Advanced Machine Learning Models for Online Travel-time Prediction on Freeways

A Thesis
Presented to
The Academic Faculty

By

Adeel Yusuf

In Partial Fulfillment
Of the Requirements for the Degree
Doctorate of Philosophy in the School of Electrical & Computer Engineering

Georgia Institute of Technology

December 2013

Advanced Machine Learning Models for Online Travel-time Prediction on Freeways

Approved by:

Dr. Prof. Vijay K. Madisetti, Advisor

School of Electrical & Computer Engineering

Georgia Institute of Technology

Dr. David V Anderson

School of Electrical & Computer Engineering

Georgia Institute of Technology

Dr. Ayanna MacCalla Howard

School of Electrical & Computer Engineering

Georgia Institute of Technology

Dr. Branislav Vidakovic

School of Biomedical Engineering

Georgia Institute of Technology

Dr. Thomas M. Conte

School of Electrical & Computer Engineering

Georgia Institute of Technology

Date Approved: November 08, 2013

Dedicated to all the people in my life who helped me become the man I am today

ACKNOWLEDGEMENTS

First of all I would thank and praise the Almighty God for establishing me to complete this PhD.

I would like to express my deep gratitude to my parents for their unconditional support and love throughout my life, which made me believe in myself. The inspiration, which I took from them, helped me exert efforts in my research at times when I was exhausted and lost.

My wife has helped me throughout the duration of my PhD studies. She took most of the burden at home while I was busy with my studies. I appreciate her efforts for making it possible and standing by me in times when the tide was rough.

My kids have been a source of inspiration for me. Whenever, I felt dejected their innocent smiles made my day.

My research supervisor, Prof Vijay K. Madisetti, helped me both towards my academic accomplishment through his professional advice and also on various other non-academic matters, which I faced during the course of my PhD. I thank him for the patience he has shown while supervising my research. He took me in his research group at a time when I had little research background. He guided me throughout the course of the last five years. There were times when I was not convinced of my abilities but he always showed faith in me and helped me pass through the obstacles I faced in my research.

Finally, I thank my research committee. They have always encouraged me on the efforts I made towards my research. Their constructive advice and positive criticism of

my work is what eventually has made this all possible. I also place on record, my sincere gratitude to all who have lent me a helping hand in my research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

DWT	≜	Discrete Wavelet Transform
MAPE	≜	Mean Absolute Percentage Error
MODWT	≜	Maximum Overlap Discrete Wavelet Transform
RMSE	≜	Root Mean Squared Error
SVM	≜	Support Vector Machines
SVR	≜	Support Vector Machine for Regression
WPD	≜	Wavelet Packet Decomposition
WPSVR	≜	Wavelet Packet Support Vector Regression
FTP	≜	File Transfer Protocol
ANN	≜	Artificial Neural Network
SQL	≜	Sequential Query Language
DCT	≜	Discrete Cosine Transform
WDSVR	≜	Wavelet Decomposed Support Vector regression
ATIS	≜	Advanced Traveler Information System
ITS	≜	Intelligent Transportation System
Caltrans	≜	California Department of Transportation
LIBSVM	≜	Library for Support Vector Machines

MATLAB	≜	Matrix Laboratory
DOT	≜	Department of Transportation
TMC	≜	Traffic Management Centers

SUMMARY

The objective of the research described in this dissertation is to improve the travel-time prediction process using machine learning methods for the Advanced Traffic Information Systems (ATIS). Travel-time prediction has gained significance over the years especially in urban areas due to increasing traffic congestion. The increased demand of the traffic flow has motivated the need for development of improved applications and frameworks, which could alleviate the problems arising due to traffic flow, without the need of addition to the roadway infrastructure.

In this thesis, the basic building blocks of the travel-time prediction models are discussed, with a review of the significant prior art. The problem of travel-time prediction was addressed by different perspectives in the past. Mainly the data-driven approach and the traffic flow modeling approach are the two main paths adopted viz. a viz. travel-time prediction from the methodology perspective. This dissertation, works towards the improvement of the data-driven method.

The data-driven model, presented in this dissertation, for the travel-time prediction on freeways was based on wavelet packet decomposition and support vector regression (WPSVR), which uses the multi-resolution and equivalent frequency distribution ability of the wavelet transform to train the support vector machines. The results are compared against the classical support vector regression (SVR) method. Our results indicate that the wavelet reconstructed coefficients when used as an input to the support vector machine for regression (WPSVR) give better performance (with selected wavelets on-

ly), when compared against the support vector regression (without wavelet decomposition).

The data used in the model is downloaded from California Department of Transportation (Caltrans) of District 12 with a detector density of 2.73, experiencing daily peak hours except most weekends. The data was stored for a period of 214 days accumulated over 5 minute intervals over a distance of 9.13 miles. The results indicate an improvement in accuracy when compared against the classical SVR method.

The basic criteria for selection of wavelet basis for preprocessing the inputs of support vector machines are also explored to filter the set of wavelet families for the WDSVR model. Finally, a configuration of travel-time prediction on freeways is presented with interchangeable prediction methods along with the details of the Matlab application used to implement the WPSVR algorithm.

The initial results are computed over the set of 42 wavelets. To reduce the computational cost involved in transforming the travel-time data into the set of wavelet packets using all possible mother wavelets available, a methodology of filtering the wavelets is devised, which measures the cross-correlation and redundancy properties of consecutive wavelet transformed values of same frequency band.

An alternate configuration of travel-time prediction on freeways using the concepts of cloud computation is also presented, which has the ability to interchange the prediction modules with an alternate method using the same time-series data.

Finally, a graphical user interface is described to connect the Matlab environment with the Caltrans data server for online travel-time prediction using both SVR and WPSVR modules and display the errors and plots of predicted values for both methods.

The GUI also has the ability to compute forecast of custom travel-time data in the offline mode.

1. INTRODUCTION

1.1 Motivation

Accurate travel-time forecast information has become a fundamental component of all ATIS (Advanced Traveler Information Systems). ATIS in Intelligent Transportation Systems framework, focuses towards, providing information to the traveler, pre-trip or en-route for making informed decision for the journey. This information includes route guidance, traffic conditions and other information focused on the traveler needs. ATIS use the data gathered from the ITS infrastructure components and converts it into information using intelligent algorithms. This information is then dispersed through the Traffic Management Centers (TMC) for the commuters.

Currently, drivers demand an accurate travel-time calculator, which can provide precise information of the future traffic conditions. This forecast becomes even more significant in the morning and evening hours, when the traffic flow increases the capacity of the roadways resulting in congestion and gridlocks. Presently, to facilitate commuters most of the State Department of traffic (DOT) websites provide the current traffic conditions; some sites even calculate a forecast of the travel-time based on the historical data and/or current data by employing a suitable algorithm [1, 2].

The traffic does not follow a specific pattern every day because the rate of flow at a certain part of the freeway might change with an accident downstream or a certain event at a place upstream of the freeway could cause an increase in the traffic demand. The traffic flow is dependent on multiple factors that are related through a complex-dependent relationship with one another. Such factors include weather conditions, driver

behavior, and time of the day etc. This complex-dependence makes the traffic data both non-linear and non-stationary. Consequently, accurate prediction of travel-time becomes a challenging task.

Travel-time prediction method can be classified from different perspectives as shown in figure 1. From the algorithmic viewpoint the traffic models and data-driven methods are the two significant techniques used for prediction. Traffic flow models work towards figuring out the complex-dependent relationship between traffic parameters and travel-time and then estimate and predict travel-time based on the current and past traffic parameters. On the other hand, data-driven methods compute a relationship between the model parameters based on historical data, and through that model compute the predicted travel-time.

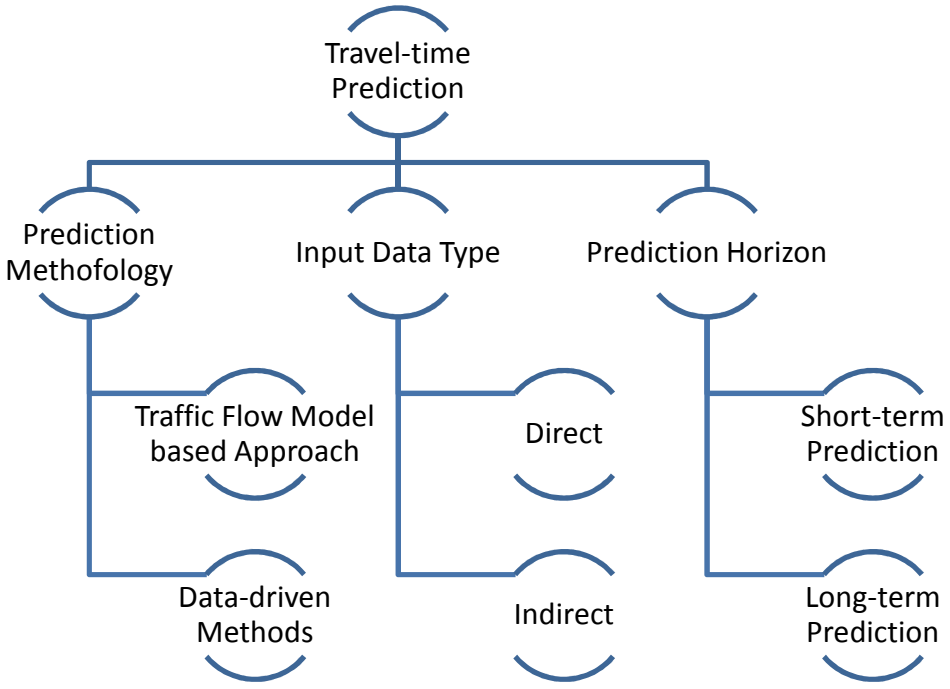


Figure 1: Taxonomy of Travel-time Prediction

Two types of data are used for traffic data collection i.e. direct, which mainly includes GPS sensors and vehicle probes. Direct sensors are present inside the vehicle and they record or communicate the vehicle location. Indirect sensors such as Inductive-loop detectors, tag matching, infrared sensors and microwave radar are all sensors, which detect the presence of the vehicle. Direct sensors are more accurate as they are capturing the vehicle data from within the vehicle and the sensors used like GPS and vehicle probes are more sensitive than their indirect sensor counterparts. Their data can also be used efficiently in real-time scenarios as GPS technology, which is now readily available in smartphones and almost every vehicle has one installed in it. With the advent of connected car frameworks, the access to this information would become trivial. However, these concepts are currently in the research and prototyping phases and some consumers are also concerned about their privacy issues. Currently, the cost of collecting data through them does not make them a feasible choice for such systems. Indirect methods especially the Inductive Loop Detectors are widely used for traffic data collection.

The prediction horizons set for travel-time prediction range from as few as 3-5 minutes to multiple hours. The work on travel-time prediction is mostly focused towards offline data processing. In the offline mode factors such as the delays involved in moving and processing data from the measurement source to the data server and processing time of data aggregation from multiple detectors into a data file are not considered. These delays are not negligible in the online mode, especially when considering the short time prediction horizons. In broad terms the time horizons of 60 minutes and under are termed as short-term prediction while 60 plus minutes prediction horizon fall under the category of long-term prediction. In general the total delay involved when using loop detectors as

a data source aggregated over 5 minutes is approximately 15 minutes. This does not include the data delay from the data server to the local machine where the forecast is implemented. While, a brief overview of all types is given in Chapter 2, the focus of this thesis is on improving the accuracy of a short-term data-driven prediction method.

Table 1 shows a brief overview of the prior art in the area of data-driven travel-time prediction. The table covers major research in the realm of short-term travel-time prediction only. The methods used for travel-time prediction range from black box approach like neural networks to signal tracking methods like Kalman filters. Since, the travel-time data is a time-series data, the typical prediction methods like regression methods and its variants are all applicable in the domain of travel-time prediction and most of them have been implemented in the past. However, only the main data-driven algorithms used for travel-time prediction are mentioned in Table 1.

The prediction accuracy is not only dependent on the prediction algorithms alone, but the length of the roadway, statistics of the freeway data and prediction horizon etc., also play a significant role in affecting the overall accuracy of the system. For example if the freeway in consideration does not experience congestion or the congestion area is irrelevant when compared against the area where the traffic is under freeflow conditions, then the variance in travel-times would be minimal. Hence such a case cannot be compared against a roadway where the congestion area is more as compared to non-congested length of freeway. Similarly, the errors of forecast horizon of 5 minutes are irrelevant when compared against the errors of 60 minutes horizon. The combination of the above-mentioned factors makes an accurate comparison even more complex. Therefore, it can be inferred that the accuracies mentioned in Table 1 are not representative of the preci-

sion of the method used. Unfortunately, there is no standard dataset to compare the results and efficiency of travel-time prediction algorithms. However, we implemented the methods mentioned in Table 1 along-with other techniques in the literature on our dataset. The results varied under different traffic conditions and prediction horizons. There was also a significant variance in accuracy of similar methods with different configurations. Therefore, a conclusive decision on the overall accuracy of a particular method under all traffic conditions with varying prediction horizons could not be deducted. A definite conclusion of the superiority of a particular method under all traffic conditions and prediction horizons also could not be made. We therefore decided to select one prediction method and focus on its improved configuration, overall accuracy and robustness.

Non-linear regression, Kalman filtering, nearest neighbor, neural network and support vector regression are the main methods used for short-term travel-time prediction. The prediction horizons in Table 1 range from 5 minutes to 60 minutes. However, there is always a network delay of 15 minutes approximately before the data is collected compiled and made available on the data server form the individual ILDs for computation. This inherent delay in the ATIS data collection process reduces the significance of prediction models with lower forecast horizons in the real-world scenario.

The process diagram of the prediction process is shown in figure 2. The process is divided into three major parts: The first part covers the routines and framework related to data associated issues. Data acquisition, preprocessing and storage are covered in this section. Data acquisition process using inductive loop detectors is explained in figure 3. The speed and traffic count data measured by the inductive loop detectors is processed at the regional traffic management centers after being processed through the individual con-

troller cabinet located beside each loop detector. The traffic management centers then transfers this data to the central data server for onward preprocessing and storage. The second part covers travel-time estimation, which houses the algorithms for data conversion from the traffic data into travel-time. This estimated travel-time data is ground truth against which the predicted travel-times are computed for errors. Lastly the travel-time prediction section houses the algorithms used for forecast of the future travel-times and compare them with the estimated values for errors.

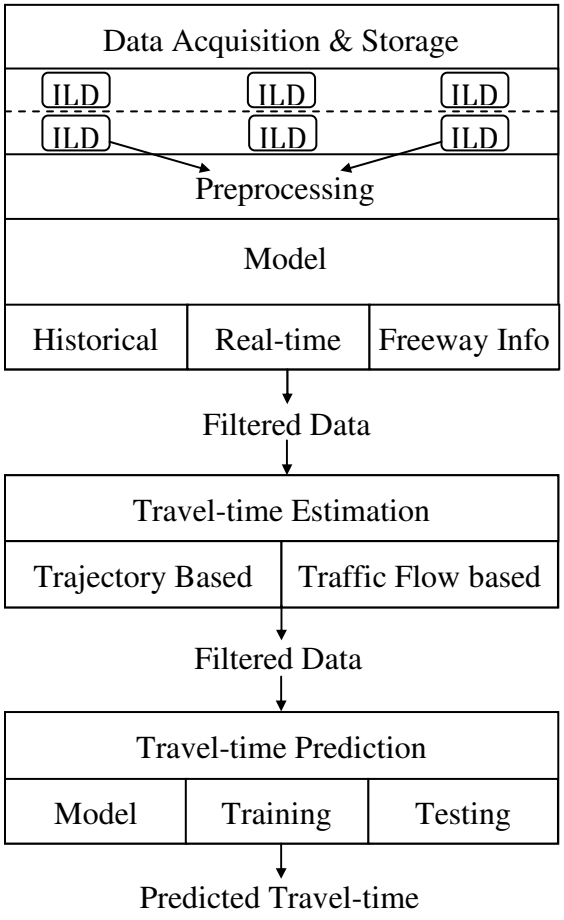


Figure 2: Process diagram of travel-time prediction using machine learning methods

Machine learning methods were extensively used in travel-time prediction [3-6]. At longer horizons the relevance of the input data decreased with the actual future travel-time and methods like the historical data predictor become more accurate than the data-driven methods. Artificial Neural Networks were mainly used for short-term travel-time prediction.

Another machine learning method, the Support Vector Regression (SVR), has shown superior performance when compared with other traditional methods for prediction of non-linear data. It was not applied aggressively in the area of travel-time prediction.

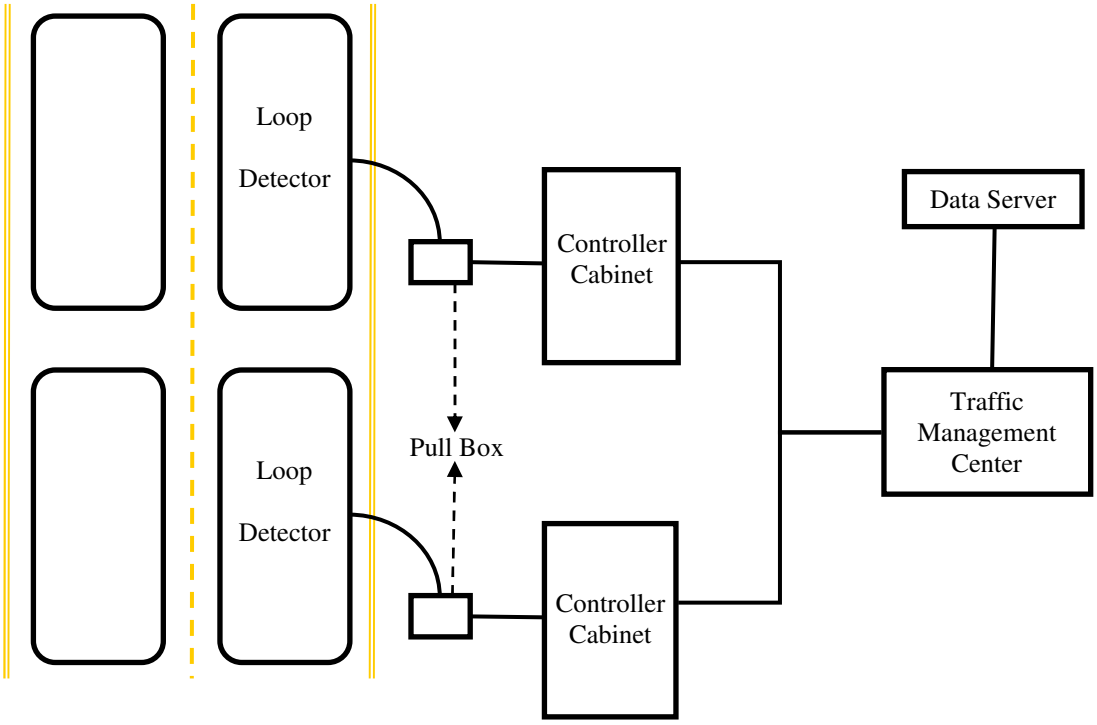


Figure 3: Data Acquisition framework using Inductive loop detectors.

Support vector machines since their inception by Vapnik [7, 8] were extensively used in classification and prediction problems. SVM uses a simple geometric interpreta-

tion and gives a sparse solution. The solution of SVM is also global and unique as SVM employs the structural-risk-minimization principle. SVR method [9] approaches the linear regression forecast by addressing it as an optimization problem (details in Chapter 4). Its performance in financial time series forecast [10], bioinformatics [11] and various other areas of research also makes it a viable method in intelligent transportation systems (ITS) applications. SVR application as a forecasting tool in ITS was first done by Wu [14], who predicted short-term travel-time on the basis of past and current values. Recently, Wang in [16], used wavelet kernel support vector machine for regression to predict traffic flow in ITS applications.

Table 1: Comparison of prior art on data-driven travel-time prediction

Prior Art related to Short-term Travel-time prediction			
Prediction Methods	**Author/Year of Publication**	**Length of Roadway**	**Accuracy / Prediction Horizon**
Neural Networks	J.W.C. Van Lint (2004) [12]	5.28 Mi (8.5 Km)	RMSEP: 7.7% MRE: 0.49% SRE 6% Horizon: 15 min
Kaman Filter	Chen and Steven Chien (2001) [13]	8 Mi (12.88 Km)	MARE: 0.0173-0.0208 Horizon: 5 min
Support Vector Regression	Wu, Ho and Lee (2004) [14]	28 – 217.5 Mi (45 – 350 Km)	RME:0.96 – 4.42%, RMSE 1.33-7.35% Horizon: 3 min
PCA/Nearest Neighbor	Rice and Zwet (2004) [1]	48 Mi (77.25 Km)	RMSE: 2.6 – 11 (Approx) Horizon: 60 min
Regression	Kwon, Coifman and Bickel (2000), [15]	6.2 Mi (10 Km), 20 Mi (32.19 Km)	MAPE: (Tree Method) 6.9 – 28.7%, (Regression) 7.7 – 23.3% Horizon 10-60 min

In the recent years many researchers decomposed time series into more informative domains like the wavelets transform [17], S-transform [18] etc., as an input to the SVR that showed more accurate results than the non-decomposed method. This improved performance of SVR along with the ability of SVR to predict non-linear data, formed the motivation of our research to explore the effectiveness of travel-time prediction using wavelet transformed travel-time values as an input to SVR.

The accurate wavelet selection for the wavelet decomposed methods reduces the significance of the method. However, an analytical method towards the selection of the wavelet is not published in literature. This dissertation makes an attempt to figure out some of the variables, which affect the efficiency of the support vector machines.

1.2 Outline

This dissertation summarizes the research for travel-time prediction on freeways and provides the model and framework to improve the accuracy for small-term travel-time prediction. The research conducted on the process of filtering out the wavelets, which do not provide improved results when compared against the classical support vector regression method are described. The method was investigated to reduce the computational cost involved in calculating the wavelet packet decomposed data for each wavelet. Thirdly, an alternate configuration for calculating and broadcasting the travel-time information is explained which exploits the cloud based technology. The motivation to work on the cloud framework came from the extensive work done in the automobile industry on the connected car framework. Finally, the graphical user interface for the online travel-time prediction calculation was presented to forecast the travel-time prediction result in real-time.

The outline of the rest of the thesis is as follows: the problem statement along with some highlights of the past research is given in Chapter 2. Chapter 2 also explains each subsection of the travel-time prediction process along with the brief of the significant work done in each area. The current approaches used to solve the travel-time prediction problem are explained in Chapter 3. The work done on each area along with the brief overview of each method is presented. The basic concept of Wavelets and Support vector

regression are explained in Chapter 4 and 5, respectively. In Chapter 6 the transform based support vector regression methods are explored. The transform based SVM is mainly used for prediction, classification, compression and forecasting problems. The methods and different configuration used in each model presented. The Chapter 7 explained the methodology proposed for optimal wavelet selection, while Chapter 8 proposed the WPSVR model along-with its procedural details and the steps involved for its implementation. Then we show the results of our model and compare it with classical SVR implementation in Chapter 9 along with the details of the alternate configuration of the framework of the travel-time prediction process. The details of the graphical user interface are given in Appendix B. Finally, the thesis is concluded in Chapter 10, with a summary of the claims made in this thesis and future work direction.

2 PROBLEM BEING SOLVED

In this chapter, the details of the travel-time prediction problems are explained with a brief overview of the classification of the prediction process. The fundamental parts of the data-driven travel-time methodology with the analysis of the traffic data are also explored.

The predicted travel-time is the projected mean time of vehicles traversing the section of the roadway in consideration, in a future time period. It can be mathematically expressed as

$$\tau_{T+y} = \sum_{t=T}^{T-x} f(\tau_t^k) + c\,, \tag{1}$$

where τ_T^k is the current travel-time on roadway k. x represents the index of past calculated travel-time values and c represents the error of $f(\tau_t^k)$.

The travel-time is calculated from the traffic data collected from various sensors. Sensors like the Inductive-loop detectors measure the speed of individual vehicles passing over them. Their accumulated mean speed is called the time mean speed. This speed is not representative of the average speed of the roadway section as it is a spot mean speed and the vehicles over the whole section are travelling at different speeds during that time period. The time mean speed and space mean speed are represented in equations (2) and (3) respectively.

$$v_t = \frac{1}{M}\sum_{i=1}^{M} v_i, \tag{2}$$

$$v_s = N\left(\sum_{i=1}^{N}\left(\frac{1}{v_i}\right)\right)^{-1}, \quad (3)$$

where M and N represent the total number of vehicles passing through the loop detector and the roadway section respectively.

The relationship between the time mean speed and the space mean speed can be mathematically represented as:

$$v_t = v_s + \frac{\sigma^2}{v_s}, \quad (4)$$

where v_t and v_s are the time mean speed and space mean speed respectively and σ^2 is the standard deviation of the spot mean speed. Since the standard deviation cannot be a negative value, therefore time mean speed would always be more than the space mean speed. The relationship presented in (4) is derived from the algebraic manipulation of the fundamental equation of traffic flow, (2) and (3). The complete derivation of (4) is detailed in Appendix A. For accurate travel-time estimation the time mean speed is converted into the space mean speed before calculating the travel-time.

The travel-time prediction problem can be viewed from the perspective of input data type, prediction methodology and prediction horizon as shown in figure 1. Irrespective of the class of travel-time prediction, the fundamental components of the process are similar as shown in figure 2. Next we explain each component with a review of the significant published work done in each area.

2.1 Data Acquisition and Storage (ILD)

Formulation of an accurate predictive inference relies significantly on the quality of traffic data. A typical speed plot constructed using a portion of the dataset we used is shown

in figure 3. The red color in the speed plot represents the freeflow traffic while the transition from yellow to blue represents the transition from freeflow to congestion.

Inductive Loop Detector (ILD) data based on its abundance and known quality issues has been used as input data in most travel-time prediction research [15, 19-25]. The scalability of the model also biased the choice of the researcher towards choosing ILD as a data source. Other forms of datasets include probe vehicle data, traffic camera feeds, and satellite data, data obtained from microwave radar, license plate matching, and automated vehicle tag matching.

To use ILD data, certain known issues required attention in context of the site selection and data pre-processing phases. Spacing between consecutive loop detectors directly affects the quality of the data captured. The standard spacing requirement between consecutive loop detectors is not defined in literature. However, [26] concluded that the

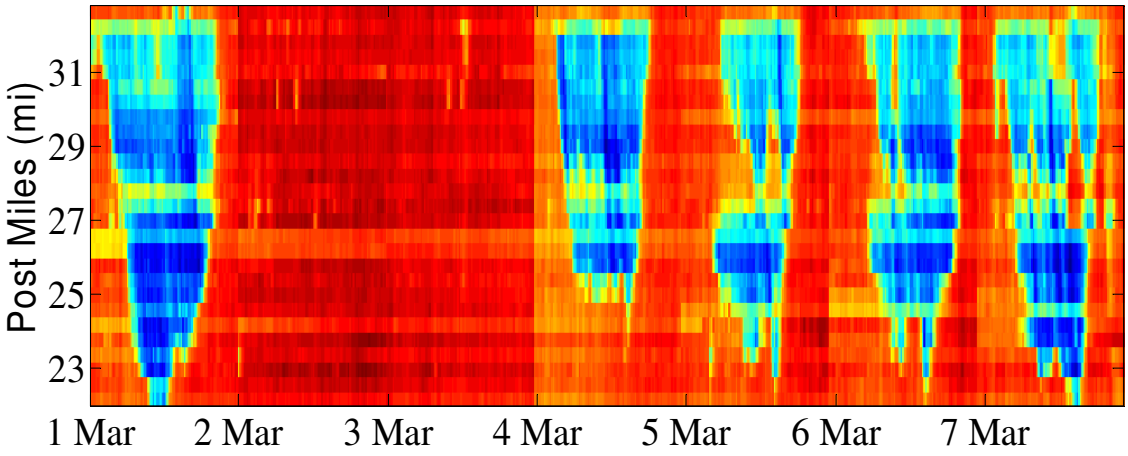


Figure 4: Speed plot of a portion of the dataset

detector spacing of 1 to 1.5 km is optimum for the use of short-term forecasting of traffic parameters. In [27], it was shown that a detector spacing of 0.33 to 1 mile does not desta-

bilize the travel-time estimation errors, while [28] concluded that a detector spacing of 0.5 miles is sufficient to represent traffic congestion with acceptable accuracy.

After data acquisition preprocessing steps are performed on this data to ensure its validity. ILDs are prone display a number of errors [29]. These data errors are usually detected and removed using imputation methods [29, 30]. [29] gives a linear model based on historical data using neighboring detectors to detect faulty values and through linear regression imputes the missing or bad values. The method proposed in [29] is adopted by Caltrans for data processing of the loop detector data in California roadways.

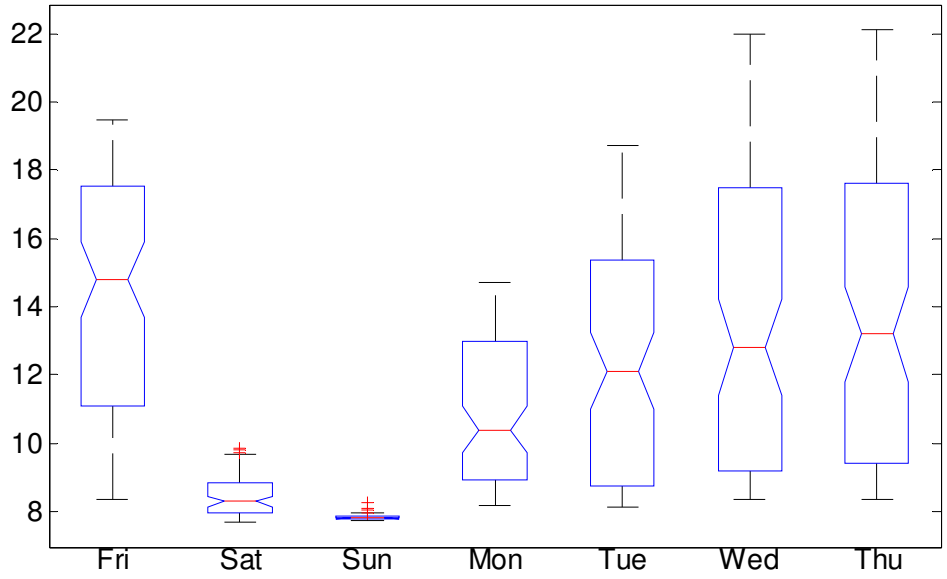


Figure 5: Box-plot of travel-time data classified by days of the week

To summarize the statistical descriptors of the travel-time data, we divided the dataset based on the weekdays and computed the box-and-whisker plot as shown in figure 5 of the travel-time data. It is evident from the plot that the traffic on each day has its own unique properties. Similarly, the spread-out quartile data of each box represents the variation in the travel-times of each day. This is due to the fact that the data selected is from 1pm to 8pm daily, which accounts for both congested and freeflow values. The smaller box of Saturday and Sunday with a lower median value and bunched quartile values than the other weekdays indicate free flowing traffic on the weekends.

Figure 6 shows the plot of the mean travel-times of weekdays for the subject dataset. The plot shows a daily congestion of traffic in the roadway for all weekdays except Saturdays and Sundays, which is representative of a typical urban roadway traffic profile.

2.2 Travel-time Estimation

The travel-time estimation from ILD data source requires conversion of speed data collected from the ILDs into space mean speed and then into travel-time. Like any prediction problem, the ground truth (estimated travel-time) is essential to evaluate the results (predicted travel-time). The travel-time estimation methods are divided into two broad categories: *trajectory-based* and *flow-based*.

2.2.1 Trajectory-based methods:

The trajectory based methods calculate travel-times between all links of the roadway between the reference points, and then by adding the travel-times of individual links accumulate the travel-time between the two points on the roadway.

The trajectory-based methods convert the time-mean speeds collected from detectors to space-mean speed. Different methods are proposed to calculate link travel-time from this speed. The three common methods are the mid-point method, minimum speed method and the average speed method.

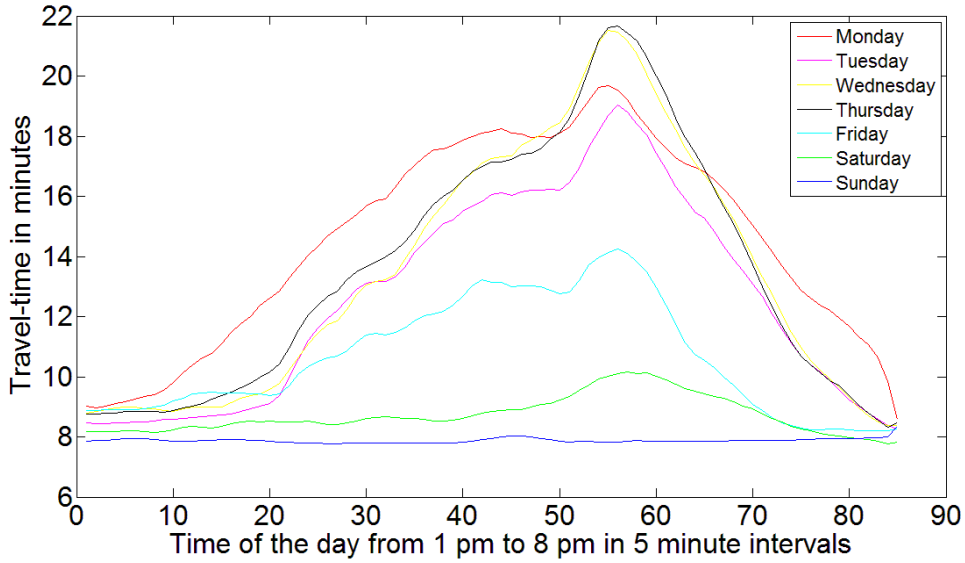


Figure 6: Mean travel-time data for all weekdays

2.2.1.1 Mid-point method Mid-point method or the half distance method uses the distance between mid-points of adjacent detectors on both sides of the subject detector to measure the travel-time. Mathematically, it can be represented as:

$$\tau_{1-2} = \frac{1}{2}\left(\frac{D_{1-2}}{v_1} + \frac{D_{2-3}}{v_2}\right),$$

where D_{1-2} is the distance between the detector D_1 and D_2 and v_1 is the space mean speed detected at D_1. The diagram illustrating this configuration is shown in figure 7.

2.2.1.2 Average speed method The average speed method assumes that the speed between two detectors is the average of the speed measured at both ends.

$$\tau_{1-2} = \frac{2D_{1-2}}{(v_1 + v_2)}$$

2.2.1.3 Minimum speed method The minimum speed method as the name implicates uses the minimum speed between the two consecutive detectors to calculate the travel-time between them.

$$\tau_{1-2} = \frac{D_{1-2}}{Min(v_1, v_2)}$$

The above-mentioned methods assume a constant speed between links, which in reality is never the case especially when traffic is in transition from freeflow to congestion or vice versa. Hence, the algorithms, which propose a constant speed lose their accuracy with the increase in congestion [31]. Van Lint proposed an alternate approach, the "Piecewise Linear Speed" method [32], which solved the function of the travel-time based on time mean speed using an ordinary differential equation to calculate the trajectory of the vehicle in the section based on space mean speed. The PLSB method solves both issues of converting the spot speed into space mean speed and the issue of representation of speed as a function instead of a constant.

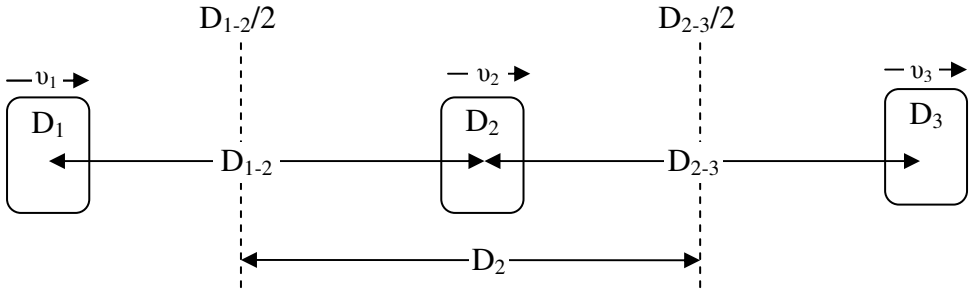


Figure 7: Configuration of detector placement on roadways

Flow-based methods:

An alternate way of estimating travel-time is through flow-based models, which focus on capturing the dynamics of traffic using traffic-flow theory concepts, and through traffic data simulation, draw the travel-time of the segment. Accurate flow information is also required for a precise estimation; however, in most cases it is difficult to collect data from all on-ramps and off-ramps using the existing infrastructure, which becomes a bottleneck for flow-based estimation methods. The current ILD placements in USA roadways were not done based on the constraints imposed by the traffic flow models. These models are, however, more popular in traffic flow simulation research where traffic flow data is simulated to conform to the flow model requirements.

The fundamental traffic flow diagram is shown in figure 8. The figure explains the relationship between flow density and speed on roadways. The traffic flows in every model irrespective of its design methodology follow the characteristics defined in figure 8.

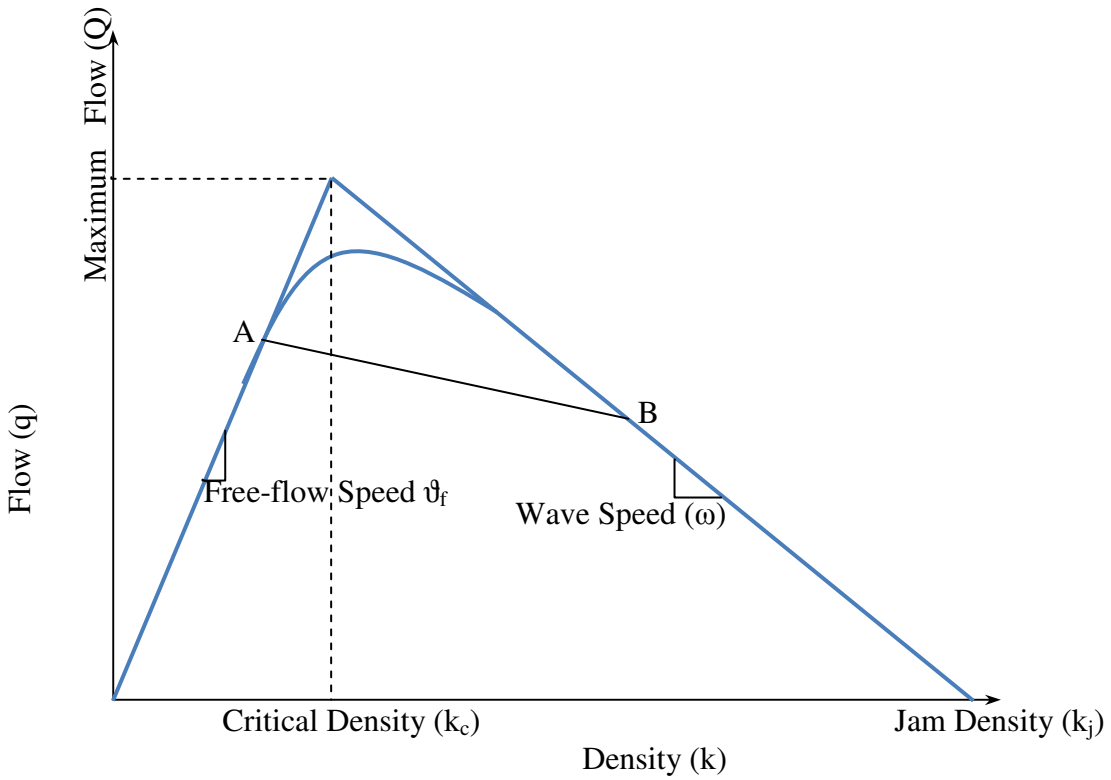


Figure 8: Fundamental diagram of traffic-flow

2.3 Travel-time Prediction

The Travel-time prediction approach is mainly classified w.r.t. the prediction horizon, modeling approach and type of input data as shown in figure 1. Further classification is also possible w.r.t. the road type (freeways, arterials); but, since the scope of this thesis is confined to freeways; we would not discuss the arterial travel-time prediction problem. Below, the major methods used for data-driven travel-time prediction are discussed.

The historical data of traffic parameters can represent a traffic profile, which could be implemented to predict future values, in similar traffic conditions. This approach demands *offline* processing. The data is classified into different subtypes based on their

characteristics. In [33] the data was sub-classified into the “type of day”, for prediction of travel-time. This prediction method does not take into account the dynamics of traffic during the travel-time which makes this method inherently erroneous. Consequently, it produces low accuracy results, when the current traffic is not representative of its historical profile. Historical predictor is normally used for long-term prediction.

A hybrid approach of combining historical data with current data was used in [34] where real time data was captured directly from the road side terminals, and using it with aggregated historical data showed comparable results. [1] used principal component analysis and windowed nearest neighbor, while combining historical and instantaneous data.

Traffic data shares similarities when compared with historical data of the same day and time as the current data. Regression methods with coefficients varying with the time of day were used by [1], [35] and [36] to predict travel-time. [15] also used linear regression with step wise variable selection method. Regression models involve the examination of historical data, thereby, extracting parameters which represent traffic characteristics, and projecting them into the future to predict travel-time. ARIMA was introduced by [37] and [38] as an alternate to model the stochastic nature of traffic. [39] used auto-regression model to predict travel-time. Non-linear time series with multifractal analysis was implemented in [40] and [41] for travel-time prediction.

Kalman and Extended Kalman Filters used in [2, 42] provide good performance in predicting travel-time for one time-step, which is normally not more than 5 minutes, as the state model needs real observations to calculate each error term. This makes the filtering approach inefficient to implement in the real world.

Artificial Neural Networks (ANN) was extensively used for marking non-linear boundaries. To address the problem of a time series forecast, a subtype of ANN called the recurrent neural network (RNN) was considered suitable [19, 24, 43]. RNN has an internal state, which keeps track of the temporal behavior between classes. Different architectures of the Multilayer perceptron have been used to predict travel-time with an acceptable accuracy [3-6, 19, 20, 23, 24, 43-51]. However, they only address the short-term travel-time prediction problem. The Support Vector Regression method was also investigated in [14, 52].

Traffic flow models work on the concept of correlating the theory of fluid dynamics with vehicular flow. From the perspective of traffic flow models, travel-time prediction is more of a boundary condition prediction problem, because the flow model is designed offline, and it would predict the time based on the values of demand and supply at on ramps and off ramps respectively. The model is run using a simulation scheme, which is based on assumptions of car-following, gap acceptance, and risk avoidance parameters. The simulation model predicts the aggregated parameters of simulated vehicles to display the predicted travel-time [53, 54]. This makes traffic flow models very complex and requires a high degree of expertise and long man-hours for design and maintenance.

Traffic flow models give us a better understanding of the traffic flow dynamics, but as far as their accuracy for travel-time prediction is concerned, they demand a precise infrastructure of input detectors, whose location would be defined by the flow model. To manage the supply and demand parameters, the flow models require additional detectors on each off and on ramp. Traffic Flow based models are a good method to evaluate the cause and effect of traffic phenomenon, but applying them for travel-time prediction

would entail a huge design and maintenance cost for every freeway section. Due to their modular design, precision of traffic flow models, for long term travel-time prediction, would be as accurate, as the precision of the predicted inputs and boundary conditions.

In summary the travel-time prediction can either work on traffic flow based models or on the data-driven models. However, the traffic-flow based models are much more complex than their data-driven counterparts. Also the data driven models are more robust when compared against traffic-flow model prediction.

3. CURRENT APPROACHES

In this chapter the models, which are published in scholarly articles related to short-term travel-time prediction are explained. We have limited ourselves to the work based on data-driven methods. Table 1 shows the major work done in the realm of short-term travel time prediction. The techniques used to predict short-term travel time include historical predictor, instantaneous predictor, Principal Component Analysis, Neural networks, Nearest Neighbor, Kalman filtering and Regression. As mentioned before there is no standard dataset for comparison of the efficiency of each method, and since, there are many factors which influence the prediction accuracy other than the prediction method, therefore a true comparison of the methods is not possible. However, the details of each along with their review are presented below.

3.1 Historical Predictor

The travel-time profile of a certain section of roadway follows a similar pattern on workdays and holidays. This similarity is based on the similarity in the daily O-D (Origin-destination) tables of the vehicles on the roadway. The origin-destination tables give the approximate start and end location of the route followed by the individual vehicles with the timestamp. If similar numbers of vehicles are travelling from one point to another on every workday at similar time then the travel-time in that section of the roadway would also be similar. This similarity was observed in figure 4 on the travel-time plot of our dataset on consecutive weekdays, which follows a similar congested pattern in the evening hours. Similar patterns are observed in datasets of every weekend also.

Figure 9 below shows the 3d and contour plot of the dataset for one month. It is evident form the 3d plot that the daily congestion pattern is followed in the evening, except weekends. The variation or duration of congestion varies between days. The contour plot signifies the difference between the congestion of similar weekdays also. However, the weekends or holidays follow a similar pattern every week.

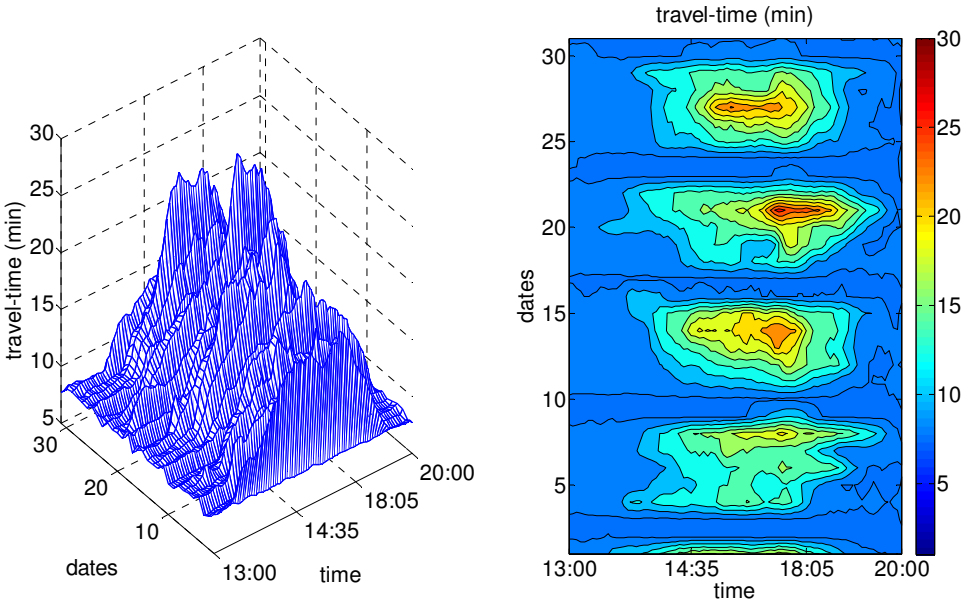


Figure 9: 3d plot and contour plot of one month of travel-time data

It is also evident that the Euclidean distance between similar data points of workdays is neither constant nor linear, which makes the historical predictor less accurate. The traffic data is not only dependent on traffic inflow and outflow from a section of the roadway but the data variation in the short-term is dependent on factors like the accidents, event occurring upstream of the traffic flow and individual driving patterns etc. Since, these factors are interrelated to each other. A change in one would cause a change

in the flow. Hence, for short-term predictions, algorithms which have better tracking ability produce better results than the historical predictor method.

The concept of using a hybrid approach also showed improved results. However, in most cases the variance of the data would lie within a certain range, which makes this algorithm suitable for long-term predictions.

3.2 Instantaneous Travel-time

An even simpler prediction method is using the last estimated travel-time value as the future value. Before explaining the pros and cons of this method let's review the constant factors, which affect the travel-time value.

The main three deterministic factors are the length of the roadway in consideration, the statistics of daily congestion and freeflow hours and the prediction horizon. The stochastic factors include the individual driving patterns, accidents, roadway work etc., which cannot be determined. Analyzing the effect on the accuracy of Instantaneous prediction method due to deterministic factors, the duration of congestion is inversely proportional to the accuracy of the instantaneous prediction method. The variability of travel-time increases with the increase in congestion, which results in the increase in prediction errors.

Another major disadvantage of this method is that the tracking ability reduces severely with the increase in prediction horizon. This reduction in reliability in travel-time prediction is equivalent to the difference in current travel-time estimation and the estimated time at the selected time horizon.

However, Instantaneous predictors are very simple to use and require minimal computational cost. They are also more reliable to predict travel-times for short horizons where congestion is minimal. Their major disadvantage is their capacity to be implemented in real scenario due to the time delays invoved in relaying of traffic data from the detectors to the data center and processing delay. The variation in traffic in this period especially during the congestion hours makes this method indeasivble to be implemented in real-scenario.

3.2 Principal Component Analysis

Principal component analysis (PCA) is an orthogonal transformation to decompose a multidimensional dataset into multidimensional orthogonal dataset. The resulting uncorrelated variables are called the principal components. The process of computation of PCA is based on eigenvalue decomposition of a cross correlated or singular value decomposed dataset with zero empirical mean. The process finds its significance in explaining the original dataset with reference to a particular feature(s). The following steps are required to compute the principal components Y of correlated dataset.:

1. Let us assume a correlated $m \times m$ data set $X = \begin{bmatrix} 1 & 5 & 10 \\ 2 & 10 & 20 \\ 3 & 15 & 30 \end{bmatrix}$. The zero empirical mean of X is calculated as

$$\bar{X}_j = \frac{1}{N}\sum_{i=1}^{N} X_{i,j}\,, \quad j = 1, \ldots\ldots, K \;=\; [2 \quad 10 \quad 20]$$

Subtract the mean from the original data gives us

$$X - \bar{X} = \begin{bmatrix} -1 & -5 & -10 \\ 0 & 0 & 0 \\ 1 & 5 & 10 \end{bmatrix}$$

2. Calculate the covariance matrix of $X - \bar{X}$

$$\begin{bmatrix} -1 & -5 & -10 \\ 0 & 0 & 0 \\ 1 & 5 & 10 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & 1 \\ -5 & 0 & 5 \\ -10 & 0 & 10 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 10 \\ 5 & 25 & 50 \\ 10 & 50 & 100 \end{bmatrix}.$$

3. We then calculate the eigenvector v of the covariance matrix which in our case is

$$\begin{bmatrix} 0.1747 & 0.9806 & 0.0891 \\ 0.8736 & -0.1961 & 0.4454 \\ -0.4543 & 0 & 0.8909 \end{bmatrix}.$$

   These eigenvectors are orthogonal to each other, which gives us additional information on the patterns in the data.

4. To express the data in terms of these eigenvectors we multiply the transpose of the eigenvector matrix with the mean subtracted data. This would again be a scaled version of the eigenvectors and each dimension would be perpendicular to the other.

The principal components can also be computed with singular value decomposition method and z-score method. To use PCA for prediction the conditional expectation of the future value is computed to give the forecast value.

Similar method was also used in predicting travel-time. [1] computed the covariance of historical and current data to compute the principal components by the singular value decomposition method and by retaining only the significant eigenvectors computed the principal components and used the expectation maximization method for future value

prediction, thereby exploiting the properties of both historical and instantaneous travel-times.

3.3 Neural Network

The artificial neural network (ANN) is a mathematical model composed of interconnected layers of nodes, which through weighted inputs solve a classification or prediction problem. Normally the three layers as depicted in figure 10 below are the input layers the hidden layer and the output layer. The input layer gathers the input data from the system and the values to the hidden layer which is composed of neurons. Neurons compute the weighted sum of the input values for the each node in the next layer. Usually there is one hidden layer for most problems. The number of hidden layers is defined by the user. The number of outputs defines the number of neurons in the output layer, which again based on the weighted sum of the inputs they receive from the hidden layer neurons compute

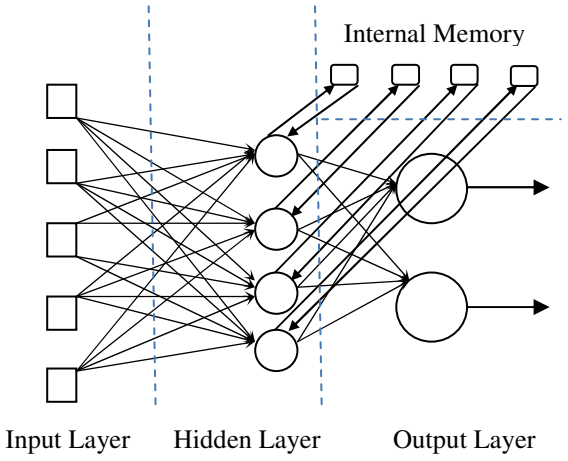


Figure 10: Configuration of a typical Recurrent Neural Network

the output for the system.

The neural networks could be defined in different configurations. In the context of time series prediction a configuration of neural nets called the recurrent neural network has proved more effective. The recurrent neural networks have an additional short term memory, which helps the network better understand the spatiotemporal properties of the signal. Hence, the output of the recurrent neural network in terms of the travel-time prediction is the predicted value in terms of the previous traffic states of the roadway.

3.4 Nearest Neighbor

The nearest neighbor and the k-nearest neighbor (KNN) methods is a simple machine learning method used for classification. It is a non-parametric method and is based on the distance of the features of the training data w.r.t. the testing data. In k-nearest neighbor method the features of the input dimension are projected in the feature space and plotted along with the features of the test data. Then based on the majority vote of the features, where the vote is given based on the minimum distance of the training feature to the test data feature, the data is classified.

In travel-time prediction this method is used to select the travel-time pattern. The input data is profiled based on its instantaneous value and historical value. Both these values are the features of travel-time. Now the current travel-time value is matched against both and the nearest neighbor is used to classify the travel-time based on the current travel-time and the historical travel-time. In other words the nearest neighbor method when implemented for travel time prediction would find the distance between the current travel-time and the historical travel-time at the same value on the time axis and based on

the minimum distance would select the day d. The selected travel-time profile would then be used to forecast future values.

In this method every day in the database serves as the individual profile of the traffic data. It can improve the travel-time prediction errors when compared to the historical predictor if the profile of the historical daily dataset is correlated with each other.

3.5 Kalman Filtering

Kalman filter is a linear quadratic estimation algorithm for tracking noisy data values. It is defined as

$$\hat{X}_k = K_k . Z_k + (1 - K_k) . \hat{X}_{k-1}$$

where $\hat{X}_k$ = current estimation

K_k = Kalmna gain

and Z_k = measured value

To use Kalman filter the data must fit the basic Kalman model defined below

$$x_k = Ax_{k-1} + Bu_k + w_{k-1}$$

$$z_k = Hx_k + v_k$$

where u_k is the control signal and w is the process noise.

The Kalman filter iterates between a set of equations classified as *Time update* and *Measurement update*. The implementation of Kalman filter can be explained with the below mentioned steps:

1. After defining initial estimates for A, B, H, R and Q the prior estimate $\hat{x}_k$ and prior covariance P_k is calculated.

$$\hat{x}_k = A\hat{x}_{k-1} + Bu_k$$

$$P_k = AP_{k-1}A^T + Q$$

2. Compute the Kalman gain, update the estimate and error covariance using equations below:

$$K_k = P_k H^T (HP_k H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k + K_k(z_k - H\hat{x}_k)$$

$$P_k = (1 - K_k H)P_k$$

This posterior estimate and covariance would now be used to calculate future prior estimates.

In general the Kalman filter is used to solved tracking problems where the current value has the correlation with its previous value. The process looks simple but modelling the noise, which in our case represents the non-linearity and non-stationarity of the data set is a challenging task as a simple Gaussian function would predict future values but the error would increase significantly as the prediction horizon increases. The bottle neck in Kalman filter implementation for travel-time prediction is also the time delays involved in relaying and processing the detector data.

3.6 Regression

Linear regression is a simple model for computing a single line through a set of data points such that the sum of the least square of the minimum distances between the data points and the line is minimum.

The simple linear regression is more effective for dataset which exhibit linearity. In our case the travel-time is both non-linear and non-stationary, which makes the use of simple linear regression in feasible. Multiple linear regression with past travel time val-

ues was used to project the future values. However, the values of α and β are variable and their values are based on the current time and prediction horizon.

Solving the non-linear problem with linear algorithms would not yield significant accuracy. The non-linear regression is therefore a more suitable option. Non-linear methods for implementation require a kernel method which could implement the algorithm with minimal computation. Non-linear methods are inherently more complex and computational intensive. Therefore neural networks and support vectors were more successful in prediction future travel-times.

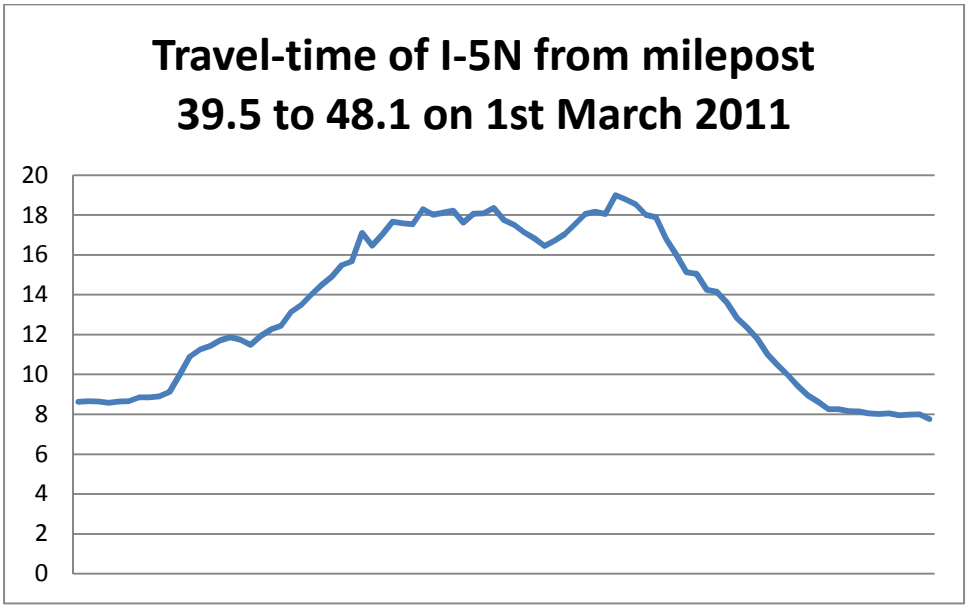


Figure 11: Travel-time of the roadway section from milepost 39.5 to 48.1 on 1st March 2011 from 1pm to 8pm

Our method is also a hybrid approach, which used the data is pre-processed using the wavelet transformation and then each array of the wavelet transformed data is given

to the separate support vector machine for regression of future travel-times. The wavelet based SVM model presented in our dissertation focuses on the smart processing of data based on the wavelet packet transform. The no-linear regression is based on the linear kernel of the support vector machine.

In the next two sections the overview of both wavelets and support vector machines is presented.

4. WAVELETS OVERVIEW

The purpose of a transform is to transform the data in a different set of axes with the aim to add information, simplify or remove noise from the original datsaset. A transform can be defined as a mathematical operator (linear or non-linear), which when applied on a data converts it into a different set of values. The inverse operator of the same transform would recover the original values back.

Researcher have used multiple transforms with different objectives; eg. The famous Fourier transform is used to convert the time domain data into frequency domain and the binomial transform is used to compute the forward difference of a number series etc. In time series analysis the transformation of a data is often required in the data preprocessing phase either to remove noise from the data, or to add or reduce dimensionality in the dataset.

The realization of the wavelet transform was motivated from the short-coming of the fourier transform. In fourier transform we basically compare the time signal with a series of sinusoidal of different frequencies to gain frequency information of the signal. However, for many applications this frequency application is not very effective with the location knowledge of the specific frequencies. This issue was initially addressed with the short-time transform by dividing the input signal into short signals and then computing the fourier transform to approximate the location of the desired frequency. It also suffered from the disadvantage of poor frequency resolution in case of short time window.

The short term fourier transform could not solve the problem because it was not focusing on the frequency and amplitude of the sinusoid signals. Wavelets achieved the time-frequency resolution of the signal by decomposing the signal with stretched and shifted versions of the wavelet signal. The wavelet in itself is also not a sinusoid but it has to obey certain properties, which would be discussed later.

Wavelets are finite signals of limited duration with zero mean. They are used for time-frequency representation of the signal. Wavelet series is defined as a square-integrable function with respect to a complete, orthonormal set of basis functions.

The wavelet function preset a multi-resolution decomposition of a signal using a mother function ψ and a linear combination of its dilated and/or shifted versions.

$$\psi_{s,n}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-n}{s}\right) \qquad (5)$$

where n defines the dilation and s defines the shift. To ensure orthonormalilty of basis functions [55] the time scale parameters are sampled on a dyadic grid on the time-scale plane. Thus (5) becomes

$$\psi_{j,n}(t) = \frac{1}{\sqrt{2^j}}\psi\left(\frac{t-n}{2^j}\right)$$

The orthonormal wavelet transform is then given by

$$\langle x(t), \psi_{j,n}(x)\rangle = \frac{1}{\sqrt{2^j}}\int_{-\infty}^{\infty} x(t)\psi_{j,n}(2^{-j}\,t-n)dt$$

4.1 Multi-Resolution Analysis

To make the transform computationally effective the concept of sub band coding [56] is used. Mallat presented the framework for multi-resolution analysis to describe the wave-

let basis. Multi-resolution is a sequence of increasing nested subspaces such that $V_{-n} \subset V_{-n+1} \subset \cdots \ldots V_0 \subset V_1 \ldots\ldots \subset V_m$. These subspaces lie in the square integrable function space called the Hilbert space. Let V_j be spanned by the scaling function ϕ_j. This means that V_{j+1} being a subspace of V_j would be a linear combination of the function ϕ_j. $\phi_{j+1}(x) = \sqrt{2}\phi_j(2x)$. Hence,

$$\phi_j = \sum_k h(k)\,\sqrt{2}\phi_j(2x-k)$$

The coefficient of $h(k)$ are the inner product of $< \sqrt{2}\phi_j(2x-k), \phi_j >$. Lets assume an orthogonal W_j of V_j such that $W_{-n} \subset W_{-n+1} \subset \cdots \ldots W_0 \subset W \ldots\ldots \subset W_m$. Now,

$$\psi_j = \sum_k g(k)\,\sqrt{2}\phi_j(2x-k)$$

The relationship between high pass filter and low pass filter is

$$g(L-1-n) = (-1)^n\, h(1-n).$$

$g(n)$ and $h(n)$ can be treated as high pass and low pass respectively and L is the length of the filter. Consequently they are also orthogonal to each other. Also $\sum h(k) = \sqrt{2},$ and $\sum g(k) = 0$.

In order to decompose the signal in wavelet domain at nth level, the high pass filter $g(k)$ and low pass filter $h(k)$ will have to be applied recursively to filter the signal with a series of high pass and low pass filters to analyze its high frequency and low frequency components respectively. The input signal $x(t)$ can now be represented in discrete domain as

$$x(t) = \sum_{n\epsilon\mathbb{Z}} c_{J,n}\emptyset_{J,n}(t) + \sum_{j=J}^{\infty}\sum_{n\varepsilon\mathbb{Z}} d_{j,n}\psi_{j,n}(t)$$

The scaling $c_{j,n}$ and wavelet coefficients $d_{j,n}$ can now be defined using high pass g_l and low pass filters h_l

$$c_{j,n} = \sum_{l\epsilon\mathbb{Z}} h_l c_{j-1,2n-1},$$

$$d_{j,n} = \sum_{l\epsilon\mathbb{Z}} g_l c_{j-1,2n-1.}$$

The wavelet filter for the undecimated wavelet transform are shown in figure 12. The high pass decimation and reconstruction filter together form the highpass halfband filter. Similarly, the lowpass halfband filter is formed combining the low pass decimation and reconstruction filter.

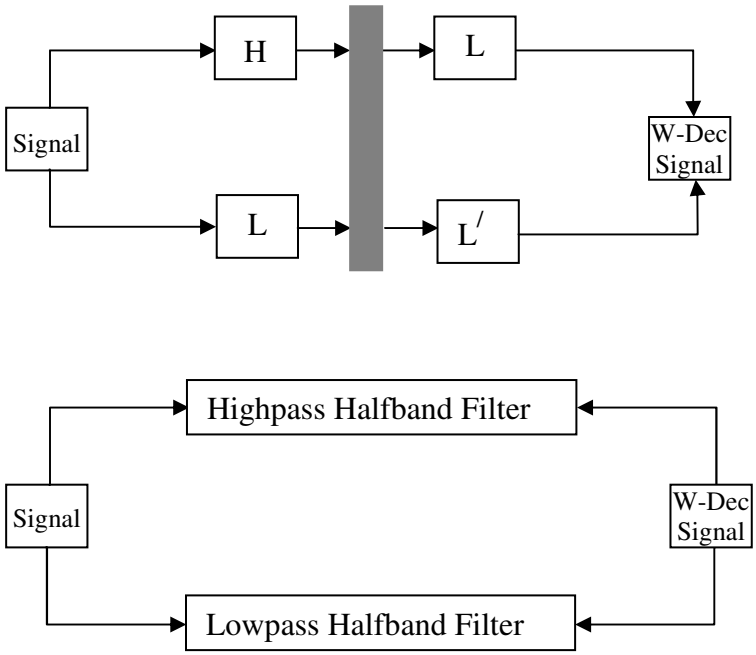


Figure 12: The block diagram of undecimated wavelet transform

To add translation-invariance in DWT, MODWT was introduced which instead of down sampling and up sampling the signal introduces high and low pass filters with up sampled by a factor of 2^{j-1}. The up sampling of filters also introduces redundancy in the

output but now the number of samples at output in every level is equal to the number of samples in the input signal. This makes multiresolution analysis much more effective especially from the perspective of using this transform as an input to another system.

$$d_{j,n}^{(M)} = \sum_{l=0}^{L-1} \tilde{g}_l \, c_{j-1,n-2^{j-1}l \, modN}^{(M)}$$

$$c_{j,n}^{(M)} = \sum_{l=0}^{L-1} \tilde{h}_l \, c_{j-1,n-2^{j-1}l \, modN}^{(M)}$$

The figure 13 shows the process diagram of the decimated discrete wavelet transform.

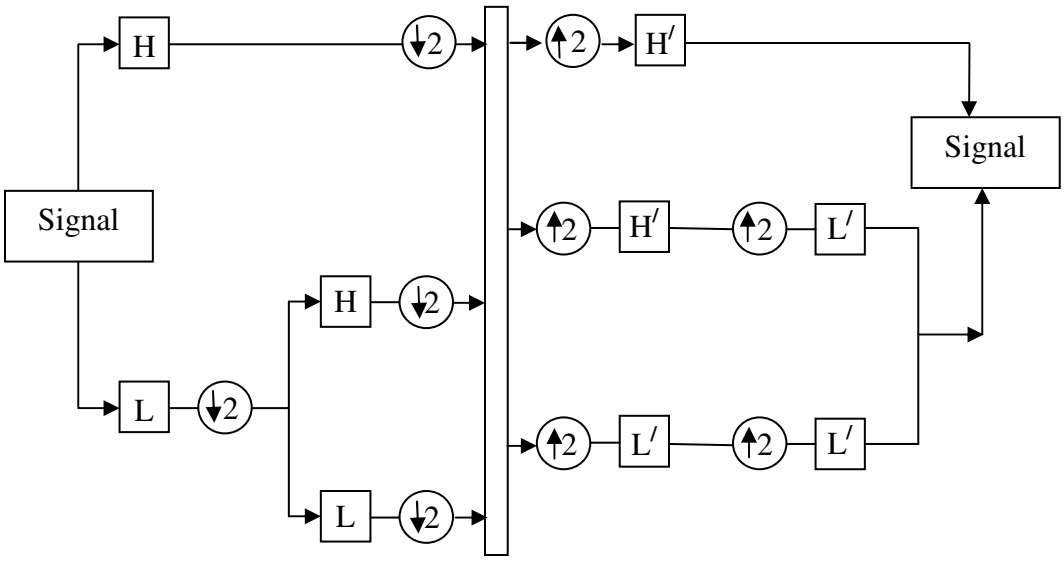


Figure 13: The block diagram of decimated wavelet transform

The filters can now be represented as a circular filter of original time series.

$$d_{j,n}^{(M)} = \sum_{l=0}^{L_j-1} \tilde{g}_{j,l} x_{n-l \, mod \, N}$$

$$c_{j,n}^{(M)} = \sum_{l=0}^{L_j-1} \tilde{h}_{j,l} x_{n-l \bmod N}$$

4.2 Wavelet Packet Decomposition

Finally, to generate the wavelet packet tree, both the approximation and detail coefficients are decomposed instead of just the approximation coefficients as in the case of the DWT. This decomposition of the low pass component into approximation and details help us in cases where the energy levels in low pass signal are very high and they need to be pre-processed into multiple bands of lower energy levels.

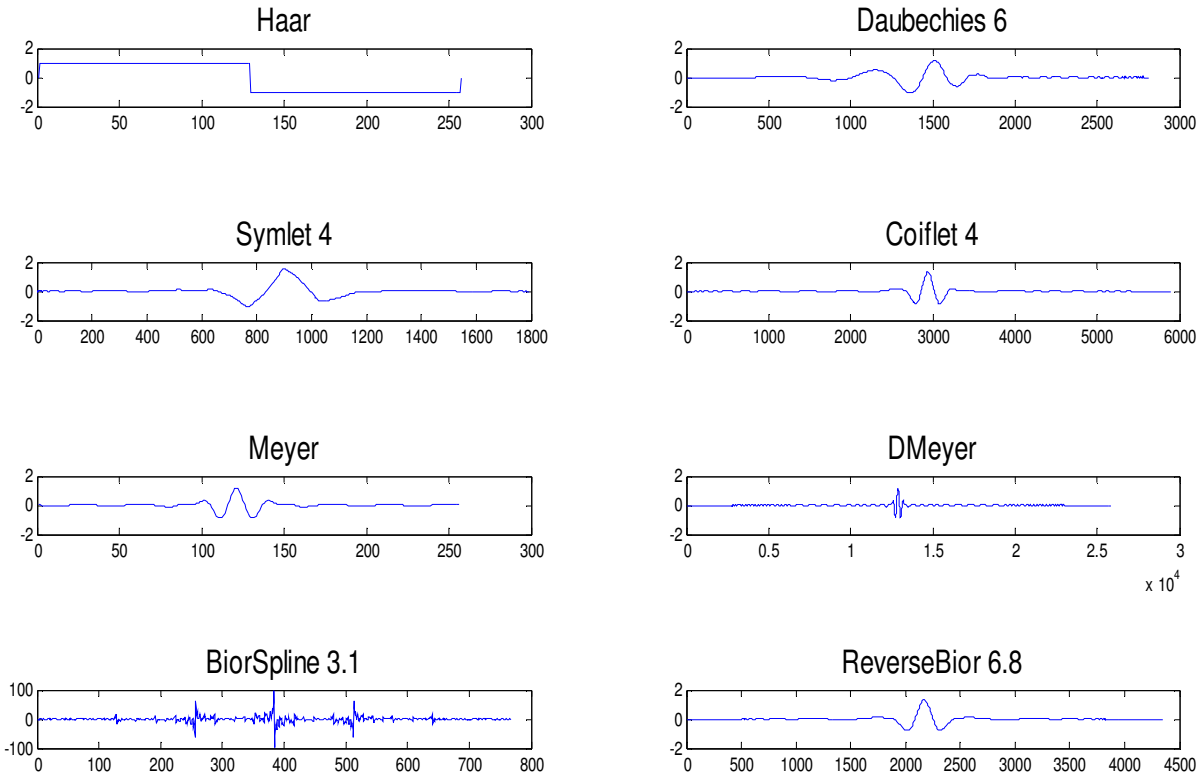


Figure 14: Plot of mother wavelet of different wavelet types

Figure 15 shows how both the high frequency component and low frequency components are filtered through the wavelet and scaling filters and then the reconstruction filters are applied in the reverse order to reconstruct the original signal.

Hence the wavelet packet distributes the frequency of the original signal evenly between all coefficients as opposed to the wavelet transform where 50% of the signal frequency is in the first detail as shown in figure 16 a and b. It is worth mentioning here that the significant portion of the energy of the original signal remains in the low pass component of the wavelet packet decomposed signal also and the energy is not divided evenly in the sub-bands.

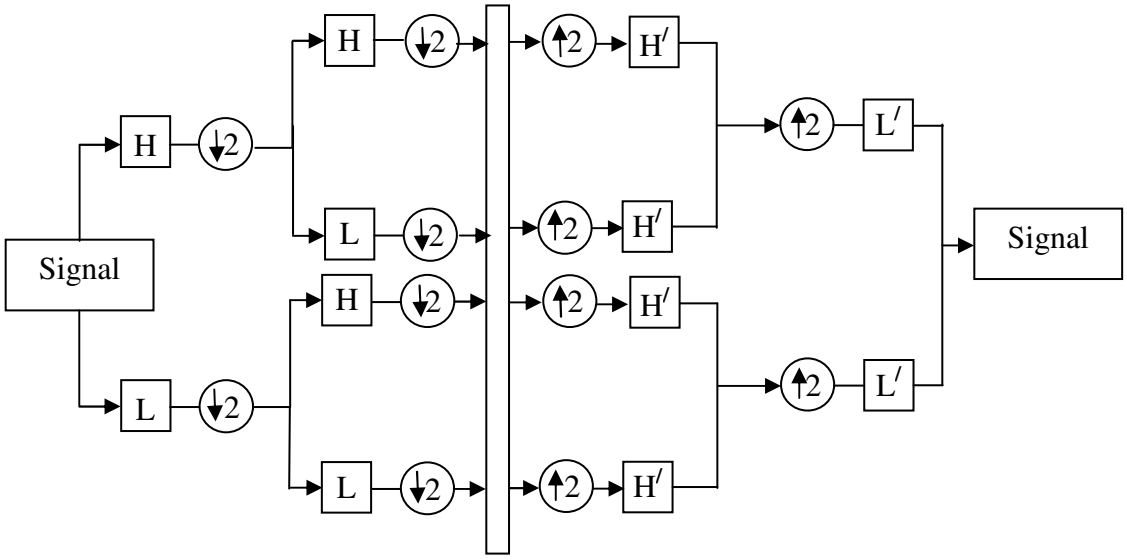


Figure 15: The wavelet packet decomposition structure at level 2

In our model also, we chose the wavelet packet transform to evenly distribute the signal frequency in each WDSVR module.

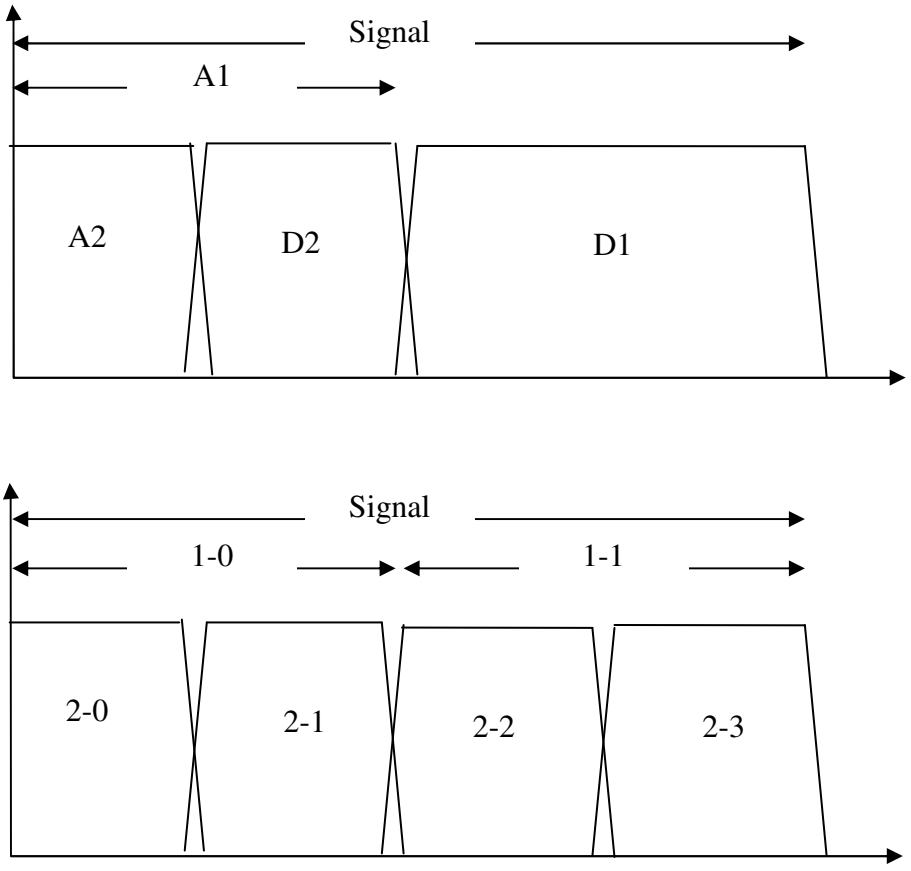


Figure 16: a) Frequency distribution of Wavelet transform **b)** Frequency distribution of Wavelet Packet transform

4.3 Biorthogonal and Reverse Biorthogonal Wavelets

The biorthogonal filters are designed in a set of four different wavelets viz. Highpass decomposition and reconstruction filter and Low pass decomposition and reconstruction filter represented by H, H', L and L' respectively.

In Biorthogonal filters are generated from two sets of wavelets $\psi(t)$ and $\tilde{\psi}(t)$ which span two different subspaces W and $\widetilde{W}$ respectively. Since the bases are orthogonal to each other; the two MRAs are said to be *biorthogonal* to each other.

Reverse Biorthogonal filters are similar to biorthogonal filter except that the H and H' and L and L' are swapped. Since, $conv(H, H') = conv(H', H)$ and $conv(L, L') = conv(L', L)$, therefore the halfband highpass and lowpass filters remain unchanged.

When we translate the effect of filter swapping as in the case of biorthogonal and reverse biorthogonal filters, the values of high pass reconstructed sub-bands have a visible effect. This difference in reconstructed values gives different results of forecasted values. An analytical approach to the selection of biorthogonal and reverse biorthogonal for wavelet decomposed support vector regression is not possible. The details of the same are discussed in chapter 7.

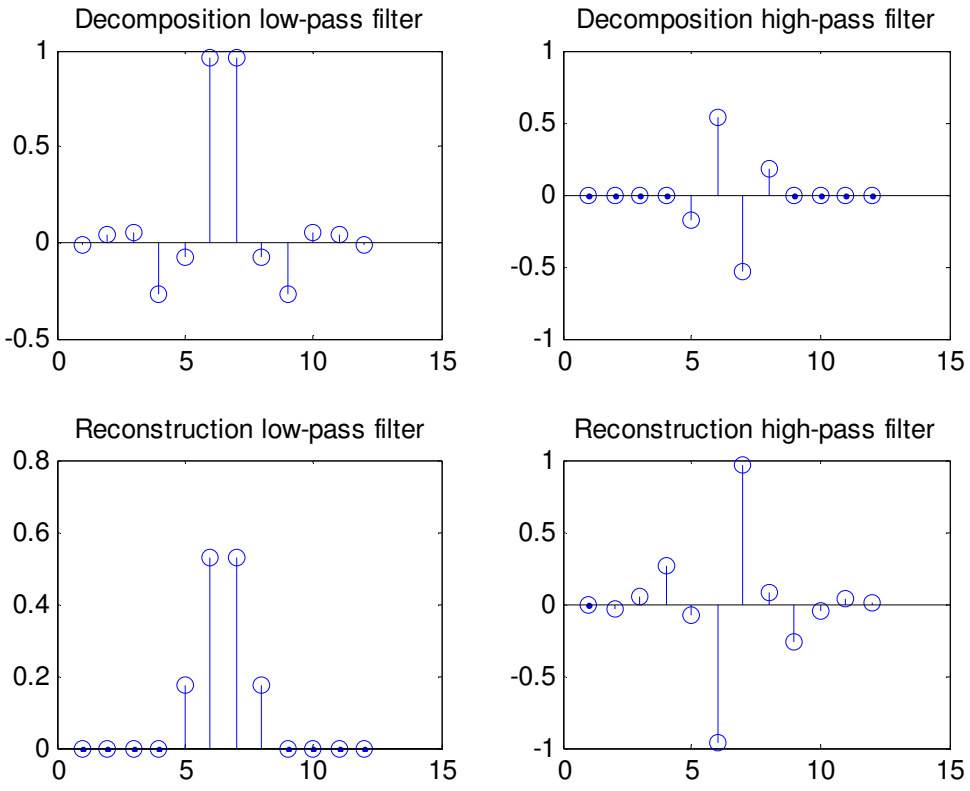


Figure 17: The filter coefficient values of Biorthogonal 3.5 wavelet

Hence the biorthogonal and reverse biorthogonal filters have the similar filters but based on the information content in the low and high frequency bins of the original data, there results would be different.

In figure 17 and 18 the biorthogonal filters 3/5 and reverse biorthogonal filters 3/5 are shown. Note that the high pass reconstruction filter and high pass filter of biorthogonal 3/5 are swapped when compared with reverse biorthogonal 3/5. Similarly, the low pass filter and low pass reconstruction filter of biorthogonal 3/5 are also swapped in comparison with reverse biorthogonal 3/5.

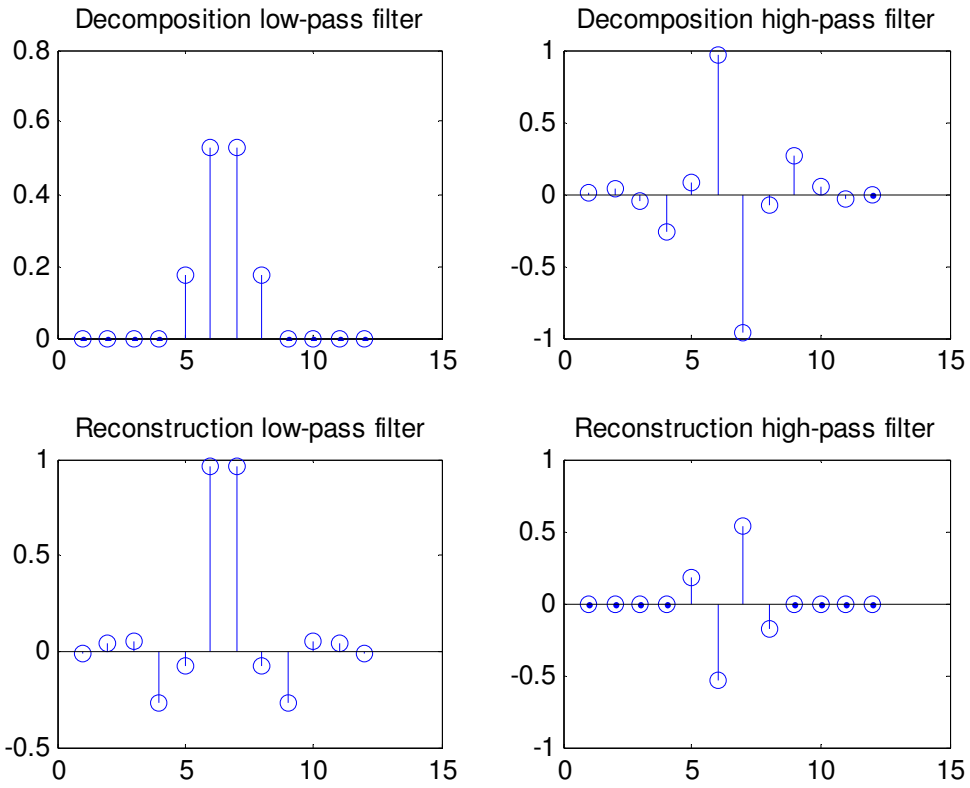


Figure 18: The filter points of reverse biorthogonal 3/5 filter

In this chapter the basics of wavelet theory were established and the details of biorthogonal and reverse biorthogonal filters were explored. Now we would move towards the concepts of support vector machines, which would enable us understand the wavelet decomposed support vector regression model.

5 SUPPORT VECTOR MACHINE

Support vector machines (SVM) are kernel machines that implement maximum margin methods. The maximum margin is generated by the kernel using a set of weighted vectors of training data called support vectors.

SVM uses the quadratic approach to define the problem of maximizing separability between classes. The margin is subject to constraint of the smoothness of the solution. The input data corresponds to a member of the class set, which is generally called the label vector. The hyperplane is created between the members of the class set to define the boundaries of each class. The support vector machine was initially built to solve the binary classification case. However, it was later extended for multiclass classification and prediction problems. To have a better understanding of the SVM we have explained it in a step by step case starting with the formulation of the basic concept and then moving to the linearly separable case. Eventually the non-separable case and regression problem is explained.

5.1 Formulation of the SVM

We would start the formulation considering the simple binary classification case. The input data in the form of $X = \{x, r\}$, where X either corresponds to $r = -1 \; \forall \; x \in C_1$ or $r = +1 \; \forall \; x \in C_2$. The goal is to find a hyperplane w it divides the two classes such that the distance from the hyperplane to the nearest data point of each class is equal from the hyperplane.

Mathematically the problem can be represented as

$$f(x) = \sum_{i=1}^{N} w_i \phi_i(x) + b \tag{6}$$

where, $w_i \phi_i(x) + b \geq 1 \; \forall \; r \epsilon C_1$ and $w_i \phi_i(x) + b \leq -1 \; \forall \; r \epsilon C_2$. This can be rewwritten as

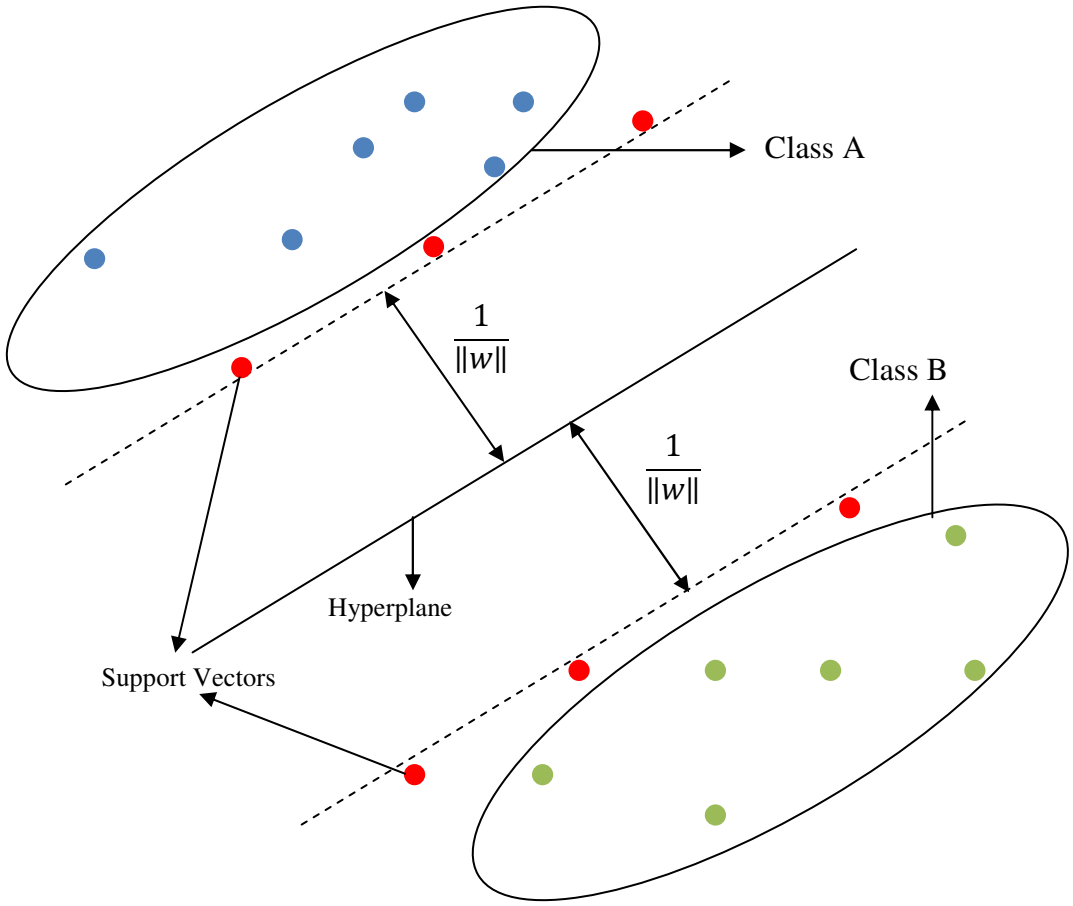


Figure 19: Main components of a binary Support Vector Machine

Now to calculate the distance of all x to the margin we take the discriminant approach. Thus,

$$\frac{r_i(w_i x_i + w_0)}{\|w\|} \geq \rho, \forall \; i$$

Now we add the condition of maximum margin by constraining the solutions to obey $\|w\| \; \rho = 1$. Hence, to maximize the margin we minimize $\|w\|$

$$\frac{1}{2}\|w\|^2 + r_i(w_i x_i + w_0) \geq 1$$

This is now the quadratic optimization problem, whose solution would give a $\frac{2}{\|w\|}$ margin with $\frac{1}{\|w\|}$ on either side of the hyperplane. The concept is explained with the help on an example below:

5.2 Linearly Separable Case

Let us consider a set of data points X=[{(1,1),-1},{(1,-1),-1}, {(0,0),-1}, {(-1,-1),-1}, {(3,0),1}, {(4,-1),1},{(4,1),1},{(5,0),1}]

Plotting the values in figure 20, the support vectors are shown circled in yellow.

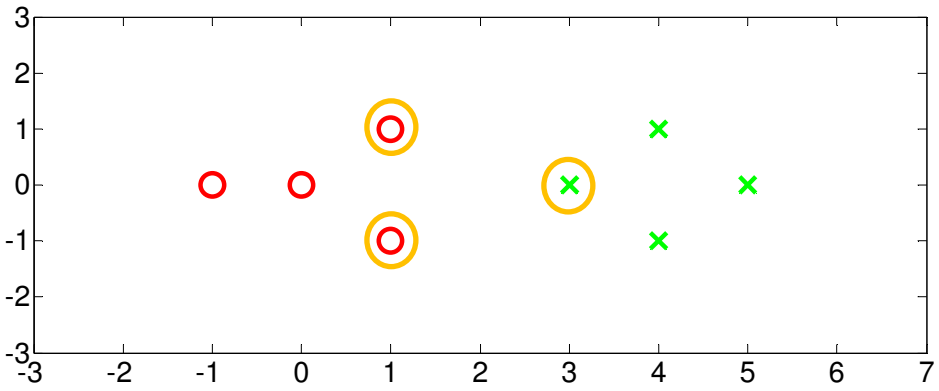


Figure 20: Sample data-points of a linearly separable SVM classification case

Since, the classes are linearly separable we need a hyperplane to optimally divide the two class zones. From the figure it is clear that the yellow marked points are the support vectors. These three support vectors would contribute towards the formulation of the hyperplane.

The value of the hyperplane line w is calculated by adding an offset of 1 to the support vectors. Hence, the margin value is calculated by solving the three equations of the support vectors.

$$\alpha_1 < s_1, s_1 > +\alpha_2 < s_1, s_2 > +\alpha_3 < s_1, s_3 \geq -1$$

$$\alpha_1 < s_1, s_2 > +\alpha_2 < s_2, s_2 > +\alpha_3 < s_2, s_3 \geq 1$$

$$\alpha_1 < s_1, s_3 > +\alpha_2 < s_2, s_3 > +\alpha_3 < s_3, s_3 \geq 1$$

In SVM implementation this margin is solved using Langrangian multipliers, the solution of which renders the alphas relating to input values, which do not contribute to the support vectors machine to be zero. In other words only the alphas relating to the support vectors would have non zero values. Solving the equations we get

$$3\alpha_1 + 1\alpha_2 + 4\alpha_3 = -1$$

$$1\alpha_1 + 3\alpha_2 + 4\alpha_3 = -1$$

$$4\alpha_1 + 4\alpha_2 + 10\alpha_3 = 1$$

Solving the above equations we get the values of $\alpha_1 = -1.75, \alpha_2 = -1.75$ and $\alpha_3 = 1.5$

Now to determine the hyperplane we take the affect of all 3 support vectors.

$$w = \sum_i \alpha_i s_i$$

$$= -1.75\begin{pmatrix}1\\1\\1\end{pmatrix} - 1.75\begin{pmatrix}1\\-1\\1\end{pmatrix} + 1.5\begin{pmatrix}3\\0\\1\end{pmatrix}$$

$$= \begin{pmatrix}1\\0\\-2\end{pmatrix}$$

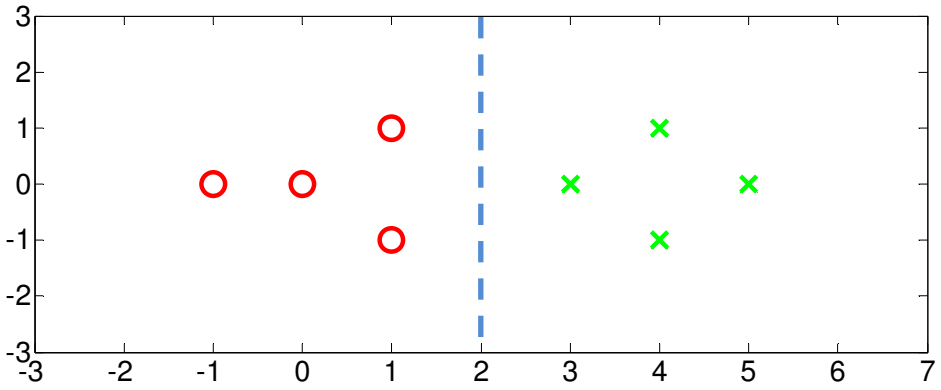


Figure 21: The hyperplane of the linearly separable SVM case

Now the hyperplane can be represented with $w = (1,0)$ and $b = -2$. The line representing these values is plotted in figure 21, which represents the optimal hyperplane between the two classes.

The margin using support vectors is implemented in SVM using Langrangian multipliers.

$$L_p = \frac{1}{2}\|w\| + \sum_i \alpha_i - \sum_i \alpha_i r_i (w^T x_i + b)$$

The dual problem is to maximize L_p subject to the constraint that gradient of the langrangian with respect to the hyperplane and the constant both are zero. The values of alphas also must be greater than or equal to zero.

$$\frac{\partial L_p}{\partial w} = 0 \Rightarrow w = \sum_i \alpha_i r_i x_i$$

$$\frac{\partial L_p}{\partial w_0} = 0 \Rightarrow \sum_i \alpha_i r_i = 0$$

Substituting the above values into equation we get

$$L_p = -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j r_i r_j x_i x_j + \sum_i \alpha_i$$

Using quadratic optimization methods we can solve the above equation. Now we have the complexity of the problem is dependent on the sample size and not on the input dimensionality. Since, the support vectors are only those values, which have a positive alpha values. Therefore, the computation requirement is further reduced.

In the above example, the dual Langrangian is maximized subject to $\alpha_i \geq 0$. However, the data points which are not on the class margin would have zero alpha values and hence would not effect the final value.

5.3 Non- separable Case

Now we would consider the case where the boundaries of the binary classes are not distinct. In other words the case where the data points of both classes are overlapping. In such cases, the data points are projected into feature space to attain linear seperability. Now in equation we would have to induce the soft error variable ξ

$$r_i(w_i x_i + w_0) \geq 1 - \xi_i$$

So, in order for the SVM to classify the data point in the soft margin the range of the soft error ξ_i must lie between 0 and 1. $\forall\, \xi_i \geq 1$ the data point is misclassfifed. The total soft error is sum of errors of the individual soft error of each x_i Hence, adding this penalty term

$$L_p = \frac{1}{2}\|w\| + C\sum_i \xi_i,$$

where C is the penalty constant, which represents the tradeoff between the complexity and accuracy. With the above mentioned constraint the Langrangian now becomes

$$L_p = \frac{1}{2}\|w\| + C\sum_i \xi_i, - \sum_i \alpha_i[r_i(w^T x_i + b) - 1 + \xi_i] - \sum_i \mu_i \xi_i \qquad (7)$$

Note that other than the addition of the ξ_i there is a new term μ_i, which is to ensure the positivity of ξ_i. Similar to the linear separable case we now maximize the Laplacian wrt w, ξ_i and $w_0$to rewrite (7) in terms of α_i, r_i and x_i. The gradient gives us the following terms

$$w = \sum_i \alpha_i r_i x_i \qquad (8)$$

$$\sum_i \alpha_i r_i = 0$$

$$C - \alpha_i - \mu_i = 0$$

Since, we introduced $\mu_i \geq 0$, it implies that $0 \leq \alpha_i \leq C$. Now maximizing w.r.t. α_i we again get

$$L_p = -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j r_i r_j x_i x_j + \sum_i \alpha_i$$

5.4 SVM Kernel

The real world classification problems mostly belong to the non-seperable class or the soft margin case. The increased number of misclassified data reduces the effectiveness of the SVM method. To overcome this issue the support vector machine incorporated idea of projecting the data from input space to another non-linear feature space. The idea behind projection of data into another space was to address the data points which were pre-

viously lying in the other class range to be projected in the region of their own class boundary. Such a projection would require computation of every data point to get an equivalent value in the feature space using the basis functions of the feature space.

Consider the new dimension z calculated through basis function $\emptyset$. Now the discriminant in feature space z is represented as

$$f(z) = \sum_{j=1}^{M} w_j \ \emptyset_j(x) \tag{9}$$

Comparing (9) with (6) we observe that w_0 is missing from (8). This is because the dimensionality M of feature space z is much larger than the input space N. On the other hand the complexity of the problem also increased from N to M.

Using the same method of Langrangian multipliers we computed the dual, which became

$$L_d = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_s \alpha_i \alpha_s r_i r_s \emptyset(x_i)^T \emptyset(x_s).$$

The computation of $\emptyset(x_i)^T \emptyset(x_s)$ is computational intensive. Kernels were brought in to replace this computation. With kernels instead of mapping the function using the basis function a much simpler kernel function in the original input space computes the value of the projected input in the feature space. Hence the dual now becomes

$$L_d = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_s \alpha_i \alpha_s r_i r_s K(x_i, x_s).$$

Comparing equation (8) and (9) the discriminant can be rewritten as

$$f(z) = \sum_i \alpha_i r_i \emptyset(x_i)^T \emptyset(x_i)$$

Since

$$K(x_i, x_s) = \emptyset(x_i)^T \emptyset(x_i),$$

therefore,

$$f(z) = \sum_i \alpha_i r_i K(x_i, x_s)$$

This means that the step of projecting the input into feature space through basis function is now done using the kernel. This is the reason for naming this method as vector machine method. The kernel $K(x_i, x_s)$ is represented as the symmetric Gram matrix.

5.5 SVM for Regression

The support vector machine was extended for solving prediction problems using non-linear regression techniques. The regression method is curve fitting method, which uses some goodness of fit criterion. Linear regression is the most common form of regression which uses the least squares method as a goodness of fit criterion. Employing linear regression to our problem would yield an error

$$\varepsilon\big(r_i, f(x_i)\big) = [r_i - f(x_i)]^2.$$

This quadratic error is sensitive to outliers. To control the effect of outliers using the linear regression method the data would have to be filtered for the outlier values. The method itself treats noise and real inputs linearly for errors. On the other hand SVR incorporates a linear approach for error calculation. This not only makes the SVR robust against outliers but it also uses an ϵ-insenstitve error function which makes the goodness of fit function insensitive to small errors in the range of ϵ set by the user, The ϵ-insensitive loss function is defined as

$$\varepsilon(r_i, f(x_i)) = \begin{cases} 0 & if\ |r_i - f(x_i)| < \epsilon \\ |r_i - f(x_i)| - \epsilon & otherwise \end{cases}.$$

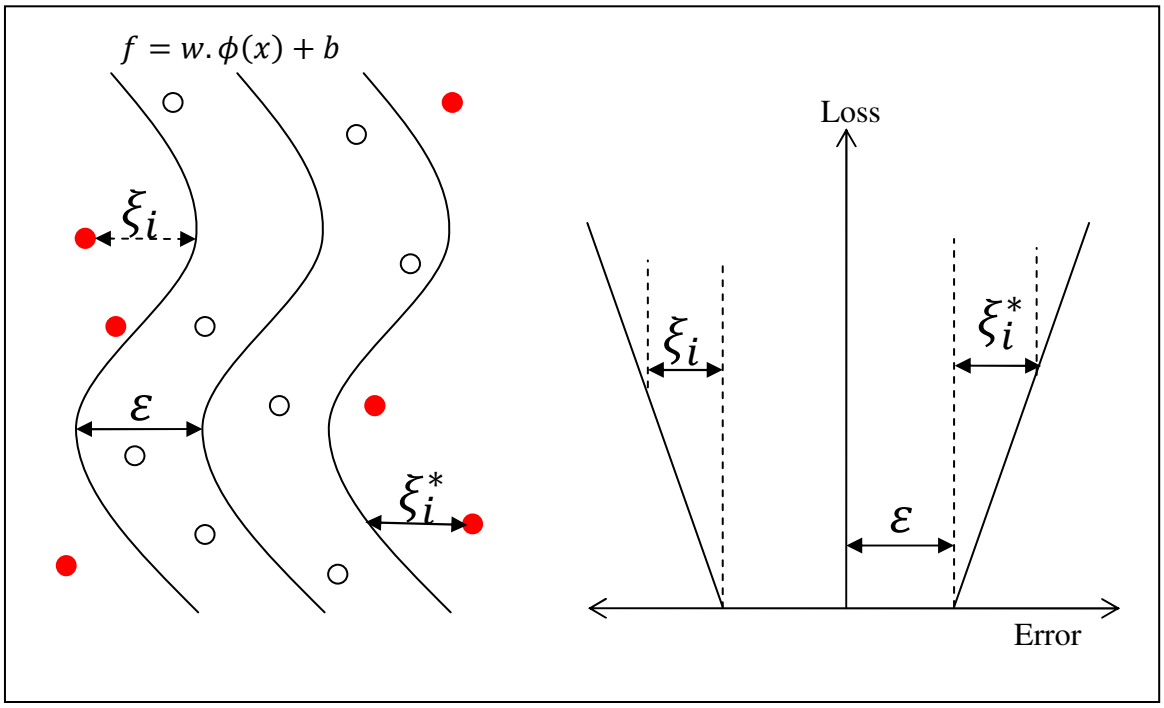


Figure 22: The ϵ-sensitive support vector machine for regression

The ϵ-sensitive loss function defined above gives the range of error $\mathcal{E}$ in which the SVR hyperplane is not effected by the error and the linear error term which comes into play outside the ϵ boundary.

To account for the errors outside the ϵ boundary two slack variables ξ_i^+ and ξ_i^- to measure the error of data points outside the ϵ boundary in both positive and negative directions respectively. The minimization function now becomes

$$min\frac{1}{2}\|w\|^2 + C\sum_i \xi_i^+ + \xi_i^-$$

subject to

$$r_i - (w^T x + b) \leq \epsilon + \xi_i^+$$

$$(w^T x + b) - r_i \leq \epsilon + \xi_i^+$$

$$\xi_i^+, \xi_i^- \geq 0$$

Similar to the classification problem now we minimize the Langrangian wrt w, b, ξ_i^+ and ξ_i^- to simplify the equation in terms of langrangian multipliers. The Langrangian dual now becomes

$$L_p = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i^+ + \xi_i^- - \sum_i \alpha_i^+ [r_i - (w^T x + b) - \epsilon - \xi_i^+]$$

$$- \sum_i \alpha_i^- [(w^T x + b) - r_i - \epsilon - \xi_i^-] - \sum_i \mu_i^+ \xi_i^+ + \mu_i^- \xi_i^-$$

Partial derivates wrt w, w_0, ξ_i^+ and ξ_i^- are

$$\frac{\partial L_p}{\partial w} = w - \sum_i (\alpha_i^+ x + \alpha_i^-)\, x_i \stackrel{\supset}{\Rightarrow} w = \sum_i (\alpha_i^+ - \alpha_i^-)\, x_i$$

$$\frac{\partial L_p}{\partial b} = \sum_i (\alpha_i^+ - \alpha_i^-)\, x_i = 0$$

$$\frac{\partial L_p}{\partial \xi_i^+} = C - \alpha_i^+ - \mu_i^+ = 0$$

$$\frac{\partial L_p}{\partial \xi_i^-} = C - \alpha_i^- - \mu_i^- = 0$$

The dual is now

$$L = \frac{1}{2}\sum_i \sum_s (\alpha_i^+ - \alpha_i^-)(\alpha_s^+ - \alpha_s^-)\, x_i x_s - \epsilon \sum_i (\alpha_i^+ + \alpha_i^-) - \sum_i r_i(\alpha_i^+ + \alpha_i^-)$$

subject to

$$0 \leq \alpha_i^+ \leq C$$

$$0 \leq \alpha_i^- \leq C$$

$$\sum_i (\alpha_i^+ - \alpha_i^-) = 0$$

Now the support vectors are determined for all values of α_i^+ and α_i^- greater than or equal to C. The support vectors are the weighted sum of feature space in terms of langrangian multipliers

$$f(x) = \sum_i (\alpha_i^+ - \alpha_i^-) x_i^T \, x_i + b.$$

Here $x_i^T x_i$ can be replaced with the kernel.

The SVR algorithm works as a regression problem of a data projected in the feature space. The support vectors are formed for data points outside the ε boundary. The cost value determines the amount of deviation caused by the outlier. A higher cost value would also cause overfitting of the data, while a lower cost would underfit the outlier values. In support vector regression the boundaries formed by the support vectors in the classification case are replaced by data values for the outliers.

6 TRANSFORM BASED MACHINE LEARNING MODELS

Researchers have tried multiple transforms for the purpose of preprocessing the input data for machine learning methods. The objective of processing the data can have multiple objectives depending upon the transform applied and type of dataset.

Machine learning methods for preprocessing the data using transforms were used for the purpose of achieving improved accuracy for compression, classification and forecasting. The major transforms which were used to enhance the features of the dataset or alternately reduce its dimensionality were the S-transform, Discrete Cosine transform, K-L transform, Contourlets, Hough transform and Stanlet transform etc.

The transform could only be effective if the transformed inputs exploit the properties of the underlying machine learning method to make them more efficient. This could be achieved by decomposing the transform and remove the noisy component from the input data, or by separating the noisy and periodic and trend components. The type of transform to be used depends on the properties of the dataset and the machine learning algorithm.

For the purpose of this thesis we have only focused on the transforms used in conjunction with the support vector machines for regression. Similar models have also been reported using neural networks as the machine learning method.

Below we have discussed a few models, which were used and published in scholarly journals and conference proceedings related to support vector machines.

6.1 Transform based SVM for Forecasting Problem:

The forecasting models based on transformed time series and support vector regression focus on the division of information from one time series to multiple time series in such a way that the combined errors of the transformed SVR model is lower than the individual support vector regression machine. This condition is not trivial to achieve, therefore, models in different configurations have been applied to achieve this condition.

6.1.1 Discrete Wavelet Transform and SVR

The wavelet transform projects the input data in the time-frequency space. In time frequency space the wavelet decomposed data represents the frequency related events, while keeping the time information. The purpose of dividing the data into multiple projections is to distribute the frequency of the incoming signal into frequency bins for better resolution. Not every wavelet basis has the ability to achieve the condition mentioned in (10). The selection of wavelet basis and the configuration of the different wavelet transform based support vector machine models are described below.

6.1.1.1 Configurations of Wavelet Transform and SVR Models

The wavelet transform based SVR models are developed in different configurations as shown in figure 23. The two main types are the wavelet kernel based SVR and the wavelet transform based SVR.

Wavelet Transform based SVR The wavelet decomposed method decomposes the input into wavelet domain before sending the signal to the support vector machine. Each decomposed signal has a separate support vector machine associated

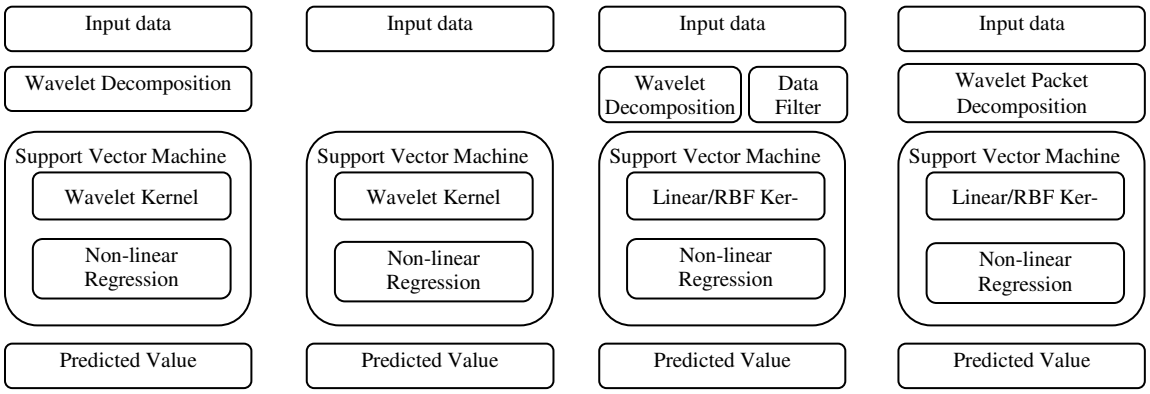
with it. It is not necessary to use all decomposed inputs as input to the SVR. The wavelet reconstructed signal, which decomposes the signal with maximum frequency normally does not follow any specific pattern. Researcher have used averaging or similar filters to be used as its substitute [57].

The wavelet decomposed SVR method has the capacity to use all wavelet basis for the decomposition phase. If a wavelet selection criteria is determined, the wavelets selection could be made based on that criteria.

Wavelet Kernel based SVR SVR with Wavelet kernel is another method used to take advantage of the wavelet transform without decomposing the input into wavelet domain. The support vector machine projects the input data into the feature space. The function used to project the input space into feature space is the support vector machine kernel. These kernels are functions for projecting the input space into feature space with some additional limitations discussed in Chapter 5. Lately wavelet based kernels have been developed which project the input space into wavelet space for classification/prediction purposes. The kernels are interchangeable; with different functional projections through different wavelets. However, SVM is limited by the choice of wavelets, which could be designed into support vector kernels.

To overcome this shortcoming wavelet decomposition and wavelet kernel were used together as shown in figure 23a below. However, the wavelet transform of wavelet

projected data could not improve the accuracy over the wavelet decomposed RBF kernel method.



(a) (b) (c) (d)

Figure 23: Different configurations of wavelet transform and support vector regression models

6.2 Transform based SVM for Compression Problem:

The concept of using support vector machine for data compression is based on the sparse representation property of the SVM. Support vector macines model the data with a minimum number of support vectors for a predefined accuracy. The transform is used to further simplify the data for the support vector machines.

6.2.1 Wavelet based SVR for Compression

The support vectors machines use support vectors and their corresponding weights to train the input data. For a predefined level of accuracy the support vector machine has the ability to model the data using a minimum number of support vectors. The number of support vectors depends on the properties of the input data and the values of cost and epsilon set for the support vector machine.

The model for the image compression follows a similar pattern as shown in figure 23c. The input image is divided into tiles and the 2D wavelet transform using a suitable wavelet is computed. The wavelet reconstructed coefficients are then trained using their individual support vector machines which model the input wavelet coefficients based on the parameters set for the SVM. The trained data for the individual support vectors is represented in the form of support vectors and their corresponding weights, which is compressed using a quantization encoding method. [58] used Huffman coding to compress the support vector values and their corresponding weights. [59] also uses a similar models but uses the Daubechies 9/7 instead of the Haar wavelet and instead of using SVMs for all high and low frequency coefficients used Diffential Pulse-code Modulation to encode the highest frequency data of the wavelet reconstructed coefficients. [60] also used the similar model but used multiple kernels to analyze their accuracies in terms of compression ratio and PSNR of the compressed images.

6.2.2 Curvelet based SVR for Compression

The Curvelet transform is a multiscale directional transform, which efficiently computes sparse representation of geometric objects with singularities. Unlike the wavelet transform, Curvelets have the ability to vary their degree of localization in orientation with scale. Therefore, they are more effective to sparsely represent data which is globally geometric with important singular information at high frequencies.

Using this property of sparse representation, curvelets were used to compress images. The model displayed at figure 23c was used by [61, 62] where the input image was

transformed into curvelet reconstructed coefficients before passing to the support vector machine for regression to compress the data.

6.2.3 DCT based SVR for Compression

The discrete cosine transform was used in [63] to compress image data in similar configuration as other transforms. The discrete cosine transform of an image maps the pixel values of the image from the spatial domain to the frequency domain. DCT diminishes the effect of higher frequencies in the image, hence reducing the image size. However, the effect of this removal of high frequencies is not visible to the naked eye. The coefficients of the discrete cosine transform are compressed using the SVM as it has the ability to model large dataset with sparse support vectors.

6.3 Transform based SVM for Classification Problem:

Support vector machines were originally designed for binary classification. Their ability to classify data by projecting it into a high level feature space using kernel method made them popular among researchers. The concept of binary classification was extended to multiple classification and regression problems. To further improve the accuracy of support vectors the input data is transformed into a specific domain to further increase the distance between multiple classes.

6.3.1 Fourier based SVR for Classification

Fourier transform gives the frequency content of the signal in the frequency domain. However, the spatial information of the signal is lost in the transformation process. Short time fourier transform was used to preserve the approximate spatial location with the fre-

quency transformation. [64] used a short time fourier transform to localize the frequency information and then using support vector machines classified the input signal as faulty or otherwise. The fourier transform has its significance in the frequency domain but when spatial information is also required in the classification system, wavelets become more useful.

6.3.2 Wavelet based SVR for Classification

Preprocessing the data into wavelet domain before classification through SVM is similar to the model explained in Section 6.1.1 except the objective of the SVM is not compression or prediction but classification. [65-69] used wavelet transform on the input data to extract features from the data set. These features were then given as an input to the support vector machine for classification of the signal.

6.3.3 PCA based SVR for Classification

Principal component analysis is used to orthogonalize the input data into convert a correlated data set into uncorrelated input. The classification of support vector machine would improve with uncorrelated dataset as there would be more distance between the different features.

[70, 71] transformed the input spectroscopy signals into principal components and then used them into the support vector machine for classification. [72] initially used the wavelet transform to convert the input data into wavelet coefficient data and then computed the principal components to uncorrelated the data before classifying it into out classes using SVM.

6.3.4 Slantlet Transform based SVR for Classification

Slantlet transform is an orthogonal DWT with two zero moments and improved time localization. The Slantlet transform unlike the wavelet transform does not have shifted and scaled versions of the mother wavelet, but the Slantlet filters use different filters on each scale. Hence, Slantlet filters have more degrees of freedom as compared against the wavelet transform. Due to this property of the Slantlet transform the filters at higher scales can be implemented using shorter supports, while keeping all the abilities of the equivalent wavelet filter.

The Slantlet filters are used to compute the features from the input data. [73] used this transform to extract features from the intensity histogram of MRI images. These features are used for the input to the SVM for classification.

6.4 Transform based SVM for Denoising:

The denoising is based on the concept of filtering out the frequency components in a specific range. Usually noise component belongs to the high frequency range. The transforms are used to divide the image or data into frequency bins. Subsequently the high frequency bins are filtered out to produce denoised data.

6.4.1 Slantlet Transform based SVR for Denoising

[74] proposed a model using slantlet transform for denoising. However, the noisy component was not removed from the dataset. The SVR has the ability to model the stochastic data using support vectors. By setting low values for epsilon and high cost values more support vectors can be forced to create, which would over fit the data. However, if

the training data gives a complete or near complete range of the input data and the testing dataset follows a similar pattern of input data then there a good chance that the SVR model would produce improved results for forecasting the stochastic process.

A similar model was followed in [74] where the dataset was decomposed using the slantlet transform into deterministic and stochastic components. The deterministic component was forecasted using the ARMA model and the stochastic component was sent to the SVR for forecasting the next value. Both deterministic and forecasted values were added to produce the final results.

6.4.2 Wavelet Transform based SVR for Denoising

The wavelet transform as a denoising method was used in [75, 76]. The input signal was first decomposed into wavelet coefficients and the high frequency components i.e. the detail coefficients were thresholded based on adaptive or fixed thresholding to remove the noisy component in the signal. The resultant signal was modeled using the support vector machine for regression. The output of the signal was passed through the inverse wavelet function to recover the denoised data.

6.4.3 Empirical Model Decomposition based SVR for denoising

Empirical model decomposition is used to transform the input data into the energy-frequency-time domain by using finite and small number of intrinsic model functions. [77] used the empirical mode decomposition method to decompose the traffic data into frequency bins. The noisy component or the high frequency component was removed and

the remaining was data was reconstructed to get the denoised traffic data in the spatial domain. The future value was then forecasted by the SVM for regression.

This method is significant for noisy data. Traffic data on the other hand represents non-linear and non-stationary patterns in all frequency range. The so called noisy component is also representing an event on the roadway. To remove it from the SVM input effectively, reduces the workload of the SVM as we are now giving less information to process in the machine learning module. A similar approach was demonstrated in [78] where the high frequency component was not removed but each frequency bin was allocated a separate SVM to predict the future value.

7 OPTIMAL WAVELET SELECTION

In this chapter we define the wavelets selection process. The wavelet selection process is based on the selection of mother wavelet, the level at which the wavelet filter decompose the signal and the type of wavelet decomposition. The factors involved in selecting the wavelets are discussed for both stand-alone wavelet models and the wavelet based machine learning models.

7.1 Selection of Mother Wavelet

The wavelet library holds many wavelet families. The choice of wavelet for a given model holds significant value as the wavelet decomposition using the complete set of wavelets is a time consuming and computational intensive task. Generally wavelet selection is based on computing the data using a set of popular wavelet like daubechies, haar etc. and based on the results the optimal wavelet is selected. However, using this hit and try method a generalized rule for wavelet selection cannot be made. Hence for every new dataset the procedure would have to be repeated again to select the best performing wavelet. Nevertheless, different approaches have been adapted for selecting wavelets for wavelet based models.

To describe the wavelet selection process we would classify the wavelet based methods based on the configuration in which they are used. The wavelet based methods are either used alone or in conjunction with another machine learning method. For simplicity we have named the standalone wavelet method as the wavelet based method and

the one which is used in conjunction with a machine learning module as the wavelet based machine learning method.

7.1.1 Selection of Mother Wavelets in Wavelet based Machine Learning Models

The selection of an exact method for optimal mother wavelet selection is not reported in literature for wavelet decomposed support vector regression methods. This is due to the black box behavior of machine learning methods on non-linear data. The support vector machine however, works on a more intuitive concept of structural risk minimization. The user has the capacity to change the kernels to improve the results of the objective function. Wavelet kernels have already been designed for use in support vector machines. The kernel which decomposes the dataset into feature space is based on wavelet transform. The dataset is still non-linear and non-stationary and we do not know the statistical properties of the wavelet decomposed signal. Therefore an initial decomposition of the training data using all possible wavelets is still required.

A wavelet selection scheme was proposed in [79], which used the genetic algorithm to select the mother wavelet and parameters of the SVM based on the training data. Such approaches loose the prime objective of saving computational cost as in general 70% of the data is used in training and genetic algorithms by their implementation are iterative. [80] also concluded that the absence of any analytical justification for selection of suitable wavelet for classification of data using wavelet based SVM classification method. The choice was made by experimenting with different wavelet families and based on the result the optimal wavelet was selected. The wavelet when used has an input to the

another machine learning method changes the dynamics of the selection process. The typical wavelet properties like vanishing moments, filter length, symmetry etc. do not directly affect the accuracy of the machine learning method which is using the wavelet transformed input. Since the objective is to improve the accuracy of the machine learning method therefore the properties of the wavelet decomposed signal with respect to the machine learning objective should be studied. The mother wavelet selection would therefore be dependent on the both properties of the dataset and the objective function of the machine learning method.

7.1.2 Selection of Mother Wavelets in Wavelet based Models

In the case of wavelet based methods, the selection of wavelets was mainly done based on the characteristics of the wavelets in relation to the input signal. The main properties, which were considered include energy and cross-correlation. In general there is no analytical justification, which could generalize the wavelet selection process [80] for classification or prediction models based on the configurations given in figure 23. Mostly, the reason of selection of a specific wavelet is given based on a relative value of the objective function for which the wavelet based model was used. The methods used for wavelet selection for wavelet based methods are described below:

7.1.1 Cross-Correlation based methods

Cross correlation is a function of similarity between the two signals and is measured as a sliding dot product or convolution of the two signals. Mathematically, cross correlation is represented as:

$$\rho_{x,y} = \frac{\sum(X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum(X - \bar{X})^2(Y - \bar{Y})^2}}$$

where $\bar{X}$ and $\bar{Y}$ are the mean values of X and Y respectively.

Correlation based method was used by [81] where the cross correlation of the signal and the wavelet filter was computed and the wavelet filter with the maximum cross correlation value was selected for the objective of denoising the ECG signal. The measure of accuracy of the wavelet filter was based on the preservation of peaks and RMSE. Denoising was performed using the threshold method. Here the minimum RMSE of the thresholded reconstructed value of the wavelet signal and the original signal represents the similarity of the wavelet filter with the original signal. The cross correlation method selected also represents the value of similarity between the two signals. A similar method was also used in [82]. Similar results would be produced if the selection criterion of entropy was selected.

However, the above mentioned selection method would not be applicable to wavelet based compression, classification and prediction using any of the configurations as shown in figure 23.

7.1.2 Energy based methods

Energy based selection criterion use retained energy and relative energy as the measure of energy captured by the wavelet coefficients from the original signal. Retained energy is the measure of similarity of the wavelet decomposed signal with the original signal. Mathematically, retained energy can be expressed as:

$$Retained\ Energy = \frac{\|x(n)\|^2}{\|x(n) + y(n)\|^2} \times 100$$

where $x(n)$ and $y(n)$ are the original and reconstructed noise signal arrays respectively. Relative energy is the measure of signal information retained in the wavelet decomposed approximation signal w.r.t. the approximation and detailed signals.

$$Relative\ Energy = \frac{\sum_k {a_{j,k}}^2}{\sum_k {a_{j,k}}^2 + \sum_j \sum_k {d_{j,k}}^2}$$

where j represents the level of wavelet decomposition selected for the denoising the signal. The hypothesis for using relative energy is based on the assumption that information content in the signal lies in the frequency band of the approximate signal of the selected wavelet, whereas the noise components lie in the high frequency range of the detail signals. [83, 84] have used the retained energy and relative energy criterion respectively, to select the optimal wavelet filer for denoising. The wavelet, which retains maximum energy, was selected for denoising.

7.1.2 Hybrid methods

A minmax approach proposed in [85] used both cross correlation and energy based functions to devise a function for wavelet selection. The minmax function was defined as:

$$MinMax(s) = \frac{H(X,Y) \times H(Y|X) \times D(X||Y)}{I(X:Y) \times C(X,Y)}$$

where $H(X,Y), H(Y|X)\ and\ D(X||Y)$ represent the joint entropy, conditional entropy and relative entropy respectively and $I(X:Y) and\ C(X,Y)$ stand for mutual information and cross correlation.

The minimum value using the above equation gave the optimal wavelet from the set of wavelets, which would denoise the input data with greater efficiency.

7.2 Selection of Wavelet Decomposition Level

The selection of wavelet decomposition level in wavelet and wavelet based machine learning systems is again based on the computation of error indicators of the training data relevant to the objective function of the problem. The training data is used to compute the wavelet transform at different decomposition levels and then the objective function is computed. The accuracy of the objective function is determined based on different parameters. The best results of the different wavelet decomposition level data determine the optimal level of decomposition.

7.2.1 Selection of Wavelet decomposition level in Wavelet based Models

The wavelet decomposition level selection model was defined in [86], which used quality index of the reference image and the normalized weighted performance metric (NWPM) in addition to the two general indicators peak signal to noise ratio (PSNR) and the root mean squared error (RMSE) to calculate the optimal level of wavelet decomposition for region level fusion of multi-focused images.

The quality index of the reference image is defined as:

$$QI = \frac{4\sigma_{ab}ab}{(a^2 + b^2)(\sigma_a^2 + \sigma_b^2)}$$

where a and b are the mean of images R and F respectively and σ and σ^2 depend covariance and variance respectively. The quality index of the image defines the image distortion in terms of loss of correlation, luminance distortion and contrast distortion. This is also called the universal image quality index and showed that it performs better than the RMSE methods under different images of varying distortions [87].

The Normalized Weighted Performance Metric is defined as

$$NWPM = \frac{\sum \forall_{i.j} Q_{ij}^{AF} W_{ij}^{A} + Q_{ij}^{BF} W_{ij}^{B}}{\sum \forall_{i.j} W_{ij}^{A} + W_{ij}^{B}}$$

NWPM is a measure of pixel level image fusion performance defined in [88]. It is accepted as a universal measure for objectively assessing the quality of the visual information obtained from the fusion image.

Another method was proposed in [89], which is based on systematically quantifying the energy in each sub-band of the wavelet decomposed signal. It decomposes the image into it's Fourier transform and reconstructs the same image after removing the frequencies which are less than 1% of the main peak. Now the energy E of the reconstructed image is compared against the energy E_i of the wavelet decomposed sub-band image energy at decomposition levels i. The ratio $\frac{E_i}{E}$ is compared against a threshold, which determines the optimal level of decomposition of the image for image information mining.

7.2.2 Selection of Wavelet decomposition level in Wavelet based Machine Learning Models

The level of wavelet decomposition depends on the number of points in the input signal. The level of decomposition also depends on the frequency content of the input signal. Lower decomposition levels through scaling and shifting produce filters which efficiently represent events in the higher frequency range of the input signal in the time-frequency wavelet space. Therefore, for a signal with more noise content or high frequency content a lower level of decomposition would be required as compared to the dataset with less high frequency content.

The level of decomposition would not improve the results if the level of decomposition is increased beyond the frequency content of the input signal. The optimal level of decomposition is the one which corresponds to the frequency content of the signal. Similar results were shown in [90].

7.3 Selection of Wavelet Transform Method

The wavelet transform can be computed in the discrete domain in one of the 3 main forms: Undecimated discrete wavelet transform, discrete wavelet transform and the wavelet packet transform. Each type of transform has its own advantages and disadvantages described below:

7.3.1 Undecimated Discrete Wavelet Transform

The undecimated discrete wavelet transform as shown in figure 12 does not downsample and upsample the low pass and high pass filtered sub-bands, but the filters at each decomposition level are stretched and dilated following the wavelet concept. In our case since, we are only using the reconstructed values, therefore the reconstructed filter values using the wavelet packet transform and the undecimated wavelet packet transform at decomposition level 2 would yield the same result.

7.3.2 Discrete Wavelet Transform

In discrete wavelet transform the high pass component of the original signal is not further decomposed in the sub-bands. The DWT only focuses on the subband coding in the time-frequency domain of the low pass signal. Implementation of the reconstructed DWT signal to the support vector regression module would always give lower accuracy than the

equivalent wavelet packet transform signal as the low energy and 50% of the high frequency information is embedded in the high pass reconstructed signal at decomposition level 1 represented as D1 in figure 16. The energy in D1 is in general more than the energy in D2, D3 and D4. The energy of D4 of the DWT is not comparable to the energy of sub-band 2-2 of the WPT. Therefore, the discrete wavelet transform in general does not give improved results as compared to the wavelet packet transform support vector regression model.

7.3.3 Discrete Wavelet Packet Transform

The wavelet packet transform divides the input signal into equal frequency components among its sub-bands. The energy of the reconstructed DWT signal in lower decomposition levels for high frequency is very low and does not contribute towards the overall accuracy of the WDSVR model. It is therefore that [57] substituted the lowest energy sub-band with the moving average of the low energy signal.

For optimal results using the current wavelet transforms an iterative approach would yield optimal results using different combinations of the reconstructed wavelet signals.

8. WAVELET PACKET SUPPORT VECTOR REGRESSION

The wavelet packet support vector regression (WPSVR) model is a multiple support vector regression model, which forecasts the value of wavelet reconstructed input data. The structure of wavelet packet support vector regression is schematically outlined in figure 24. The model works by evenly distributing the original signal's frequency using the wavelet packet transform into the SVR modules.

The wavelet packet transform evenly distributes the input signal frequency into its

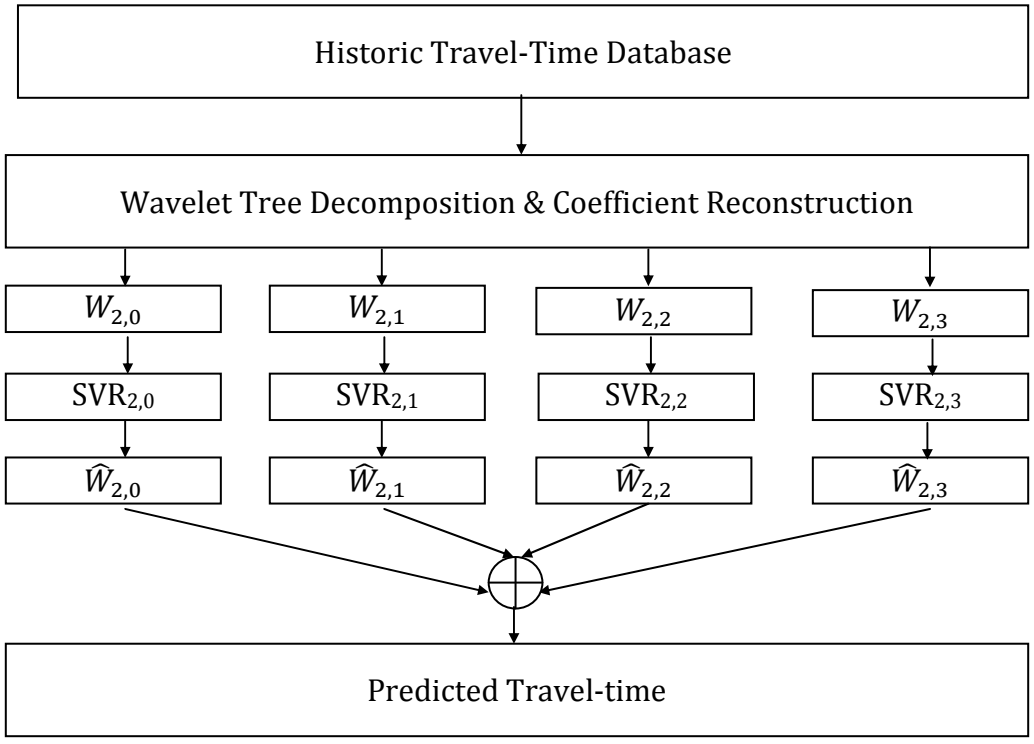


Figure 24: Flow diagram of the Wavelet Decomposed support vector regression model

wavelet decomposed signals. The frequency distribution was done based on the hypothesis that the original signal has equal traffic information in all frequency bands.

The time series signal $x(t)$, which represents the travel-time of the freeway was sampled from the database based on the prediction horizon selected. The time signal was then transformed using wavelet packet decomposed signals, such as $\sum_{n=0}^{2^j-1} W_{j,n}$, where j is the level of the decomposition. The wavelet decomposition was calculated using a sliding window as shown in figure 25. The window size determines the number of input features given to the support vector machine. In our case the window size of 8 was selected and the decomposition was done at level 2. These wavelet coefficients were stored for the support vector regression module. The four frequency components were processed through their respective support vector machines leading to one time-step ahead output, where the step was equal to the time interval between the consecutive input values. The support vector regression output was finally aggregated to calculate the travel-time forecast. Table 2 below gives the step by step implementation of the wavelet decomposed support vector regression algorithm.

8.1 Collection and Storage of Traffic data:

The traffic data was collected from the Inductive loop detectors, which measure speeds and count of individual vehicles passing over them. The speed of the indivisual vehicles was aggregated over a specified period. CALTRANS stores these speed aggregated over an interval of 30 seconds and 5 minutes. If, v_i^A is the speed of vehicle i passing over detector A, between $t1$ and $t2$ then the aggregated speed of vehicles passing over the detector during the specified period would be

$$V_{t2}^{A} = \frac{1}{N}\sum_{i=1}^{N} v_i^A$$

where N is the total number of vehicles passing over the detector A between times $t1$ and $t2$.

The aggregated speeds of the vehicles called the time mean speed of detector A. This time mean speed is stored in the form of flat files along with the metadata of the roadway.

The data from different loop detectors is stored in a centralized database, which is updated periodically. A length of 9.13 miles was chosen on I-5N with a detector density of 2.73 to ensure proper coverage of the traffic pattern on the freeway stretch.

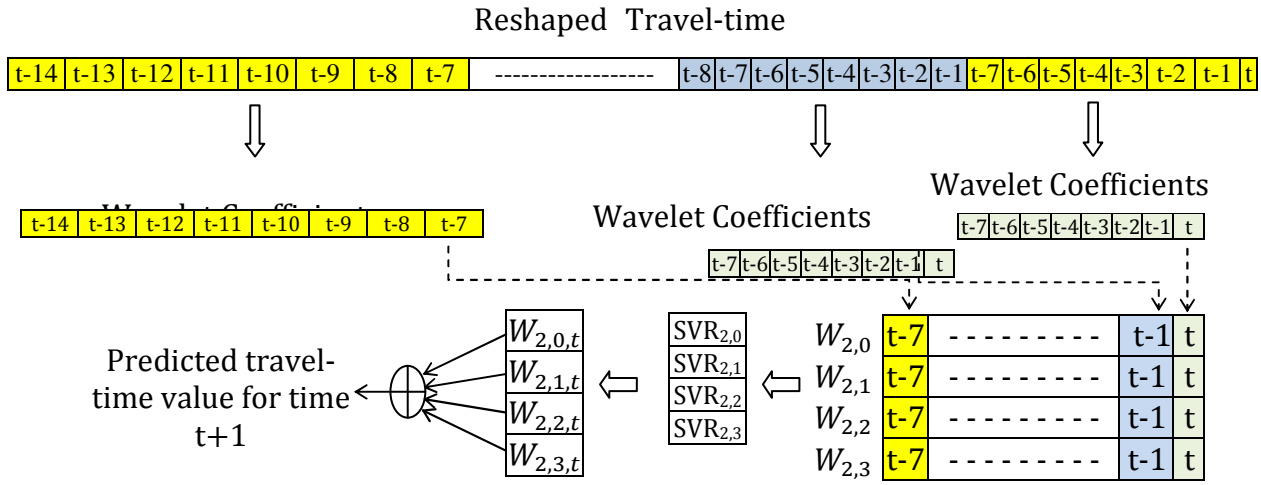


Figure 25: Algorithm for wavelet decomposed support vector regression

An ftp (file transfer protocol) connection was made with the data server to transfer the relevant data related to the section of the roadway. The speed data of the loop detectors was transferred in the form of flat files and migrated into the SQL server database. SQL scripts were run on the database to filter out the speed data from 1 pm to 8 pm daily. This output data variable was used to calculate the travel time.

8.2 Calculation of travel-time

The speed data collected from the California Department of Transportation is the time mean speed or spot speed of the loop detectors accumulated over a period of 5 minutes. This speed data is converted to space mean speed (reasons explained in Section 3 above). The information on the distance between consecutive detectors was already known. Consequently, the travel-time was calculated from the space mean speed array and the distance array. This time array is then updates the travel-time variable.

8.3 Sampling of Input data

The travel-time data was sampled for different time horizons for the wavelet packet transformation. The sampling period was proportional to the prediction horizon. For example if the travel-times were updated with an average of the aggregated speeds of all vehicles passing over a period of 5 minutes and the selected forecast horizon is 10 minutes then every second travel-time value in the input array would be sampled.

Mathematically,

$$x_{\delta}(\mathrm{t}) = \tau(\frac{\delta \mathrm{t}}{f})$$

Table 2: Algorithm for wavelet decomposed support vector regression

1. Sample travel-time array into subsets for their respective prediction horizons using

$$y[T] = \sum_{k=0}^{N} x\left[\frac{hkT}{5} + 1\right]$$

where h is the prediction horizon in minutes.

2. Initialize p=0 and decompose the sampled signal using wavelet packet decomposition at level j=2

$$W_{j,n}\left[\sum_{k=p}^{p+7} y\,[t-k]\right].$$

3. Store $W_{j,n}$ computed in step 2 for the SVR module and increment $p = p + 1$.

4. Repeat steps 2 and 3 until the end of the input array $y[T]$.

5. Increment $n = n + 1$ and repeat steps 2-4 until $n = 2^j$.

6. Divide SVR_{in}^{n} into training and testing sets and compute one step ahead prediction value using their respective SVR modules.

7. Aggregate the predictions of all 4 SVR modules to calculate the predicted travel-time.

where δ is the forecast horizon in minutes and f is the interval in minutes at which the data is collected. This sampled input data is then transformed into a $u * v$ matrix with $u = l - 7$ and $v = 8$ where l represents the Nth value of the time series. The decomposed and reshaped travel-time was represented as

$$X = \begin{bmatrix} x(t-1) & \cdots & x(t-8) \\ \vdots & \ddots & \vdots \\ x(l-7) & \cdots & x(l) \end{bmatrix}$$

The index of travel-time of each row and column is incremented by one.

8.4 Wavelet Packet transformation

The sampled and reshaped travel-time matrix was transformed in the wavelet packet domain at level 2. Each row of the matrix X was individually decomposed using the wavelet packet transform. Since, the wavelet packet transform was applied at level 2 the out would give four signals of the wavelet packet reconstructed coefficients. The wavelet reconstructed coefficients were stored in four different matrices representing $W^{2,0}, W^{2,1}, W^{2,2}$ and $W^{2,3}$. Each matrix can be represented as.

$$W_{j,n} = \begin{bmatrix} W_{j,n,t-1}(x) & \cdots & W_{j,n,t-8}(x) \\ \vdots & \ddots & \vdots \\ W_{j,n,l-7}(x) & \cdots & W_{j,n,l}(x) \end{bmatrix}$$

where j is the number of decomposition level and the number of wavelet decomposed matrices generated by wavelet packet decomposition is 2^j.

Support vector regression machine is given a training set of generation of support vectors to train the machine learning algorithm. The training set is composed of input dimensions and their corresponding labels or in the case of SVR the expected values. The vector of true values against which the SVR is trained is the label vector. The label vector in the the wavelet decomposed support vector regression model is the $t + 1$ value of the travel time in the input dimension space. The traiing label is represented as

$$label_{j,n} = \begin{bmatrix} W_{j,n,t} \\ W_{j,n,t-1} \\ \vdots \\ W_{j,n,l+1} \end{bmatrix}$$

The number of label vectors in WDSVR is 2^j. As a general rule 70% of the data is used for training and 30% of the remaining data for testing. Hence, the four matrices

were given as input to their respective support vector machines with $(l-7)*0.7$ rows for training, while the remaining 30% for evaluation.

A set of 42 wavelets were used for the wavelet transformation. Each wavelet transformed data had a dimension of N-δ rows and 4 columns.

The four columns correspond to the frequency division at level 2 through the wavelet packets. The data was stored for the computation in the support vector regression module after the wavelet selection process.

8.5 Wavelet Selection

The purpose of the wavelet selection process module was added to reduce the computational cost of the model. In literature no analytical justification is given for the selection of the optimal wavelet for best results for the wavelet decomposed machine learning models. In this section we made an effort to define two measures, which help us in separating a subset of the wavelets, which always produce lower accuracy prediction results when compared against the classical support vector regression method.

The wavelet selection models for standalone wavelet models as explained in chapter 7 were based on the wavelet properties w.r.t the objective function for that specific dataset. In the wavelet decomposed machine learning models the objective function is computed after the support vector machine module. The purpose of the wavelet function in the model is to pre-process the data for the support vector machine for regression. Therefore, the properties of the support vector regression module were studied w.r.t. the properties of the support vector machine instead of the objective function.

8.5.1 Cross-correlation of trend of the wavelet reconstructed data

The travel-time follows a similar trend of congestion in the evening hours on weekdays for which the data is sampled. Figure 26a shows the cyclic pattern of the travel-time. The travel-time data is reshaped for the wavelet module using the moving window as shown in the figure 26b. The reshaped travel-time data has a very high cross-correlation among the successive rows as the time-series data is advanced by a single value in every iteration.

The purpose for pre-processing the data was to decompose the data into different time-frequency bins, so that the multiple support vector machines could learn more patterns related to the dataset than one support vector machine. The downside of this process are the errors associated with each SVM also aggregate by increasing the number of SVMs and to improve the overall result, (10) must be satisfied.

The first difference of the time series gives the trend of the data. A similar trend in the consecutive time series would indicate that the wavelet reconstructed data is passing trend information to the support vector machine.

The SVM input is in eight dimensions and since we are using the linear kernel only the feature space of SVM is eight dimensional. Four support vector machines with an input of eight dimensional data would enable the machines to train for 32 dimensions in the four wavelet signals.

The dimensional information cannot be established analytically, therefore, we have established the measure that if the wavelet signal is generating one measure repeatedly then the SVM would work to optimize that measure alone. A pattern in the data would imply redundant information being given to the SVM. The idea behind running

multiple SVM was motivated by the fact that the time domain signal has information in the data, which can be linearized by transforming the data in another domain. If a certain wavelet basis has linearized only one aspect using multiple data points then it is highly unlikely that more information is also produced using the same variables.

Summarizing the discussion we established that an exact linear change in trend, which is calculated with computing the first difference of the wavelet data would be an indicator of an inefficient SVM. The model, after the wavelet decomposition computed the first difference between the successive rows of $W_{j,n}$.

$$Trend(W_{j,n}) = \sum_{c=1}^{l-8} (W_{j,n})_{c,d} - (W_{j,n})_{c+1,d}$$

where c and d represent the rows and columns of the wavelet reconstructed coefficient matrix $W_{j,n}$.

Figures 27 and 28 show the wavelet reconstructed coefficients of two wavelet basis biorthogonal 1/3 and reverse biorthogonal 6/8. The reverse biorthogonal wavelet produced the best results for the prediction horizon of 1 hour. It can be observed that unlike the reverse biorthogonal wavelet basis the first difference of the biorthogonal wavelet 1/3 is repeatedly producing similar values as shown in figure 31 and 32.

The measure of cross correlation between the successive rows of the matrix $W_{j,n}$ gives us the similarity between the two arrays. A high cross correlation value was expected as the wavelet reconstructed values in the successive rows of $W_{j,n}$ slides by one value. Sliding here indicates the moving window concept, which was illustrated in figure 26b. The basic idea was to analyze if the wavelet filter captures the effect of the moving window in the time-frequency domain. The filters which capture the effect of the differ-

ence between $(W_{j,n})_{c,d}$ and $(W_{j,n})_{c+1,d}$ gave low cross correlation values, whereas the signals of wavelet reconstructed arrays were similar for wavelets with high correlation values.

8.5.2 Recurrence Relationship of wavelet decomposed data

The recurrence relation here defines a similarity in the value at a specific point in every iteration. The support vector machine using the linear kernel has the ability to only linearly project the data in the feature space. Since, the input data is in the time-frequency space, therefore the feature projections would also be a linear combination of the input space.

In figure 27 it is shown that the wavelet reconstructed coefficients at W[2,1] is producing zero value at a similar location in the time-frequency domain. However, if we look at the signal in figure 28 the reverse biorthogonal filter at 6.8 is also generating zero value in the time-frequency domain at W[2,2]. However, the significance of the wavelet reconstructed signal is directly proportional to the energy of the wavelet packet transform at each level and W[2,2] retaining minimum energy amongst all four wavelet reconstructed coefficients has minimal contribution to the overall accuracy.

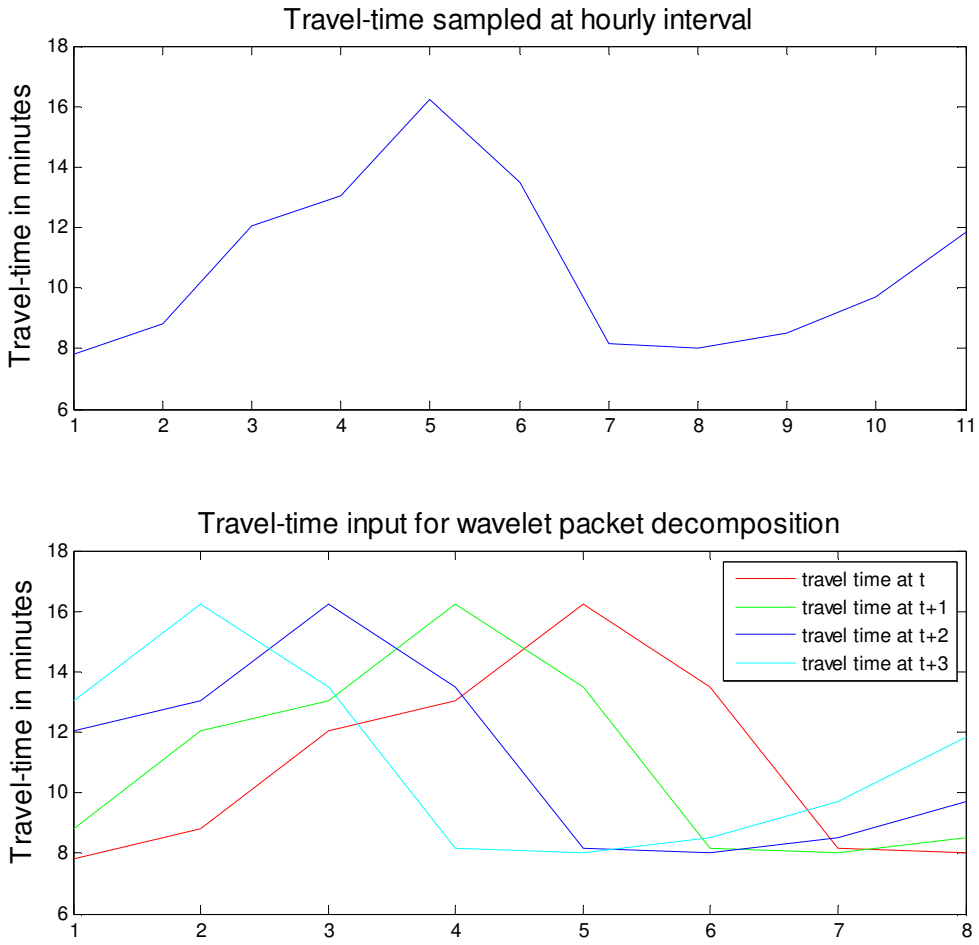


Figure 26: a) Travel-time plot of the dataset **b)** the plot for consecutive rows of the re-shaped travel-time data for wavelet packet module

Figure 29 indicates the proportion of energy of the original signal which each of the component of the wavelet packet reconstructed signal posses. There is an uneven distribution of the energy of the wavelet signal, the W[2,0] component which is the low pass signal at level 2 as shown in figure 16b has approximately 95% of the signals energy and W[2,2] has the least amount of energy at 0.37%. Therefore, the filtration based on the results W[2,0], W[2,1] and W[2,3] are important.

8.6 Support Vector Regression

The wavelet time-frequency space is the input space for the support vector regression machine. The number of input dimensions indicates the number of factors which are influencing the output label. Each input dimension value is projected in to the feature space using the wavelet kernel. For the WDSVR each row of the matrices $W_{j,n}$ corresponds to a

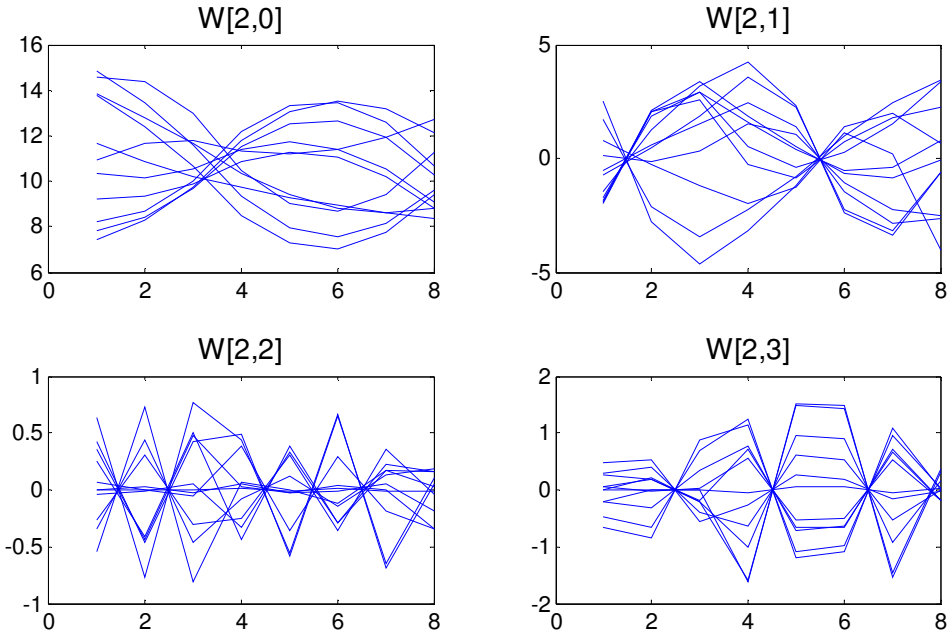


Figure 27: Wavelet Packet Reconstructed signal using Bior 1/3 wavelet at level 2

input array for the support vector machine and each row of $label_{j,n}$ is the label output. In terms of the SVM concept explained in Chapter 5 the SVM for regression can be modeled with the following equation

$$f(x) = \sum_{i=1}^{8} w_i W_i(x) + b$$

where $\emptyset_i(x)$ is now replaced by $W_i(x)$, which is the row of the Wavelet reconstructed input matrix $W_{j,n}$. Since, we have used the linear kernel, therefore, the function f re-

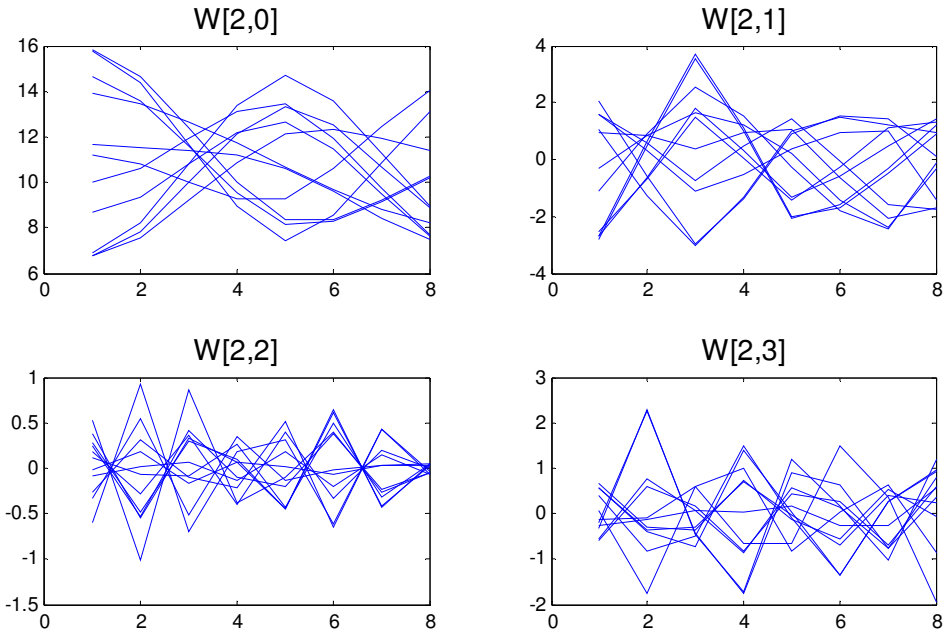


Figure 28: Wavelet Packet Reconstructed signal using Rbio 6/8 wavelet at level 2

mained a function of x.

In the linear SVM for regression case the next step is to compute the regression error, which uses the least squares method to calculate the goodness of fit. But the support vector machine for regression employs the ϵ-insensitive loss function as the good ness of fit criterion.

$$\varepsilon\big(W_i(x), f(x_i)\big) = \begin{cases} 0 & if\ |W_i(x) - f(x_i)| < \epsilon \\ |W_i(x) - f(x_i)| - \epsilon & otherwise \end{cases}$$

The ϵ-insensitive loss is linearly sensitive towards outliers, which in our case are all values of $|W_i(x) - f(x_i)| \geq \epsilon$. For each outlier a support vector would be generated,

which would store the weight associated with the outlier w.r.t. the cost value C selected

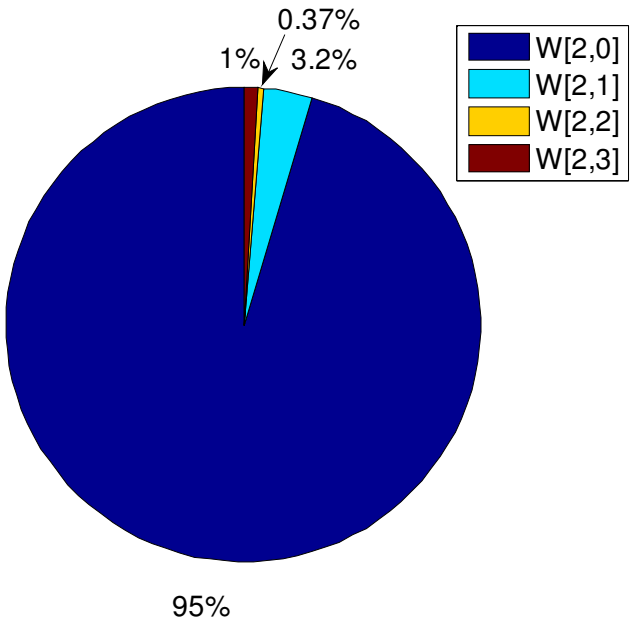


Figure 29: Pie chart of average energies at level 2 using multiple wavelet basis

by the user.

The minimization function for regression would now become

$$min\frac{1}{2}\|w\|^2 + C\sum_i \xi_i^+ + \xi_i^-$$

The outliers can be show in either of the two sides of the ϵ boundary. ξ_i^+ and ξ_i^- in the above equation are slack variables, which were introduces to compensate for errors on either side of the ϵ boundary. The values of ξ_i^+ and ξ_i^- are graphically represented in figure 22.

The Langrangian is now implemented to solve the dual problem. The minimized Langrangian wrt w, b, ξ_i^+ and ξ_i^- in terms of langrangian multipliers is given by

$$f(x) = \sum_i (\alpha_i^+ - \alpha_i^-) x_i^T \, x_i + b.$$

The computation of the above equation is also conditional to the only those cases where $\xi_i^+, \xi_i^- \geq 0$ or in other words when the value in the feature space goes outside the ϵ boundary. This makes the solution of SVM sparse.

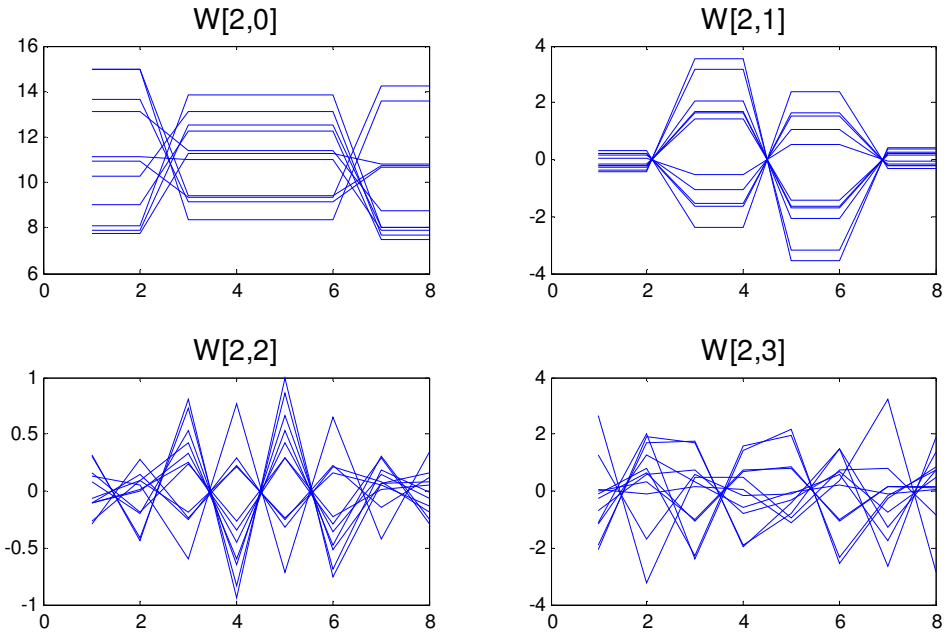


Figure 31: First difference of values of each input using biorthogonal 3/3 wavelet

Finally the kernel function $k(x, x')$ replaces $x_i^T x_i$, which minimizes the cost of computation. The function eventually becomes

$$f(x) = \sum_i (\alpha_i^+ - \alpha_i^-)\, k(x, x') \; + b.$$

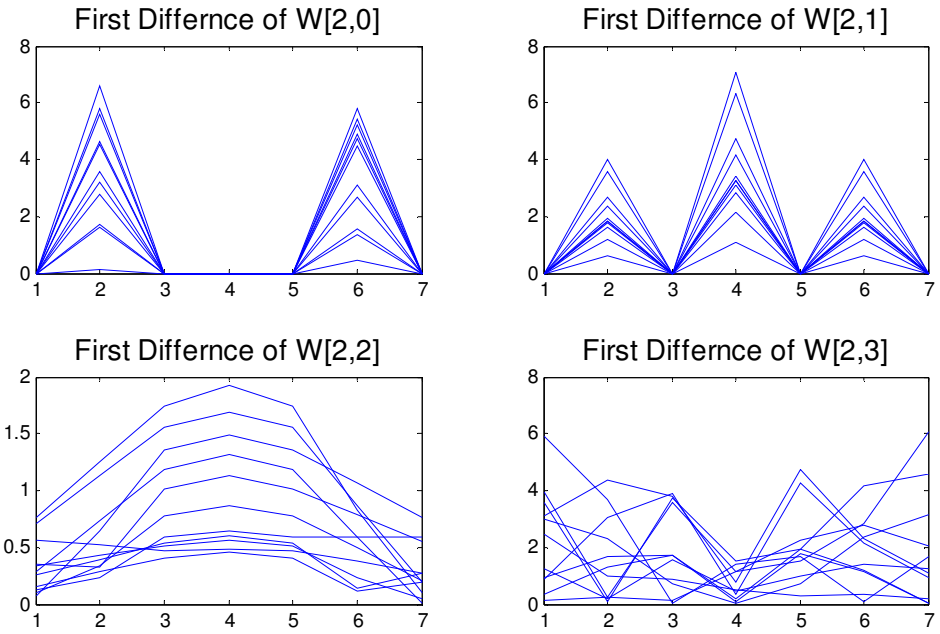


Figure 32: First difference of values of each input using biorthogonal 1/3 wavelet

This function $f(x)$ generated the predicted values which were compared against the true values in the label vector.

The process is repeated for all matrices of the wavelet reconstructed signals. The forecasted values of each row of the four matrices are aggregated at the output to form an array of the predicted travel-time.

$$\hat{T} = \widehat{W}^{2,0} + \widehat{W}^{2,1} + \widehat{W}^{2,2} + \widehat{W}^{2,3}$$

This predicted travel-time is compared against the PLSB estimates of travel-time based on the ILD data to produce the model error.

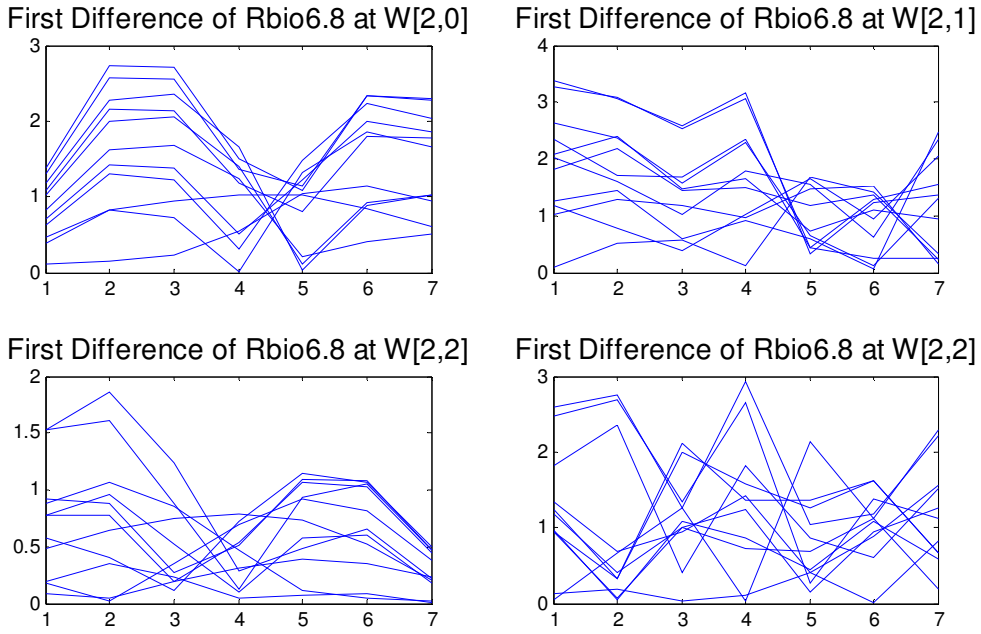


Figure 33: First difference of values of each input using reverse biorthogonal 6/8 wavelet

9 EXPERIMENT AND RESULTS

9.1 Selection of Mother Wavelet for WPSVR Model

The major computational load of the proposed travel-time prediction model is divided into two parts: computation of the wavelet packet reconstructed time-series data, and training of the support vector regression machines using the optimal cost and epsilon values. The grid search method was used for searching for epsilon and cost values.

A definite procedure for selection of mother wavelets is yet to be established for wavelet decomposed support vector regression models. However, analyzing the wavelet reconstructed signal in context to the characteristics of support vector machines helped us in filtering the relevant wavelets basis.

The accuracy of the proposed model is superior to the classical SVR model, if the condition in equation (4) is met.

$$\varepsilon_{SVR} > \varepsilon_{SVR(2,0)} + \varepsilon_{SVR(2,1)} + \varepsilon_{SVR(2,2)} + \varepsilon_{SVR(2,3)} \qquad (10)$$

Where, ε_{SVR} is the error of the classical support vector method. It is clear from equation (10) that WPSVR would not produce more accurate results than SVR for shorter time horizons. In our datasets, the WPSVR started giving more accurate results than the SVR method for prediction time horizons of 45 minutes or more.

We conducted two basic tests for the admissibility of a wavelet for the support vector machine module.

9.1.1 Effect of Cross-correlation of wavelet decomposed data:

For accurate predictions of a non-linear and non-stationary dataset the reconstructed wavelet coefficients of successive windows should not be correlated with one another. To test our hypothesis we computed the cross-correlation of each window with the other.

9.1.2 Effect of Recurrence Relationship:

The second test was to detect if the reconstructed wavelet coefficients were following a certain pattern. The input data of the successive windows is highly non-linear. The existence of a unique pattern would reduce the chances of the wavelet to produce more accurate results. First difference of each successive window was calculated and it was revealed that two wavelets showed zero value at a certain point in every iteration.

To identify the above characteristics in the wavelet signal we used a subset of the data chosen at random ranging four days. In figure 30 the wavelet reconstructed difference signal converged to zero at a similar point in every iteration. Based on our tests we filtered 10 wavelet basis out of a total of 42, hence reducing the computational load of our project by 23.8%. While a detailed study on wavelet selection for WDSVR is needed, our results on the use of the support vector machines have shown encouraging results.

9.2 An alternate configuration for interchangeable

The WDSVR and SVR have both proven suitable for travel-time prediction depending on the selected forecast horizon. In our dataset, we observed that SVR is more accurate for prediction horizons of less than 45 minutes.

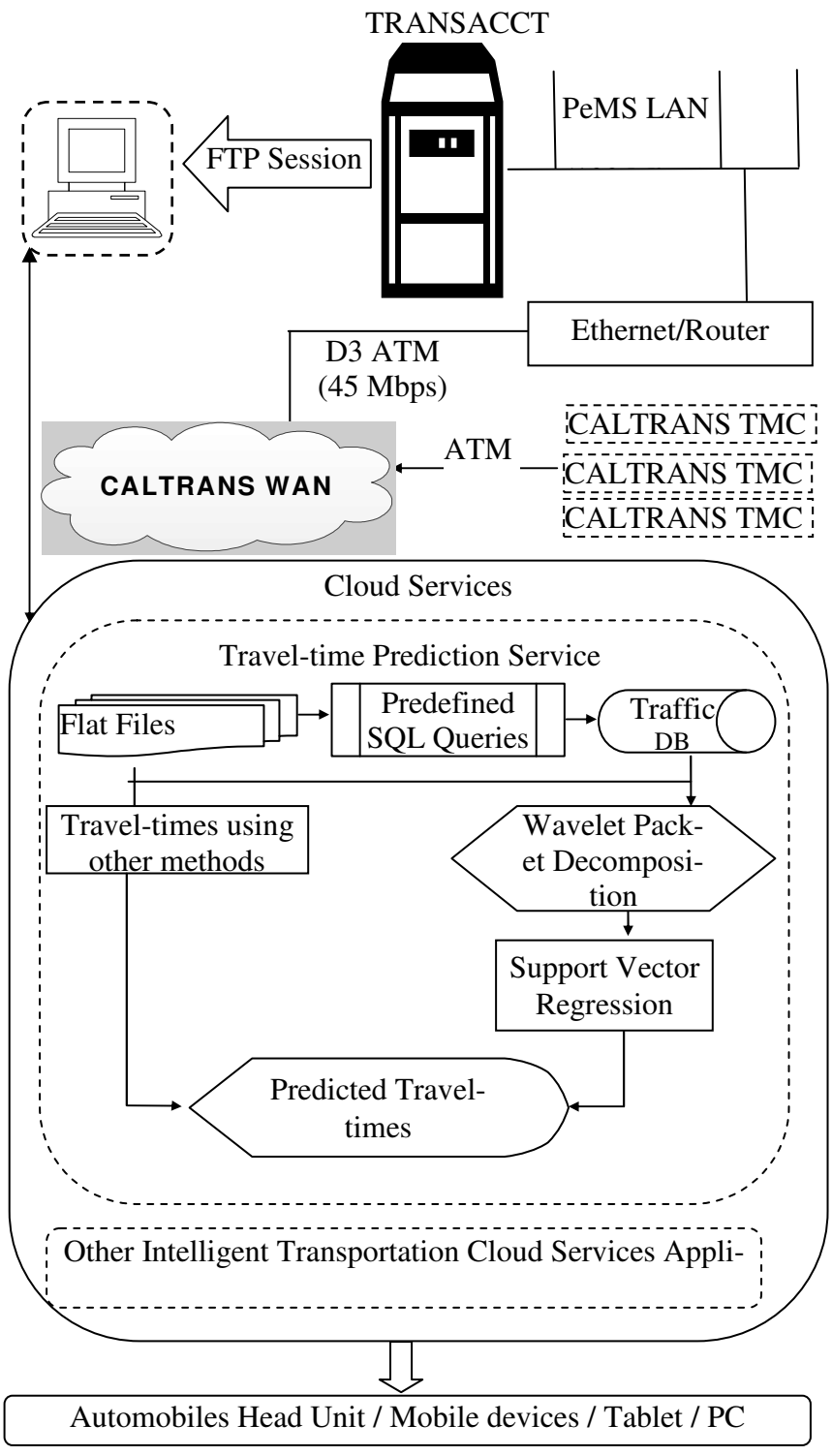
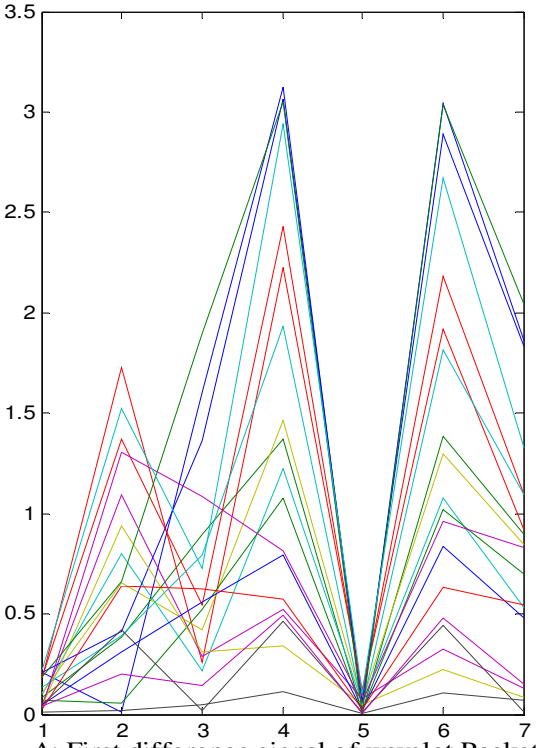
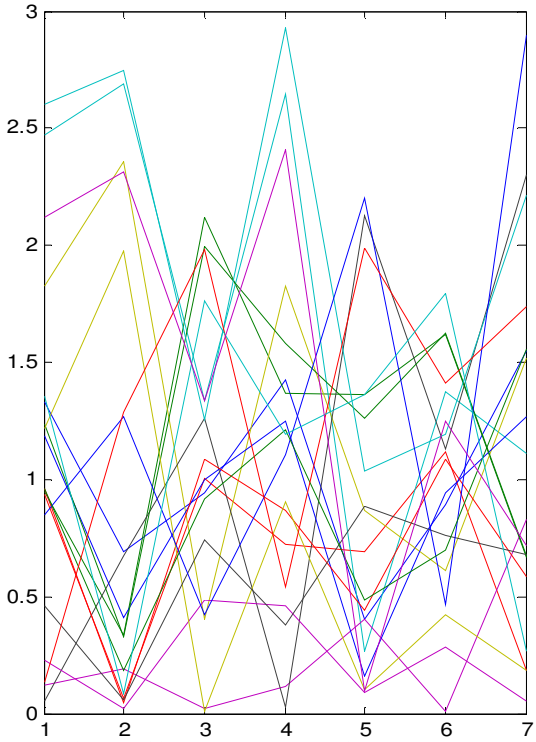


Figure 34: Proposed configuration for travel-time prediction for ATIS

From 45 minutes onwards, WDSVR gives more accurate results. Considering the effectiveness of both these models in different horizons, we have proposed an interchangeable configuration in figure 33, where we can chose to compute the travel-times in parallel and *switch* to the configuration for active use depending the selected prediction horizon. The cloud component, which houses both the prediction models is flexible and can be either scaled horizontally or vertically to accommodate for the computation overhead.



A: First difference signal of wavelet Packet Reconstructed time series at level 2,3 using Biorthogonal 3.3



B: First difference signal of wavelet Packet Reconstructed time series at level 2,3 using Reverse Biorthogonal 6.8

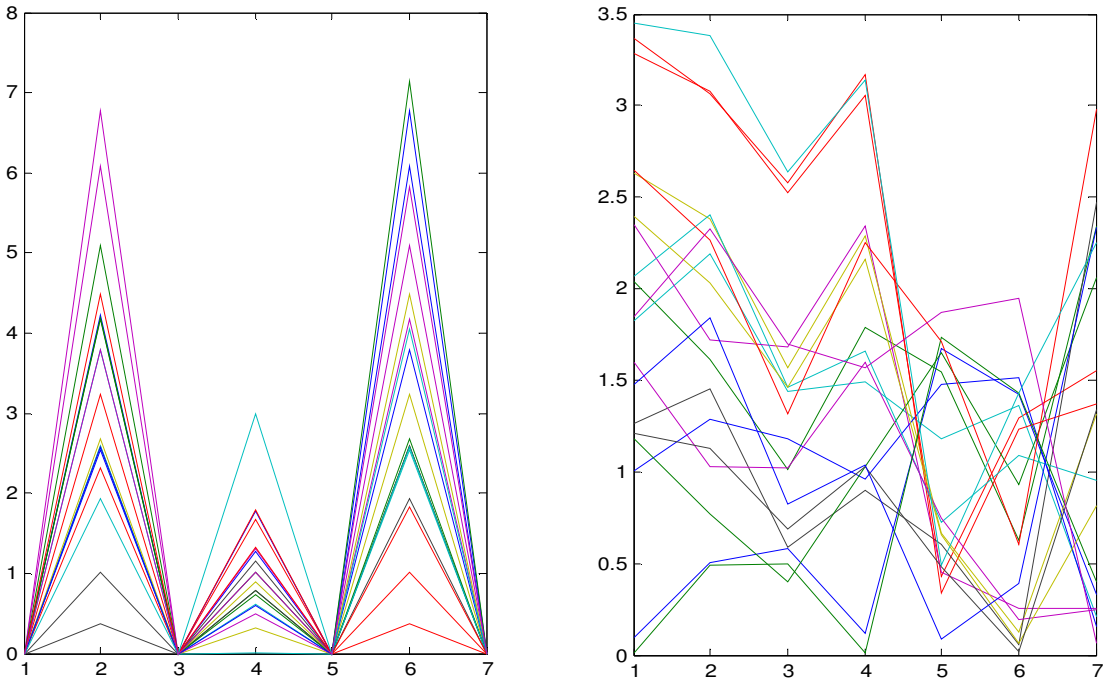


Figure 354: A comparison of wavelet recurrence relationship of better and worse performing wavelets

9.3 Experimental Setup

The data for our model validation and testing was collected from the Caltrans Performance Measurement System (PeMS) website [2]. The route of 9.13 miles on I-5N was selected with a detector density of 2.73. The data was observed for 214 consecutive days commencing from March 01, 2011 to September 30, 2011 from 1 pm to 8 pm. The time slot was selected after observing the daily pattern of congestion during this period. The data revealed daily congestion in the evening hours except holidays and most weekends. This loop detector data was collected over an interval 5 minutes.

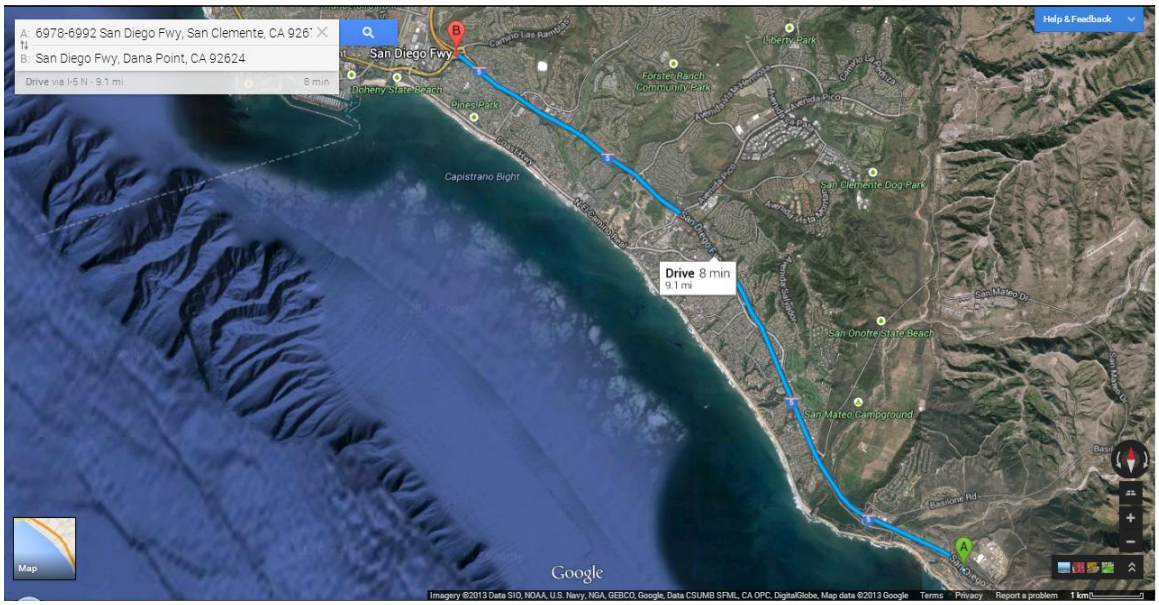


Figure 36: Map of the test site of I-5N freeway

The speed data was converted to travel-time series using the PLSB travel-time estimation method [32]. We decomposed the time series using wavelet packet decomposition at level 2. The data was then reshaped into a $u * v$ matrix with $u = l - 7$ and $v = 8$ where l represents the Nth value of the time series. The decomposed and reshaped wavelet transform of travel-time matrix gave us 2^j matrices at level j represented as

$$W_{j,n} = \begin{bmatrix} W_{j,n,t-1} & \cdots & W_{j,n,t-8} \\ \vdots & \ddots & \vdots \\ W_{j,n,l-7} & \cdots & W_{j,n,l} \end{bmatrix}$$

The four matrices were given as input to their respective support vector machines with $(l-7)*0.7$ rows for training, while the remaining 30% for evaluation. The evaluation matrix for each $W_{j,n}$ above was represented as

$$label_{j,n} = \begin{bmatrix} W_{j,n,t} \\ W_{j,n,t-1} \\ \vdots \\ W_{j,n,l+1} \end{bmatrix}$$

The predicted labels of each support vector machine were aggregated to compute the forecast time value. Finally the values generated by SVR were evaluated for errors. Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) were the two indicators chosen for evaluation of our model and for comparison with the classical Support Vector Regression model.

Mean Absolute Percentage Error is mathematically defined as:

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{\tau_t - \widehat{\tau_t}}{\tau_t}\right| , \tag{11}$$

where τ_t and $\widehat{\tau_t}$ represents actual travel-time and predicted travel-time respectively.

Root Mean Squared Error is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\tau_t - \widehat{\tau_t})^2} , \tag{12}$$

Table 3: Comparison of RMSE between SVR and SVR with Wavelet Decomposed Inputs (our approach).

Prediction	Prediction Horizon							
	45-min		50-min		55-min		60-min	
Wavelet Packet	**coif3**	**ε=0.1, C=100**	**bior6.8**	**ε=0.01, C=100**	**coif5**	**ε=0.1,C=100**	**db6**	**ε=0.001, C=100**
	2.2		2.31		2.41		2.46	
SVR Predictor	**ε=0.01, C=100**		**ε=0.1, C=1**		**ε=0.001, C=100**		**ε=0.1, C=10**	
	2.26		2.4		2.48		2.88	

Table 4: Comparison of MAPE (%) between SVR and SVR with Wavelet Decomposed Inputs.

Prediction	Prediction Horizon							
	45-min		50-min		55-min		60-min	
Wavelet Packet	**bior2.6**	**ε=0.1, C=1**	**rbio2.8**	**ε=0.1, C=100**	**rbio2.8**	**ε=0.001,C=100**	**rbio6.8**	**ε=0.01, C=100**
	12.35		13.1		13.66		14.74	
SVR Predictor	**ε=0.01, C=10**		**ε=0.01, C=100**		**ε=0.1, C=1**		**ε=0.1, C=100**	
	12.57		13.5		13.96		15.06	

Table 5: Comparison of RMSE Between SVR And SVR with Wavelet Decomposed Inputs

Prediction Methods	Prediction Horizon							
	1-Hour		3-Hour		5-Hour		7-Hour	
Wavelet Packet SVR	**db6**	**ε=0.001, C=100**	**db6**	**ε=0.001, C=100**	**bior1.5**	**ε=0.1, C=10**	**bior2.2**	**ε=0.01, C=1**
	2.46		4.24		3.63		4.39	
SVR Predictor	**ε=0.1, C=10**		**ε=0.1, C=100**		**ε=0.1, C=10**		**ε=0.001, C=1**	
	2.88		4.5		4.18		5.1	

Table 6: Comparison of MAPE(%) between SVR and SVR with Wavelet Decomposed Inputs

Prediction Methods	Prediction Horizon							
	1-Hour		3-Hour		5-Hour		7-Hour	
Wavelet Packet SVR	**rbio6.8**	**ε=0.01, C=100**	**db6**	**ε=0.001, C=100**	**bior1.5**	**ε=0.1, C=10**	**bior2.2**	**ε=0.01, C=1**
	14.75		22.18		19.59		18.74	
SVR Predictor	**ε=0.1, C=10**		**ε=0.1, C=100**		**ε=0.1, C=10**		**ε=0.001, C=1**	
	15.15		22.66		23.19		25.55	

We tested our model using Debauchies, Coiflets, Symlets, Reverse Biorthogonal and Biorthogonal wavelets in 42 different configurations, with different values of cost and epsilon. It was also observed that not all wavelets gave better results than the bench-

mark SVR predicted values. However, some of the worse performing wavelets were filtered out beforehand using our wavelet selection process to save computational cost. The best outputs in each time horizon sub-category are shown in tables 3-6.

Our results indicate that the wavelets decomposed support vector regression model consistently showed better performance for prediction horizon of 45 minutes and above, shown in tables 3-6. Below 45 minutes the classical SVR method is more accurate. figure 36 also shows the better tracking ability of the proposed model in comparison with SVR model.

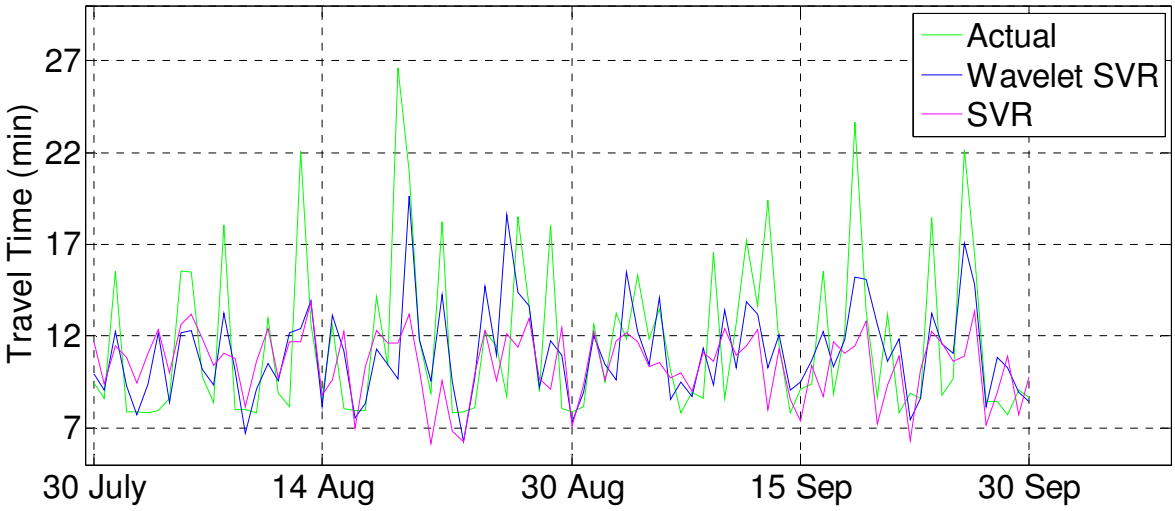


Figure 37: Comparison of actual travel-time, predicted travel-time by Support Vector Regression and Wavelet decomposed Support Vector Regression methods

9.4 Online-Implementation of WDSVR

The model of wavelet decomposed support vector regression was implemented in an online configuration using pre-computed support vectors and an active ftp connection.

The database was connected with the data server at Caltrans. A time delay of 5 minutes was induced to transfer the data from the loop detectors to the Data center via the traffic management centers (TMC). A further time delay of 5 minutes was caused due to the storage and accumulation of the 30 seconds data into 5 minutes interval flat files. The data was then copied to the local machine where the predicted travel-times were computed following the steps given in table 2.

9.5 GUI for the Online WPSVR Prediction Model

The graphical user interface was developed using the Matlab GUI tool. The development effort was focused towards both offline and online travel-time prediction user interface. The offline prediction has the option to load the pre-computed travel-time data file to forecast one time step ahead travel-times using both SVR and wavelet packet decomposed SVR methods.

The offline predictor uses the file transfer protocol to connect to the Caltrans server and downloads predefined data from the relevant stations to the local drive. Then similar to the offline method computes the values for the future travel-times based on both methods.

The details of the GUI are described in Appendix B.

10 SUMMARY OF RESULTS AND FUTURE WORK

10.1 Summary of Results

We have shown that wavelet packet transformed travel-time data when used as an input the support vector regression for prediction of travel-times is more accurate for longer time horizons when compared against the support vector regression results with similar data.

The factors, which affect the accuracy of the WPSVR module were analyzed for the purpose of optimal wavelet selection. It was shown that cross correlation and the existence of any recurrence relationship amongst the consecutive wavelet transformed values effect the accuracy of the support vector regression.

The online and offline implantation of the support vector regression and WPSVR models was carried by connecting to the Caltrans traffic data server. The computed results by both modules were analyzed for errors in a graphical user interface generated in Matlab.

10.2 Future Work

The wavelet packet decomposed SVR method has shown confidence in data prediction and the non-linearity of traffic data makes it a suitable technique for prediction. For accurate state estimation large datasets are needed which are now available online. Their training would require computation cost but since training is done offline therefore it is not a prime concern. Our model also signifies the effectiveness of support vector ma-

chines for smoothed datasets. However, there is need of further investigating the wavelet properties in conjunction with the effectiveness for support vector machines.

Further improvements to our model could be made by subdividing the data set based on their patterns, some examples are by congested and freeflow parts or by day of the week or both. The scalability of the model also makes it a viable option for its application to calculate arterial travel-times.

APPENDIX A

Appendix A: Relation between Time Mean Speed and Space Mean Speed

The mathematical relationship between time means speed and space mean speed is derived below using the fundamental equation of traffic flow.

$$q = k \times v_s \tag{13}$$

Let a set of vehicles i is passing through the roadway with velocities v_i and flow q_i.

$$q_i = k_i \times v_i, \tag{14}$$

where k represent the traffic density. Hence, total flow and density can be defined using equations below

$$q = \sum q_i, \tag{15}$$

$$k = \sum k_i, \tag{16}$$

The ratio of individual vehicle density and total traffic density is defined by

$$f_i = \frac{k_i}{k}, \tag{17}$$

From the definition of space mean speed we know that space mean speed is the average of the speed of vehicles passing over the space (roadway). Hence, for k_i vehicles with v_i speed, space mean speed can be defined as

$$v_s = \frac{\sum k_i v_i}{k}, \tag{18}$$

Similarly, the time mean speed is the average of speed over time. Therefore, considering vehicular flow q is the number of vehicles passing through a certain point, we can represent time mean speed as

$$v_t = \frac{\sum q_i v_i}{q}, \tag{19}$$

Substituting the value of q_i from (14) in (19) time mean speed can now be written as

$$v_t = \frac{\sum k_i v_i^2}{q}, \tag{20}$$

Now, substitute the value of q from (13) in (20)

$$v_t = \frac{\sum k_i v_i^2}{k v_s}, \tag{21}$$

Comparing value of f_i from (17) in (21)

$$v_t = \frac{\sum f_i v_i^2}{v_s}, \tag{22}$$

Now, add and subtract v_s in (22) and through algebraic manipulations time mean speed can be represented as

$$v_t = \frac{\sum f_i (v_s + (v_i - v_s)^2}{v_s}$$

$$= \frac{\sum f_i (v_s)^2 + (v_i - v_s)^2 + 2v_s(v_i - v_s)}{v_s}$$

$$v_t == \frac{\sum f_i v_s^2}{v_s} + \frac{\sum f_i (v_i - v_s)^2}{v_s} + \frac{2v_s \sum f_i (v_i - v_s)}{v_s}. \tag{23}$$

In equation (23) the third term $\frac{2v_s \sum f_i (v_i - v_s)}{v_s}$ will be zero if $\sum (v_i - v_s)$ is zero. Since, the space mean speed v_s is the mean speed of the vehicles passing over the roadway, there-

fore the total sum of the individual vehicles minus their average speed would always be zero.

We know that standard deviation is the variation from the mean or expected value of the data. Note that $(v_i - v_s)^2$ in the second term of v_t represents the variance of the speed on the roadway. Therefore, (23) can be represented as

$$v_t = v_s \sum f_i \frac{\sigma^2}{v_s} + 0$$

Also $\sum f_i$ would always be 1. Hence,

$$v_t = v_s + \frac{\sigma^2}{v_s}$$

The above relationship shows that time mean speed would always be greater than space mean speed. Both speeds can only be equal to each other if the variance of the vehicular speeds is zero, i.e. all vehicles travel at same mean speed through the roadway.

APPENDIX B

Appendix B: GUI for Online and Offline Travel-Time Prediction

The GUI was designed using the Matlab GUI development environment (GUIDE). The functionality of the Matlab travel-time calculator GUI was divided into online and offline panels. Both offline and online panels of the travel-time predictor compute travel-tie forecast using the SVR and WPSVR models.

Figure 387: Screenshot of Matlab Travel-time Predictor GUI

The offline prediction panel has the load data button, which is used to load the dataset in the form of matlab data file into the Matlab environment.

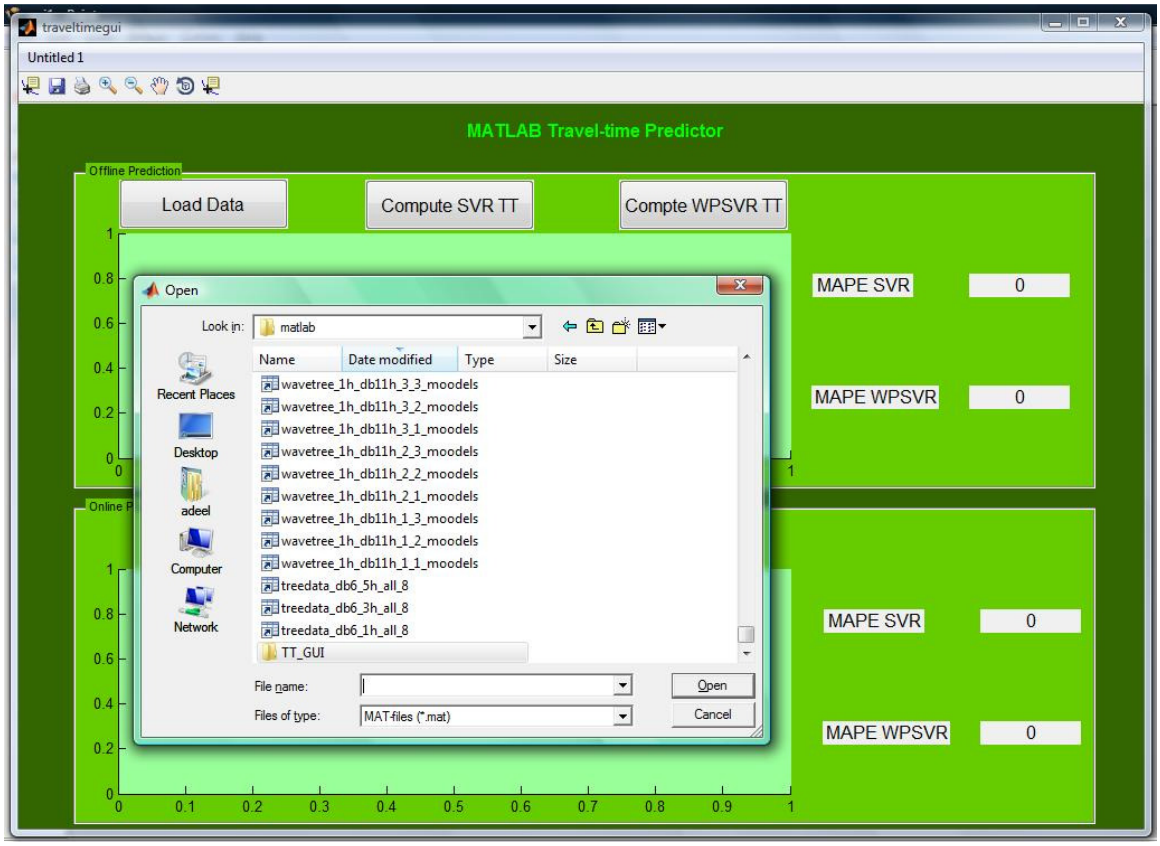


Figure 398: Screenshot of Load Data dialog box

The pre-computed travel-times in the offlne mode are divided into training and testing sets of 70% and 30% respectively. The *Compute SVR TT* and the *Compute WPSVR TT* buttons are used to compute the support vectors using the training set and perform predictions on the testing set.

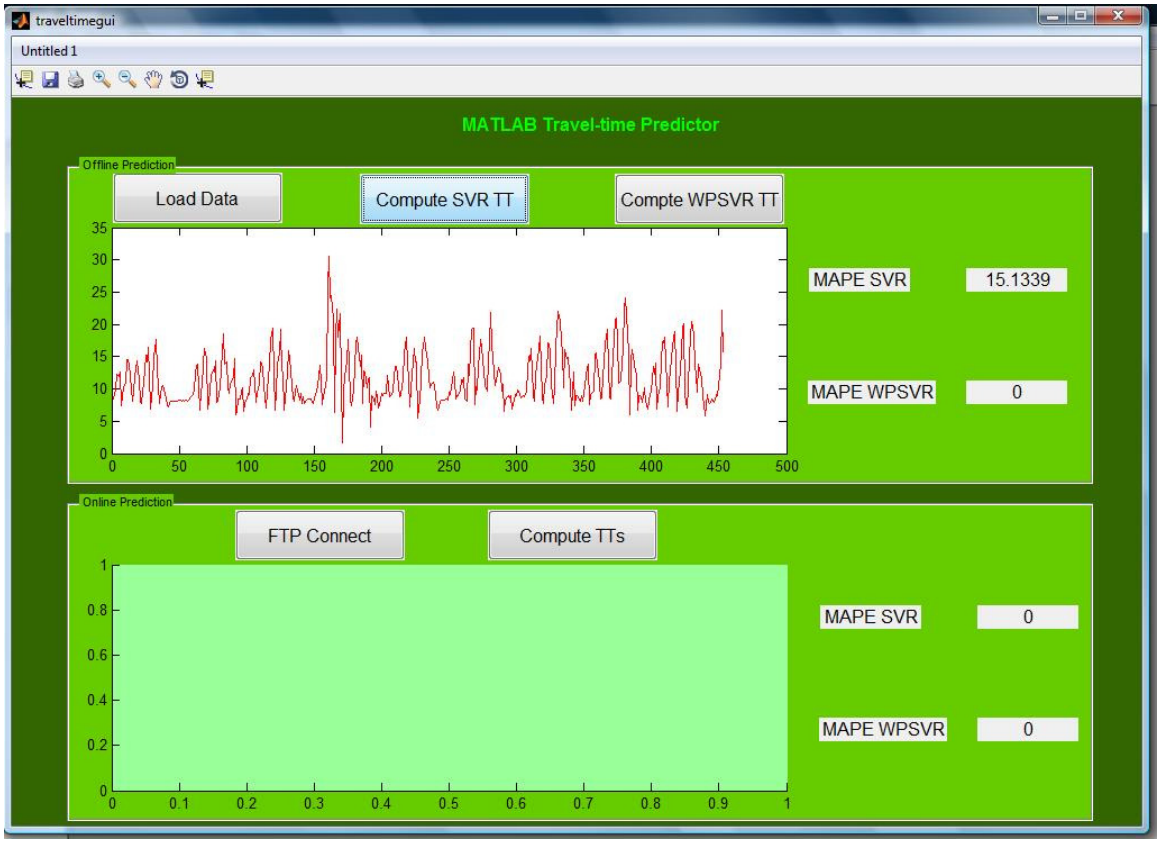


Figure 40: Screenshot of Compute SVR TT function of GUI

The output of the SVR and WPSVR models are computed for errors and the Mean Absolute percentage error as defined in (11) is shown in their respective textboxes in the GUI as shown in figure 42.

Figure 410: Screenshot of Compute WPSVR TT function of GUI

The online module of the GUI performs two main functions. It has a FTP Connect button with is pressed to initiate a file transfer protocol connection between the Caltrans server at UC Berkeley and the local machine.

The Caltrans server houses the data files in the form of flat files. The dataset in both 30 second and 5 minutes aggregated forms are archived everyday in separate folders. However, during the span of 24 hours only the current data i.e. the data of the latest timestamp is accessible from the current folder

The FTP Connect is a batch command file, which after connecting to the Caltrans data server retrieves the flat files of the preconfigured District roadway. The flat are or-

ganized in the data server at the California District level. Therefore, after every 5 minutes 12 data files representing each of the 12 California Districts is updated on the server.

Figure 41: Screenshot of FTP Connect function of GUI

The Compute TTs button on the online prediction panel also performs a batch function, It computes the wavelet packet transform of the reshaped dataset and subsequently calculates the travel-time using both models. Finally it displays the MAPE of each against their respective textbox.

Figure 42: Screenshot of Compute TTs function of GUI

REFERENCES

[1] J. Rice, and E. Van Zwet, "A simple and effective method for predicting travel times on freeways," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 5, no. 3, pp. 200-207, 2004.

[2] C. M. Kuchipudi, S. I. J. Chien, and Trb, "Development of a hybrid model for dynamic travel-time prediction," *Transportation Data Research: Planning and Administration*, no. 1855, pp. 22-31, 2003.

[3] A. Dharia, and H. Adeli, "Neural network model for rapid forecasting of freeway link travel time," *Engineering Applications of Artificial Intelligence,* vol. 16, no. 7-8, pp. 607-613, Oct-Dec, 2003.

[4] D. Park, L. R. Rilett, and G. Han, "Spectral basis neural networks for real-time travel time forecasting," *Journal of Transportation Engineering-Asce,* vol. 125, no. 6, pp. 515-523, Nov-Dec, 1999.

[5] D. Park, L. R. Rilett, and G. H. Han, *Forecasting multiple-period freeway link travel times using neural networks with expanded input nodes*, 1998.

[6] L. R. Rilett, D. Park, and Trb, "Direct forecasting of freeway corridor travel times using spectral basis neural networks," *Travel Patterns and Behavior; Effects of Communications Technology: Planning and Administration*, Transportation Research Record 1752, pp. 140-147, 2001.

[7] C. Cortes, and V. Vapnik, "Support-vector networks," *Machine learning,* vol. 20, no. 3, pp. 273-297, 1995.

[8] V. N. Vapnik, *The nature of statistical learning theory*: Springer Verlag, 2000.

[9] V. Vapnik, S. E. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing."

[10] T. B. Trafalis, and H. Ince, "Support vector machine for regression and applications to financial forecasting." pp. 348-353 vol. 6.

[11] M. Song, C. M. Breneman, J. Bi, N. Sukumar, K. P. Bennett, S. Cramer, and N. Tugcu, "Prediction of protein retention times in anion-exchange chromatography systems using support vector regression," *Journal of chemical information and computer sciences,* vol. 42, no. 6, pp. 1347-1357, 2002.

[12] H. van Lint, *Reliable travel time prediction for freeways*: Netherlands TRAIL Research School, 2004.

[13] M. Chen, and S. I. Chien, "Dynamic freeway travel-time prediction with probe vehicle data: Link based versus path based," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1768, no. 1, pp. 157-161, 2001.

[14] C. H. Wu, J. M. Ho, and D. Lee, "Travel-time prediction with support vector regression," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 5, no. 4, pp. 276-281, 2004.

[15] J. Kwon, B. Coifman, and P. Bickel, "Day-to-day travel-time trends and travel-time prediction from loop-detector data," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1717, no. -1, pp. 120-129, 2000.

[16] F. Wang, G. Tan, and Y. Fang, "Multiscale wavelet support vector regression for traffic flow prediction." pp. 319-322.

[17] S. Yao, C. Hu, and W. Peng, "Server Load Prediction Based on Wavelet Packet and Support Vector Regression." pp. 1016-1019.

[18] M. Faisal, and A. Mohamed, "A New Technique to Predict the Sources of Voltage Sags using Support Vector Regression based S-Transform."

[19] H. van Lint, S. P. Hoogendoorn, and H. J. van Zuylen, "State space neural networks for freeway travel time prediction," *Artificial Neural Networks - Icann 2002*, Lecture Notes in Computer Science J. R. Dorronsoro, ed., pp. 1043-1048, 2002.

[20] S. Innamaa, "Short-term prediction of travel time using neural networks on an interurban highway," *Transportation,* vol. 32, no. 6, pp. 649-669, Nov, 2005.

[21] J. W. C. van Lint, "Reliable real-time framework for short-term freeway travel time prediction," *Journal of Transportation Engineering-Asce,* vol. 132, no. 12, pp. 921-932, Dec, 2006.

[22] N. Zou, J. W. Wang, G. L. Chang, and Ieee, *A Reliable Hybrid Prediction Model for Real-time Travel Time Prediction with Widely Spaced Detectors*, 2008.

[23] N. Zou, J. W. Wang, G. L. Chang, and J. Paracha, "Application of Advanced Traffic Information Systems Field Test of a Travel-Time Prediction System with Widely Spaced Detectors," *Transportation Research Record*, no. 2129, pp. 62-72, 2009.

[24] J. W. C. van Lint, S. P. Hoogendoorn, and H. J. van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data," *Transportation Research Part C-Emerging Technologies,* vol. 13, no. 5-6, pp. 347-369, Oct-Dec, 2005.

[25] H. J. M. Van Grol, M. Danech-Pajouh, S. Manfredi, and J. Whittaker, *DACCORD: On-line travel time prediction*, 1999.

[26] H. Chen, M. S. Dougherty, and H. R. Kirby, "The effects of detector spacing on traffic forecasting performance using neural networks," *Computer-Aided Civil and Infrastructure Engineering,* vol. 16, no. 6, pp. 422-430, 2001.

[27] P. Yi, S. Ding, H. Wei, and G. W. Saylor, "Investigating the Effect of Detector Spacing on Midpoint-Based Travel Time Estimation," *Journal of Intelligent Transportation Systems,* vol. 13, no. 3, pp. 149-159, 2009, 2009.

[28] J. Kwon, K. Petty, and P. Varaiya, "Probe Vehicle Runs or Loop Detectors?: Effect of Detector Spacing and Sample Size on Accuracy of Freeway Congestion Monitoring," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2012, no. -1, pp. 57-63, 2007.

[29] C. Chen, J. Kwon, J. Rice, A. Skabardonis, and P. Varaiya, "Detecting errors and imputing missing data for single-loop surveillance systems," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1855, no. -1, pp. 160-167, 2003.

[30] L. N. Jacobson, N. L. Nihan, and J. D. Bender, "Detecting erroneous loop detector data in a freeway traffic management system," *Transportation Research Record*, no. 1287, 1990.

[31] C. D. R. Lindveld, R. Thijs, P. H. L. Bovy, and N. J. Van der Zijpp, "Evaluation of online travel time estimators and predictors," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1719, no. -1, pp. 45-53, 2000.

[32] J. W. C. van Lint, and N. Van der Zijpp, "Improving a travel-time estimation algorithm by using dual loop detectors," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1855, no. -1, pp. 41-48, 2003.

[33] M. Saito, and T. Watanabe, "Prediction and Dissemination system for travel time utilizing vehicle detectors."

[34] I. Steven, J. Chien, and C. M. Kuchipudi, "Dynamic travel time prediction with real-time and historic data," *Journal of transportation engineering,* vol. 129, pp. 608, 2003.

[35] X. Zhang, and J. A. Rice, “Short-term travel time prediction,” *Transportation Research Part C: Emerging Technologies,* vol. 11, no. 3-4, pp. 187-210, 2003.

[36] H. Sun, H. X. Liu, H. Xiao, R. R. He, and B. Ran, “Use of local linear regression model for short-term traffic forecasting,” *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1836, no. -1, pp. 143-150, 2003.

[37] M. S. Ahmed, and A. R. Cook, “Analysis of freeway traffic time-series data by using Box-Jenkins techniques,” *Transportation Research Record*, no. 722, 1979.

[38] M. Levin, and Y. D. Tsao, “On Forecasting Freeway Occupancies and Volumes (Abridgment),” *Transportation Research Record*, no. 773, 1980.

[39] T. Oda, "An algorithm for prediction of travel time using vehicle sensor data." pp. 40-44.

[40] M. P. D'Angelo, H. M. Al-Deek, and M. C. Wang, “Travel-time prediction for freeway corridors,” *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1676, no. -1, pp. 184-191, 1999.

[41] S. Ishak, and H. Al-Deek, “Performance evaluation of short-term time-series traffic prediction model,” *Journal of transportation engineering,* vol. 128, no. 6, pp. 490-498, 2002.

[42] H. F. Ji, A. G. Xu, X. Sui, and L. Y. Li, *The Applied Research of Kalman in the Dynamic Travel Time Prediction*, 2010.

[43] J. W. C. van Lint, S. P. Hoogendoorn, H. J. van Zuylen, and Trb, “Freeway travel time prediction with state-space neural networks - Modeling state-space dynamics with recurrent neural networks,” *Advanced Traffic Management Systems for Freeways and Traffic Signal Systems 2002: Highway Operations, Capacity, and Traffic Control*, no. 1811, pp. 30-39, 2002.

[44] C. P. I. van Hinstiergen, J. W. C. van Lint, and H. J. van Zuylen, “Bayesian Training and Committees of State-Space Neural Networks for Online Travel Time Prediction,” *Transportation Research Record*, no. 2105, pp. 118-126, 2009.

[45] D. J. Park, L. R. Rilett, and C. Natl Res, "Forecasting multiple-period freeway link travel times using modular neural networks," *Land Use and Transportation Planning and Programming Applications*, Transportation Research Record 1617, pp. 163-170, 1998.

[46] Y. Lee, and Ieee, “Freeway Travel Time Forecast Using Artifical Neural Networks With Cluster Method,” *Fusion: 2009 12th International Conference on Information Fusion, Vols 1-4*, pp. 1331-1338, 2009.

[47] H. Liu, R. H. He, K. Zhang, and J. Li, *A Neural Network Model for Travel Time Prediction*, 2009.

[48] V. Turchenko, V. Demchuk, and U. Lviv Polytech Natl, *Neural-based vehicle travel time prediction noised by different influence factors*, 2006.

[49] C. F. Shao, Y. L. Gu, and K. L. Zhang, *A study on dynamic travel time forecast with neural networks*, 2002.

[50] J. Ya, G. L. Chang, H. W. Ho, Y. Liu, and Ieee, *Variation Based Online Travel Time Prediction Using Clustered Neural Networks*, 2008.

[51] G. Ghiani, D. Gulli, F. Mari, R. Simino, and R. Trunfio, "Weather-dependent road travel time forecasting using a neural network," *Urban Transport Xiv: Urban Transport and the Environment in the 21st Century*, Wit Transactions on the Built Environment C. A. Brebbia, ed., pp. 505-514, 2008.

[52] L. Vanajakshi, L. R. Rilett, and Ieee, *Support vector machine technique for the short term prediction of travel time*, 2007.

[53] M. Ben-Akiva, M. Bierlaire, H. Koutsopoulos, and R. Mishalani, "DynaMIT: a simulation-based system for traffic prediction."

[54] M. Ben-Akiva, M. Bierlaire, D. Burton, H. N. Koutsopoulos, and R. Mishalani, “Network state estimation and prediction for real-time traffic management,” *Networks and Spatial Economics,* vol. 1, no. 3, pp. 293-318, 2001.

[55] S. G. Mallat, *A wavelet tour of signal processing*: Academic Pr, 1999.

[56] M. Vetterli, and J. Kovačević, *Wavelets and subband coding*: Prentice Hall PTR Upper Saddle River, NJ, 1995.

[57] S. Yao, and C. Hu, "Prediction of Server Load Based on Wavelet-Support Vector Regression-Moving Average." pp. 833-837.

[58] R. Ahmed, "Wavelet-based Image Compression using Support Vector Machine Learning and Encoding Techniques." pp. 162-166.

[59] Y. Li, and H. Hu, "Image compression using wavelet support vector machines," *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues*, pp. 922-929: Springer, 2007.

[60] J.-M. Chen, L. Li, and L.-Y. Nie, "Wavelet image compression by using hybrid kernel SVM." pp. 3056-3060.

[61] Y. Li, Q. Yang, and R. Jiao, "A Novel Image Compression Algorithm Using the Second Generation of Curvelet Transform and SVM." pp. 117-121.

[62] Y. Li, Q. Yang, and R. Jiao, "Image compression scheme based on curvelet transform and support vector machine," *Expert Systems with Applications,* vol. 37, no. 4, pp. 3063-3069, 2010.

[63] J. Robinson, and V. Kecman, "Combining support vector machine learning with the discrete cosine transform in image compression," *Neural Networks, IEEE Transactions on,* vol. 14, no. 4, pp. 950-958, 2003.

[64] T. P. Banerjee, S. Das, J. Roychoudhury, and A. Abraham, "Implementation of a new hybrid methodology for fault signal classification using short-time fourier transform and support vector machines." pp. 219-225.

[65] S.-y. Zhang, X. Xue, and X. Zhang, "Feature extraction and classification with wavelet transform and support vector machines." pp. 3795-3798.

[66] Z. Moravej, A. Abdoos, and M. Pazoki, "Detection and classification of power quality disturbances using wavelet transform and support vector machines," *Electric Power Components and Systems,* vol. 38, no. 2, pp. 182-196, 2009.

[67] T.-S. Li, "Applying wavelets transform and support vector machine for copper clad laminate defects classification," *Computers & Industrial Engineering,* vol. 56, no. 3, pp. 1154-1168, 2009.

[68] F.-M. Schleif, M. Lindemann, M. Diaz, P. Maaß, J. Decker, T. Elssner, M. Kuhn, and H. Thiele, "Support vector classification of proteomic profile spectra based on feature extraction with the bi-orthogonal discrete wavelet transform," *Computing and visualization in science,* vol. 12, no. 4, pp. 189-199, 2009.

[69] X. Li, P. Nie, Z.-J. Qiu, and Y. He, "Using wavelet transform and multi-class least square support vector machine in multi-spectral imaging classification of Chinese famous tea," *Expert Systems with Applications,* vol. 38, no. 9, pp. 11149-11159, 2011.

[70] Z.-y. Liu, J.-j. Shi, L.-w. Zhang, and J.-f. Huang, "Discrimination of rice panicles by hyperspectral reflectance data based on principal component analysis and support vector classification," *Journal of Zhejiang University SCIENCE B,* vol. 11, no. 1, pp. 71-78, 2010.

[71] M.-L. O'Connell, T. Howley, A. G. Ryder, M. N. Leger, and M. G. Madden, "Classification of a target analyte in solid mixtures using principal component analysis, support vector machines, and Raman spectroscopy." pp. 340-350.

[72] M. Khelil, M. Boudraa, A. Kechida, and R. Drai, "Classification of defects by the SVM method and the principal component analysis (PCA)," *Transactions on Engineering, Computing andTechnology,* vol. 9, pp. 226-231, 2005.

[73] M. Maitra, A. Chatterjee, and F. Matsuno, "A novel scheme for feature extraction and classification of magnetic resonance brain images based on Slantlet Transform and Support Vector Machine." pp. 1130-1134.

[74] K. He, K. K. Lai, and J. Yen, "A hybrid slantlet denoising least squares support vector regression model for exchange rate prediction," *Procedia Computer Science,* vol. 1, no. 1, pp. 2397-2405, 2010.

[75] S. Abbasion, A. Rafsanjani, A. Farshidianfar, and N. Irani, "Rolling element bearings multi-fault classification based on the wavelet denoising and support vector machine," *Mechanical Systems and Signal Processing,* vol. 21, no. 7, pp. 2933-2945, 2007.

[76] H. Cheng, J. Tian, J. Liu, and Q. Yu, "Wavelet domain image denoising via support vector regression," *Electronics Letters,* vol. 40, no. 23, pp. 1479-1481, 2004.

[77] Y. Qian, J. Xia, K. Fu, and R. Zhang, "Network traffic forecasting by support vector machines based on empirical mode decomposition denoising." pp. 3327-3330.

[78] A. Yusuf, and V. K. Madisetti, "Configuration for Predicting Travel-Time Using Wavelet Packets and Support Vector Regression," *Journal of Transportation Technologies,* vol. 3, pp. 220-231, 2013.

[79] M. Zavar, S. Rahati, M.-R. Akbarzadeh-T, and H. Ghasemifard, "Evolutionary model selection in a wavelet-based support vector machine for automated seizure detection," *Expert Systems with Applications,* vol. 38, no. 9, pp. 10751-10758, 2011.

[80] K. Ferroudji, N. Benoudjit, M. Bahaz, and A. Bouakaz, "Selection of a suitable mother wavelet for microemboli classification using SVM and RF signals." pp. 1-4.

[81] B. N. Singh, and A. K. Tiwari, "Optimal selection of wavelet basis function applied to ECG signal denoising," *Digital Signal Processing,* vol. 16, no. 3, pp. 275-287, 2006.

[82] X. Ma, C. Zhou, and I. Kemp, “Automated wavelet selection and thresholding for PD detection,” *Electrical Insulation Magazine, IEEE,* vol. 18, no. 2, pp. 37-45, 2002.

[83] K. Ranjeet, "Retained signal energy based optimal wavelet selection for Denoising of ECG signal using modifide thresholding." pp. 196-199.

[84] J. Li, T. Jiang, S. Grzybowski, and C. Cheng, “Scale dependent wavelet selection for de-noising of partial discharge detection,” *Dielectrics and Electrical Insulation, IEEE Transactions on,* vol. 17, no. 6, pp. 1705-1714, 2010.

[85] R. Yan, *Base wavelet selection criteria for non-stationary vibration analysis in bearing health diagnosis*: ProQuest, 2007.

[86] K. Kannan, and S. A. Perumal, “Optimal Decomposition Level of Discrete, Stationary and Dual Tree Complex Wavelet Transform for Pixel based Fusion of Multi-focused Images.”

[87] Z. Wang, and A. C. Bovik, “A universal image quality index,” *Signal Processing Letters, IEEE,* vol. 9, no. 3, pp. 81-84, 2002.

[88] C. Xydeas, and V. Petrovic, “Objective image fusion performance measure,” *Electronics Letters,* vol. 36, no. 4, pp. 308-309, 2000.

[89] V. P. Shah, N. H. Younan, S. S. Durbha, and R. L. King, “A systematic approach to wavelet-decomposition-level selection for image information mining from geospatial data archives,” *Geoscience and Remote Sensing, IEEE Transactions on,* vol. 45, no. 4, pp. 875-878, 2007.

[90] N. Doghmane, Z. Baarir, N. Terki, and A. Ouafi, “Study of effect of filters and decomposition level in wavelet image compression.”