11-1-2014

Harvesting Data from Advanced Technologies

Xindong Wu

University of Vermont, xwu@uvm.edu

REPORT 2014 - 11

VERMONT AGENCY OF TRANSPORTATION

Research and Development Section
Research Report

HARVESTING DATA FROM ADVANCED TECHNOLOGIES

Report 2014 – 11

November 2014

HARVESTING DATA FROM ADVANCED TECHNOLOGIES

Report 2014 – 11

November 2014

Reporting on SPR-RAC-723

STATE OF VERMONT
AGENCY OF TRANSPORTATION

RESEARCH & DEVELOPMENT SECTION

BRIAN R. SEARLES, SECRETARY OF TRANSPORTATION
CHRIS COLE, DIRECTOR OF POLICY, PLANNING AND INTERMODAL DEVELOPMENT
JOE SEGALE, P.E./PTP, PLANNING, POLICY & RESEARCH
WILLIAM E. AHEARN, P.E., RESEARCH & DEVELOPMENT

Prepared By:

University of Vermont, Transportation Research Center
Xindong Wu, Ph.D., Professor, Department of Computer Science

Transportation Research Center
Farrell Hall
210 Colchester Avenue
Burlington, VT 05405
Phone: (802) 656-1312
Website: www.uvm.edu/transportationcenter

The University of Vermont

The information contained in this report was compiled for the use of the Vermont Agency of Transportation (VTrans). Conclusions and recommendations contained herein are based upon the research data obtained and the expertise of the researchers, and are not necessarily to be construed as Agency policy. This report does not constitute a standard, specification, or regulation. VTrans assumes no liability for its contents or the use thereof.

Technical Report Documentation Page

1. Report No. 2014-11	2. Government Accession No. - - -	3. Recipient's Catalog No. - - -

4. Title and Subtitle HARVESTING DATA FROM ADVANCED TECHNOLOGIES	5. Report Date November 2014 6. Performing Organization Code
7. Author(s) Xindong Wu, Ph.D.	8. Performing Organization Report No. 2014-11
9. Performing Organization Name and Address UVM Transportation Research Center Farrell Hall] 210 Colchester Avenue Burlington, VT 05405	10. Work Unit No. 11. Contract or Grant No. RSCH017-723
12. Sponsoring Agency Name and Address Vermont Agency of Transportation Materials and Research Section 1 National Life Drive National Life Building Montpelier, VT 05633-5001 Federal Highway Administration Division Office Federal Building Montpelier, VT 05602	13. Type of Report and Period Covered Final 2012 – 2013 14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Data streams are emerging everywhere such as Web logs, Web page click streams, sensor data streams, and credit card transaction flows. Different from traditional data sets, data streams are sequentially generated and arrive one by one rather than being available for random access before learning begins, and they are potentially huge or even infinite that it is impractical to store the whole data.

To study learning from data streams, we target online learning, which generates a best–so far model on the fly by sequentially feeding in the newly arrived data, updates the model as needed, and then applies the learned model for accurate real-time prediction or classification in real-world applications. Several challenges arise from this scenario: first, data is not available for random access or even multiple access; second, data imbalance is a common situation; third, the performance of the model should be reasonable even when the amount of data is limited; fourth, the model should be updated easily but not frequently; and finally, the model should always be ready for prediction and classification. To meet these challenges, we investigate streaming feature selection by taking advantage of mutual information and group structures among candidate features. Streaming feature selection reduces the number of features by removing noisy, irrelevant, or redundant features and selecting relevant features on the fly, and brings about palpable effects for applications: speeding up the learning process, improving learning accuracy, enhancing generalization capability, and improving model interpretation. Compared with traditional feature selection, which can only handle pre-given data sets without considering the potential group structures among candidate features, streaming feature selection is able to handle streaming data and select meaningful and valuable feature sets with or without group structures on the fly.

In this research, we propose 1) a novel streaming feature selection algorithm (GFSSF, Group Feature Selection with Streaming Features) by exploring mutual information and group structures among candidate features for both group and individual levels of feature selection from streaming data, 2) a lazy online prediction model with data fusion, feature selection and weighting technologies for real-time traffic prediction from heterogeneous sensor data streams, 3) a lazy online learning model (LB, Live Bayes) with dynamic resampling technology to learn from imbalanced embedded mobile sensor data streams for real-time activity recognition and user recognition, and 4) a lazy update online learning model (CMLR, Cost-sensitive Multinomial Logistic Regression) with streaming feature selection for accurate real-time classification from imbalanced and small sensor data streams. Finally, by integrating traffic flow theory, advanced sensors, data gathering, data fusion, feature selection and weighting, online learning and visualization technologies to estimate and visualize the current and future traffic, a real-time transportation prediction system named VTraffic is built for the Vermont Agency of Transportation.

17. Key Words Data Streaming, Data Noise, Predictive Modeling, VTraffic	18. Distribution Statement No Restrictions.

19. Security Classif. (of this report) - - -	20. Security Classif. (of this page) - - -	21. No. Pages	22. Price - - -

Abstract

Data streams are emerging everywhere such as Web logs, Web page click streams, sensor data streams, and credit card transaction flows. Different from traditional data sets, data streams are sequentially generated and arrive one by one rather than being available for random access before learning begins, and they are potentially huge or even infinite that it is impractical to store the whole data.

To study learning from data streams, we target online learning, which generates a best–so far model on the fly by sequentially feeding in the newly arrived data, updates the model as needed, and then applies the learned model for accurate real-time prediction or classification in real-world applications. Several challenges arise from this scenario: first, data is not available for random access or even multiple access; second, data imbalance is a common situation; third, the performance of the model should be reasonable even when the amount of data is limited; fourth, the model should be updated easily but not frequently; and finally, the model should always be ready for prediction and classification. To meet these challenges, we investigate streaming feature selection by taking advantage of mutual information and group structures among candidate features. Streaming feature selection reduces the number of features by removing noisy, irrelevant, or redundant features, and selecting relevant features on the fly, and brings about palpable effects for applications: speeding up the learning process, improving learning accuracy, enhancing generalization capability, and improving model interpretation. Compared with traditional feature selection, which can only handle pre-given data sets without considering the potential group structures among candidate features, streaming feature selection is able to handle streaming data and select meaningful and valuable feature sets with or without group structures on the fly.

In this research, we propose 1) a novel streaming feature selection algorithm (GFSSF, Group Feature Selection with Streaming Features) by exploring mutual information and group structures among candidate features for both group and individual levels of feature selection from streaming data, 2) a lazy online prediction model with data fusion, feature selection and weighting technologies for real-time traffic prediction from heterogeneous sensor data streams, 3) a lazy online learning model (LB, Live Bayes) with dynamic resampling technology to learn from imbalanced embedded mobile sensor data streams for real-time activity recognition and user recognition, and 4) a lazy update online learning model (CMLR, Cost-sensitive Multinomial Logistic Regression) with streaming feature selection for accurate real-time classification from imbalanced and small sensor data streams. Finally, by integrating traffic flow theory, advanced sensors, data gathering, data fusion, feature selection and weighting, online learning and visualization technologies to estimate and visualize the current and future traffic, a real-time transportation prediction system named VTraffic is built for the Vermont Agency of Transportation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Traffic congestion is a situation of transportation systems that occurs as the saturation of road network capacity, due to increased traffic volume or interruptions, and is characterized by slower speeds, increased vehicular queuing, and longer trip times. Congestion is one external cost of transport and the reduction of its impact is often one of the primary objectives for transport policy makers. Traffic congestion, continuously one of the major problems in various transportation systems, has many negative effects on travelers, businesses, agencies and cities. One significant aspect is the value of the wasted fuel and additional time. The top 15 urban areas include about 58 percent of the delay estimated for 2010, and the top 20 areas account for over 65 percent of the annual delay. Based on wasted time and fuel, traffic congestion costs about $115 billion in the 439 urban areas in 2010 [1].

The negative impacts of traffic congestion may be alleviated by providing timely and reliable prediction information to system dispatchers and motorists [2, 3, 4, 5, 6]. However, traffic situations vary significantly depending on the weather situation, the season of the year, the day of the week, and even the time of day. In addition, the capacity, which is often mistakenly considered to be constant, may vary because of weather, work zones, or traffic incidents and so on [7, 8, 9]. Furthermore, those conditions are not independent of each other, and most of them are interrelated explicitly or implicitly. Therefore, there are a great number of differing and changing circumstances which cause or aggravate congestion. It is a great challenge to estimate under which conditions a “congestion” may occur suddenly.

The use of advanced technologies such as wireless communication and sensors on transportation networks has made a significant increase in traffic data available in recent years. A series of infrastructure-based in-road detectors, roadside sensors, bridge sensors, videos and floating vehicles have facilitated better transportation data collection for both operations and planning. Many transportation agencies have developed techniques for collecting these data realtime and storing them historically, while few have been making a full use of them. The Federal Highway Administration (FHWA) has shown significant interest in this topic as described in the recent study of Cross-Cutting Studies and State of the Practice Reviews: Archived and Use of Intelligent Transportation Systems (ITS) Generated Data (http://www.its.dot.gov/). With this data, traffic prediction has the ability to improve traffic conditions and to reduce travel delays by facilitating better utilization of available capacity. By integrating traffic flow theory, advanced sensors, data gathering, data fusion, data mining and visualization technologies, the current and future traffic can be estimated and visualized.

In this project, acoustic sensors were installed to monitor and to collect real-time data. Reliable predictions can be obtained from historical data and be verified and refined by the current and near future real-time data. This topic is emerging primarily at the federal level as a core strategy for better transportation system monitoring and management. Some unique characteristics of northern communities such as limited rural communication infrastructure compounded by congested primary arterial roadways, inclement weather conditions, and heavy seasonal tourism traffic demand also motivate us to examine in depth the impact of real-time data on transportation network management. Over the last few years, University of Vermont (UVM) researchers have begun to study the opportunities and challenges of ITS deployment in northern and rural communities.

1.2 Project Description

Three factors come together to motivate this research to harvest data for reliable real-time transportation network management: the availability of a great amount of data from a variety of sources; readily accessible advanced technologies and the desire to develop a knowledge-based system to improve transportation planning and operation, serving such purposes as congestion management, emergency evacuation, operation, planning and policy decision making; and the urgency of providing an integrated reliable rural and urban primary arterials traffic monitoring and management system in northern communities.

Objectives: The primary goal is to design and develop state-of-the-art data mining and fusion techniques and modeling tools that provide reliable and real-time transportation network management including traffic congestion prediction, incident identification and bridge structural health monitoring. This suite will be achieved by designing and using advanced data gathering, processing and mining tools to estimate the current and future transportation system performance. The resulting modeling tools will be enabled by emerging technologies suitable for implementation by transportation agencies in northern communities. A secondary goal is to quantify the effect of travel behavior changes and consequent planning and modeling challenges in the use of advanced technologies. Research findings will be disseminated to multidisciplinary academic outlets, transportation agencies, and emergency management and planning organizations. Specifically, the project will probe how data mining paradigms can be utilized to address three critical research problems that are at the heart of transportation operations at the present time:

1. An on-line architecture integrating roadway, bridge, traffic, and crash information for data usage;

2. Novel data fusion algorithms accounting for heterogeneous data sources and for real-time congestion and bridge structural performance monitoring, and incident detection on freeways and major rural and urban arterials with a certain degree of reliability; and

3. A Geographic Information Systems (GIS)-based visualization tool enabled with the proposed data fusion models and also as a function of landscape and meteorological variables.

1.3 Report Organization

Chapter 2 presents the methodology used in this project. Chapter 3 provides a description of the implementation. Chapter 4 shows the experimental results of the study. Finally, discussions and conclusions are given in Chapter 5.

Chapter 2

Methodology

To achieve the goals and objectives of this project, a work plan with various tasks has been developed, permitting a thorough investigation of harvesting real-time data for reliable congestion management, structural health monitoring and incident detection. In this chapter, we present solution methods for our project.

2.1 Data Fusion

2.1.1 Architecture for Data Sources

This project develops an on-line architecture integrating roadway, bridge, traffic, sensor data, and crash information for data usage, archiving and sharing over a freeway and arterial network. A web server is applied to store, analyze and visualize traffic and structural bridge data of a congestion map. A Google fusion table and the Apache Tomcat real-time Web tools are used in the server.

Typically, interstate highways are the busiest and most important main roads. There are two interstate highways I89 and I91 across the state of Vermont. Therefore, we choose I89 and I91 to deploy our system. Dozens of acoustic sensors have already been installed on the two highways.

This project adopts the Vaisala nu-metrics portable traffic analyzer NC-200 shown in Fig. 2.1 to collect real-time traffic data. The sensor can be installed quickly and easily, which is designed to detect accurate vehicle count, speed, and classification by utilizing vehicle magnetic imaging (VMI) technology. After data has been collected from a sensor, the data can be easily exported to the highway data management (HDM) software shown in Fig. 2.2, where it can be presented in the form of reports, charts and graphs [10].

As we all know, the impact of weather on traffic is very serious. Weather includes visibility, precipitation, wind, and temperature, which affects driver capabilities, vehicle performance, pavement friction, and roadway infrastructure to impact the state of the transportation system [11].

The Clarus system was established in 2004 to provide weather information to transportation managers and users to alleviate the effects of adverse weather

Figure 2.1: NC-200

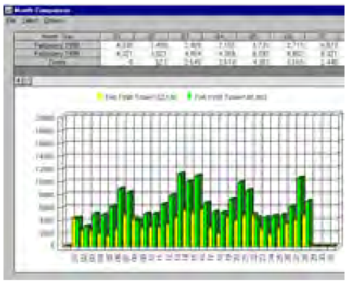


Figure 2.2: HDM Software

[12, 13]. This system belongs to Federal Highway Administration Research & Innovative Technology Administration, and it is provided as a public service. This project gets historical and real-time weather data from the Clarus system.

2.1.2 Innovative Data Fusion

Two areas have been covered in this task to summarize and synthesize the findings of existing literature: data fusion techniques; and the achievement of reliable results and feasibility for realtime implementation.

A data fusion strategy has been developed to improve the quality of available raw data gathered from different sensors and probe vehicles to ensure the information can be used for traffic state estimation. They are

a) data quality control, including error detection and data cleansing;

b) missing information acquisition; and

c) conflict resolution when overlapping data from different sources are inconsistent.

2.2 Predicting Transportation System State

2.2.1 Reliable Congestion Estimation

There are several major challenges in predicting a transportation system state using probe vehicle data. One is to determine its accuracy level with which sampled data for a given probe vehicle ratio can match the aggregated travel time experienced by the overall vehicle population on the roadway. Second, the probe data points are inherently complex as they are collected by diverse sources with different temporal and spatial attributes and are not related to a well defined common denominator. Third, the collected data points may not all represent the same field conditions. To respond to these challenges, the research team identifies the evolution of the individual travel time measurements between and within the aggregation periods based on data variability and use statistical inference methods, including Bayesian methods, to fuse highly variable and noisy data points to achieve a reliable and robust estimation of travel time.

Using the novel data fusion techniques, this project develops a methodology to obtain individual-link traffic state estimates for an integrated freeway and urban network. Estimation accuracy with the percentage of probe vehicles is also explored.

2.2.2 Incident Management

The provision of decision support tools to address non-recurring congestion in large complex networks is necessary for transportation system management. Developing new techniques to capitalize on the data obtained from advanced technologies is the major goal of this task. For instance, when a lane-blocking incident occurs, traffic backs up and a bottleneck forms where the incident has taken place. The data collected by the probe vehicles and in-road sensors are essentially time-space trajectory data sampled at a certain frequency. Analysis of these sampled trajectories produces an estimate of bottleneck locations and a decision on whether these locations are in unexpected places that might be attributed to an incident. However, it is not always easy to determine, especially when the incident does not significantly impact trajectories, e.g., when traffic demand is low. Furthermore, there is the possibility that false alarms may result from unexpected slowdowns not necessarily due to an incident. Therefore, the method developed for incident detection should carefully balance detection and false alarm rates by maximizing the use of all available information. The impacts of probe vehicle density along a corridor on detection accuracy is evaluated using mathematical optimization techniques.

2.2.3 Bridge Structural Health Monitoring

Bridges constitute critical nodes of transportation systems, and, therefore, ensuring their continuous operation is of the utmost importance for safe and efficient transportation. A novel approach to bridge condition assessment is Structural Health Monitoring (SHM), defined as the measurement of an operating and loading environment through use of a sensor system to track and evaluate incidents, anomalies, damage and deterioration. The ultimate goal of applying SHM is to detect, localize, and quantify the accumulated structural damage in real time.

Traditionally, damage assessment techniques are primarily based on the observation of the changes of model parameters, such as modal frequencies and vibration shapes. However, several investigations have pointed out that this technique can be efficient only if very precise measurements are available or a large level of damage exists. It is obvious that a more robust damage assessment algorithm must be developed. The objective of this task is to develop a Fuzzy Expert System of SHM with integrative information system design to perform damage detection, localization, and severity estimation for bridge structures subject to traffic loading. The proposed procedure is used for safety assessment of structures when the data for safety analysis is insufficient or when a decision for safety management is urgently required. In case exact measurement such as when a nondestructive test cannot be utilized to assess a damaged point for any reason, an expert's estimation with visual inspection based on the data of a nondestructive test at a near point can be used. The system reliability-based safety assessment may then be performed approximately but rationally by converting the qualitative expert's estimation to quantitative terms in the proposed procedure. The fuzzy expert system will also be used as a classification tool to show that the proposed method can identify different structural conditions as compared to other methods based on non-reduced and ordinary feature extraction. The system includes statistical pattern recognition algorithms that analyze statistical distributions of the measured or derived features to enhance the damage identification process.

2.3 Feature Selection and Weighting

Our real-time traffic predicting algorithm (Algorithm 1) is based on the k-NN algorithm, and k-NN in its most basic form operates under the implicit assumption that all features are of equal importance. In fact, different features possess different information with different importance. Especially, when irrelevant, noisy and redundant features influence the neighborhood search to the same degree as highly relevant features, the accuracy is likely to deteriorate. Therefore, removing irrelevant, noisy and redundant features and assigning important features higher weight values in the preprocessing phrase before learning begins are very important for achieving a good prediction accuracy.

2.3.1 Feature Selection

Feature selection, a process of selecting an optimal subset from the original candidate feature set according to certain criteria, removes noisy, irrelevant, or redundant features, reduces the number of features, and brings about palpable effects for applications: speeding up the learning process, improving learning accuracy, enhancing generalization capability, and improving model interpretation [14]. It is noticeable that among different evaluation criteria, information metric seems to be more comprehensively studied. The main reason is that information entropy is a good measurement to quantify the uncertainty of a feature [15].

There are many feature selection algorithms using information metric such as MIFS [16] and mRMR [17]. These algorithms can efficiently remove irrelevant, noisy and redundant features and select relevant features (such as traffic speed, number of vehicles and visibility in this project) for producing accuracy prediction.

2.3.2 Feature Weighting

Feature weighting is a technique for approximating the optimal degree of the influence of individual features. When successfully applied, relevant features are attributed high weight values, whereas irrelevant, noisy and redundant features are given weight values close to zero.

RELIEF [18, 19] is a famous feature weighting algorithm, it arranges features in a descending order and assigns weight values for features according to their priorities. Therefore, RELIEF can be used for feature weighting and also discarding features with weight values below a certain threshold.

MIFS and mRMR arrange features in a descending order according to their priorities too, but they do not assign weight values for selected features. In this project, the weight values of their selected features are simply set to $1/rank$, where $rank$ is the order of a selected feature. Obviously, this strategy can guarantee the more important features get higher weight values.

2.4 Visualization

The visualization tool is developed in a GIS web-enabled format with the analytical models developed. GIS is also applied to manage spatial information on the location of bridges on a regional map and the position of sensors on a bridge model. It provides users with an interactive interface between the user and the system so that users cannot only access the raw data of various sensors from the central database at any time but also visualize or further process the data. For instance, the map allows users to select specific roadway segments to compute their travel time and speed estimates. The real-world data provided by transportation agencies (including the weather data) and additional data generated from well-calibrated, microscopic traffic simulation models can produce sufficient data sets for these implementations.

2.5 Other Technologies

- K-nearest neighbors classifier

  The k-nearest neighbor algorithm (k-NN) is a method for classifying objects based on the closest training examples in the feature space. In our data model, we use the k-NN approach to deal with noisy data and integrate heterogeneous traffic data sources.

- Oracle data management and the Apache Tomcat real-time Web tool are used in our server.

- A Google fusion table in the cloud is employed for data storage.

- A Google map is adopted for visualization.

Chapter 3

System Design and Implementation

The main framework of our system VTraffic is given in Fig. 3.1. In this project, acoustic sensors were installed on highways, I89 and I91, within the territory of Vermont to monitor and to collect real-time data.

Firstly, we design a data fusion strategy to improve the quality of raw data gathered from different sensors and other available historical data to ensure the information can be used for traffic estimation. Secondly, we use data fusion on the data which comes from heterogeneous data sources to get useful knowledge and store in a Google fusion table in the cloud [20]. Thirdly, a prediction model is built from the knowledge set for real-time traffic predicting, and the real-time data is employed for verifying previous predictions and refining the model. Finally, a web portal is implemented using Google maps for visualization [21, 22]. All components in Fig. 3.1 are controlled through the set of system parameters.

3.1 Data Collection

The traffic and weather historical data recorded by transportation managers and sensors are obtained at the very beginning. Considering the data transmission efficiency and computational performance, NC-200 sensors continuously gather real-time traffic data, but only report data once every five minutes. In general, the weather does not occur great changes in a relatively short period of time. Therefore, the real-time weather data are fetched once every five minutes from the Clarus system.

3.2 Data Fusion of Heterogeneous Sources

Many transportation agencies have developed techniques for collecting the real-time data and storing it historically, while few have been making full use of it.

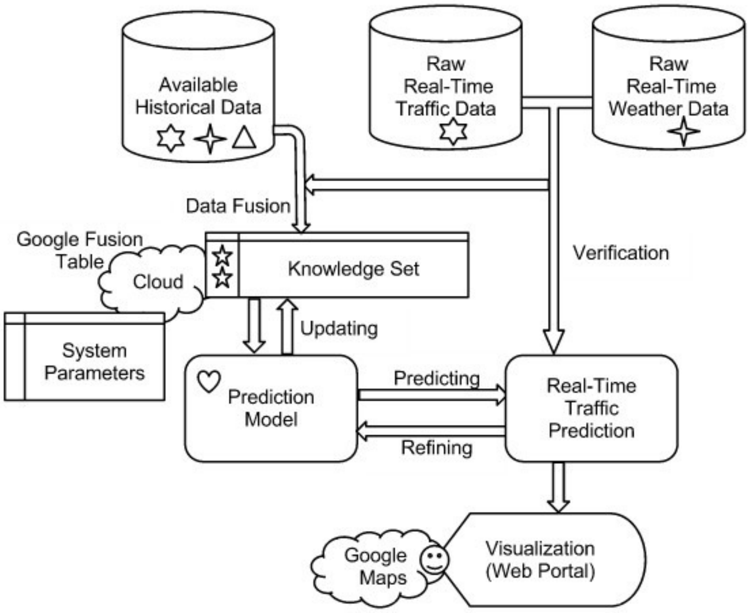


Figure 3.1: VTraffic: The Main Framework

In order to make full use of all available data, we have designed a strategy to integrate data from heterogeneous sources.

3.2.1 Data Preprocessing

The raw attributes of traffic data collected by NC-200 sensors and those of weather data obtained from the Clarus system are listed in Table 3.1. We also have some historical traffic data filled by transportation managers, however, this data only provides general traffic information about some particular sites, we can only use them for verification during preprocessing.

Table 3.1: Data Formats

Data Source	Data Format								
NC-200	Location	Date	Time	VehicleCount	Volume	Speed	Report time	Occup	...
Clarus	Location	Date&Time		Temperature	Humidity	WindSpeed	WindDirect	Visibility	...
Integration	Location	Date	Time	Confidence	Parents	TrafficAttrs	WeatherAttrs	Children	...

The historical data is preprocessed with the following steps:
1) Remove useless attributes;
2) Guess missing values;
3) Correct wrong data which can be corrected;
4) Remove wrong data which cannot be corrected;
5) Remove redundant data.

3.2.2 Data Fusion

The traffic data and the weather data are different in source, sensor type, location, time, and data format. Therefore, we have to fuse them first, and the format of the integration data is shown in Table 3.1.

Data fusion consists of the following steps:

1. As mentioned in Section 3.1, the data reported by NC-200 sensors contain the real-time traffic of the last five minutes. We first split the data into twenty pieces, each interval of fifteen seconds, and the data format remains the same.
2. From Table 3.1 we can easily observe that weather data has the attribute "Date & Time" while there are two attributes "Date" and "Time" in traffic data. We split "Date & Time" into "Date" and "Time".
3. By adding the weather data to each traffic data piece, we fuse those two data sets into one. Since the location and time between the weather data and the traffic data may not be exactly the same, we just choose the weather data with the closest time and location.
4. The integration data generated in Step 3, also called the knowledge set, is stored in a Google fusion table in the cloud [20].

3.3 Traffic Modeling

As we discussed in Section 3.1, our sensors collect real-time traffic data once every five minutes. Therefore, we do not know the real-time traffic between any two consecutive data collection until the next data has arrived. However, five minutes on the highway is a relatively long time, we have to model the real-time traffic. The continuous real-time traffic can be modeled according to the real-time data collected by the sensors and all other available knowledge generated by the data fusion process in Section 3.2. The model will be used for real-time traffic prediction and travel-time guidance.

3.3.1 Modeling

In our VTraffic system, there are two interstate highways, I89 and I91, and obviously, each of them has two directions south and north, which means there are four highways: I89N, I89S, I91N, and I91S. Therefore, we have to model the real-time traffic of the four highways. To simplify, we assume that the real-time traffics of the four highways are independent of each other.

The prediction model for a highway is a 4-tuple (Σ, Υ, Ξ, Δ), where

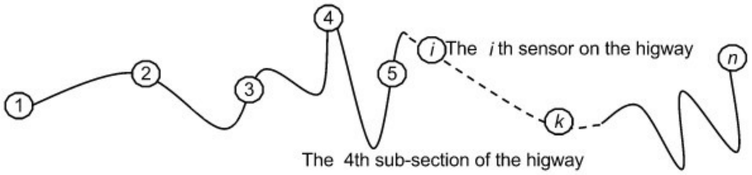

1
2
3
4
5
i
The i th sensor on the higway
k
n
The 4th sub-section of the higway

Figure 3.2: The Topology of a Highway

1. Σ is a finite set of sensors, each of which represents the real-time traffic of its location. As Fig. 3.2 shows, the sensors are sequentially arranged from 1 to n.
2. Υ is a finite set of roads, each road owning a state represents the real-time traffic of that road. The ith road is the sub-section between the ith sensor and the $(i+1)$th sensor, therefore $|\Upsilon| = |\Sigma| - 1$.
3. Ξ is a set of knowledge items from the historical data and the real-time data (mentioned in Section 3.2). Ξ keeps constantly updating because the real-time data turns into the historical data over time.
4. $\Delta : \Sigma \times \Xi \times \Upsilon \rightarrow (\Sigma' \times \Xi' \times \Upsilon')$ is the predictor.

Furthermore, we use four separate models to model the four highways respectively, then the four models together form the prediction model of our VTraffic system.

3.3.2 Real-Time Prediction

We collect the real-time traffic data once every five minutes, and predict the real-time traffic once every "fifteen seconds", where the prediction interval "fifteen seconds" is a system parameter.

Predicting The main idea of our prediction model is: the real-time traffic at a particular site associates with time, weather, location, the previous traffic situation, the traffic situation some distance before and so on. Based on this idea, we use the k-nearest neighbor (k-NN) algorithm [23] to get k knowledge items from Ξ. The prediction algorithm is shown in Algorithm 1, and uses k-NN twice: Step 2 uses k-NN to find a set of k-nearest knowledge items N_0 at the prediction site S based on the current weather and time, while Step 4 finds N_1 at other sites based on the current weather, time, and traffic condition at the site S. Steps 3 and 5 predict the traffic based on N_0 and N_1 respectively. Here, the confidence of the predictions $\Re_0$ and $\Re_1$ comes from the confidence of items in N_0 and N_1 respectively. Step 6 integrates the predictions $\Re_0$ and $\Re_1$ to obtain the final prediction $\Re$. Obviously, the items in N_0 and N_1 are the parents of $\Re$; on the other hand, $\Re$ is a new child of the items in N_0 and N_1. Therefore, Step 7 adds their relationships in their "parents" and "children" attributes. Finally, Step 8 adds the prediction $\Re$ into the knowledge set Ξ, and this algorithm returns the prediction $\Re$ at Step 10.

Algorithm 1 Predicting

```
Require:
    Set $K$ =user-specified; $\Theta$ =user-specified; $S = Site$;
        $\Re = NULL$; $\Xi' = \{\hbar | \hbar.location = S \& \hbar \subseteq \Xi\}$; $\overline{\Xi'} = \Xi - \Xi'$;
Ensure:
if $S \in \Sigma \& \Theta \leq 1 \& \Theta \geq 0$ then
    $N_0 \leftarrow knn(K, \Xi', Weather(S), Time(S))$;
    $\Re_0 \leftarrow predicting(N_0)$;
    $N_1 \leftarrow knn(K, \overline{\Xi'}, Weather(S), Time(S), Traffic(S))$;
    $\Re_1 \leftarrow predicting(N_1)$;
    $\Re \leftarrow Sum(\Re_0 \times \Theta, \Re_1 \times (1 - \Theta))$
    $AddRelationship(N_0 \bigcup N_1, \Re)$
    $\Xi \leftarrow \Xi \cup \{\Re\}$;
end if
return   $\Re$;
```

The traffic of the ith road is the arithmetic mean of sensor conditions at its both ends shown in Formula (3.1).

$$\Upsilon[i] = avg(\Sigma[i], \Sigma[i+1]) \tag{3.1}$$

Verification Our prediction model updates the traffic of the entire transportation system immediately when sensors collect and report the real-time traffic data. Meanwhile, the real-time data also verifies the previous predictions: if a prediction passed the verification, the knowledge items which generated the prediction will get a bonus; otherwise they will get a penalty. The bonus or penalty will affect the confidence of the corresponding knowledge items. A knowledge item will be removed from the knowledge set if its confidence is less than a certain threshold. The prediction model will stabilize after a certain time based on the assigned bonuses and penalties.

3.3.3 Predicting with Selected and Weighted Features

The original predicting algorithm is given in Algorithm 1, in order to reuse it for predicting with selected and weighted features, we simply integrate the weight information into the data instances in the preprocessing phase.

Predicting with Selected and Weighted Features consists of the following steps:

1. **Discretization** It is often difficult to compute the integral in the continuous space based on a limited number of instances, when mutual information is being estimated. Thus, those continuous features were discretized into nominal ones using the CAIM discretization algorithm [24].

2. **Normalization** As the k-NN algorithm assumes that all features are of equal importance, each feature is normalized independently.
3. **Feature Selection and Weighting** MIFS, mRMR, or RELIEF is employed for feature selection and weighting.
4. **Integrating Weights** The values of the selected features are updated by multiplying with their corresponding weight values.
5. **Prediction** The predicting algorithm given in Algorithm 1 is employed for prediction.

It is easy to notice that with Steps 3-4, it performs predicting with selected and weighted features, without them it degrades as the original predicting algorithm in Section 3.3.2.

3.3.4 Travel-Time Estimation

When a travel-time request occurs, our model uses the real-time traffic state for estimation shown in Formula (3.2).

$$travel\_time(A, B) = \sum_{r \in Roads} \frac{r.distance}{r.state.speed}, \tag{3.2}$$

where r is a road between site A and site B, $r.distance$ is the length of the road r, and $r.state.speed$ is the predicted traffic speed of the road r.

Firstly, the model identifies the starting and ending sites from the request. Secondly, the two sites are mapped into several adjacent roads on a highway. Thirdly, the travel-time of each road is calculated according to the real-time traffic of that road. Finally, the overall travel-time is the sum of all travel-time on every road.

3.4 Visualization

This project displays the traffic of the entire transportation system as geographical objects on a map, then changes the color, size, and displays meaningful markers and curves based on the real-time traffic prediction, to allow users to quickly grasp the traffic of the entire system or some particular location.

As we mentioned in our prediction model in Section 3.3.1, a highway is divided into $n - 1$ adjacent roads by n sensors. We can get the real-time traffic from the model, and then display on a map in a user-friendly form of visualization. There are two important algorithms for the visualization of our system: Initialization (shown in Algorithm 2) and Updating (Algorithm 3).

The Initialization algorithm is invoked when a user launches or refreshes the visualization web portal. Its main function is to create and initialize the essential objects, and then display them on the map. For each sensor in Σ, the algorithm creates a marker to represent the sensor. In Step 2, according to the prediction of the *model*, it chooses a color C to represent the traffic state at that

Algorithm 2 Initialization

```
Require:
    Set $A = \Sigma$; $\xi = Congestion_{Threshold}$; $Markers = \{\emptyset\}$; $Roads = \{\emptyset\}$;
Ensure:
 1: for $(I \leftarrow 0; I < A.size(); I \leftarrow I + 1)$ do
 2:   $C \leftarrow Model.predict(A[I]) \geq \xi\ ?\ COLOR_{Free} : COLOR_{Congested}$;
 3:   $Visible \leftarrow Model.predict(A[I]) \geq \xi\ ?\ False : True$;
 4:   $Markers \leftarrow Markers \cup newMarker(A[I], C, Visible)$;
 5:   if $I > 0$ then
 6:     $S \leftarrow Model.predict(A[I]) + Model.predict(A[I-1])$;
 7:     $C \leftarrow S/2 \geq \xi\ ?\ COLOR_{Free} : COLOR_{Congested}$
 8:     $Roads \leftarrow Roads \cup newRoad(A[I-1], A[I], C)$;
 9:   end if
10: end for
```

Algorithm 3 Updating

```
Require:
    Set $\xi = Congestion_{Threshold}$; $M = Markers$; $R = Roads$;
Ensure:
 1: for $(I \leftarrow 0; I < M.size(); I \leftarrow I + 1)$ do
 2:   $C \leftarrow Model.predict(M[I]) \geq \xi\ ?\ COLOR_{Free} : COLOR_{Congested}$;
 3:   if $C \neq M[I].getColor()$ then
 4:     $M[I].setColor(C)$;
 5:     $M[I].setVisible(!M[I].getVisible())$;
 6:   end if
 7:   if $I > 0$ then
 8:     $S \leftarrow Model.predict(M[I]) + Model.predict(M[I-1])$;
 9:     $C \leftarrow S/2 \geq \xi\ ?\ COLOR_{Free} : COLOR_{Congested}$;
10:     $R[I-1].setColor(C)$;
11:   end if
12: end for
```

site. In Steps 3 and 4, a new *Marker* is created, in order to make important information more eye-catching, $Visible = True$ only when the traffic at that site is congested. From Steps 5 to 9, a *Road* is created and assigned a color C according to the traffic prediction of the *model* to represent its traffic state.

The Updating algorithm is automatically invoked to update the traffic at regular intervals. Therefore, it is very important and effective to optimize this algorithm. This algorithm is very similar to Algorithm 2, and there are only two differences: a) this algorithm does not create any new object, and it just reuses the *Marks* and *Roads* created by Algorithm 2. b) this algorithm only updates those markers and roads which need to be updated. With a), it does not consume any additional memory, with b) it tries to do as little as possible. By a) and b), the Updating algorithm has been well optimized.

We have also designed a *Zoom_Change* function and a *Click_Event* monitor. The *Zoom_Change* deals with some details when the zoom size of the map is changing, and the *Click_Event* monitors the click events and shows more detailed information about a particular site where the marker is clicked.

Chapter 4

Experimental Results

4.1 Experiments

4.1.1 Experimental Setting

To validate performance fairly, the traffic data collected by the RITIS system [25] (https://www.ritis.org/) and the weather data collected by the Clarus system [13] (https://www.clarus-system.com/) were adopted in our experiments. We selected 10 different sites and downloaded 30000 traffic data instances and the corresponding weather data instances from the RITIS system and the Clarus system, respectively. Then, using the data fusion strategy mentioned in Section 3.2, we built the original dataset. The original dataset was used to build a smaller training dataset with 10000 instances and a testing dataset with 20000 instances. All instances were chosen randomly without replacement such that the two sets are disjoint.

Feast [26] is a feature selection toolbox for C and Matlab, which provides implementations of common mutual information based feature selection algorithms, and an implementation of RELIEF. Spider [27] is an object orientated environment for machine learning in Matlab, which provides implementations of NaiveBayes [28], k-NN [23], C45 [29] and RandomForest [30]. In this report, the implementations in the Feast toolbox were employed for feature selection and weighting, and the comparison prediction algorithms were the implementations in the Spider environment.

The experiments were conducted on a computer with Windows 7, 3.33 GHz dual-core CPU, and 2GB memory.

4.1.2 Experimental Results

As Algorithm 1 shows, there are two user-specified parameters k and θ in our predicting algorithm, where k is the number of neighbors for the k-NN algorithm, and θ is the weight of the result of the first k-NN algorithm, and obviously, the weight of the result of the second one is $1 - \theta$.

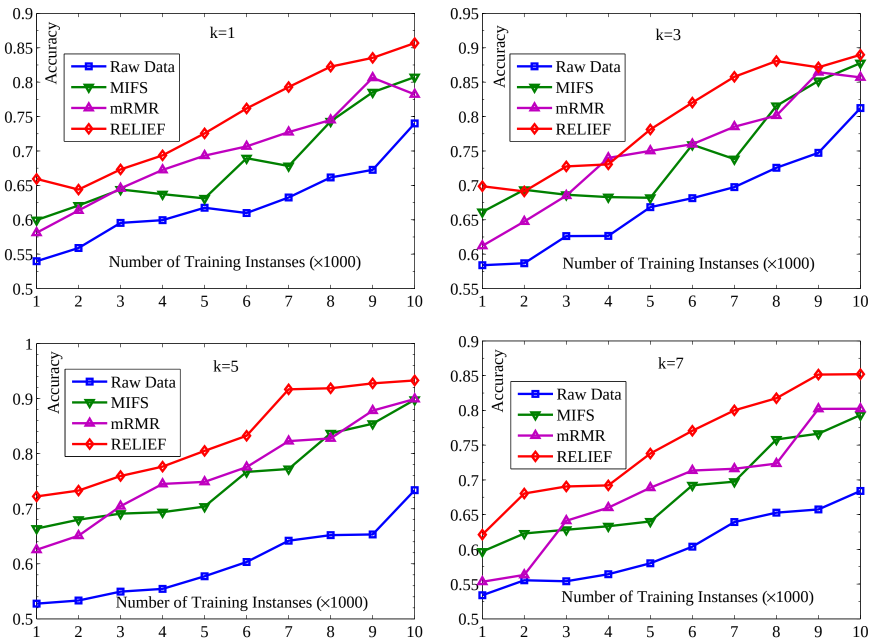


Figure 4.1: Accuracies of Our Predicting Algorithm Using Different k Values

To illustrate the performances of our predicting algorithm with different k values, we performed a set of experiments by varying the value of the parameter k from 1 to 10 for traffic prediction in the next 5 minutes. The experiments were based on the raw dataset and the datasets selected and weighted by MIFS, mRMR and RELIEF. The accuracies of the predicting algorithm using different k values are shown in Fig. 4.1. Due to space limitations, only a part of the experimental results are presented.

Several observations can be drawn from the results of Fig. 4.1. First, the accuracies of almost all k values on all datasets increase with the number of training instances. Second, the performances on the raw datasets are the worst, while the performances on the datasets selected and weighted by RELIEF are the best. This indicates feature selection and weighting can improve the performances of the predicting algorithm efficiently, in other words, the predicting with selected and weighted features algorithm in Section 3.3.3 is better than the original predicting algorithm in Section 3.3.2. Third, $k = 3$ or $k = 5$ archives better performance than others.

To illustrate the performances of our predicting algorithm with different θ values, we also performed a set of experiments by varying the value of the parameter θ from 0.1 to 1 in increments of 0.1 for traffic prediction in the next 5 minutes. Similar to the exploring of the parameter k, the raw dataset and the datasets selected and weighted by MIFS, mRMR and RELIEF were employed. The accuracies of the predicting algorithm using different θ values are shown

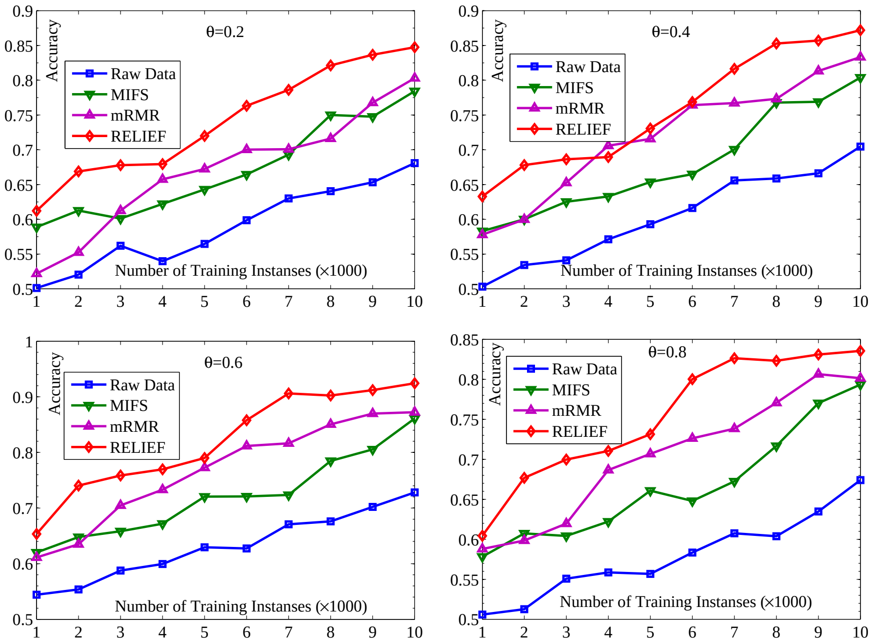


Figure 4.2: Accuracies of Our Predicting Algorithm Using Different θ Values

in Fig. 4.2. For demonstration purposes, Fig. 4.2 only outlines a part of the experimental results. From Fig. 4.2, it is easy to notice that $\theta = 0.6$ achieves the best performance. One can also observe that the accuracies increase with the number of training instances and the accuracies on the datasets selected and weighted by RELIEF are the best, which is consistent with the results of Fig. 4.1.

In our experiments, four popular classifiers, namely, NaiveBayes [28], k-NN [23], C4.5 [29], and RandomForest [30], were chosen to compare with our k-NN based predicting algorithm to predict the traffic in the next 5 minutes, and the results are shown in Fig. 4.3. It is easy to notice that the predicting algorithm, with higher accuracies on the four datasets, clearly surpasses other comparison algorithms. One can also easily observe that the accuracies increase with the number of training instances and the accuracies on the datasets selected and weighted by RELIEF are the best, which is consistent with the results of Figs. 4.1 and 4.2.

The experimental results about prediction accuracies (%) of different prediction time-lengths (from 10 to 30 minutes in increments of 5 minutes) using 5 prediction algorithms are presented in Table 4.1. In the experiments, the entire training set was employed for training. Predicting (raw) is the original predicting algorithm (Algorithm 1) using the raw dataset without feature selection and weighting. For the other prediction algorithms, the RELIEF algorithm is employed for feature selection and weighting as one can notice its performances

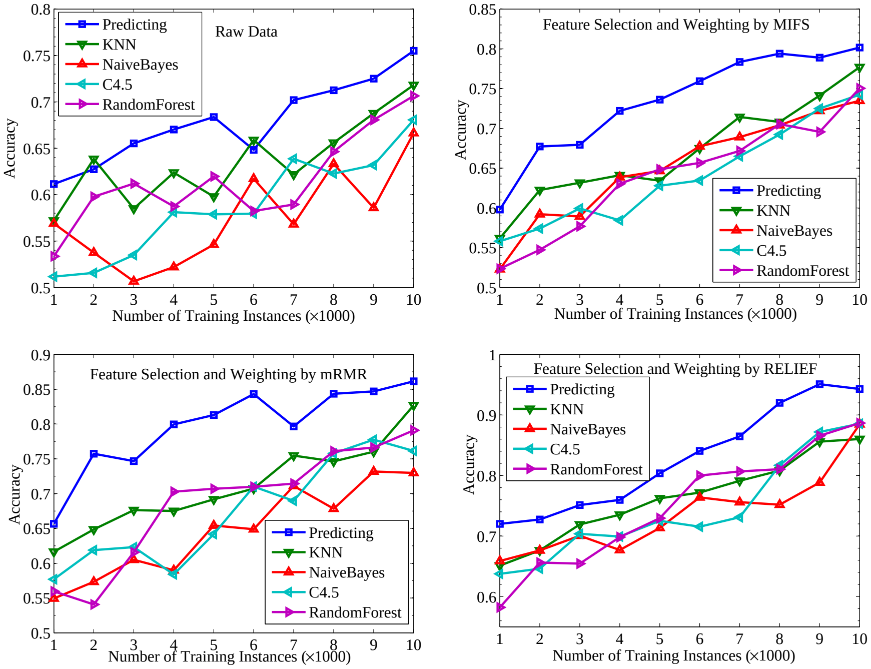


Figure 4.3: Accuracies of Different Prediction Algorithms

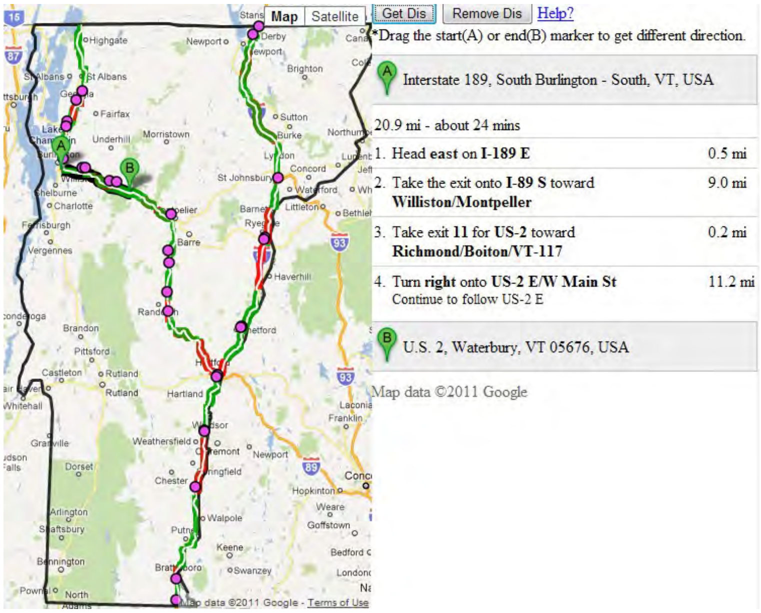
Table 4.1: A Comparison of Accuracies (%) of Different Prediction Time-Lengths

		Predicting	Predicting(raw)	k-NN	NaiveBayes	C4.5	RandomForest
Prediction Time-Length	10	**90.32**	70.16	87.91	84.19	81.52	84.02
	15	**86.52**	66.12	82.31	76.44	77.19	75.20
	20	**78.13**	60.72	**78.13**	63.18	62.11	61.09
	25	**71.29**	52.54	64.59	51.12	55.12	54.98
	30	**62.18**	46.14	55.14	40.59	47.02	42.19
WTL		**4/1/0**	0/0/5	0/1/4	0/0/5	0/0/5	0/0/5

are the best from Figs. 4.1, 4.2 and 4.3. In addition, the bold value in entries means that it is the best one among these 6 prediction algorithms. The “WTL” (win/tie/loss) represents that the number of runs where the corresponding algorithm has higher (or equal to, lower) accuracy than others.

The results in Table 4.1 show that the accuracies of the predicting algorithm are better than others in most cases. One can easily observe that the accuracies of all algorithms decrease with the growth of the prediction time-length, and the predictions of the comparison algorithms are useless when the prediction time-length is approaching to 30 minutes as their accuracies are too low.

4.2 Visualization



*Drag the start(A) or end(B) marker to get different direction.

Interstate 189, South Burlington - South, VT, USA

20.9 mi - about 24 mins

1. Head **east** on **I-189 E** 0.5 mi
2. Take the exit onto **I-89 S** toward **Williston/Montpeller** 9.0 mi
3. Take exit **11** for **US-2** toward **Richmond/Boiton/VT-117** 0.2 mi
4. Turn **right** onto **US-2 E/W Main St** 11.2 mi
   Continue to follow US-2 E

U.S. 2, Waterbury, VT 05676, USA

Map data ©2011 Google

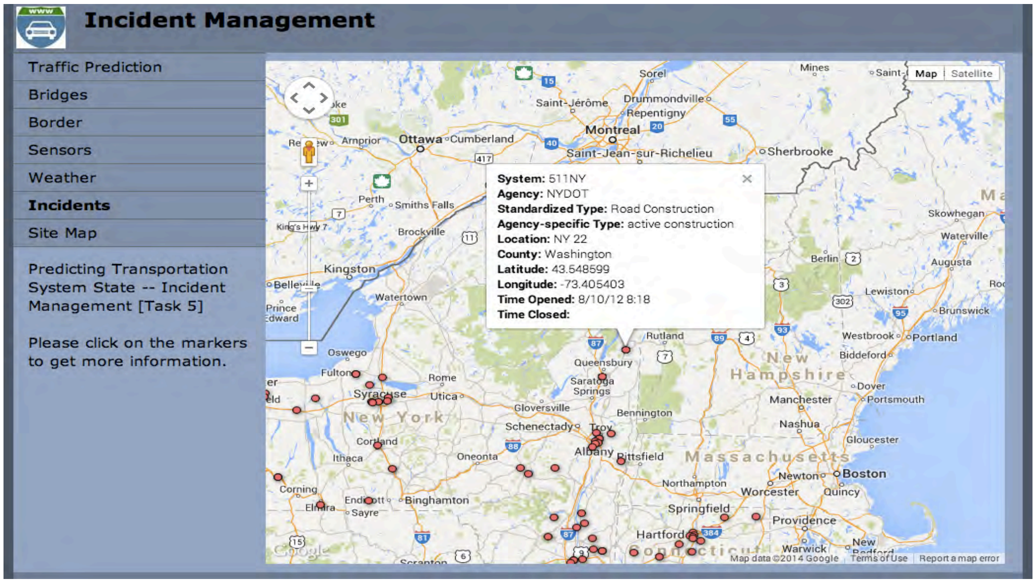
Figure 4.4: Visualization Demonstration

Fig. 4.4 is a run-time screenshot. The real-time traffic prediction is displayed over the highway as color-coded lines. The colors indicate the traffic on the road (green: free, red: congested), and the small pink circles (which come from Step 3 of Algorithm 2) indicate the traffic at a particular sensor site is congested.

By simply clicking on a pink circle, users can get an info window to show more information in detail. Users can zoom in to have a more specific vision with more details or zoom out to get a broader perspective with less details. The list on the right shows the travel information from A to B (also marked in the left map), which is estimated by our model in Section 3.3.4.

4.2.1 Incident Management

The running time screenshots of the incident management are given in Figure 4.5 and Figure 4.6. An incident is shown on the map by a red circle, and by a simple clicking on that red circle, an info window pops up to show more information in details.

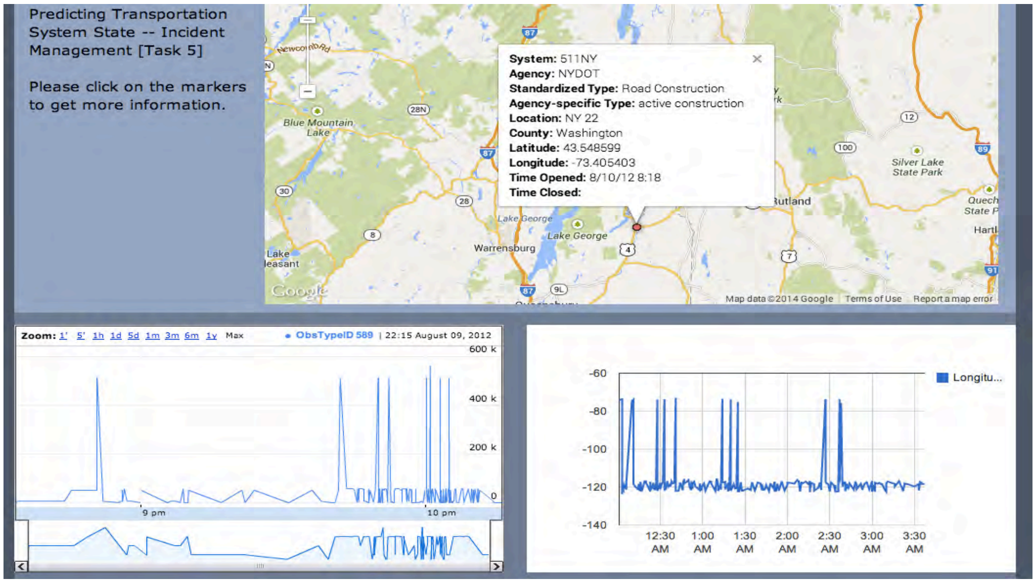


Predicting Transportation System State -- Incident Management [Task 5]

Please click on the markers to get more information.

System: 511NY
Agency: NYDOT
Standardized Type: Road Construction
Agency-specific Type: active construction
Location: NY 22
County: Washington
Latitude: 43.548599
Longitude: -73.405403
Time Opened: 8/10/12 8:18
Time Closed:

Figure 4.5: Incident Management I



Predicting Transportation System State -- Incident Management [Task 5]

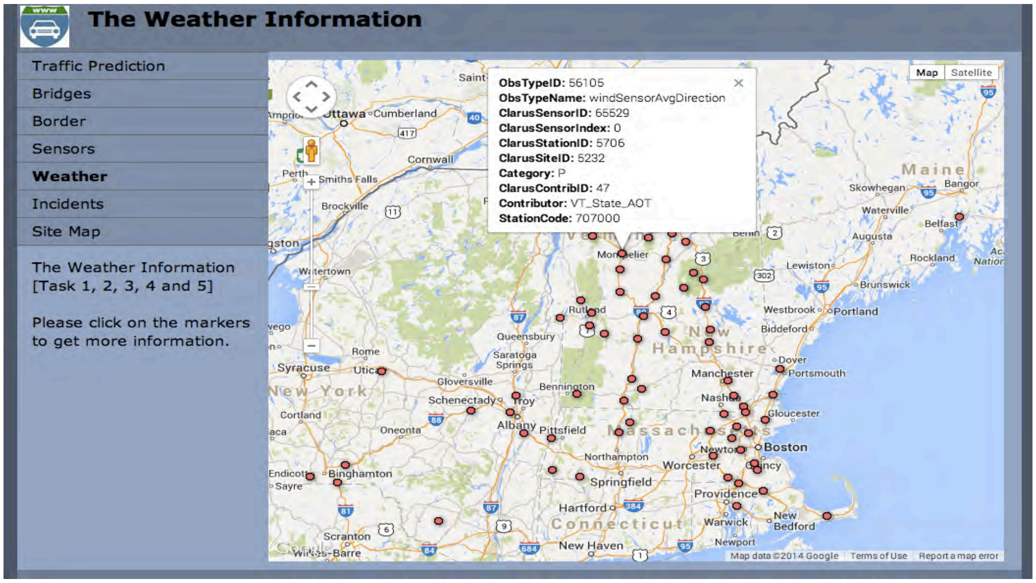
Please click on the markers to get more information.

System: 511NY
Agency: NYDOT
Standardized Type: Road Construction
Agency-specific Type: active construction
Location: NY 22
County: Washington
Latitude: 43.548599
Longitude: -73.405403
Time Opened: 8/10/12 8:18
Time Closed:

Figure 4.6: Incident Management II

4.2.2 Weather Information

The running time screenshots of the Weather Information are given in Figure 4.7 and Figure 4.8. A weather reporting location is shown on the map by a red circle, and by a simple clicking on that red circle, an info window pops up to show more information in details.



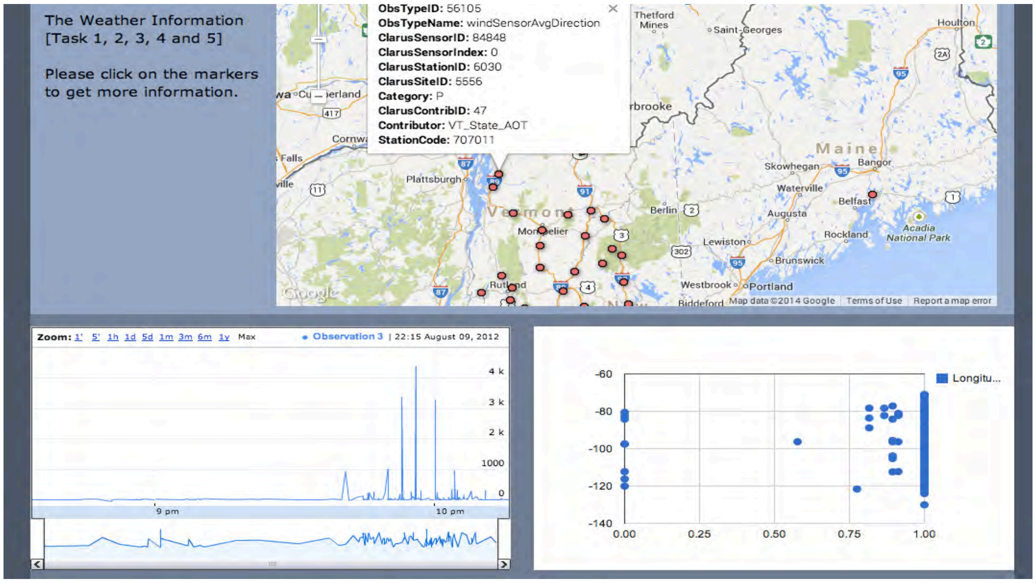
The Weather Information [Task 1, 2, 3, 4 and 5]

Please click on the markers to get more information.

ObsTypeID: 56105
ObsTypeName: windSensorAvgDirection
ClarusSensorID: 65529
ClarusSensorIndex: 0
ClarusStationID: 5706
ClarusSiteID: 5232
Category: P
ClarusContribID: 47
Contributor: VT_State_AOT
StationCode: 707000

Map data ©2014 Google Terms of Use Report a map error

Figure 4.7: Weather Information I


Zoom: 1' 5' 1h 1d 5d 1m 3m 6m 1y Max
Observation 3 | 22:15 August 09, 2012
4 k
3 k
2 k
1000
0
9 pm
10 pm
-60
-80
-100
-120
-140
0.00
0.25
0.50
0.75
1.00
Longitu...

The Weather Information [Task 1, 2, 3, 4 and 5]

Please click on the markers to get more information.

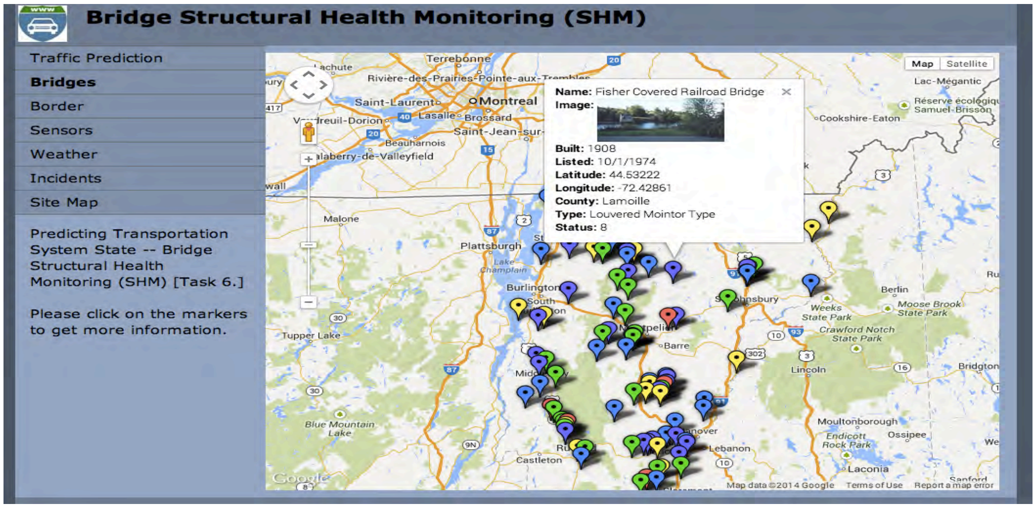
ObsTypeID: 56105
ObsTypeName: windSensorAvgDirection
ClarusSensorID: 84848
ClarusSensorIndex: 0
ClarusStationID: 6030
ClarusSiteID: 5556
Category: P
ClarusContribID: 47
Contributor: VT_State_AOT
StationCode: 707011

Map data ©2014 Google Terms of Use Report a map error

Figure 4.8: Weather Information II

4.2.3 Bridge Structural Health Monitoring

The running time screenshots of the Bridge Information are given in Figure 4.9 and Figure 4.10. A bridge is shown on the map by a marker, and by a simple clicking on that marker, an info window pops up to show more information in details.



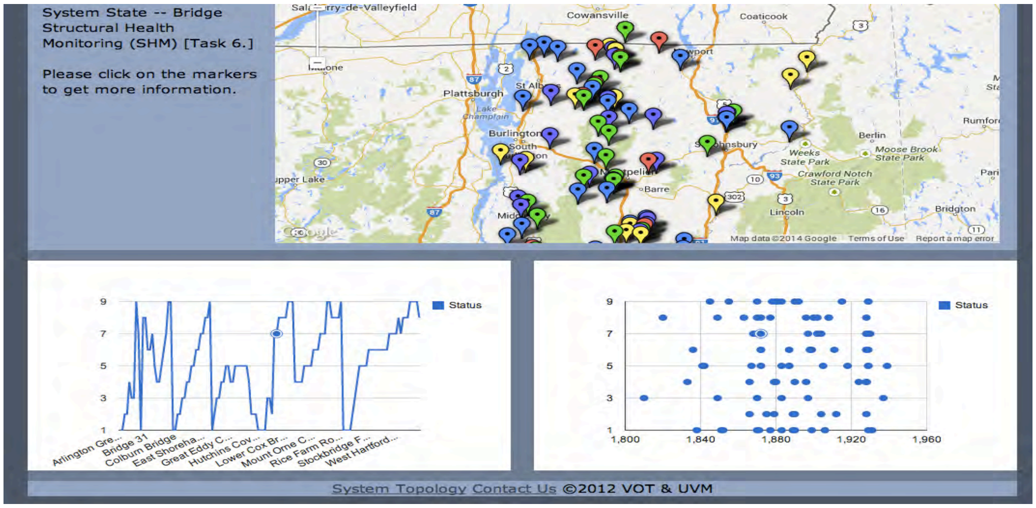
Bridge Structural Health Monitoring (SHM)

Predicting Transportation System State -- Bridge Structural Health Monitoring (SHM) [Task 6.]

Please click on the markers to get more information.

Name: Fisher Covered Railroad Bridge
Image:
Built: 1908
Listed: 10/1/1974
Latitude: 44.53222
Longitude: -72.42861
County: Lamoille
Type: Louvered Mointor Type
Status: 8

Figure 4.9: Bridge Structural Health Monitoring I

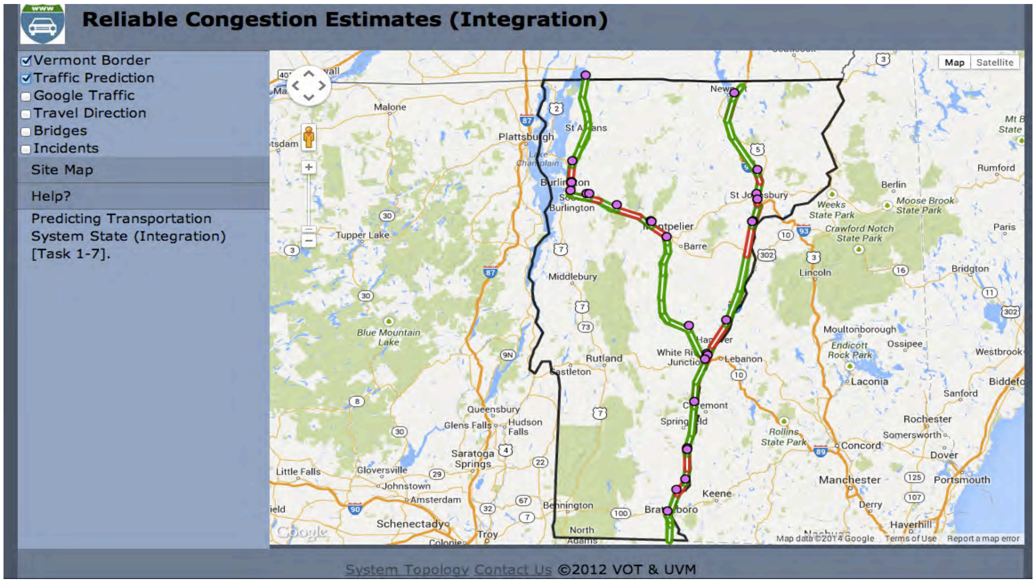


System State -- Bridge Structural Health Monitoring (SHM) [Task 6.]

Please click on the markers to get more information.

Figure 4.10: Bridge Structural Health Monitoring II

4.2.4 Traffic Prediction



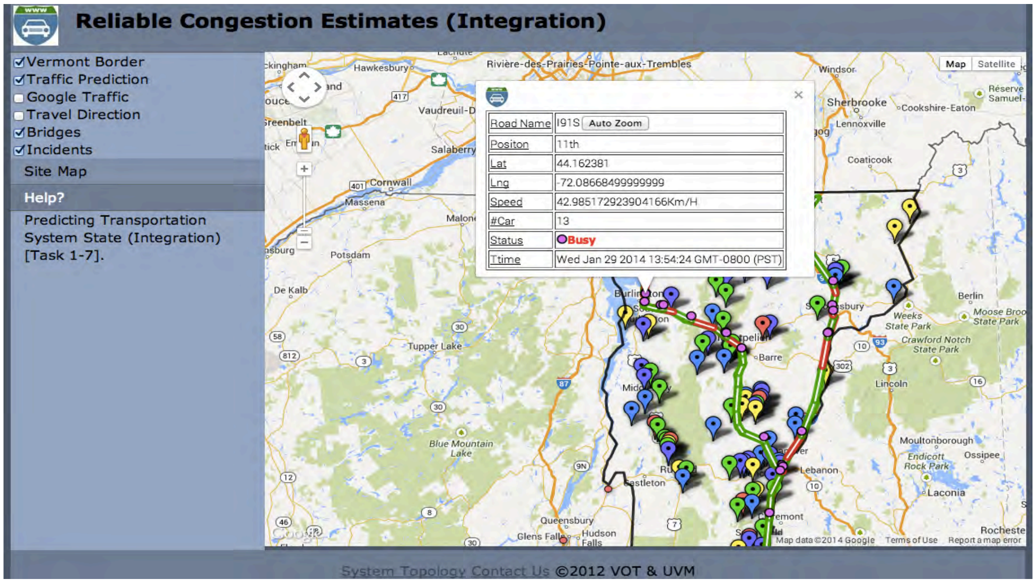
Reliable Congestion Estimates (Integration)

☑ Vermont Border
☑ Traffic Prediction
□ Google Traffic
□ Travel Direction
□ Bridges
□ Incidents

Predicting Transportation System State (Integration) [Task 1-7].

System Topology Contact Us ©2012 VOT & UVM

Figure 4.11: Traffic Prediction I



Reliable Congestion Estimates (Integration)

☑ Vermont Border
☑ Traffic Prediction
□ Google Traffic
□ Travel Direction
☑ Bridges
☑ Incidents

Predicting Transportation System State (Integration) [Task 1-7].

Road Name	I91S Auto Zoom
Positon	11th
Lat	44.162381
Lng	-72.08668499999999
Speed	42.985172923904166Km/H
#Car	13
Status	● Busy
Ttime	Wed Jan 29 2014 13:54:24 GMT-0800 (PST)

System Topology Contact Us ©2012 VOT & UVM

Figure 4.12: Traffic Prediction II

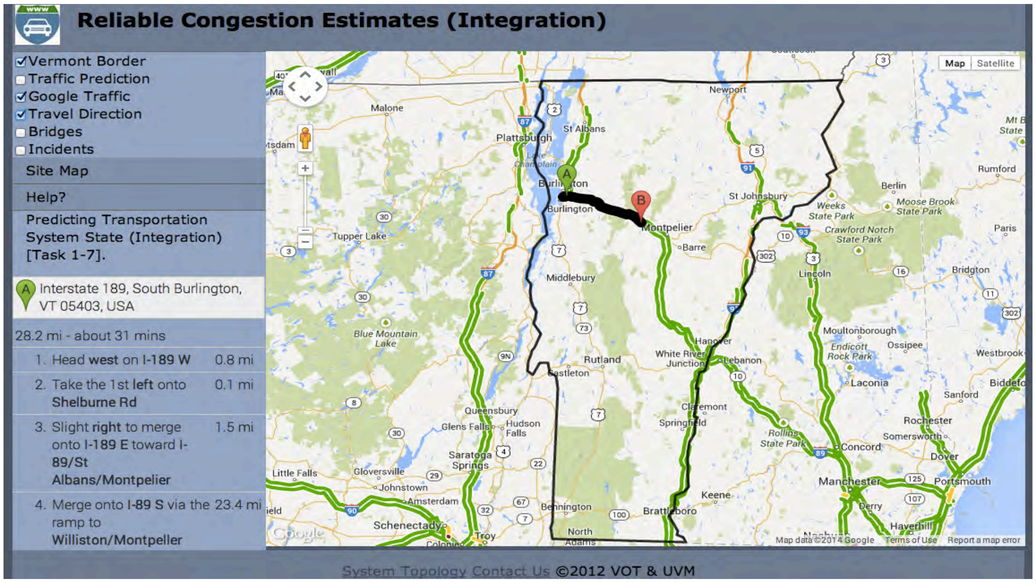
The running time screenshots of the traffic prediction are given in Figure 4.11

and Figure 4.12. The real-time traffic prediction is displayed over the highway as color-coded lines. The colors indicate the traffic on the road (green: free, red: congested), and the small pink circles indicate the traffic at a particular sensor site is congested.

By simply clicking on a pink circle, users can get an info window to show more information in detail. Users can zoom in to have a more specific vision with more details or zoom out to get a broader perspective with less details.

4.2.5 Route Planning

The running time screenshot of the route planning is given in Figure 4.13. The list on the left shows the travel information from A to B (also marked in the right map).



Reliable Congestion Estimates (Integration)

☑ Vermont Border
□ Traffic Prediction
☑ Google Traffic
☑ Travel Direction
□ Bridges
□ Incidents

Site Map

Help?

Predicting Transportation System State (Integration) [Task 1-7].

A Interstate 189, South Burlington, VT 05403, USA

28.2 mi - about 31 mins

1. Head **west** on **I-189 W** 0.8 mi
2. Take the 1st **left** onto **Shelburne Rd** 0.1 mi
3. Slight **right** to merge onto **I-189 E** toward **I-89/St Albans/Montpelier** 1.5 mi
4. Merge onto **I-89 S** via the ramp to **Williston/Montpeller** 23.4 mi

System Topology Contact Us ©2012 VOT & UVM

Figure 4.13: Route Planning

Chapter 5

Discussion & Conclusion

5.1 Discussion

In this section, we further analyze our algorithms. Firstly, as we described in Algorithm 1 in Section 3.3.2, the predicting algorithm is based on the famous k-NN algorithm. Therefore, the same as the k-NN algorithm, Algorithm 1's performance associates with the user-specified parameter k. A suitable value for parameter k can greatly improve the performance (Fig. 4.1). Meanwhile, the parameter θ determines the weight of the first k-NN algorithm's result, and as Fig. 4.2 shows, $\theta = 0.6$ archives the best performance, which indicates the result of the first k-NN algorithm should be slightly more important than the second. Furthermore, Figure 4.2 suggests that, for traffic prediction at a site, the data instances from the same site are more valuable than these from other sites.

Secondly, with selected and weighted features, the predicting algorithm can achieve a better performance. As discussed in Section 2.3, feature selection removes irrelevant, noisy and redundant features, while feature weighting identifies the important features and assigns them high weight values. Obviously, this just makes up the drawback of the k-NN algorithm, which assumes all features are of equal importance. Since the predicting algorithm is based on the k-NN algorithm, feature selection and weighting improves its performance greatly (Figs. 4.1, 4.2 and 4.3).

Thirdly, the prediction performance with selected and weighted features is better than the comparison algorithms. From Fig. 4.3 and Table 4.1, one can easily observe that the performance of the k-NN algorithm is better than the other comparison algorithms, which indicates that the k-NN algorithm is more suitable for our problem. The predicting algorithm is based on the k-NN algorithm (Algorithm 1 in Section 3.3.2), which splits the data instances into two disjoint sets according to the site where the data instance comes from, and treats them differently with the parameter θ. Therefore, it can identify valuable instances more accurately and achieves a better performance.

Overall, the proposed predicting algorithm is suitable for our traffic prediction problem. Furthermore, with selected and weighted features, the performance can be improved greatly. The prediction accuracy of the predicting with selected and weighted features algorithm is about 5 percent higher than the comparison algorithms, and it performs well for traffic prediction.

5.2 Conclusions

This project developed a suite of state of the art modeling tools that provides real-time congestion monitoring, facilitates incident management and accurate network state estimation. This is achieved by using advanced data gathering, processing and mining tools to estimate the current and future transportation system performance. In order to make a full use of all available data, we built a data fusion strategy to integrate data from heterogeneous data sources. To maximize the quality of the prediction, a dynamic prediction model was implemented. The predictions will be automatically verified by real-time data, and the model will be refined dynamically. A web portal was also built for visualization, using it users can easily observer the traffic and get useful guidance.

Chapter 6

Publications of This Project

[1] H. Li, X. Wu, Z. Li, and W. Ding. Online Group Feature Selection from Feature Stream. In: *Proceedings of the AAAI'13*, 2013.

[2] H. Li, X. Wu, Z. Li, and W. Ding. Group Feature Selection with Streaming Features. In: *Proceedings of the ICDM'13*, 2013.

[3] H. Li, Z. Li, R. T. White, and X. Wu. A Real-Time Transportation Prediction System. *International Journal of Applied Intelligence*, 2013.

[4] H. Li, Z. Li, R. T. White, and X. Wu. A Real-Time Transportation Prediction System. In: *Proceedings of the IEA-AIE'12*, 2012.

Bibliography

[1] David Schrank, Tim Lomax, and Bill Eisele. 2011 urban mobility report. Technical report, Texas Transportation Institute, September 2011.

[2] Steven I. J. Chien, Xiaobo Liu, and Kaan Ozbay. Predicting travel times for the south jersey real-time motorist information system. *Transportation Research Record: Journal of the Transportation Research Board*, 1885/2003:32–40, January 2007.

[3] Alper Aksa, Erkam Uzun, and Tansel zyer. A real time traffic simulator utilizing an adaptive fuzzy inference mechanism by tuning fuzzy parameters. *Applied Intelligence*, 36:698–720, 2012.

[4] Jia Wu, Abdeljalil Abbas Turki, and Abdellah ElMoudni. Cooperative driving: an ant colony system for autonomous intersection management. *Applied Intelligence*, 37:207–222, 2012.

[5] Eric Lu, Wangchien Lee, and Vincent Tseng. Mining fastest path from trajectories with multiple destinations in road networks. *Knowledge and Information Systems*, 29:25–53, 2011.

[6] Enrique Castillo, Mara Nogal, Jos Mara Menndez, Santos Snchez-Cambronero, and Pilar Jimnez. Stochastic demand dynamic traffic models using generalized beta-gaussian bayesian networks. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):565–581, 2012.

[7] FHWA. Fhwa operations - operations story. August 2011.

[8] Ilija Basicevic, Dragan Kukolj, and Miroslav Popovic. On the application of fuzzy-based flow control approach to high altitude platform communications. *Applied Intelligence*, 34:199–210, 2011.

[9] Tobias Warden and Ubbo Visser. Real-time spatio-temporal analysis of dynamic scenes. *Knowledge and Information Systems*, 32:243–279, 2012.

[10] Vaisala Homepage. http://www.vaisala.com/.

[11] Paul A. Pisano, Lynette C. Goodwin, and Michael A. Rossetti. U.S. highway crashes in adverse road weather conditions. In *Proceedings of the 88th*

American Meteorological Society Annual Meeting, New Orleans, Louisiana, January 2008.

[12] Paul Pisano. Clarus success stories: Using clarus data to improve operations. Technical Report FHWA-JPO-10-005, U.S. Department of Transportation Road Weather Management, 2009.

[13] Clarus System. http://www.clarus-system.com/.

[14] Huan Liu, Hiroshi Motoda, Rudy Setiono, and Zheng Zhao. Feature selection: An ever evolving frontier in data mining. *J. Mach. Learn. Res. - Proceedings Track*, 10:4–13, 2010.

[15] Hakan Altinay and Zafer Erenel. Using the absolute difference of term occurrence probabilities inbinary text categorization. *Applied Intelligence*, 36:148–160, 2012.

[16] Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5:537–550, 1994.

[17] Hanchuan Peng, Fuhui Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, August 2005.

[18] Kenji Kira and Larry A. Rendell. The feature selection problem: traditional methods and a new algorithm. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 129–134, 1992.

[19] Shuanghong Yang and Baogang Hu. Discriminative feature selection by nonparametric bayes error minimization. *IEEE Trans. on Knowl. and Data Eng.*, 24(8):1422–1434, 2012.

[20] Google Fusion Table. https://www.google.com/fusiontables/.

[21] Google Maps. http://code.google.com/apis/maps/.

[22] Jess Tan, Eric Lu, and Vincent Tseng. Preference-oriented mining techniques for location-based store search. *Knowledge and Information Systems*, pages 1–23, 2012.

[23] Vassilis Athitsos, Jonathan Alon, and Stan Sclaroff. Efficient nearest neighbor classification using a cascade of approximate similarity measures. In *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition*, pages 486–493, Washington, DC, USA, April 2005.

[24] Lukasz A. Kurgan and Krzysztof J. Cios. Caim discretization algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 16:145–153, 2004.

[25] RITIS System. https://www.ritis.org/.

[26] G. Brown, A. Pocock, M. J. Zhao, and M. Luján. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *J. Mach. Learn. Res.*, 13:27–66, 2012.

[27] Spider. http://people.kyb.tuebingen.mpg.de/spider/.

[28] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pages 338–345, CA, USA, 1995.

[29] J. R. Quinlan. Bagging, boosting, and c4.5. In *Proceedings of the 13th National Conference on Artificial Intelligence*, pages 725–730, 1996.

[30] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.