489.7	12553.5
	Normalized Polynomial Kernel	0.8466	17695.9	25191.8	0.8734	12544.8	22703.5
	PUK	0.516	22759.4	38825.2	0.6102	18104	35799.2
	RBF Kernel	0.8559	16940.3	23936.9	0.9387	10934.2	15895.8
Glendo	**Quadratic Kernel**	0.9426	17308.9	27705.3	0.963	13440.7	20973.1
	Normalized Polynomial Kernel	0.9483	17667.3	28357	0.925	19561.3	29833.4
	PUK	0.7756	36207.3	49579.5	0.746	34756.4	49702.9
	RBF Kernel	0.938	19215.6	29659.9	0.97	10008.9	15976.3

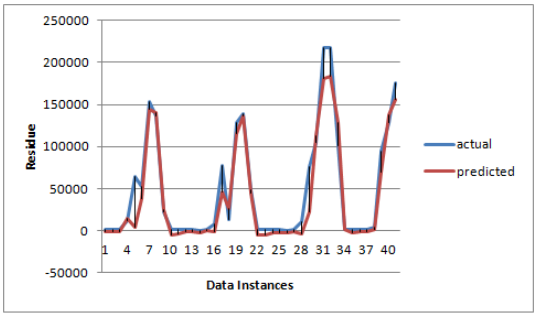


Figure 6.4: **Performance efficiency of Glendo Reservoir for lag3 dataset**

6.2.2 Performance Analysis of Support Vector Regression Spatio-temporal data model

Figures 6.5 (a)-(h) demonstrate the data trend between the actual and predicted value for the Seminoe reservoir. It can be concluded from Table 6.4 that there is a considerable decrease in the degree of deviation in the Spatio-Temporal dataset in comparison with the Spatial dataset for all the kernel functions (Normalized Quadratic, RBF and the Pearson universal Kernel).

For the spatio-temporal model, the predicted data trend completely overlaps with the actual data trend in comparison with the spatial data model ((a)-(b)). Table 6.4 shows that the Quadratic kernel displays the least prediction errors for the Seminoe reservoir with the reported correlation coefficient value as high as 0.9879 and an RMSE value of 3416.91, meaning that it can easily map the input features to result in useful prediction rules. Next reservoir along the course is the Pathfinder and the data trends for spatial and spatio-temporal data models are given in Figure 6.5 (a)-(h).The spatio-temporal data model using the Quadratic kernel again demonstrates an improvement. An improved correlation coefficient value of 0.9669, a low MAE value of 6489.7 and RMSE value of 12553.5 establish the Spatio-temporal model employing the quadratic kernel as a good fit (Figure 6.6 (a)-(b)). Employing Quadratic kernel in the SVR data model for the Glendo reservoir reports an improvement in performance; increase in correlation coefficient value from 0.9426 to 0.963 and decrease in MAE value from 17308.9 to 13440.7. However, the best fit is reported for using the RBF kernel with a correlation coefficient of 0.97 and an MAE value of

10008.9 thus demonstrating improvements with Spatio-Temporal over Spatial Data. Figures 6.7 (g)-(h) illustrate the drastic improvement of Spatio-temporal Data over Spatial Data when the RBF kernel is employed. It can be concluded that the Quadratic Kernel works well when the variance is comparatively high whereas the RBF kernel suits well when the variance is not that much pronounced. However, this point needs to be validated with data pertaining to other reservoir systems.

The fact that Quadratic kernel gives a better fit for all the Seminoe and Pathfinder reservoirs indicates that the true function of the input data vectors can be well approximated by that of a quadratic function. This highlights the underlying simple quadratic (rather than a complex logarithmic or exponential) relationship between the inflow, initial storage, and storage and the release values of each of the reservoirs. Therefore, a parametric Quadratic model can be concluded as the best fit for predictive mining for Seminoe and Pathfinder reservoirs.

However, there exists a more complex relationship between the input vectors for the Glendo Reservoir. Here the model developed with RBF kernel provides the best fit for predicting the reservoir release operation. This may be due to the fact that the RBF kernel works well for the normalization of the data taking into account the fact that the variability can be reduced by the RBF Kernel function which employs the logarithmic or exponential transformation of the data and resulting data pertains to normal distribution. Also, another interesting observation is that there is a high level of compliance between the predicted and the actual values when the release values were typically high and a good compliance between the predicted and the actual when the demand-meeting release values were relatively low. These results show that Support Vector Regression model has performed very well on Spatio-temporal data.

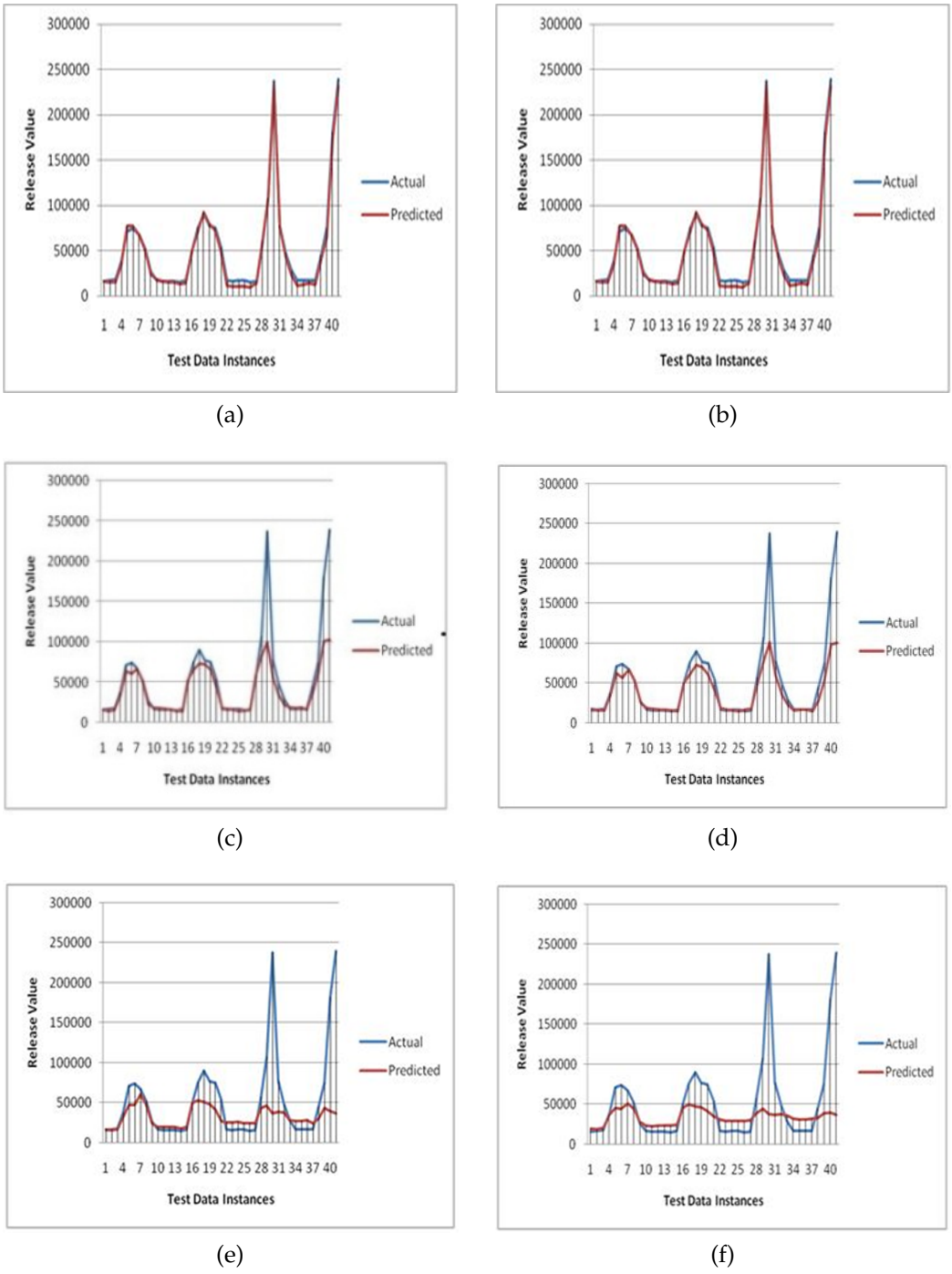
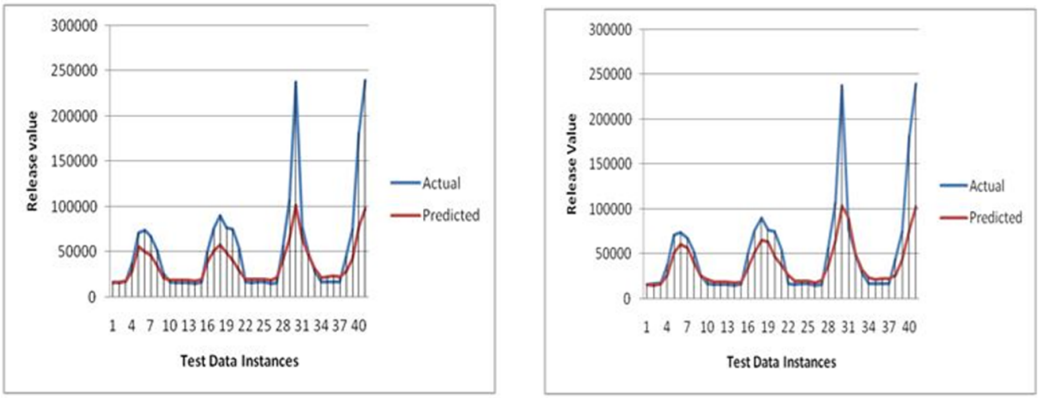


Figure 6.5: **Performance comparison for the Seminoe Reservoir**



(g) (h)

Figure 6.5: **Performance comparison for the Seminoe Reservoir**

6.2.3 Performance Analysis of Reduced Feature Support Vector Regression (RF-SVR)

The input data (CSV Worksheet) consists of 138 instances. By the principle of parsimony, PCA essentially reduces the dimensionality by extracting the smallest number of components that account for most variability of the original dataset and 39 attributes is read into MATLAB. The resulting is the set of PCA components. The aim of PCA is to explain as much of the variance of the observed variables as possible using few composite variables (usually referred to as components).For all of the reservoir data, a total of 39 Principal components were generated. If 100% of the variance in the correlation matrix was to be accounted for, then all of the 39 components would need to be retained. However, this would lead to over fitting and would be counter-productive. This prompted the use of explained variance [56]. The performance of the data model for variance between 0.95 and 0.98 is reported for evaluation. We chose a set of 4 and 6 principal components for variances of 0.95 and 0.98, respectively. The performance of the data model for each of the reservoirs using the PCA data model is reported in Table 6.5.

For the Seminoe reservoir, since the Quadratic kernel gives better results we may say that the best performance (MAE of 12348.3 and correlation coefficient of 0.93) was achieved when variance is 0.95. For the Pathfinder reservoir also the same conclusion can be derived and the best

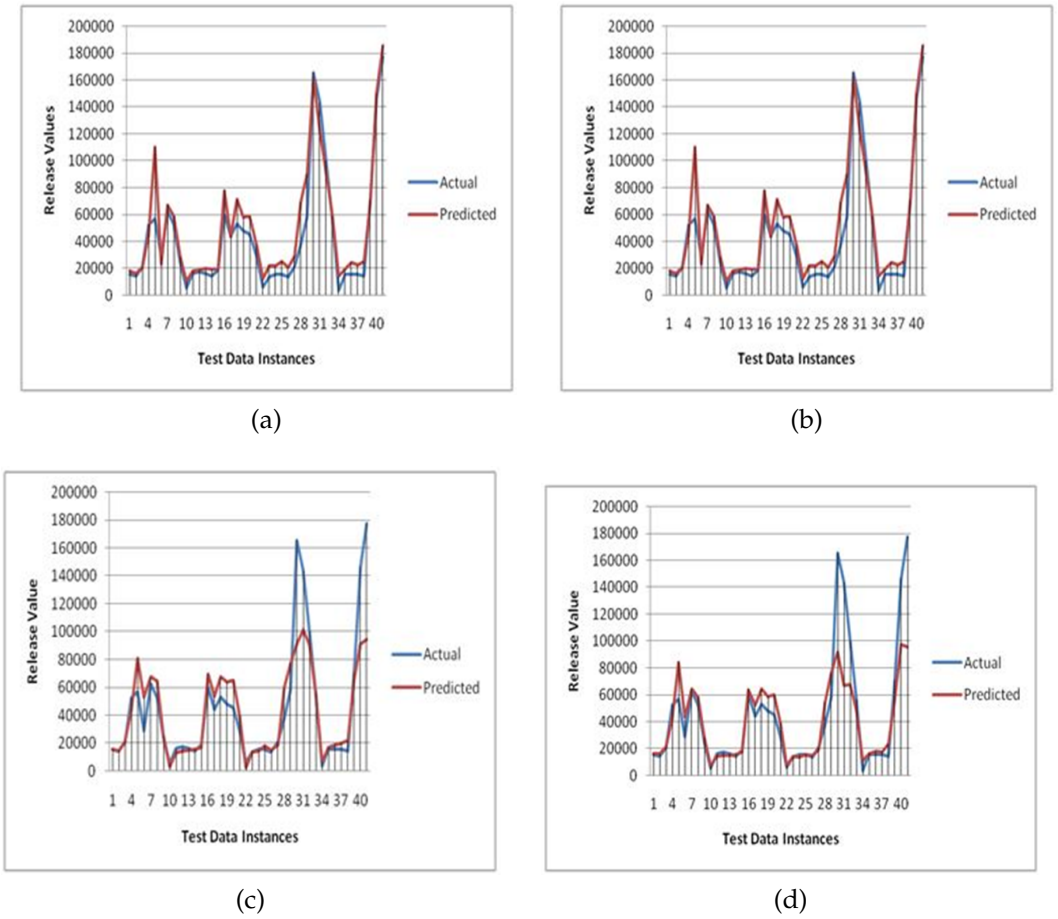
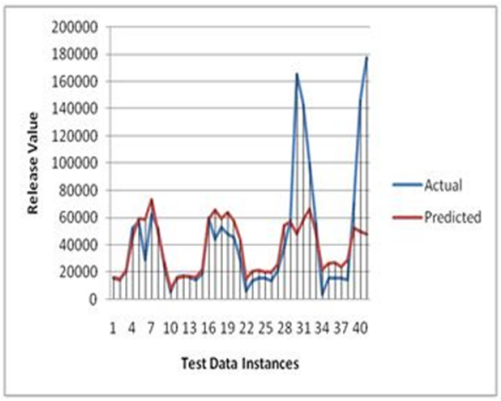


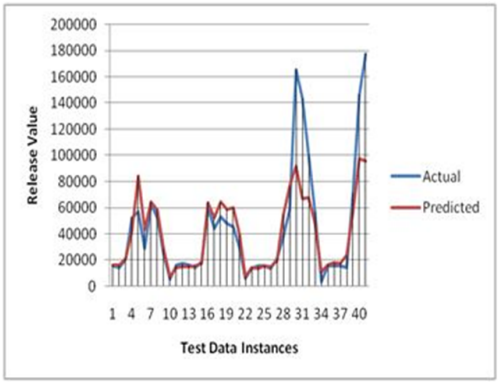
Figure 6.6: **Performance comparison for the Pathfinder Reservoir**

performance was achieved when variance is 0.95.

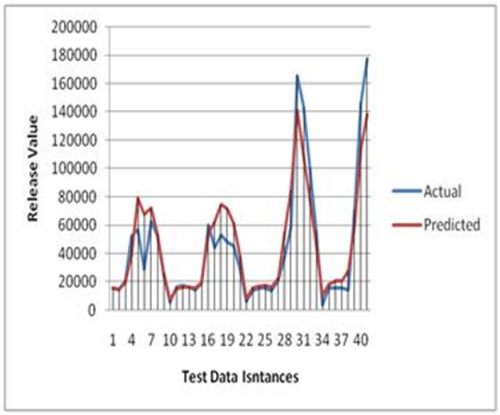
The Glendo reservoir reports best results when the variance is 0.98 and quadratic kernel function reports the highest efficiency with a MAE of 16270.6 and a correlation coefficient of 0.9456. Figure 6.8 gives trend lines of the actual (blue) and the predicted (red) values with the best RF-SVR model for each of the reservoirs. Since the PCA components generates a minimal set of components that are most descriptive, the Quadratic kernel function is sufficient to model the reservoir operation in RF-SVR Spatio-Temporal model as opposed to a RBF kernel required to capture the relationship in the high dimensional Spatio-Temporal data.



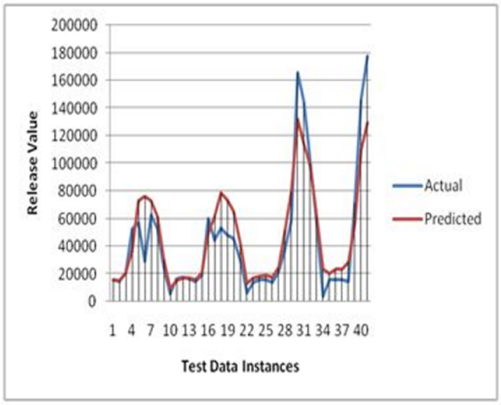
(e)



(f)

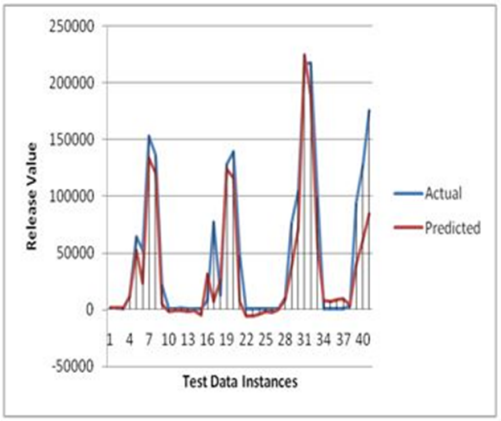


(g)

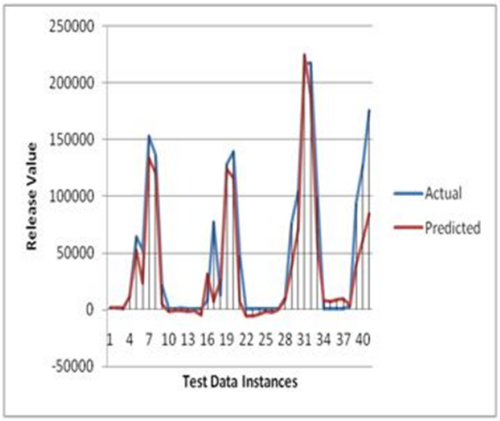


(h)

Figure 6.6: **Performance comparison for the Pathfinder Reservoir**



(a)



(b)

Figure 6.7: **Performance comparison for the Glendo Reservoir**

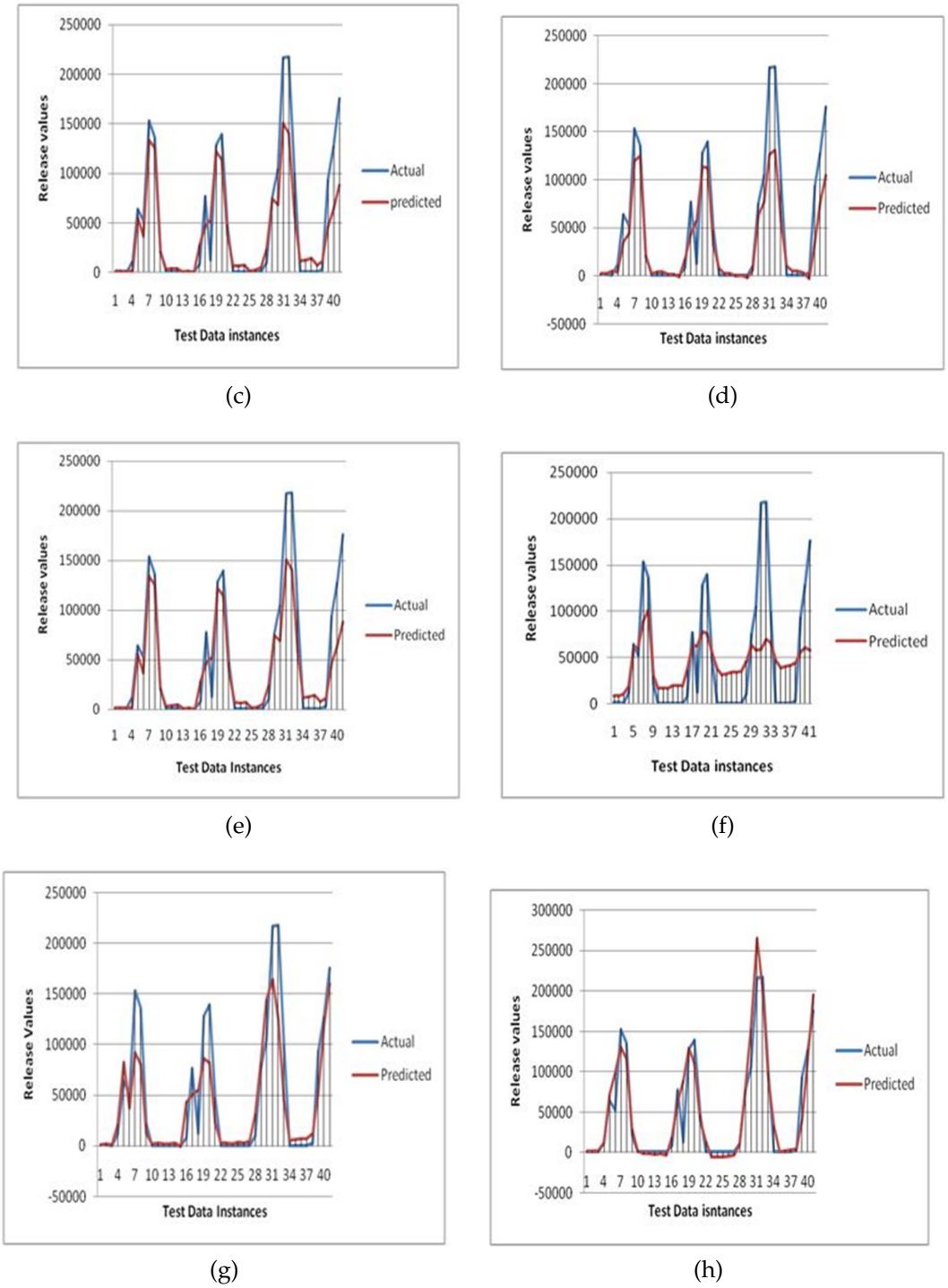
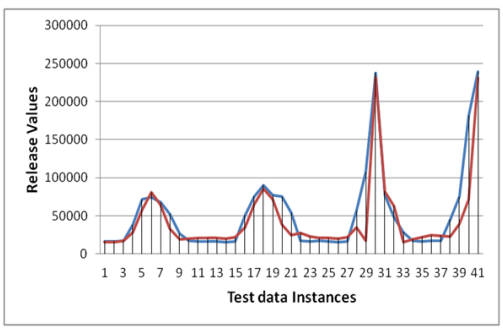


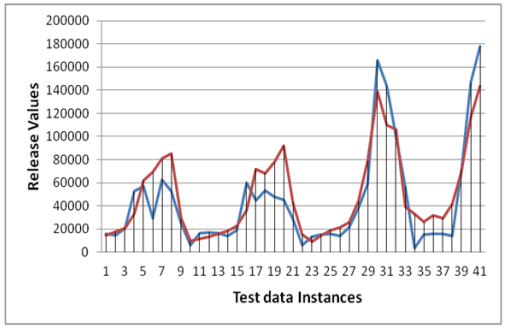
Figure 6.7: **Performance comparison for the Glendo Reservoir**

Table 6.5: **Performance of RF-SVR spatio-temporal data model for each of the reservoirs**

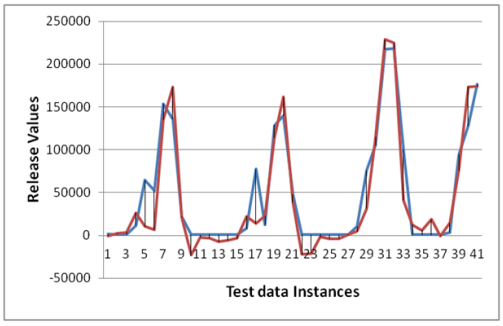
RESERVOIR	SMOREG KERNEL	VARIANCE =0.95			VARIANCE=0.98		
		Correlation Coefficient	Mean Absolute Error	Root Mean Squared Error	Correlation Coefficient	Mean Absolute Error	Root Mean Squared Error
Seminoe	Quadratic Kernel	0.93	12348.3	22475.2	0.779	19217.6	35823.9
	Normalized Polynomial Kernel	0.708	21648	44098.6	0.76	21775.8	43615.9
	PUK	0.553	24908	51750.3	0.557	26549.3	52951.4
	RBF Kernel	0.62	32430.2	57858.9	0.67	32487.5	5790.8
Pathfinder	Quadratic Kernel	0.924	13752.4	19376.7	0.879	17621.9	27549.4
	Normalized Polynomial Kernel	0.843	15039.9	25476.2	0.802	21803.3	27898.2
	PUK	0.562	20250.5	36768	0.561	23803.3	37257.3
	RBF Kernel	0.652	23899.5	41717.8	0.655	23815.2	41844.4
Glendo	Quadratic Kernel	0.896	23994.8	35959.9	0.9456	16270.6	23497.3
	Normalized Polynomial Kernel	0.9209	22204.9	32100.3	0.8996	18537.7	29696.7
	PUK	0.8124	27599.6	38180.3	0.7985	31592.9	46953.1
	RBF Kernel	0.8667	38364	61406.9	0.8608	38125.5	61332.2



(a)



(b)



(c)

Figure 6.8: **Performance of the RF-SVR model with Quadratic Kernel**

Chapter 7

CONCLUSIONS AND FUTURE WORK

7.1 CONCLUSIONS

As was stated in chapter 1, the major goal of this thesis was to device an effective spatio-temporal data mining model in a data short multivariate environment and to check the applicability of both temporal data mining and spatio-temporal data mining algorithms for mining the operation rules for reservoir operation both in case of single reservoir and multi-reservoir systems. It was discovered that in terms of performance, best results for temporal data was reported by the Multi layer perceptron classifier.

It was also found that number of instances and number of attributes of the data sets do not have strong influence on the performance of the data mining algorithms as high accuracy of prediction was observed in data-short environment as well with large data sets. Employing PCA for the spatio-temporal datset resulted in a drastic- nearly 8 fold decrease in the input feature vector size (From 42 to 5). Based on the results, it can be concluded that the converting the data from a high dimensional to one with few attributes has indeed resulted in an effective model. The RF-SVR model performs much better that the spatial data model in general.

In comparison spatio-temporal SVR model the spatio-temporal RF-SVR model reports slightly lower performance but above average accuracy. The SVR data model however employs a relatively

large dimensional multivariate data. When employing the SVR spatio-temporal data model for application with more spatially distributed data points (in this case more number of reservoirs), it might result in over-fitting. This might also reflect on the execution time that might cause a bottleneck. However, it is to be noted that the training time for the SVR data model was considerably high - upto 10 fold in comparison with the RF-SVR data model (SVR- 132.9 secs and RF-SVR- 11.3 secs). This could also result int the RF-SVR data model being more efficient when extending it to larger number of instances. It is this tradeoff between accuracy, efficiency and speed that makes the RF-SVR a better model for a spatio-temporal application.

Multivariate spatio-temporal applications have numerous interactions which must be efficiently modeled for a fully functional data model. In this case study, decision on reservoir water release is crucial for efficient reservoir operation, both in single and multi-reservoir systems. In addition, both the spatial and the temporal relationships between the data cannot be easily identified. In this study Support Vector Regression based data mining technique that employed a Quadratic kernel function is applied. The model performed better on a high dimensional spatio-temporal data model in comparison with a spatial model. The method was further enhanced by employing a Reduced Feature-Support Vector Regression model that employed Principal component analysis. The best kernel function and the Variance value suitable for a particular reservoir was then carefully determined. For most cases, the Quadratic kernel function was able to perfectly capture the variable dependency especially in the SVR model except for one case where the RBF kernel reported better results. In case of the RF-SVR model, the Quadratic kernel function proved to be efficient in capturing the relationship to predict the release values. The evaluation results of the RF-SVR data model suggests that dimension reduction with PCA does not drastically change (decrease) the efficiency of a SVR based data model. Though it has some tradeoffs, the RF-SVR proves to be a frontrunner algorithm for model spatio-temporal application modeling.

7.2 FUTURE WORK

This investigation has revealed a number of interesting data mining results with experimentation on the reservoir operation data. It suggests that a larger investigation, as outlined below, using more data sets and data set characteristics would be worthwhile.

- Using more data sets:

  The use of a large number of data sets would allow an increase in size of the data sets generated for clustering analysis. This will allow the clustering algorithms to consider more cases for the formation of clusters. In the present study, because of the data-short environment, the cluster analysis was not that effective.

- Increased number of the data sources:

  The data sets used in this thesis came mainly from the very limited data collection, a multi-reservoir system from USA, and one single reservoir system in India . The use of a larger variety of real data sets from different climatic conditions may allow the mining of more accurate knowledge discovery and to make it for general applicability with different data mining algorithms.

- Using optimal parameter values by fine-tuning the settings of each algorithm:

  The residue rates can be decreased by fine-tuning the different options available for optimal classification. Especially domain specific expertise can help identify the decision parameter that have a more pronounce effect on the prediction variable thereby further enhancing performance.

- Employing visualization tools to analyze the generated data set:

  Visualization of the generated data set may provide important information and may allow better analysis of the reservoir operation rules thus formed.

Bibliography

[1] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In Proceedings of the Eleventh International Conference on Data Engineering, ICDE '95, pages 3–14, Washington, DC, USA, 1995. IEEE Computer Society. ISBN 0-8186-6910-1. URL http://dl.acm.org/citation.cfm?id=645480.655281.

[2] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. SIGMOD Rec., 22(2):207–216, June 1993. ISSN 0163-5808. doi: 10.1145/170036.170072. URL http://doi.acm.org/10.1145/170036.170072.

[3] Claudia M Antunes and Arlindo L Oliveira. Temporal data mining: An overview. KDD 2001 Workshop on Temporal Data Mining.

[4] V. Babovic, R. Canizares, H. R. Jensen, and A. Klinting. Artificial neural networks as a routine for updating of numerical models, 2001.

[5] PradipKumar Bala. Data mining for retail inventory management. In Sio-Iong Ao and Len Gelman, editors, Advances in Electrical Engineering and Computational Science, volume 39 of Lecture Notes in Electrical Engineering, pages 587–598. Springer Netherlands, 2009. ISBN 978-90-481-2310-0. doi: 10.1007/978-90-481-2311-7_50. URL http://dx.doi.org/10.1007/978-90-481-2311-7_50.

[6] Gideon Berger and Alexander Tuzhilin. Discovering unexpected patterns in temporal data using temporal logic. In Temporal Databases, Dagstuhl, pages 281–309, 1997.

[7] George Edward Pelham Box and Gwilym Jenkins. Time Series Analysis, Forecasting and Control. Holden-Day, Incorporated, 1990. ISBN 0816211043.

[8] Antony Browne and Shuang Yang. Neural network ensembles: combining multiple models for enhanced performance using a multistage approach. Expert Systems, 21(5):279–288, 2004. ISSN 1468-0394. doi: 10.1111/j.1468-0394.2004.00285.x. URL http://dx.doi.org/10.1111/j.1468-0394.2004.00285.x.

[9] Chris Chatfield. The analysis of time series: an introduction. CRC Press, Florida, US, 6th edition, 2004.

[10] Stefano Concaro, Lucia Sacchi, Carlo Cerra, Pietro Fratino, and Riccardo Bellazzi. Mining healthcare data with temporal association rules: Improvements and assessment for a practical use. In Proceedings of the 12th Conference on Artificial Intelligence in Medicine: Artificial Intelligence in Medicine, AIME '09, pages 16–25, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-02975-2. doi: 10.1007/978-3-642-02976-9_3. URL http://dx.doi.org/10.1007/978-3-642-02976-9_3.

[11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. Mach. Learn., 20(3):273–297, September 1995.

[12] P. Coulibaly, M. Hache, V. Fortin, and B. Bobee. Improving daily reservoir inflow forecasts with model combination., 2001.

[13] J. Dong. Research on nonlinear combination forecasting method based on wavelet network, 2000.

[14] Robin A. Dubin. Spatial autocorrelation: A primer. Journal of Housing Economics, 7(4): 304–327, 1998. URL http://EconPapers.repec.org/RePEc:eee:jhouse:v:7:y:1998:i:4:p:304-327.

[15] Usama Fayyad and Paul Stolorz. Data mining and kdd: Promise and challenges. Future Gener. Comput. Syst., 13(2-3), November 1997. ISSN 0167-739X.

[16] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. AI Magazine, 17(3):37–54, 1996.

[17] Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. Mach. Learn., 37(3):277–296, December 1999. ISSN 0885-6125.

[18] Thomas J. Glezakos, Theodore A. Tsiligiridis, Lazaros S. Iliadis, Constantine P. Yialouris, Fotis P. Maris, and Konstantinos P. Ferentinos. Feature extraction for time-series data: An artificial neural network evolutionary training model for the management of mountainous watersheds. Neurocomputing, 73(13):49 – 59, 2009.

[19] Tejada Guibert, J. Alberto, Sharon A. Johnson, and Jery R. Stedinger. The value of hydrologic information in stochastic dynamic programming models of a multireservoir system. Water Resources Research, 31(10):2571–2579, 1995. ISSN 1944-7973. doi: 10.1029/95WR02172. URL http://dx.doi.org/10.1029/95WR02172.

[20] Ralf Hartmut Güting. An introduction to spatial database systems. The VLDB Journal, 3 (4):357–399, October 1994. ISSN 1066-8888. URL http://dl.acm.org/citation.cfm?id=615204.615206.

[21] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. SIGKDD Explor. Newsl., 11(1):10–18, November 2009.

[22] Jiawei Han and Yongjian Fu. Discovery of multiple-level association rules from large databases. In Proceedings of the 21th International Conference on Very Large Data Bases, VLDB '95, pages 420–431, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-379-4. URL http://dl.acm.org/citation.cfm?id=645921.673134.

[23] D. J. Hand and K. Yu. Idiots bayes not so stupid after all?., 2001.

[24] Paris C. Kanellakis, Gabriel M. Kuper, and Peter Z. Revesz. Constraint query languages. Journal of Computer and System Sciences, 51(1):26 – 52, 1995.

[25] Bekir Karlik and A Vehbi Olgac. Performance analysis of various activation functions in generalized mlp architectures of neural networks. International Journal of Artificial Intelligence And Expert Systems (IJAE).

[26] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. Improvements to platt's smo algorithm for svm classifier design. Neural Comput., 13(3):637–649, March 2001. ISSN 0899-7667.

[27] K. J. Kim. Financial time series forecasting using support vector machines, 2003.

[28] K. Koperski, J. Han, and N. Stefanovic. An efficient two-step method for classification of spatial data, 2000.

[29] Boris Kovalerchuk and Evgenii Vityaev. Data mining for financial applications. In Oded Maimon and Lior Rokach, editors, Data Mining and Knowledge Discovery Handbook, pages 1203–1224. Springer US, 2005. ISBN 978-0-387-24435-8. URL http://dx.doi.org/10.1007/0-387-25465-X_57.

[30] Laks V. S. Lakshmanan, Raymond T. Ng, Jiawei Han, and Alex Pang. Exploratory mining and pruning optimizations of constrained association rules. In SIGMOD Conference, pages 13–24. ACM Press, 1998. ISBN 0-89791-995-5.

[31] Aleksandar Lazarevic, Dragoljub Pokrajac, and Zoran Obradovic. Distributed clustering and local regression for knowledge discovery in multiple spatial databases. pages 129–134, 2000.

[32] Xue-Zhen Li, Li-Zhong Xu, and Yan-Guo Chen. Implicit stochastic optimization with data mining for reservoir system operation. pages 2410–2415. IEEE, 2010.

[33] Weiqiang Lin and et al. An overview of temporal data mining, 2002.

[34] D. P. Loucks, B. F. Sule, and J. R. Stedinger. Stochastic dynamic programming models for reservoir operation optimization, 2000. Water Resources.

[35] Nikos Mamoulis, Huiping Cao, and D. W. Cheung. Mining frequent spatio-temporal sequential patterns. In Data Mining, Fifth IEEE International Conference on, pages 8 pp.–, Nov 2005. doi: 10.1109/ICDM.2005.95.

[36] Jeremy L. Mennis and Jun Wei Liu. Mining association rules in spatio-temporal data: An analysis of urban socioeconomic and land cover change. T. GIS, 9(1):5–17, 2005.

[37] J. Mercer. Functions of positive and negative type and their connection with theory of integral equations, 1909. Philosophical Transaction of the Royal Society A209.

[38] Harvey J. Miller and Jiawei Han. Geographic Data Mining and Knowledge Discovery. Taylor & Francis, Inc., Bristol, PA, USA, 2001. ISBN 0415233690.

[39] A. W. Minns and M. J. Hall. Neural Networks for Hydrological Modelling, 1996. Hydrological Sciences.

[40] Thomas M. Mitchell. Machine Learning. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072.

[41] Abhinaya Mohan and Peter Z. Revesz. A new spatio-temporal data mining method and its application to the north platte river reservoir system. submitted, International Database Engineering and Applications Symposium (IDEAS) 2014.

[42] Abhinaya Mohan and Peter Z. Revesz. Temporal data mining of uncertain water reservoir data. In Proceedings of the Third ACM SIGSPATIAL International Workshop on Querying and Mining Uncertain Spatio-Temporal Data, QUeST '12, pages 10–17, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1700-9. doi: 10.1145/2442985.2442987. URL http://doi.acm.org/10.1145/2442985.2442987.

[43] Michael C. Mozer. Neural net architectures for temporal sequence processing. In A. Weigend and N. Gershenfeld, editors, Predicting the Future and Understanding the Past. Addison-Wesley, Reading, MA, 2007.

[44] S. Mukherjee, E. Osuna, and F. Girosi. Nonlinear prediction of chaotic time series using support vector machines. In J. Principe, L. Giles, N. Morgan, and E. Wilson, editors, IEEE Workshop on Neural Networks for Signal Processing VII. IEEE Press, 1997.

[45] Raymond T. Ng and Jiawei Han. Efficient and effective clustering methods for spatial data mining. In Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94, pages 144–155, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc. ISBN 1-55860-153-8.

[46] Tveito Ole, Izabela Dyras, Hartwig Dobesch, and Estelle Grueter. The use of geographic information systems in climatology and meteorology, 2002.

[47] Sankar K. Pal, Sushmita Mitra, C. A. Murthy, P. S. Sastry, and Santanu Chaudhury, editors. Pattern Recognition and Machine Intelligence, Third International Conference, PReMI 2009, New Delhi, India, December 16-20, 2009 Proceedings, volume 5909 of Lecture Notes in Computer Science, 2009. Springer. ISBN 978-3-642-11163-1.

[48] John C. Platt. Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pages 185–208. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-19416-3. URL `http://dl.acm.org/citation.cfm?id=299094.299105`.

[49] Ross J. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[50] Peter Z. Revesz. Introduction to Databases - From Biological to Spatio-Temporal. Texts in Computer Science. Springer, 2010. ISBN 978-1-84996-094-6, 978-1-84996-095-3.

[51] Peter Z. Revesz and Thomas Triplet. Classification integration and reclassification using constraint databases. Artificial Intelligence in Medicine, 49(2):79–91, June 2010. ISSN 0933-3657.

[52] Peter Z. Revesz and Thomas Triplet. Temporal data classification using linear classifiers. Information System, 36(1):30–41, March 2011. ISSN 0306-4379.

[53] John F. Roddick and Myra Spiliopoulou. A survey of temporal knowledge discovery paradigms and methods. IEEE Trans. on Knowl. and Data Eng., 14(4):750–767, July 2002. ISSN 1041-4347.

[54] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6):386–408, 1958.

[55] Hanan Samet and Walid G. Aref. Spatial data models and query processing. In Modern Database Systems, pages 338–360. ACM Press, 1994.

[56] Lorenzo Seva. How to report the percentage of explained common variance in exploratory factor analysis. Technical Report. Department of Psychology,Universitat Rovira i Virgili, Tarragona.

[57] Shashi Shekhar, Paul R. Schrater, Ranga R. Vatsavai, Weili Wu, and Sanjay Chawla. Spatial contextual classification and prediction models for mining geospatial data. IEEE Transactions on Multimedia, 4:174–188, 2002.

[58] Lindsay I Smith. A tutorial on principal components analysis. Technical report, Cornell University, USA, February 26 2002.

[59] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. Statistics and Computing, 14(3):199–222, August 2004. ISSN 0960-3174.

[60] D. P. Solomatine and Avila L. A. Torres. Neural network approximation of a hydrodynamic model in optimizing reservoir operations., 1998. Proceedings of the Hydro Informatics Conference.

[61] J. M. Spate, B. F. W. Croke, and A. J. Jakeman. Data mining in hydrology, 2003. Proceedings of the 2003 International Congress on Modelling and Simulation.

[62] Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, SIGMOD '96, pages 1–12, New York, NY, USA, 1996. ACM.

[63] V. Sudha, N. K. Ambujam, and K. Venugopal. Functions of positive and negative type and their connection with theory of integral equations, 1909. Philosophical Transaction of the Royal Society A209.

[64] Yufei Tao, George Kollios, Jeffrey Considine, Feifei Li, and Dimitris Papadias. Spatio-temporal aggregation using sketches. In ICDE, pages 214–225. IEEE Computer Society, 2004. ISBN 0-7695-2065-0.

[65] Ilias Tsoukatos and Dimitrios Gunopulos. Efficient mining of spatiotemporal patterns. volume 2121 of Lecture Notes in Computer Science, pages 425–442. Springer, 2001. ISBN 3-540-42301-X.

[66] A. W. Tucker and H. W. Kuhn. Nonlinear programming, 1951. Proceedings of 2nd Berkeley Symposium. University of California Press.

[67] C. C. Wei and N. Hsu. Optimal tree-based release rules for real-time flood control operations on a multipurpose multireservoir system, 2009. Journal of Hydrology.