

Big Data Analytics

Predicting Traffic Flow Regimes From Simulated Connected Vehicle Messages Using Data Analytics and Machine Learning

www.its.dot.gov/index.htm

Final Report — December 2016

Publication Number: FHWA-JPO-17-498



U.S. Department of Transportation

Produced by Noblis, Inc.
U.S. Department of Transportation
ITS Joint Program Office

Notice

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof. The U.S. Government is not endorsing any manufacturers, products, or services cited herein and any trade name that may appear in the work has been included only because it is essential to the contents of the work.

Technical Report Documentation Page

1. Report No. FHWA-JPO-17-498	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Predicting Traffic Flow Regimes From Simulated Connected Vehicle Messages Using Data Analytics and Machine Learning		5. Report Date December 2016	
		6. Performing Organization Code	
7. Author(s) Meenakshy Vasudevan, Chris Curtis, Alexa Lowman, and James O'Hara		8. Performing Organization Report No.	
9. Performing Organization Name And Address Noblis, 600 Maryland Ave., SW, Suite 755, Washington, DC 20024		10. Work Unit No. (TRAIS)	
		11. Contract or Grant No. DTFH61-11-D-00018	
12. Sponsoring Agency Name and Address ITS-Joint Program Office Office of the Assistant Secretary for Research and Technology, USDOT 1200 New Jersey Avenue, S.E. Washington, DC 20590		13. Type of Report and Period Covered Final Report	
		14. Sponsoring Agency Code	
15. Supplementary Notes Work Performed for: Govind Vadakpat (FHWA R&D) and Ariel Gold (COR, ITS JPO)			
16. Abstract The key objectives of this study were to: <ol style="list-style-type: none"> 1. Develop advanced analytical techniques that make use of a dynamically configurable connected vehicle message protocol to predict traffic flow regimes in near-real time in a virtual environment and examine accuracy for various levels of market penetration 2. Examine the tradeoff between information insight and cost of data processing and management <p>Data from a virtual (simulated) testbed for the I-405 corridor in Seattle was used to conduct the study. The field data and VISSIM simulation model were obtained from WSDOT. The simulation model went through rigorous calibration and validation process as part of a separate study conducted by Noblis for FHWA Traffic Analysis Tools Program. The Trajectory Conversion Algorithm (TCA V2.3), an open source tool developed by Noblis for the USDOT, was used to emulate SAE J2735 Basic Safety Messages (BSM).</p> <p>Traffic flow regimes (free flow, speed at capacity, and congested) were predicted for 100' x 100' boxes overlaid on the I-405 traffic network, every 5 minutes an hour ahead of time using the simulated BSMS. The study made use of Apache Spark's machine learning libraries for Logistic Regression, Decision Tree and Random Forest to develop models to predict the traffic flow regimes. The computational resources and analytic environment used for this work were provisioned via the Microsoft Azure cloud environment. The computing cluster used for the analysis consisted of four nodes in total: 2 head nodes for job submission and management and 2 worker nodes for computation. Prediction accuracy was tested for two types of communication technologies (Cellular, Dedicated Short Range Communications (DSRC)), two market penetrations (20%, 75%), and six traffic operational conditions. The three algorithms were tested for 6, 8, and 11 principal components. In addition, the Decision Trees and Random Forest algorithms were tested using two node impurity metrics (entropy, Gini), and Random Forest was tested for multiple ensembles of trees (10, 250, 1000). The model that used the Random Forest algorithm with 11 principal components, 250-tree ensemble, and the Gini node impurity metric, had the best results with an average F1 score of 0.83 over all scenarios. The F1 scores were 0.87 for free flow, 0.67 for at capacity and 0.95 for congested traffic regimes. The model was able to fully process an hour's worth of BSMS into the 100' x 100' grid boxes, and make a prediction for the following hour, at 5-minute intervals for each of the 100' x 100' boxes in 6 to 16 minutes.</p>			
17. Key Words Machine learning, prediction, traffic regimes, congestion, big data, connected vehicles, Spark		18. Distribution Statement	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 103	22. Price

Table of Contents

Executive Summary	7
1 Introduction	15
1.1 BACKGROUND	15
1.2 STUDY MOTIVATION	16
1.3 STUDY OBJECTIVES.....	16
1.4 REPORT ORGANIZATION	17
2 Literature Review	18
2.1 TRAVEL TIME PREDICTIONS USING PROBE DATA	18
2.1.1 Large-Scale Estimation in Cyberphysical Systems Using Streaming Data	18
2.1.2 Real-Time Estimation of Distributed Parameters Systems.....	19
2.1.3 Learning the Dynamics of Arterial Travel from Probe Data Using a Dynamic Bayesian Network.....	20
2.1.4 Arterial Travel Time Forecast with Streaming Data	20
2.1.5 Real-Time Traffic Modeling and Estimation with Streaming Probe Data using Machine Learning.....	21
2.2 KEY FINDINGS	22
3 Hypotheses and Assumptions	23
3.1 HYPOTHESES	23
3.2 ASSUMPTIONS	23
4 Data and Computing Resources	24
4.1 DATA.....	24
4.1.1 Data Used for Analysis	24
4.1.2 Experimental Design	25
4.1.3 Emulated Connected Vehicle Messages	27
4.2 COMPUTING RESOURCES.....	28
5 Technical Approach	29
5.1 DATA PRE-PROCESSING	30
5.1.1 Overlaying Box Network on Simulated Network.....	30
5.1.2 Feature Identification	31
5.1.3 Data Assembly.....	34
5.1.4 Data Normalization.....	34
5.1.5 Creating Data Sets	34
5.2 FEATURE EXTRACTION	35
5.3 MODEL DEVELOPMENT	36
5.4 MODEL EVALUATION METRICS	38
6 Data Analysis	39
7 Results	42
7.1 PREDICTION RESULTS USING VALIDATION DATA	42
7.1.1 Prediction Results by Operational Condition Using BSM.....	42
7.1.2 Summary Prediction Results Using BSM	47

7.1.3	Summary Prediction Results Using BMM.....	49
7.2	PREDICTION RESULTS USING TEST DATA	50
7.2.1	Summary Prediction Results Using BSM	50
7.2.2	Summary Prediction Results Using BMM.....	53
7.3	TRADEOFFS BETWEEN INFORMATION INSIGHT AND COST/TIMING	54
8	Conclusions.....	57
8.1	KEY FINDINGS	57
8.2	FUTURE RESEARCH.....	58
	References	59
	APPENDIX A: Prediction Results – Validation Data.....	61
	APPENDIX B: Prediction Results – Test Data.....	89

List of Tables

Table ES-1: Traffic Regimes Modeled in the Study	9
Table ES-2: Parameters varied for study and their range of values	10
Table ES-3: Cluster Configuration	11
Table 3-1: Traffic Regimes Modeled in the Study	23
Table 4-1: Parameters varied for study and their range of values	26
Table 4-2: Cluster Configuration	28
Table 5-1: Sample Incident file data	32
Table 5-2: Sample Demand File Data: Operational Condition #1	33
Table 6-1: Number of BSMs by Traffic Regime Index – Operational Condition #1, Low Demand	39
Table 6-2: Number of BSMs by Traffic Regime Index – Operational Condition #2, Low Visibility	39
Table 6-3: Number of BSMS by Traffic Regime Index – Operational Condition #3, Weather + Incidents	40
Table 6-4: Number of BSMs by Traffic Regime Index – Operational Condition #4, Many Incidents	40
Table 6-5: Number of BSMs by Traffic Regime Index – Operational Condition #5, Bottleneck Trouble	40
Table 6-6: Number of BSMs by Traffic Regime Index – Operational Condition #6, Few Incidents	41
Table 7-1: Comparing F1 Scores of Prediction Models by Communication Mode and Market Penetration for Validation Data	46
Table 7-2: Prediction Results Using Random Forest for Validation Data Comprising Basic Mobility Messages (BMM) for Cellular-20% Scenario	50
Table 7-3: Average Prediction Results Using Random Forest for Test Data Comprising BSM Across All Scenarios	53
Table 7-4: Prediction Results Using Random Forest for Test Data Comprising Basic Mobility Messages (BMM) for Cellular-20% Scenario	54

List of Figures

Figure ES-1: I-405 Geographic Network (Source: Google Maps/FHWA).....	10
Figure ES-2: Framework for Traffic Regime Prediction Using Simulated Connected Vehicle Messages	13
Figure 4-1: I-405 Geographic Network (Source: Google Maps/FHWA).....	25
Figure 4-2: Locations of Roadside Equipment at Major Interchanges on I-405.....	27
Figure 5-1: Framework for Traffic Regime Prediction Using Simulated Connected Vehicle Messages	30
Figure 5-2: An Example Illustration of the Next Box Algorithm for a Curved Road	31
Figure 5-3: Example Incident	32
Figure 5-4: VISSIM Zones on the I-405 Network.....	33
Figure 5-5: Scree Plot generated by R using the nFactor package on Many Incidents, 20% Cellular.....	36
Figure 7-1: Trellis plot of F1 scores for each prediction model by operational condition, traffic regime index and PCA k-value for DSRC-20%	43
Figure 7-2: Trellis plot of F1 scores for each prediction model by operational condition, traffic regime index and PCA k-value for DSRC-75%	44
Figure 7-3: Trellis plot of F1 scores for each prediction model by operational condition, traffic regime index and PCA k-value for Cellular-20%	45
Figure 7-4: Trellis plot of F1 scores for each prediction model by operational condition, traffic regime index and PCA k-value for Cellular-75%	46
Figure 7-5: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined DSRC-20% validation data using 11 principal components.....	48
Figure 7-6: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined DSRC-75% validation data using 11 principal components.....	48
Figure 7-7: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined Cellular-20% validation data using 11 principal components.....	49
Figure 7-8: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined Cellular-75% validation data using 11 principal components.....	49
Figure 7-9: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined DSRC-20% test data using 11 principal components	51
Figure 7-10: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined DSRC-75% test data using 11 principal components	52
Figure 7-11: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined Cellular-20% test data using 11 principal components	52
Figure 7-12: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined Cellular-75% test data using 11 principal components	53
Figure 7-13: Comparing Time taken for Model Training and Prediction.....	55

Executive Summary

BACKGROUND

The United States Department of Transportation (USDOT) connected vehicle research program has the potential to transform surface transportation system performance [1]. In a connected vehicle environment, wireless sub-second data exchange connects vehicles, the infrastructure, and travelers' mobile devices. These data have the promise to transform the geographic scope, precision, and latency of transportation system control, thereby resulting in significant safety, mobility, and environmental benefits. These vast amounts of data can help transportation system managers get a comprehensive and accurate view of their systems, understand the causality of transportation problems (e.g., crashes, bottlenecks, delays), and improve the accuracy and latency of decision-making, thereby facilitating proactive management of the transportation system. However, the new data influx also has the potential to over-burden legacy computational and communication systems. Although connected vehicle technology can facilitate ubiquitous system coverage, existing prediction methods, computational platforms, and data management methods are insufficient to process the data within a reasonable timeframe for real-time predictions. With increased market adoption of connected vehicle technology, this data explosion is imminent, thereby necessitating big data solutions to fully exploit connected vehicle data for transformational improvements to the transportation system operations and management.

The focus of this study is to develop and test analytic tools that can handle data that is of such volume, velocity, and variety that it cannot be processed or managed using traditional tools (e.g., relational database management systems), and requires technologies that support *big data*. What is big data? The most commonly accepted definition is Gartner's definition – "Big data is high volume, velocity and variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making" [2]. Additional qualifiers, such as veracity and value, are sometimes added.

The expectation is that connected vehicle data can be processed rapidly using advanced ("big data") analytics and high performance computing to create precise predictions of congestion, prior to the deterioration of roadway conditions. Consequently, public agency staff will be able to improve travel on their roadways by assessing the predicted congestion levels and undertaking suitable congestion mitigating actions. To examine if this expectation was meaningful, an initial effort (Phase 1) was jointly funded in late 2013 by USDOT's Data Capture and Management Program and Noblis internal research funds given the technical risk and the uncertain value of findings in this exploratory research. This initial effort explored the use of graph analytics and high performance computing (HPC) in predicting congestion using SAE J2735 Basic Safety Messages (BSM; [3]). The study resulted in a framework that was able to predict congestion in 100 feet (30.5 m) segments at one-minute intervals over a time window of 1 hour, 30-60 minutes in advance of actual congestion [4]. Despite sparseness of data (data represented only 2% of the vehicle population), the proposed framework predicted highly congested locations 40% of the time. Severity of congestion was predicted with an accuracy of 77%.

STUDY MOTIVATION

The initial study (Phase 1) was a first step in determining the value of collecting BSMS comprehensively. BSMS and the prediction methodology afforded the capability to predict in real time transition of traffic flow from an uncongested state to a highly congested state even on arterials where traffic detectors are typically not deployed. However, at this stage it is unknown if BSMS need to be collected from every connected vehicle at intervals of a 10th of a second. As the amount of data increases, the accuracy might increase while the computational efficiency might decrease. Research is needed to examine the tradeoff between information insight and cost of data processing and management. Thus the motivation of this study was to examine the following two key questions:

1. Should BSMS be collected from every connected vehicle in the system at intervals of a 10th of a second to predict traffic flow regimes?
2. What is the tradeoff between information insight and cost of data processing and management?

A secondary motivation for this study was to provide a practical example of how connected vehicle (CV) data can improve transportation operations with the intent of motivating others to investigate potential applied uses of CV data. Data used for this study along with data from a number of other open CV data sets can be found on the USDOT's Research Data Exchange (RDE; <https://www.its-rde.net>) and other sources. The resulting code and documentation from this study is non-proprietary and will be posted on the USDOT's Open Source Application Development Portal (OSADP; itsforge.net) alongside many other existing Open Source CV applications. Researchers and application developers are encouraged to use the code and data to further research and development in this area, and share their results via sites such as the RDE and the OSADP.

STUDY OBJECTIVES

The key objectives of the study are to:

1. Develop advanced analytical techniques that make use of a dynamically configurable connected vehicle message protocol to predict transition of traffic flow regimes in near-real time in a virtual environment and examine accuracy for various levels of market penetration
2. Examine the tradeoff between information insight and cost of data processing and management

Data from a virtual (simulated) testbed will be used to conduct the study. The Trajectory Conversion Algorithm (TCA V2.3), an open source tool developed by Noblis for the USDOT, will be used to emulate SAE J2735 Basic Safety Messages (BSM) for the virtual testbed [5]. The TCA V2.4, also an open source tool, will be used to emulate the prototype Dynamic Interrogative Data Capture (DIDC) controller in order to model a dynamically configurable connected vehicle message protocol (i.e., the Basic Mobility Messages) [6]. Note that the Basic Mobility Message (BMM) is not a published standard along the lines of the BSM but rather a concept that is being researched by multiple groups, including but not limited to the USDOT. The BMM is an event-driven, configurable message set.

A virtual testbed was used since at this stage, connected vehicle technology has not been deployed on a large scale making it difficult to assess Objective 2. Secondly, dynamically configurable connected vehicle message protocols are still under development, and hence Objective 1 cannot be accomplished without making use of a simulated testbed. Thus, this study suffers from similar limitations as any study that makes use of data that represents reality but is not reality. However, it

should be noted that the simulation models that were used in the study went through rigorous calibration and validation process, as part of a separate study conducted by Noblis for the Federal Highway Administration (FHWA) Traffic Analysis Tools Program.

HYPOTHESES

The following hypotheses will be tested in the study.

- Hypothesis #1: Proposed approach that makes use of high-volume connected vehicle data, advanced analytics, and cloud computing will meet computational speed requirements for a real-time decision support system
- Hypothesis #2: Proposed approach will be able to predict traffic flow regimes with high temporal and geographic precision and accuracy for higher market penetration of messages

ASSUMPTIONS

- Equipment/device failures are not part of the assessment
- Traffic regimes/indices are identified based on the work by researchers at Virginia Tech [12] and precipitation and visibility levels observed in the data used for the study. Table ES-1 shows the traffic regimes that were modeled in the study.

Table ES-1: Traffic Regimes Modeled in the Study

Traffic Regime	Clear to Light Rain ($\leq 0.1'$), Good to Medium Visibility (> 5 mi)	Clear to Light Rain ($\leq 0.1'$), Low Visibility (≤ 5 mi)	Index
Free Flow Regime	Speed $> 85\%$ of Free Flow Speed	Speed $> 74\%$ of Free Flow Speed	1
Speed at Capacity Regime	75% of FF Speed $<$ Speed $\leq 85\%$ of FF Speed	65% of FF Speed $<$ Speed $\leq 74\%$ of FF Speed	2
Congested Regime	Speed $\leq 75\%$ of Free Flow Speed	Speed $\leq 65\%$ of Free Flow Speed	3

DATA

Data Used for Analysis

A full-scale deployment of connected vehicle technology has not yet occurred. Hence, a representative large scale data set that was created using a traffic simulation tool was used for the analysis. The simulated data set was generated as part of a separate project that Noblis conducted for FHWA's Traffic Analysis Tools (TAT) Program [13]. The geographic network (Figure ES-1) used for the analysis in this study was the I-405 Corridor, a 29.5 mile long major commuter corridor in the Seattle area that is subject to periods of high travel demand and congestion. The I-405 corridor experiences significant travel time variability as a result of dynamic incident patterns and frequent rain and fog. The Washington State Department of Transportation (WSDOT) provided FHWA and Noblis, traffic, travel

time, incident, and weather data for 2012. After removing weekends and holidays, there were 196 weekdays left that were clustered into six operational conditions for the study – low demand, low visibility, weather and incidents, many incidents, bottleneck trouble, and few incidents – and a representative day was selected for each cluster [13]. WSDOT also provided a VISSIM model of the I-405 network as part of the TAT project. Using this VISSIM network as an initial base model, Noblis calibrated VISSIM models for each of the six operational conditions [13].

The simulated vehicle trajectories from the six calibrated VISSIM models were made available to us for use in this project. These trajectories were used as input to the Trajectory Converter Analysis (TCA) to emulate connected vehicle Basic Safety Messages (BSM).

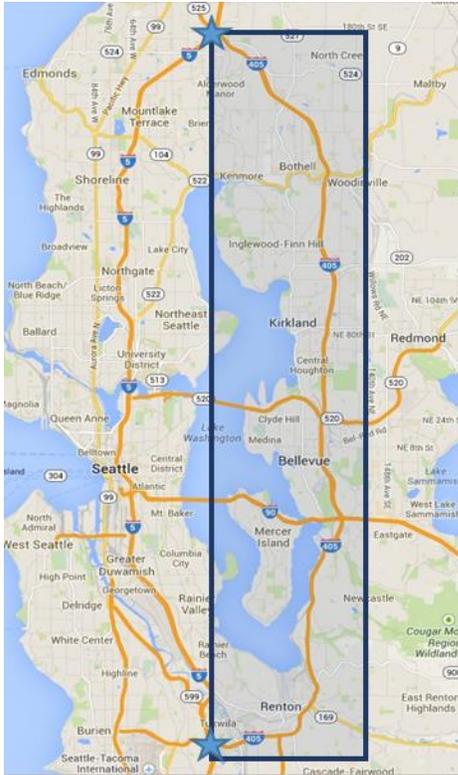


Figure ES-1: I-405 Geographic Network (Source: Google Maps/FHWA)

Experimental Design

Accuracy of the prediction model is examined by varying the market penetration, operational conditions, and communications strategies (Table ES-2).

Table ES-2: Parameters varied for study and their range of values

Parameters	Range
Study Period	AM Peak (5:30-10:30 AM)
Operational Conditions	1. Low Demand 2. Low Visibility 3. Weather + Incidents

Parameters	Range
	4. Many Incidents 5. Bottleneck Trouble 6. Few Incidents
Market Penetration of Connected Vehicles	20%, 75%
Communication Strategies	Cellular, DSRC with RSEs deployed at major interchanges (see Figure 4-2)

Basic Safety Messages were examined for all combinations of these variables, Basic Mobility Messages were examined for all Operational Conditions using just a Cellular communication strategy and 20% market penetration.

COMPUTING RESOURCES

The computational resources and analytic environment used for this work were provisioned via the Microsoft Azure cloud environment [15], and consisted of two primary elements: file storage and computing cluster.

File storage was provided using the Azure Blob Storage service, which is designed to store unstructured data in any format as objects or blobs. The Blob Storage service provides easy access to extremely large storage space up to 500TB per instance, with built-in redundancy and data protection, without requiring management of or even visibility into individual disks or volumes. This allowed simulated data files on the order of 150GB to be created and accessed routinely. The Azure Blob Storage service is also provisioned as network-local to Azure compute resources; so these data files are readily accessible with no WAN bandwidth limitations.

Compute resources were provided via the Azure HDInsight service [16], which is a cloud-native distribution of the open-source Apache Hadoop framework [17]. The HDInsight service provides template-based provisioning to allow clusters to be spun up and down on demand, without requiring manual configuration of either cluster machines or software packages. The template used for this work included the open-source Apache Spark data processing package, along with its MLlib machine learning library [18]. Code for these analyses was developed using both the Scala and Python (PySpark) languages.

The cluster used for these analyses consisted of four nodes in total: 2 head nodes for job submission and management and 2 worker nodes for computation. Specific provisioned configurations of these nodes are given in Table ES-3.

Table ES-3: Cluster Configuration

Cluster Nodes	CPU Cores	Memory (RAM)	Local Disk (HDFS)
Head nodes (2 each)	4	28 GB	200 GB
Worker nodes (2 each)	16	112 GB	800 GB

TECHNICAL APPROACH

This section describes the technical approach to predict traffic regimes (see Table ES-1) in 100' x 100' boxes overlaid on the I-405 traffic network, every 5 minutes an hour ahead of time using simulated BSMs. As the key purpose of the prediction algorithm is to classify data into three traffic regimes, this is formulated as a classification problem.

Figure ES-2 is a graphical illustration of the traffic regime prediction framework:

- *Field Data*: In this stage, field data, including demand, incident, and weather data, were obtained for the I-405 corridor from WSDOT and processed (see Section 4.1.1).
- *Simulated Data*: In the second stage, the processed field data were used as input to a VISSIM model for the I-405 network, also obtained from WSDOT, and calibrated (see Section 4.1.1). Simulated vehicle trajectories were fed into the TCA to emulate SAE J2735 BSMs (Section 4.1.3). BSMs were generated every 10th of a second, and transmitted either via DSRC-enabled RSEs or cellular networks, as defined in Table 4-1 (see Section 4.1.2). BSMs were generated and transmitted according to the DIDC Parameters as defined in Section 4.1.2.5.
- *Data Pre-Processing*: The third stage includes feature identification, data assembly, normalization, and data set creation. Speed, demand, incident, and weather were identified as the key features (Section 5.1.2). Next, a grid network of 100' x 100' boxes was overlaid on the I-405 network. For each box, the following data were assembled for the past 1 hour: average speeds at 5-minute intervals in the subject box, and two boxes upstream and downstream of the subject box; northbound and southbound I-405 demand over the past 1 hour; incident information over the past 1 hour; and precipitation levels and visibility over the past 1 hour. Once this was done, data were normalized. Data that are measured on different numeric scales are normalized or converted to a common scale so that no single feature dominates the others. Then we divided the data into 3 sets – training, validation, and test data sets. Training and validation data sets were used for model development and to prevent overfitting, and the test set was used to report out the accuracy of the prediction models.
- *Feature Extraction*: In this stage the most relevant information is extracted from the original set of features and represented in a lower dimensionality space. The previous stage resulted in 69 raw features, some potentially highly correlated. Highly correlated features effectively represent the same phenomenon, causing an overrepresentation of that phenomenon and possibly leading to poor generalization. Hence, in this stage, the 69 possibly correlated features were transformed into uncorrelated variables using Principal Component Analysis (PCA). The number of required principal components was identified by performing a scree plot analysis in R using the nFactor package [19] on the Many Incidents, 20% Cellular data. The plot showed how much each principal component contributed to the overall variance, with each point representing a principal component and its corresponding eigenvalue. The scree plot helped choose the number of principal components to use based on variance contributions. In this study 6, 8 and 11 principal components were tested as these were the three points with noticeable drops in contribution to variance.
- *Model Development and Selection*: In this stage, Spark's machine learning libraries for Logistic Regression, Decision Tree and Random Forest were used for predicting the traffic regimes. Traffic regime indices were predicted at 5-minute intervals, an hour in advance, in each of the 100' x 100' boxes on the network. For example, if the current time is 7 AM, predictions were made for 8 to 8:05 AM. The process was repeated every five minutes. The Decision Trees and Random Forest algorithms were tested using two node impurity metrics (entropy, Gini). In addition, Random Forest was tested for multiple ensembles of trees (10,

250, 1000 trees). The best models were selected based on the accuracy seen for the training and validation sets.

- *Model Evaluation:* In the final stage, the predictions models were evaluated using the test set. The selected evaluation metrics include: precision, recall and F1 score.

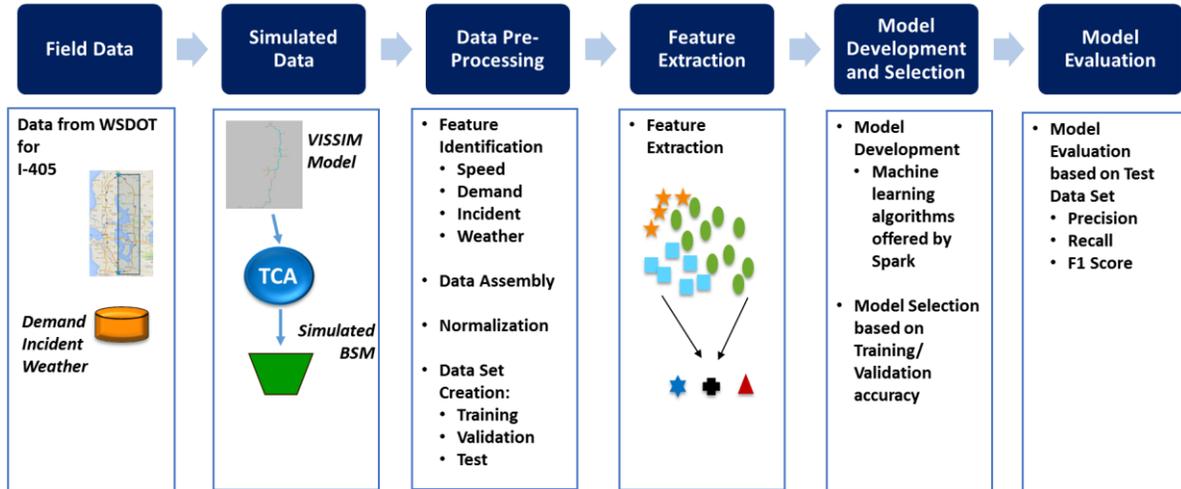


Figure ES-2: Framework for Traffic Regime Prediction Using Simulated Connected Vehicle Messages

KEY FINDINGS

The study validated the two hypotheses.

- *Hypothesis #1: Proposed approach that makes use of high-volume connected vehicle data, advanced analytics, and cloud computing will meet computational speed requirements for a real-time decision support system*
 - The best performing model (Random Forest with 250-tree ensemble) was able to fully process an hour's worth of BSMs into the 100' x 100' grid boxes, calculate the average speed for each box, direction and 5-minute interval, find average speeds in upstream and downstream boxes, join environmental features, perform normalization and Principal Components Analysis, and make a prediction for the following hour, at 5-minute intervals for each of the 100' x 100' boxes in 6 to 16 minutes.
- *Hypothesis #2: Proposed approach will be able to predict traffic flow regimes with high temporal and geographic precision and accuracy for higher market penetration of messages*
 - Across all experimental scenarios, the model that used the Random Forest algorithm with 11 principal components, 250-tree ensemble and the Gini node impurity metric, had the best results with an average F1 score of 0.83. The F1 scores were 0.87 for free flow, 0.67 for at capacity and 0.95 for congested traffic regimes. Predictions were made for 100' x 100' boxes nearly an hour in advance at 5-minute intervals.
 - More data (i.e., increase in market penetration) didn't necessarily translate into better predictions; however, more *representative* data did produce higher F1 scores as is evidenced by the higher F1 scores for free flow and congested regimes than for the speed at capacity regime.

CONCLUSIONS

In a connected vehicle environment, wireless sub-second data exchange connects vehicles, the infrastructure, and travelers' mobile devices. These data have the promise to transform the geographic scope, precision, and latency of transportation system control, thereby resulting in significant safety, mobility, and environmental benefits. However, the new data influx also has the potential to over-burden legacy computational and communication systems. Although connected vehicle technology can facilitate ubiquitous system coverage, existing prediction methods, computational platforms, and data management methods are insufficient to process the data within a reasonable timeframe for real-time predictions. With increased market adoption of connected vehicle technology, this data explosion is imminent, thereby necessitating big data solutions to fully exploit connected vehicle data for transformational improvements to the transportation system operations and management.

The focus of this analysis was to develop and test analytic tools that can handle data that is of such volume, velocity, and variety that it cannot be processed or managed using traditional tools (e.g., relational database management systems), and requires technologies that support big data.

The study presented a technical approach that combined Apache Spark's open source data analytics and machine learning techniques to predict traffic flow regimes using simulated connected vehicle messages. The computational resources and analytic environment used for this work were provisioned via the Microsoft Azure cloud environment. Predictions were made for the following hour at 5-minute intervals for 100' x 100' boxes in less than 20 minutes. The study demonstrated that connected vehicle data can be processed rapidly using advanced ("big data") analytics and high performance computing to create precise predictions of traffic flow regimes, prior to the deterioration of roadway conditions. Public agency staff will be able to improve travel within these corridors by assessing the predicted congestion levels and undertaking suitable congestion mitigating actions.

Future Research

The study showed that the model that used the Random Forest algorithm was the best overall, with an average F1 score of 0.83. While the overall score is good, for the at capacity regime it was only 0.67. Overall, at capacity BSMs were about 22% of the total BSMs – which is approximately half of what was generated for the other two regimes. This shows that the data was imbalanced. In our study, due to schedule and budget constraints, we examined the use of TCA-DIDC to oversample speed at capacity regimes using a single set of parameters. There are numerous DIDC parameters as well as ranges of possible optimal values for those parameters that can be set using a DIDC Controller. Thus there is potential for substantial improvement on prediction accuracy through the use of either different control parameters, different control values or both. Future research should focus on developing prediction models by either undersampling the majority classes (i.e., free flow and congested traffic regimes) or oversampling the minority class (i.e., speed at capacity regime) using TCA-DIDC and/or statistical techniques.

Another potential research could focus on predicting other traffic phenomena, such as queue lengths at signalized intersections, conflicts, etc.

1 Introduction

1.1 Background

The United States Department of Transportation (USDOT) connected vehicle research program has the potential to transform surface transportation system performance [1]. In a connected vehicle environment, wireless sub-second data exchange connects vehicles, the infrastructure, and travelers' mobile devices. These data have the promise to transform the geographic scope, precision, and latency of transportation system control, thereby resulting in significant safety, mobility, and environmental benefits. These vast amounts of data can help transportation system managers get a comprehensive and accurate view of their systems, understand the causality of transportation problems (e.g., crashes, bottlenecks, delays), and improve the accuracy and latency of decision-making, thereby facilitating proactive management of the transportation system. However, the new data influx also has the potential to over-burden legacy computational and communication systems. Although connected vehicle technology can facilitate ubiquitous system coverage, existing prediction methods, computational platforms, and data management methods are insufficient to process the data within a reasonable timeframe for real-time predictions. With increased market adoption of connected vehicle technology, this data explosion is imminent, thereby necessitating big data solutions to fully exploit connected vehicle data for transformational improvements to the transportation system operations and management.

The focus of this study is to develop and test analytic tools that can handle data that is of such volume, velocity, and variety that it cannot be processed or managed using traditional tools (e.g., relational database management systems), and requires technologies that support *big data*. What is big data? The most commonly accepted definition is Gartner's definition – "Big data is high volume, velocity and variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making" [2]. Additional qualifiers, such as veracity and value, are sometimes added.

The expectation is that connected vehicle data can be processed rapidly using advanced ("big data") analytics and high performance computing to create precise predictions of congestion, prior to the deterioration of roadway conditions. Consequently, public agency staff will be able to improve travel on their roadways by assessing the predicted congestion levels and undertaking suitable congestion mitigating actions. To examine if this expectation was meaningful, an initial effort (Phase 1) was jointly funded in late 2013 by USDOT's Data Capture and Management Program and Noblis internal research funds given the technical risk and the uncertain value of findings in this exploratory research. This initial effort explored the use of graph analytics and high performance computing (HPC) in predicting congestion using SAE J2735 Basic Safety Messages (BSM; [3]). The study resulted in a framework that was able to predict congestion in 100 feet (30.5 m) segments at one-minute intervals over a time window of 1 hour, 30-60 minutes in advance of actual congestion [4]. Despite sparseness of data (data represented only 2% of the vehicle population), the proposed framework predicted highly congested locations 40% of the time. Severity of congestion was predicted with an accuracy of 77%.

1.2 Study Motivation

The initial study (Phase 1) was a first step in determining the value of collecting BSMs comprehensively. BSMs and the prediction methodology afforded the capability to predict in real time transition of traffic flow from an uncongested state to a highly congested state even on arterials where traffic detectors are typically not deployed. However, at this stage it is unknown if BSMs need to be collected from every connected vehicle at intervals of a 10th of a second. As the amount of data increases, the accuracy might increase while the computational efficiency might decrease. Research is needed to examine the tradeoff between information insight and cost of data processing and management. Thus the motivation of this study was to examine the following two key questions:

1. Should BSMs be collected from every connected vehicle in the system at intervals of a 10th of a second to predict traffic flow regimes?
2. What is the tradeoff between information insight and cost of data processing and management?

A secondary motivation for this study was to provide a practical example of how connected vehicle (CV) data can improve transportation operations with the intent of motivating others to investigate potential applied uses of CV data. Data used for this study along with data from a number of other open CV data sets can be found on the USDOT's Research Data Exchange (RDE; <https://www.its-rde.net>) and other sources. The resulting code and documentation from this study is non-proprietary and will be posted on the USDOT's Open Source Application Development Portal (OSADP; itsforge.net) alongside many other existing Open Source CV applications. Researchers and application developers are encouraged to use the code and data to further research and development in this area, and share their results via sites such as the RDE and the OSADP.

1.3 Study Objectives

The key objectives of the study are to:

1. Develop advanced analytical techniques that make use of a dynamically configurable connected vehicle message protocol to predict traffic flow regimes in near-real time in a virtual environment and examine accuracy for various levels of market penetration
2. Examine the tradeoff between information insight and cost of data processing and management

Data from a virtual (simulated) testbed will be used to conduct the study. The Trajectory Conversion Algorithm (TCA V2.3), an open source tool developed by Noblis for the USDOT, will be used to emulate SAE J2735 Basic Safety Messages for the virtual testbed [5]. The TCA V2.4, also an open source tool, will be used to emulate the prototype Dynamic Interrogative Data Capture (DIDC) controller in order to model a dynamically configurable connected vehicle message protocol (i.e., the Basic Mobility Messages) [6]. Note that the Basic Mobility Message (BMM) is not a published standard along the lines of the BSM but rather a concept that is being researched by multiple groups, including but not limited to the USDOT. The BMM is an event-driven, configurable message set.

A virtual testbed was used since at this stage, connected vehicle technology has not been deployed on a large scale making it difficult to assess Objective 2. Secondly, dynamically configurable connected vehicle message protocols are still under development, and hence Objective 1 cannot be accomplished without making use of a simulated testbed. Thus, this study suffers from similar limitations as any study that makes use of data that represents reality but is not reality. However, it

should be noted that the simulation models that were used in the study went through rigorous calibration and validation process, as part of a separate study conducted by Noblis for the Federal Highway Administration (FHWA) Traffic Analysis Tools Program.

1.4 Report Organization

Section 2 provides a summary of existing literature on predicting travel times and traffic conditions using probe data and advanced analytics. Section 3 presents the hypotheses and assumptions of the study. Section 4 includes a description of the data and computing resources used for the analysis, followed by a description of the technical approach in Section 5. Section 6 presents the data analysis, and Section 7 discusses the results of the study. Finally, conclusions, including key findings and future research, are discussed in Section 8.

2 Literature Review

This section summarizes research conducted in predicting travel times and traffic conditions using probe data and advanced analytics, and assesses their strengths and limitations for possible adaptation or use in our study.

2.1 Travel Time Predictions Using Probe Data

2.1.1 Large-Scale Estimation in Cyberphysical Systems Using Streaming Data

2.1.1.1 Purpose

University of California Berkeley researchers sought to predict a driver's travel times in a large city area given sparse GPS traces [7]. The primary basis of the study addresses the utility of extracting travel time distributions from sparse, noisy GPS measurements collected in real-time from vehicles over a large network. The paper's pipeline evaluates probabilistic distribution of travel times over road segments by using GPS data from probe vehicles.

2.1.1.2 Data

The study made use of GPS traces collected at 1-minute intervals from taxi cabs in San Francisco for more than a year, creating hundreds of millions of GPS points. Ground truth was calculated using data over a 2-day period from 10 taxicabs that generated GPS data every 1 second.

2.1.1.3 Methodology

Raw GPS readings were first projected onto the road network. Feasible paths were identified between each pair of candidate points, and each path was assigned a probability using a stochastic model for vehicle dynamics and probabilistic driver preferences learned from data. Travel times were allocated to each link that was on the trajectory using Expectation Maximization (EM) algorithm. Lack of coverage was handled by including data from the same day before the current time step (between 20 minutes and 2 hours); as well as previous days, corresponding to same day of the week (1 to 10 weeks). Estimations were made every 20 minutes, and only trips of duration 10 to 30 minutes were examined. The study used Spark, which is an in-memory batch processing framework started in UC Berkeley and now open sourced through Apache.

2.1.1.4 Results

The best performance was observed when using more historical data (i.e., 2 hours prior to current time, 10 weeks prior to current day). Errors were lowest for trips of duration 4 to 11 minutes. Mean absolute deviation was approximately 0.5 minutes (percent error of 5-13%). Travel time estimates got worse when vehicles were stopped at red lights for unusually long times.

2.1.1.5 Strengths

The approach is highly scalable; given twice as many computation nodes, the algorithm performs the same task about twice as fast. Their approach was able to update the traffic state within a few seconds with sufficient computing resources.

2.1.1.6 Limitations

Their approach may not be suitable for longer trips as errors increased monotonically for durations greater than 11 minutes.

2.1.2 Real-Time Estimation of Distributed Parameters Systems

2.1.2.1 Purpose

The key purpose of this study was to develop a real-time estimation algorithm for monitoring traffic using velocity data from mobile devices [8]. Travel times were estimated using GPS-enabled cell phones.

2.1.2.2 Data

The study made use of GPS traces collected from cars driven by 77 students on I-880 NB, in San Francisco Bay Area, at virtual trip lines (VTL) creating 1100 vehicle trajectories. Loop detector data were obtained from PeMS. Ground truth travel time data were obtained from video license plate re-identification.

2.1.2.3 Methodology

First, Lighthill-Whitham-Richards (LWR) density model (which makes use of flow conservation) was transformed into a velocity-based function. Next, the LWR generalized velocity was transformed into discrete velocity evolution. Velocity at the next time step in a given cell is computed as a function of the velocity at the previous time step in the current cell and the immediately upstream and downstream cells.

Scenarios were modeled by varying:

- Number of loop detectors between 0 and 16
- Number of probe vehicle trajectories used between 0 and 1100 (100%)
- Number of virtual trip lines between 9 (spacing of 8.68 miles) and 99 (spacing of 0.79 miles)

2.1.2.4 Results

Errors of less than 10% was achieved whether using data from only loop detectors; only probe data; or a combination of both. Adding more than 8 loop detectors stations (average spacing of 0.83 miles) did not yield additional benefits; errors remained between 6% and 13%. Increasing the number of probe measurements by more than 31 VTL (i.e., spacing of 2.54 miles) did not improve the accuracy.

2.1.2.5 Strengths

The approach may be applicable for estimating travel times on freeways, especially when probe messages or BSM are sparse. While the research did not make use advanced analytics or tools, the methodology can be implemented on high performance computing due to its multi-threading capability.

2.1.2.6 Limitations

The accuracy of the algorithm is unknown on interrupted facilities. Secondly, scalability and latency of approach have not been tested.

2.1.3 Learning the Dynamics of Arterial Travel from Probe Data Using a Dynamic Bayesian Network

2.1.3.1 Purpose

The study estimated and predicted arterial travel times using probe vehicle data that were received at random times at random locations [9].

2.1.3.2 Data

The study made use of GPS traces at 1-minute intervals from a fleet of 500 probe vehicles in San Francisco.

2.1.3.3 Methodology

The researchers used Dynamic Bayesian Network to estimate and make a short term forecast of probability distribution function (pdf) of travel times. First, a probability density was assigned to each observation (made up of two GPS readings) that depends on the pdf of travel times on links traversed between the two measurements and the spatial distribution of vehicles over the links. Next, dynamic model of the dependence between travel time observations and the congestion state of a link at a given time were created. Finally, expectation-maximization algorithm and historical learning techniques were used to estimate the current state of traffic on the network and predict the probability of congestion in real time. Validation was performed by splitting data into training (70%) and validation (30%) sets.

2.1.3.4 Results

When market penetration was high, the error in estimating travel times was 6.8% and error in predicting travel times 15 minutes into the future was 24%. With sparse market penetration, the estimation error was 7.2% and the prediction error was 24.3%.

2.1.3.5 Strengths

Differences in the estimation and prediction errors at dense and sparse market penetrations were small. This is because the effect of sparse data was compensated by using a larger pool of historical data to train the model.

2.1.3.6 Limitations

Travel times were predicted only 15 minutes in advance, which is likely not enough time for a decision support system.

2.1.4 Arterial Travel Time Forecast with Streaming Data

2.1.4.1 Purpose

A key purpose of the study was to estimate and predict travel times on an arterial network using probe data with the help of a hybrid modeling framework [10].

2.1.4.2 Data

The study made use of GPS traces at 1-minute intervals from a fleet of 500 probe vehicles in San Francisco.

2.1.4.3 Methodology

The researchers developed a hybrid modeling framework that combined statistical and traffic theory models to take advantage of the availability of robust historical data. They defined a set of parameters (cycle time, red time, saturation number of vehicles, parameters of the free flow pace distribution) to characterize the probability distribution of travel times on the network. The parameters were trained using machine learning techniques with historical data. Parameters were assigned to each link and were then used to predict travel times in real time.

Validation was performed in two ways:

- The GPS traces from 500 vehicles were split into training (70%) and validation (30%) sets
- A field experiment was conducted to collect probe data every second from 20 drivers over three days on four routes.

2.1.4.4 Results

The mean percentage error of 37.67% was observed for the large scale validation set and 33.24% for the limited field test.

2.1.4.5 Strengths

Use of machine learning techniques and historical data to estimate the parameters for travel time estimation reduced errors.

2.1.4.6 Limitations

Errors are higher than what we were able to achieve in Phase 1 (23%). Latency and scalability of approach are untested.

2.1.5 Real-Time Traffic Modeling and Estimation with Streaming Probe Data using Machine Learning

2.1.5.1 Purpose

The study estimated arterial (non-highway, major city streets) traffic conditions using speed and location data provided by sparse GPS probe data from mobile devices [11].

2.1.5.2 Data

VTL data comprised data from 20 vehicles driven at the ITS World Congress in New York City representing 2% of vehicle flow as well as data from the Paramics micro-simulation software based on the SR41 corridor in Fresno, CA representing 5% of the vehicle flow.

The model was tested using probe data from a fleet of about 500 taxis in San Francisco. Each taxi provided a measurement of its location approximately once every minute (generally between 40 and 100 seconds).

2.1.5.3 Methodology

The author developed a hybrid modeling framework utilizing multiple models:

- VTL system using STARMA and logistic regression models.
- Bayesian real-time estimation model
- Graphical Coupled Hidden Markov Model

2.1.5.4 Results

The STARMA model estimation accuracy ranged from 71% to 78%. The mean percentage error of the baseline Bayesian model was 44.4%, and that of the graphical model was 30.1%.

2.1.5.5 Strengths

The Bayesian model provided a robust estimate of the general distribution of traffic patterns; although untested, the model should be able to work on larger networks. The graphical model leveraged traffic conditions over many days to identify traffic patterns.

2.1.5.6 Limitations

Both regression methods have high data requirements. Majority of Bayesian input data span several links per observation, limiting the precision of estimates. Updates are made only at 15 minute intervals. Errors are higher than what we were able to achieve in Phase 1.

2.2 Key Findings

- The reviewed literature dealt with the *estimation* of travel times, either on a freeway or on an arterial; only one approach *predicted* travel times 15 minutes in advance.
- The studies estimated or predicted travel times for the entire trip, versus 100' x 100' boxes in Phase 1 of our study.
- Errors in the reviewed literature were comparable to what was seen in Phase 1 of our study and in some cases worse than what we achieved.
- Predictions were made 15 minutes in advance in reviewed literature versus 1 hour in Phase 1 of our study; accuracy should be higher if the future time is closer to current time.

3 Hypotheses and Assumptions

3.1 Hypotheses

The following hypotheses will be tested in the study.

- Hypothesis #1: Proposed approach that makes use of high-volume connected vehicle data, advanced analytics, and cloud computing will meet computational speed requirements for a real-time decision support system
- Hypothesis #2: Proposed approach will be able to predict traffic flow regimes with high temporal and geographic precision and accuracy for higher market penetration of messages

3.2 Assumptions

- Equipment/device failures are not part of the assessment.
- Traffic regimes/indices are identified based on the work by researchers at Virginia Tech [12] and precipitation and visibility levels observed in the data used for the study. Table 3-1 shows the traffic regimes that were modeled in the study.

Table 3-1: Traffic Regimes Modeled in the Study

Traffic Regime	Clear to Light Rain ($\leq 0.1'$), Good to Medium Visibility (>5 mi)	Clear to Light Rain ($\leq 0.1'$), Low Visibility (≤ 5 mi)	Index
Free Flow Regime	Speed $> 85\%$ of Free Flow Speed	Speed $> 74\%$ of Free Flow Speed	1
Speed at Capacity Regime	75% of FF Speed $<$ Speed $\leq 85\%$ of FF Speed	65% of FF Speed $<$ Speed $\leq 74\%$ of FF Speed	2
Congested Regime	Speed $\leq 75\%$ of Free Flow Speed	Speed $\leq 65\%$ of Free Flow Speed	3

4 Data and Computing Resources

4.1 Data

4.1.1 Data Used for Analysis

A full-scale deployment of connected vehicle technology has not yet occurred. Hence, a representative large scale data set that was created using a traffic simulation tool was used for the analysis. The simulated data set was generated as part of a separate project that Noblis conducted for FHWA's Traffic Analysis Tools (TAT) Program [13]. The geographic network (Figure 4-1) used for the analysis in this study was the I-405 Corridor, a 29.5 mile long major commuter corridor in the Seattle area that is subject to periods of high travel demand and congestion. The I-405 corridor experiences significant travel time variability as a result of dynamic incident patterns and frequent rain and fog. The Washington State Department of Transportation (WSDOT) provided FHWA and Noblis, traffic, travel time, incident, and weather data for 2012. After removing weekends and holidays, there were 196 weekdays left that were clustered into six operational conditions for the study – low demand, low visibility, weather and incidents, many incidents, bottleneck trouble, and few incidents – and a representative day was selected for each cluster [13]. WSDOT also provided a VISSIM model of the I-405 network as part of the TAT project. Using this VISSIM network as an initial base model, Noblis calibrated VISSIM models for each of the six operational conditions [13].

The simulated vehicle trajectories from the six calibrated VISSIM models were made available to us for use in this project. These trajectories were used as input to the Trajectory Converter Analysis (TCA) to emulate connected vehicle Basic Safety Messages (BSM) and Basic Mobility Messages (BMM). Section 4.1.3 discusses the emulation process in detail.

Table 4-1: Parameters varied for study and their range of values

Parameters	Range
Study Period	AM Peak (5:30-10:30 AM)
Operational Conditions	1. Low Demand 2. Low Visibility 3. Weather + Incidents 4. Many Incidents 5. Bottleneck Trouble 6. Few Incidents
Market Penetration of Connected Vehicles	20%, 75%
Communication Strategies	Cellular, DSRC with RSEs deployed at major interchanges (see Figure 4-2)

Basic Safety Messages were examined for all combinations of these variables, Basic Mobility Messages (BMM) were examined for all Operational Conditions using just a Cellular communication strategy and 20% market penetration. The parameters used to generate BMMs using TCA-DIDC are given in Section 4.1.2.5.

4.1.2.2 Market Penetration of Connected Vehicles

Prediction model development and testing was performed for two market penetrations of connected vehicles 20% and 75%. In TCA, the probability that a vehicle is an equipped vehicle capable of transmitting BSMs, is equal to the specified market penetration. For example, if the market penetration is set as 20%, then in TCA a vehicle has a 20% probability that it is a connected vehicle.

4.1.2.3 Operational Conditions

Prediction model development and testing was performed for six operational traffic conditions, including:

1. Low Demand
2. Low Visibility
3. Weather + Incidents
4. Many Incidents
5. Bottleneck Trouble
6. Few Incidents

The incidents were modeled as a speed reduction over all lanes for the incident duration and incident area.

The simulation period was for the AM peak period from 5:30 AM to 10:30 AM.

4.1.2.4 Communication Strategies

Two communication strategies were tested in this study. The first was a wide area cellular network which could collect all BSMs generated and transmitted anywhere on the network. The second was a Dedicated Short Range Communication network where BSMs were collected through Roadside Equipment placed at all major interchanges. In the second case, only BSMs generated and

transmitted within range of an RSE would be captured by the network and available for analysis. Ten RSEs were placed along the entire I-405 corridor at major interchanges as shown in Figure 4-2.

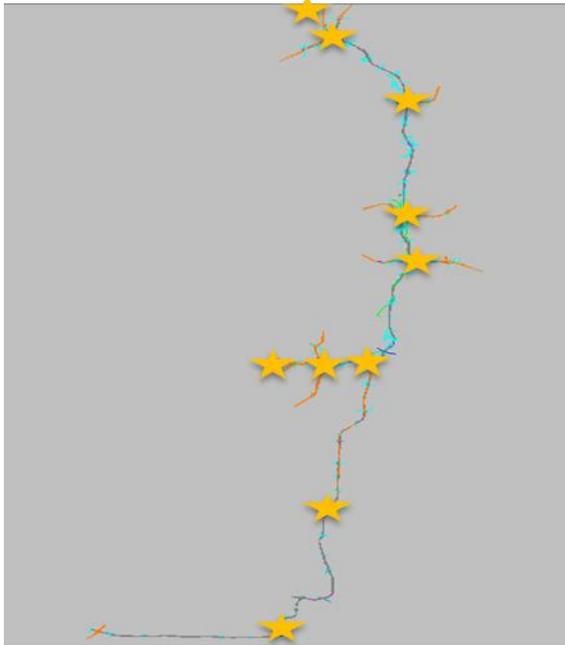


Figure 4-2: Locations of Roadside Equipment at Major Interchanges on I-405

For this study, two message types were used: a periodic message with a target of 20 BMMs per 100 feet and a Speed at Capacity event based message triggered whenever vehicle speed fell between 52.5 and 59.5 MPH (or 45.5 and 51.8 MPH for Low Visibility). Periodic message rates were optimized every 270 seconds, while Speed at Capacity messages were optimized every 15 seconds.

4.1.2.5 DIDC Parameters

For this study, two message types were used: a periodic message with a target of 20 BMMs per 100 feet and a Speed at Capacity event based message triggered whenever vehicle speed fell between 52.5 and 59.5 MPH (or 45.5 and 51.8 MPH for Low Visibility). Periodic message rates were optimized every 270 seconds, while Speed at Capacity messages were optimized every 15 seconds.

4.1.3 Emulated Connected Vehicle Messages

Basic Safety Messages (BSM) used in this analysis were emulated using the Trajectory Converter Analysis (TCA) software version 2.3, developed by Noblis. The TCA software uses vehicle trajectories (i.e., dynamic representation of vehicle kinematics) to replicate the generation of connected vehicle messages (e.g., BSMs). The messages are then transmitted via DSRC or cellular communications. The generation and transmission of messages are dictated by user-defined parameters. Please see the TCA Concept of Operations for a detailed discussion of the TCA [14].

Basic Mobility Messages (BMM) used in this analysis were emulated using the Trajectory Converter Analysis (TCA) DIDC software version 2.4, developed by Noblis. This version of the TCA software uses vehicle trajectories along with an emulated Dynamic Interrogative Data Capture (DIDC)

U.S. Department of Transportation
Intelligent Transportation System Joint Program Office

Controller to replicate the generation of prototype Basic Mobility Messages. The primary goal of DIDC Controlled Basic Mobility Messages is to dynamically throttle and/or control data capture and transmission rates from wireless entities while still providing the data set needed to optimize system management. The DIDC Controller is defined by the DIDC Parameters (Section 4.1.2.5) that specifies when event-based messages are triggered and the target number of messages desired for periodic or event based messages. Message generation rates are then adjusted to meet these thresholds at a specific optimization interval rate.

In this study, time-dependent records of vehicle position and speed as generated by the VISSIM model of the I-405 network were used as input to the TCA. Control variables, including market penetration and communication strategy, as defined in Section 4.1.2 are then used by the TCA to emulate the BSMs and BMMs that would be generated by equipped vehicles.

4.2 Computing Resources

The computational resources and analytic environment used for this work were provisioned via the Microsoft Azure cloud environment [15], and consisted of two primary elements: file storage and computing cluster.

File storage was provided using the Azure Blob Storage service, which is designed to store unstructured data in any format as objects or blobs. The Blob Storage service provides easy access to extremely large storage space up to 500TB per instance, with built-in redundancy and data protection, without requiring management of or even visibility into individual disks or volumes. This allowed simulated data files on the order of 150GB to be created and accessed routinely. The Azure Blob Storage service is also provisioned as network-local to Azure compute resources; so these data files are readily accessible with no WAN bandwidth limitations.

Compute resources were provided via the Azure HDInsight service [16], which is a cloud-native distribution of the open-source Apache Hadoop framework [17]. The HDInsight service provides template-based provisioning to allow clusters to be spun up and down on demand, without requiring manual configuration of either cluster machines or software packages. The template used for this work included the open-source Apache Spark data processing package, along with its MLlib machine learning library [18]. Code for these analyses was developed using both the Scala and Python (PySpark) languages.

The cluster used for these analyses consisted of four nodes in total: 2 head nodes for job submission and management and 2 worker nodes for computation. Specific provisioned configurations of these nodes are given in Table 4-2.

Table 4-2: Cluster Configuration

Cluster Nodes	CPU Cores	Memory (RAM)	Local Disk (HDFS)
Head nodes (2 each)	4	28 GB	200 GB
Worker nodes (2 each)	16	112 GB	800 GB

5 Technical Approach

This section describes the technical approach, including data pre-processing, feature identification, feature extraction, and development and test of machine learning algorithms, to predict traffic regimes (see Table 3-1) in 100' x 100' boxes overlaid on the I-405 traffic network, every 5 minutes an hour ahead of time using simulated BSMs and BMMs. As the key purpose of the prediction algorithm is to classify data into three traffic regimes, this is formulated as a classification problem. This section also discusses the evaluation criteria for estimating the accuracy of the prediction models.

Figure 5-1 is a graphical illustration of the traffic regime prediction framework:

- *Field Data*: In this stage, field data, including demand, incident, and weather data, were obtained for the I-405 corridor from WSDOT and processed (see Section 4.1.1).
- *Simulated Data*: In the second stage, the processed field data were used as input to a VISSIM model for the I-405 network, also obtained from WSDOT, and calibrated (see Section 4.1.1). Simulated vehicle trajectories were fed into the TCA to emulate SAE J2735 BSMs (Section 4.1.3). BSMs were generated every 10th of a second, and transmitted either via DSRC-enabled RSEs or cellular networks, as defined in Table 4-1 (see Section 4.1.2). BMMs were generated and transmitted according to the DIDC Parameters as defined in Section 4.1.2.5.
- *Data Pre-Processing*: The third stage includes feature identification, data assembly, normalization, and data set creation. Speed, demand, incident, and weather were identified as the key features (Section 5.1.2). Next, a grid network of 100' x 100' boxes was overlaid on the I-405 network. For each box, the following data were assembled for the past 1 hour: average speeds at 5-minute intervals in the subject box, and two boxes upstream and downstream of the subject box; northbound and southbound I-405 demand over the past 1 hour; incident information over the past 1 hour; and precipitation levels and visibility over the past 1 hour. Once this was done, data were normalized. Data that are measured on different numeric scales are normalized or converted to a common scale so that no single feature dominates the others. Then we divided the data into 3 sets – training, validation, and test data sets. Training and validation data sets were used for model development and to prevent over-fitting, and the test set was used to report out the accuracy of the prediction models.
- *Feature Extraction*: In this stage the most relevant information is extracted from the original set of features and represented in a lower dimensionality space. The previous stage resulted in 69 raw features, some potentially highly correlated. Highly correlated features effectively represent the same phenomenon, causing an overrepresentation of that phenomenon and possibly leading to poor generalization. Hence, in this stage, the 69 possibly correlated features were transformed into uncorrelated variables using Principal Component Analysis (PCA). In this study 6, 8 and 11 principal components were tested. See section 5.2 for details.
- *Model Development and Selection*: In this stage, Spark's machine learning libraries for Logistic Regression, Decision Tree and Random Forest were used for predicting the traffic regimes. Traffic regime indices were predicted at 5-minute intervals, an hour in advance, in each of the 100' x 100' boxes on the network. For example, if the current time is 7 AM, predictions were made for 8 to 8:05 AM. The process was repeated every five minutes. The best models were selected based on the accuracy seen for the training and validation sets.

- *Model Evaluation*: In the final stage, the predictions models were evaluated using the test set. The selected evaluation metrics include: precision, recall and F1 score.

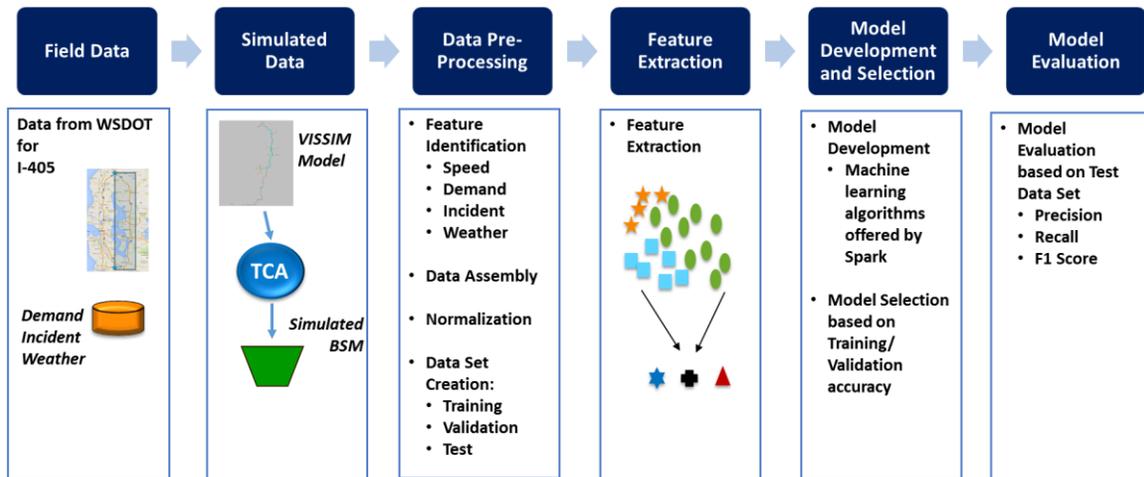


Figure 5-1: Framework for Traffic Regime Prediction Using Simulated Connected Vehicle Messages

5.1 Data Pre-Processing

5.1.1 Overlaying Box Network on Simulated Network

In order to group the simulated data for analysis, a means of relating BSMs to the road network was necessary. This results from the limited nature of the BSM data, which does not maintain any connection to the VISSIM network of roads used to generate traffic. To perform this grouping, the study area was overlaid with a 100' x 100' grid, and simulated BSMs were assigned to grid boxes by latitude and longitude, and tagged with a direction corresponding to the quadrant (North/South/East/West) the BSM heading falls into. A total of 1,332,834 boxes were defined on the network. The BSM records were also quantized into five-minute time buckets based on the time data.

Given that the BSMs are disconnected from any reference to the road network, identifying upstream and downstream traffic posed a significant challenge. More concretely, the challenge lies in determining which box should be considered “next” or “previous” for a given grid box with flow in a given direction (quadrant).

To solve this problem, the VISSIM network links were also mapped into grid boxes using the latitude and longitude coordinates. Then, the following algorithm was applied for each box/direction combination:

1. Identify the adjacent candidate boxes in the following order of preference:
 - a. Directly adjacent in the same direction
 - b. Ahead and to the left, relative to the direction
 - c. Ahead and to the right, relative to the direction
 - d. Directly to the left, relative to the direction
 - e. Directly to the right, relative to the direction
2. Select as “next” the first box in order that:
 - a. Shares a network link with the starting box, and

- b. Has flow in the correct direction
3. If no box meets the criteria, then there is no “next” box for that box/direction combination.

“Previous” (upstream) boxes are identified using the same algorithm, with the directions appropriately reversed. The algorithm is illustrated using Figure 5-2. Figure 5-2 shows a notional curved road segment with Eastbound traffic. In this example, the box labeled “1” would be the first box examined (as directly adjacent in the same direction). However, there are no VISSIM links shared with the starting box; so the algorithm would proceed to examine the box labeled “2”. In this box there are links shared with the starting box, so the algorithm would look for Eastbound traffic; if it finds any then it will select that box as “next” and stop. In the example, box 2 is identified as the next box.

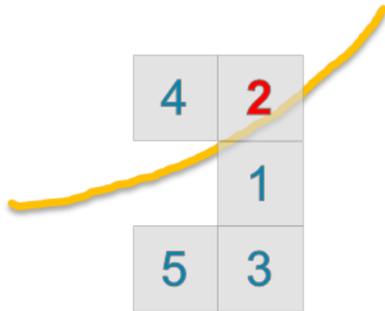


Figure 5-2: An Example Illustration of the Next Box Algorithm for a Curved Road

5.1.2 Feature Identification

5.1.2.1 Speed

Average speed is computed for each grid box for each direction quadrant: north, south, east and west and is quantized into 5-minute buckets using the Basic Safety Messages identified in that box travelling in that direction. Additional average speeds were calculated as features including average speed over 5-minute intervals for the past one hour in: the current box, two boxes upstream of the current box and two boxes downstream of the current box.

As noted in Section 5.1.1, determining upstream and downstream speeds is not trivial since grid boxes and BSMs are inherently tied to the road network. However, the next or previous box needs to be identified both by direction and roadway curvature as demonstrated in Figure 5-2. First boxes were manually mapped to VISSIM links representing roadway segments following the geographical coordinates of each box. Then code was developed that for each box-direction pair found the first box that shared a roadway segment and had flow in the correct direction. This process was then repeated for two boxes upstream and the second box downstream.

5.1.2.2 Weather

Weather data was provided by WSDOT at the request of FHWA for the representative days selected for each operational condition. This data included hourly observations reported for Seattle-Tacoma International Airport (KSEA) on the identified days. The observations of interest were visibility, reported in statute miles to the nearest tenth, and precipitation for the preceding 1 hour period, reported in inches and hundredths.

Prior to use, the weather data were pre-processed to replace “no data” observations with either the maximum (10.0 sm) for visibility or the minimum (0.00 in) for precipitation, as appropriate. The time of each observation record was also mapped to a minute-in-day offset in the period of interest, to align with the time scale used in the simulated BSM data.

5.1.2.3 Incidents

In the VISSIM models, incidents were identified as areas with reduced speed. The six models were used to manually identify the links where reduced speed areas occur. Once identified, VISSIM provided information on start and end time, duration, the length, number of impacted lanes, the free flow speed, the reduced speed, and a link ID for each incident. A Python script was used to identify the affected boxes corresponding to each link ID. The process is illustrated using an example in Figure 5-3. The red link shown in Figure 5-3 passes through four boxes (shaded); so the incident corresponding to this link would have four affected boxes associated with it. The Python script is used to identify the IDs of each of these boxes that the link passes over. The result was an incident file, showing the distinct operational condition, incident, link, and box ID combinations (Table 5-1).

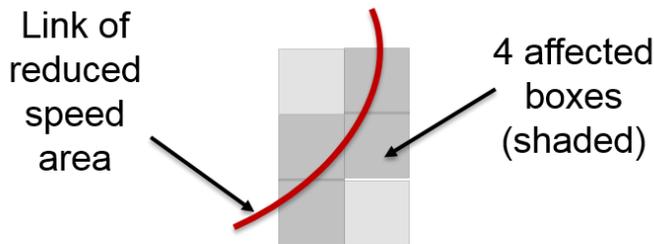


Figure 5-3: Example Incident

Table 5-1: Sample Incident file data

OC#	Incident ID	Start Time	End Time	Duration (min)	Length (ft)	Number of Impacted Lanes	Link	Affected Box ID#	Free Flow Speed (mph)	Reduced Speed (mph)
1	1	9:34 AM	9:39 AM	5	500	2	62	750805	70	2.7
1	1	9:34 AM	9:39 AM	5	500	2	62	750806	70	2.7
1	1	9:34 AM	9:39 AM	5	500	2	62	750807	70	2.7
1	1	9:34 AM	9:39 AM	5	500	2	62	750808	70	2.7
1	1	9:34 AM	9:39 AM	5	500	2	62	750809	70	2.7

5.1.2.4 Demand

The goal when representing demand was to have a separate demand file for each operational condition, each containing columns for origin zone, destination zone, and hourly flow rate in vehicles per hour, for each hour in the 5:30 AM to 10:30 AM period. The sample file in Table 5-2 shows what part of the demand file for operational condition #1 looks like – it shows hourly flow rate for vehicles starting at zone 1 and going to zones 1 through 5.

In order to develop this file, Noblis used O-D matrices for each operational condition that were provided with the I-405 VISSIM network. These were 193 x 193 matrices representing the origin and destination for the 193 zones in the network. The matrices specified the total number of vehicles traveling from one zone to another in a 30 minute period. In order to determine the hourly flow rate, the 30-minute matrices were summed to create one matrix per hour, per operational condition, for the entire period of 5:30 AM to 10:30 AM.

An issue was that over 90% of the O-D pairs had 0 vehicles traveling to and from those zones in a given hour, resulting in hourly matrices not ideal for feature identification. This issue was resolved by grouping the 193 zones into 13 larger zones, as shown in the image of the VISSIM network in Figure 5-4. Python scripts were developed to sum the values in the 193 x 193 O-D matrices by new zone to create new 13 x 13 O-D matrices, one for each hour and operational condition. Another script collapsed these matrices into the file format shown in Table 5-2.

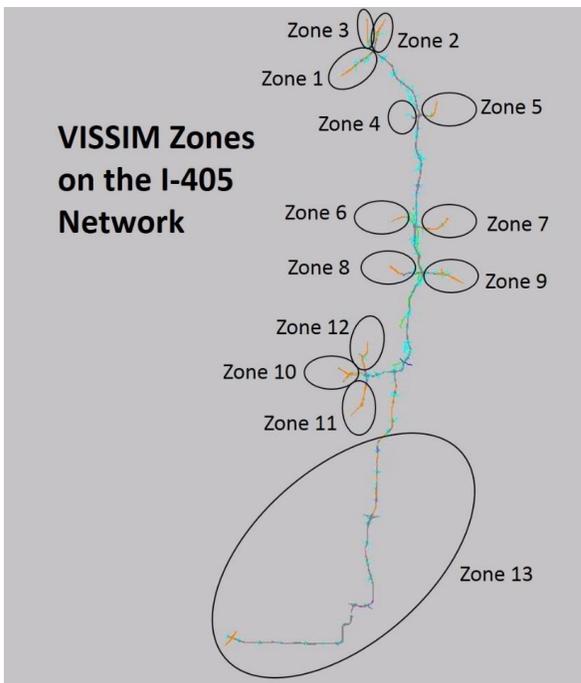


Figure 5-4: VISSIM Zones on the I-405 Network

Table 5-2: Sample Demand File Data: Operational Condition #1

Origin Zone	Destination Zone	5:30 - 6:30	6:30 - 7:30	7:30 - 8:30	8:30 - 9:30	9:30 - 10:30
1	1	1502	1689	3054	3017	2628
1	2	2365	3069	5415	4439	3633
1	3	737	803	1249	793	640
1	4	0	0	2	5	15

Origin Zone	Destination Zone	5:30 - 6:30	6:30 - 7:30	7:30 - 8:30	8:30 - 9:30	9:30 - 10:30
1	5	90	39	75	104	96

5.1.3 Data Assembly

Data corresponding to the features discussed in Section 5.1.2 were assembled into 69 raw features, including:

- Average speed in subject box at current time
- Average speeds over 5-minute intervals for the past 1 hour in:
 - Subject box
 - Two boxes immediately upstream of subject box
 - Two boxes immediately downstream of subject box
- Northbound network demand
- Southbound network demand
- Number of incidents in past 1 hour
- Reduced speed (for incident) in past 1 hour
- Number of lanes impacted by incident in past 1 hour
- Duration of incident in past 1 hour
- Precipitation levels in past 1 hour
- Visibility for past 1 hour

5.1.4 Data Normalization

In the feature extraction and prediction stages larger valued features would be overweighted. When using tools like Principal Component Analysis which maximizes variance, features that have larger values would contribute more to the overall variance. In this study, speed ranged from 0-70 MPH, while precipitation values were from 0-0.08 inches. If data are not normalized, speed would explain nearly all the variance while precipitation would not have much of an impact.

Data normalization is a process that transforms the data so that all feature values are within the same range, between 0 and 1 inclusive, while preserving variance in the data. In order to perform data normalization on Spark, the study utilizes PySpark's Normalizer API which normalizes samples individually to Normalizer API which normalizes samples individually to unit Lp norm, default is p=2 for Euclidean norm. The Euclidean norm is the length of the vector, x , as determined by the ordinary distance formula. In order to utilize the PySpark Normalizer the raw feature columns were into a single features vector and normalization was performed on these vectors.

5.1.5 Creating Data Sets

For the prediction phase, Spark Machine Learning algorithms need two inputs, a features vector (or predictors) and a label, which is what is being predicted. For this study, data is labeled by the traffic regime index calculated using the speed in the current box an hour ahead of the current time and the visibility in the current box an hour ahead of the current time to create a label-feature pair.

Data from each operational condition is combined into one master data set for each market penetration and communication strategy in order to train and test the model on the full data set. Prior to combining them, each data record is labeled by the operational condition it came from to support further analysis of individual operational conditions.

Data are then split into three sets: training, validation and test. The first split of the data creates 80% for training and validation combined and reserves 20% for testing. The 80% of the data is further randomly split into training (70% of the 80% data) and validation (30% of the 80% data) data sets for model development.

The training data set is used to develop and train the prediction model in the development phase. Using the label-feature pair created earlier, the selected model learns the mapping of the features onto the labels. The validation data set is used to fine tune the mapping of the features to the labels. The validation process in the development phase ensures that the model isn't over fitted to the training data; when training set accuracy increases and validation set accuracy decreases overfitting has occurred. The test data set is used to test the accuracy of the prediction model once the development phase is completed. In the testing phase the model is trained on the entire development data, then the traffic regime index is predicted for the test data. The predicted indices are compared to the known index, and the accuracy of the model is reported.

5.2 Feature Extraction

The last step before model prediction is performing feature extraction to select the best set of features to perform predictions with. With 69 total features and the likelihood that identified features are highly correlated, there is the risk of creating a poor performing model since the highly correlated features may be over represented. By reducing the feature space to the smallest number of features that contribute the highest amount of variance the model can more accurately map features onto labels creating better predictions.

This study utilized Principal Components Analysis (PCA), which is an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The first principal component has the largest possible variance and each subsequent component has the highest possible variance without being correlated to the previous component(s). PCA also performs dimensionality reduction, reducing the number of features to the most predictive number by selecting the top X number of principal components.

The number of required principal components was identified by performing a scree plot analysis in R using the nFactor package [19] on the Many Incidents, 20% Cellular data. Figure 5-5 shows the result of the scree plot analysis. The plot shows how much each principal component contributes to overall variance, with each point representing a principal component and its corresponding eigenvalue. The scree plot helps choose the number of principal components to use based on variance contributions; points chosen are when there is a significant drop in the contribution to overall variance by an individual component. In this study 6, 8 and 11 principal components were tested as indicated on the plot, as these were the three points with noticeable drops in contribution to variance.

Non Graphical Solutions to Scree Test

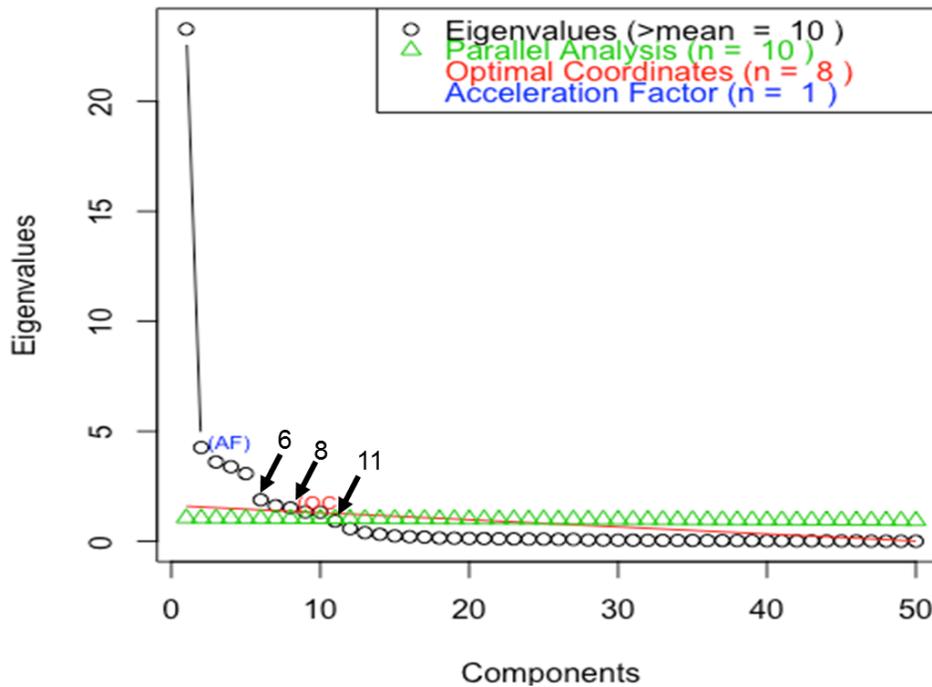


Figure 5-5: Scree Plot generated by R using the nFactor package on Many Incidents, 20% Cellular

5.3 Model Development

To analyze the preprocessed data and to develop and assess the performance of prediction models for traffic regimes, models were trained and evaluated using three different standard classification algorithms with Spark/MLlib implementations [18]: logistic regression, decision tree, and Random Forest. In all cases preprocessing of the data, including feature extraction, normalization, and principal component analysis, was performed prior to model training and evaluation, and identically-preprocessed data were used for each algorithm. Note that each algorithm was developed using 6, 8, and 11 principal components.

Logistic regression is a classification method in the family of linear regression methods, which are mathematically formalized as convex optimization problems. That is, given a vector of features, it seeks to find a set of weights which, when linearly combined, predict a dependent variable in such a way as to minimize the error between the prediction and true value. Formally, the optimization problem is, given n training data feature vectors x_i with length d and their corresponding labels y_i , to find the vector of weights w that minimizes the loss function L :

$$\min_{w \in \mathbb{R}^d} f(x) = \frac{1}{n} \sum_{i=1}^n L(w; x_i, y_i) \quad (1)$$

For training logistic regression models, the loss function given below is the logistic loss function over a linear combination of weights and features (the $\mathbf{w}^T \mathbf{x}$ term):

$$L(\mathbf{w}; \mathbf{x}, y) = \log(1 + e^{-y\mathbf{w}^T \mathbf{x}}) \quad (2)$$

Once the model is trained (i.e., the error-minimizing weights are found), the prediction step is made by applying the logistic sigmoid function below:

$$f(\mathbf{w}; \mathbf{x}) = \frac{1}{(1 + e^{-\mathbf{w}^T \mathbf{x}})} \quad (3)$$

$$\text{class}(\mathbf{w}; \mathbf{x}) = \begin{cases} 1, & f(\mathbf{w}; \mathbf{x}) > 0.5 \\ 0, & f(\mathbf{w}; \mathbf{x}) \leq 0.5 \end{cases} \quad (4)$$

To perform multiclass prediction, as in this work, a multinomial logistic regression model was built, consisting of two classifiers using the first class as a baseline: one evaluates the relative probability of class two over class one, and the other the relative probability of class three over class one. The class with the highest probability is then chosen as the final prediction (e.g., if classes two and three are both more likely than class one, the class with the highest-magnitude relative probability will be chosen). Finding the appropriate weights was accomplished using the the L-BFGS optimization algorithm, included in the Spark/MLlib implementation of logistic regression [20].

Decision trees are classification models that work by recursively splitting the solution space into binary classes, and thereby essentially creating a large tree of yes/no decision branches. The splitting criterion for each branch is determined by selecting the criterion that maximizes the difference between the two branches, according to a chosen metric. In this work, entropy and Gini impurity were evaluated as the splitting metric. The entropy metric seeks to maximize the information gain of the split; that is, to find the split that most rapidly narrows down the choice of predicted state. Formally, the split s is chosen at each tree node, applied to dataset D of size N , to create two subsets D_{left} and D_{right} of sizes N_{left} and N_{right} so as to maximize entropy $E(x)$ with respect to the number of discrete classes C , where f_i is the frequency of class i at a node:

$$E(x) = \sum_{i=1}^C -f_i \log f_i \quad (5)$$

$$\arg \max_s \left(E(D) - \frac{N_{left}}{N} E(D_{left}, s) - \frac{N_{right}}{N} E(D_{right}, s) \right) \quad (6)$$

The Gini impurity metric also seeks to maximize the information gain of a split, but instead of preferring the most efficient split, optimizes for minimizing the chance of misclassification given a particular split. The Gini impurity $G(x)$ is computed as below, and the split again chosen to maximize $G(x)$:

$$G(x) = \sum_{i=1}^C f_i (1 - f_i) \quad (7)$$

$$\arg \max_s \left(G(D) - \frac{N_{left}}{N} G(D_{left}, s) - \frac{N_{right}}{N} G(D_{right}, s) \right) \quad (8)$$

Random forests fall in the class of ensemble methods, whereby multiple classification models are trained and applied, with the highest-probability prediction being selected across all models. The intuition behind the approach is that while all classification models have error, the errors are not uniformly distributed across models, and running all the models in parallel minimizes the impact of the individual model errors.

Random forests specifically are implemented by creating an ensemble of many **partially-random** decision trees and then polling the ensemble to choose a predicted label. Randomization applied in the Spark/MLlib implementation includes bootstrapping from random subsamples, and using random subsets of features to define a split. With this randomization applied, although any individual random decision tree is likely to be a suboptimal predictor, each of them is flawed in a slightly different way, and so in aggregate these errors become diffuse and the result turns out to be surprisingly effective. In this work, multiple ensembles (i.e., 10, 250, 1000) of decision trees were used, again with both entropy and Gini impurity as information gain metrics.

5.4 Model Evaluation Metrics

Three evaluation metrics were chosen to evaluate the prediction models.

- Precision: For a given traffic regime index, how many of the predicted instances of the traffic regime index were correct?

$$Precision = \frac{t_p}{t_p + f_p} \quad (7)$$

- Recall: For a given traffic regime index, how many of the actual instances of the traffic regime index were predicted accurately?

$$Recall = \frac{t_p}{t_p + f_n} \quad (8)$$

- F1 Score: It is a balanced effectiveness measure and is measured as the harmonic mean of precision and recall.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (9)$$

6 Data Analysis

For each operational condition and experimental condition (market penetration and communication strategy combination), the total set of BSMs was classified by traffic regime index. Table 6-1 through Table 6-6 show the number of BSMs by traffic regime index for the six operational conditions.

For the Low Demand condition (Table 6-1), there were approximately three times as many '1' indices as there were '3' indices, with few '2's. It is important to note that for this operational condition (and all subsequent operational conditions), these percentages were similar for both Cellular (20%, 75%), and DSRC (20%, 75%) scenarios. However, the two DSRC conditions had only 12% the number of BSMs as the two Cellular scenarios.

Table 6-1: Number of BSMs by Traffic Regime Index – Operational Condition #1, Low Demand

Index	Cellular 20% Count	Cellular 20% Percentage	Cellular 75% Count	Cellular 75% Percentage	DSRC 20% Count	DSRC 20% Percentage	DSRC 75% Count	DSRC 75% Percentage
1	163184472	68%	618206973	68%	20927875	72%	78567703	72%
2	20874666	9%	79128929	9%	2615508	9%	9788777	9%
3	55944597	23%	214215981	24%	5383903	19%	20294171	19%

Table 6-2 and Table 6-3 show that the results for the Low Visibility condition and the Weather + Incidents condition were similar. For the Low Visibility condition, there were less than 25% '1' indices with the rest split between '2's and '3's, and for the Weather + Incidents condition, there were less than 10% '1' indices, with the rest split between the '2's and '3's. Both of these conditions represent the highest percentage of '2' indices across all operational conditions.

Table 6-2: Number of BSMs by Traffic Regime Index – Operational Condition #2, Low Visibility

Index	Cellular 20% Count	Cellular 20% Percentage	Cellular 75% Count	Cellular 75% Percentage	DSRC 20% Count	DSRC 20% Percentage	DSRC 75% Count	DSRC 75% Percentage
1	43626122	21%	164298333	21%	5815912	24%	22028094	24%
2	92349577	44%	347523341	43%	11448305	47%	42980111	47%
3	76083120	36%	288631944	36%	6912038	29%	26509479	29%

Table 6-3: Number of BSMS by Traffic Regime Index – Operational Condition #3, Weather + Incidents

Index	Cellular 20% Count	Cellular 20% Percentage	Cellular 75% Count	Cellular 75% Percentage	DSRC 20% Count	DSRC 20% Percentage	DSRC 75% Count	DSRC 75% Percentage
1	21443269	7%	80875386	7%	2651965	8%	10016519	8%
2	147389633	49%	553823879	49%	18614098	53%	69741674	53%
3	130794421	44%	497506469	44%	13725134	39%	52031882	39%

The Many Incidents (Table 6-4) operational condition was characterized by the lowest percentage of '2' indices, with the rest split between '1's and '3's. The Bottleneck Trouble (Table 6-5) condition was similar in that it also had an extremely low percentage of '2' indices, but the percentages of '1's and '3's were flipped (i.e. approximately 60% '1's for Many Incidents, and approximately 60% '3's for Bottleneck Trouble.) The Few Incidents (Table 6-6) condition had BSM numbers very similar to those of the Bottleneck Trouble condition, but slightly more '2' indices.

Looking at the entire data, traffic flow was free flow 42% of the time, at capacity 22% of the time, and congested 36% of the time. Thus the analysis revealed that the data were imbalanced.

For the scenario where only 20% of the vehicles were equipped and communication occurred via DSRC-based RSEs deployed at major interchanges, approximately 156 million BSMS were generated. When market penetration increased to 75%, the number of BSMS quadrupled to approximately 586 million BSMS. When BSMS were generated continuously by a vehicle throughout the trip and transmitted via cellular communication, the number of BSMS was 1.3 billion when the market penetration of connected vehicles was 20%. This was twice the number of BSMS generated via DSRC-based RSE at 75% market penetration. When the market penetration rose to 75%, the number of BSMS again quadrupled to nearly 4.9 billion BSMS.

Table 6-4: Number of BSMS by Traffic Regime Index – Operational Condition #4, Many Incidents

Index	Cellular 20% Count	Cellular 20% Percentage	Cellular 75% Count	Cellular 75% Percentage	DSRC 20% Count	DSRC 20% Percentage	DSRC 75% Count	DSRC 75% Percentage
1	32733591	40%	122665798	40%	4062396	35%	15217482	35%
2	912238	1%	3426433	1%	162367	1%	618845	1%
3	48886809	59%	182707769	59%	7490774	64%	28106390	64%

Table 6-5: Number of BSMS by Traffic Regime Index – Operational Condition #5, Bottleneck Trouble

Index	Cellular 20% Count	Cellular 20% Percentage	Cellular 75% Count	Cellular 75% Percentage	DSRC 20% Count	DSRC 20% Percentage	DSRC 75% Count	DSRC 75% Percentage
1	160487495	59%	601257752	59%	19958084	60%	74933985	60%

Index	Cellular 20% Count	Cellular 20% Percentage	Cellular 75% Count	Cellular 75% Percentage	DSRC 20% Count	DSRC 20% Percentage	DSRC 75% Count	DSRC 75% Percentage
2	5050257	2%	18928059	2%	861601	3%	3205688	3%
3	108006490	39%	405966647	39%	12282569	37%	46233313	37%

Table 6-6: Number of BSMs by Traffic Regime Index – Operational Condition #6, Few Incidents

Index	Cellular 20% Count	Cellular 20% Percentage	Cellular 75% Count	Cellular 75% Percentage	DSRSC 20% Count	DSRSC 20% Percentage	DSRSC 75% Count	DSRSC 75% Percentage
1	110521077	60%	414614270	59%	14222182	62%	53427423	62%
2	14983442	8%	55984497	8%	1833542	8%	6819867	8%
3	59551092	32%	226391096	32%	6887865	30%	25816717	30%

7 Results

7.1 Prediction Results Using Validation Data

7.1.1 Prediction Results by Operational Condition Using BSM

This section presents the prediction results by operational condition for the validation data comprising BSMs. Each prediction model performed about the same relative to the others across operational conditions. For most operational conditions Logistic Regression had the lowest F1 scores, while Random Forest and Decision Tree performed nearly equally, with small differences in favor of Random Forest. With respect to principal component k-values, results for predictions using 11 principal components were substantially and consistently better than using 6 or 8 principal components. Results for the latter two were near equal across operational conditions, traffic regimes and prediction models. While relative results were the same across operational conditions, the actual values varied, with substantially better results for certain traffic regime indices depending on the operational condition. With respect to the node impurity metrics, both Gini and entropy produced nearly similar results, with Gini outperforming entropy. With respect, to the tree configurations, the differences were nominal; however, the model that used random algorithm with 250 trees was the best overall.

Figure 7-1 to Figure 7-4 show a Trellis plot [21] of F1 scores for three prediction models ((i) Logistic Regression, (ii) Decision Trees with entropy node impurity metric, and (iii) Random Forest with entropy node impurity metric and 10 trees) by operational condition, traffic regime index and PCA k-value for DSRC-20%, DSRC-75%, Cellular-20%, and Cellular-75% scenarios, respectively.

For most experimental scenarios, models that used the Random Forest algorithm was the best performing model. outperforming the models that used the Decision Tree algorithm in F1 score by an average of 0.4% and Logistic Regression algorithm by an average of 11.2%. There were exceptions for specific cases.

For example, as shown in Figure 7-1, for the DSRC-20% experimental scenario all three models performed equally across traffic regimes for the Many Incidents (OC 4) operational condition.

For most operational conditions Logistic Regression had the lowest F1 scores, however, in a few instances, it was the best performing model. For Weather+Incidents (OC 3), Logistic Regression was the best at predicting free flow and speed at capacity regime for DSRC-75% (Figure 7-2) and Cellular-75% (Figure 7-4) scenarios.

All three models performed poorly in predicting speed at capacity regimes across operational conditions and across all scenarios, except for the Low Visibility (OC 2) and Weather+Incidents (OC 3) conditions. This divergence was because for these two operational conditions (Table 6-2 and Table 6-3), the data were imbalanced in favor of the at capacity regime. There were more than twice as many “2” than “1” and “3”. For these two operational conditions, index 2 data points were 44 and 49 percent of the total number of recorded BSMs respectively, while for the other operational conditions,

index 2 data points ranged between 1 and 9 percent of the total number of recorded BSMs. Increased representation of index 2 in the training set allowed for a more accurate prediction.

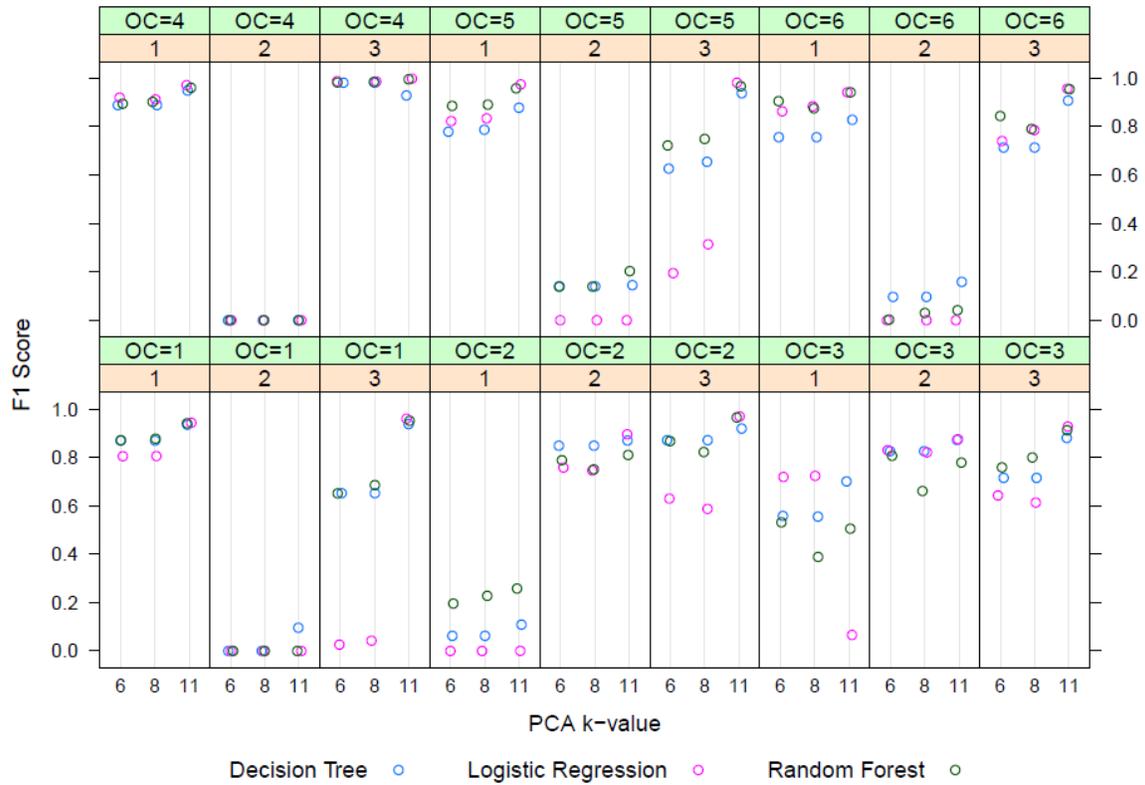


Figure 7-1: Trellis plot of F1 scores for each prediction model by operational condition, traffic regime index and PCA k-value for DSRC-20%

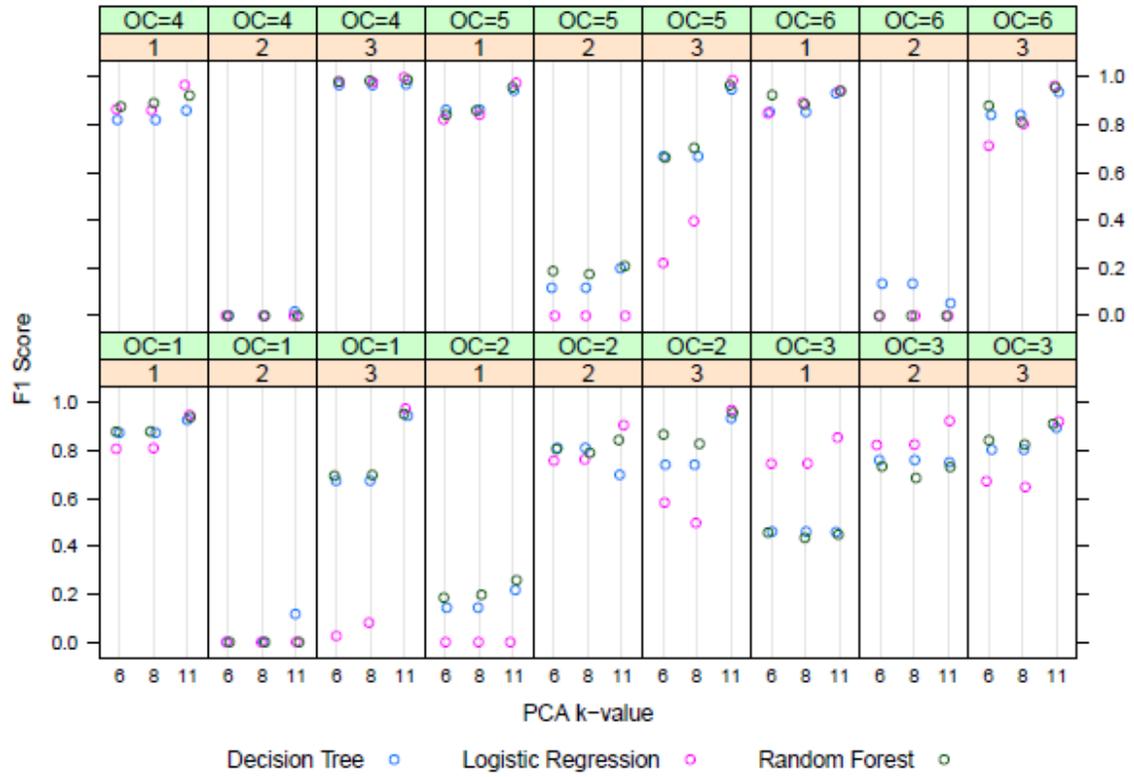


Figure 7-2: Trellis plot of F1 scores for each prediction model by operational condition, traffic regime index and PCA k-value for DSRC-75%

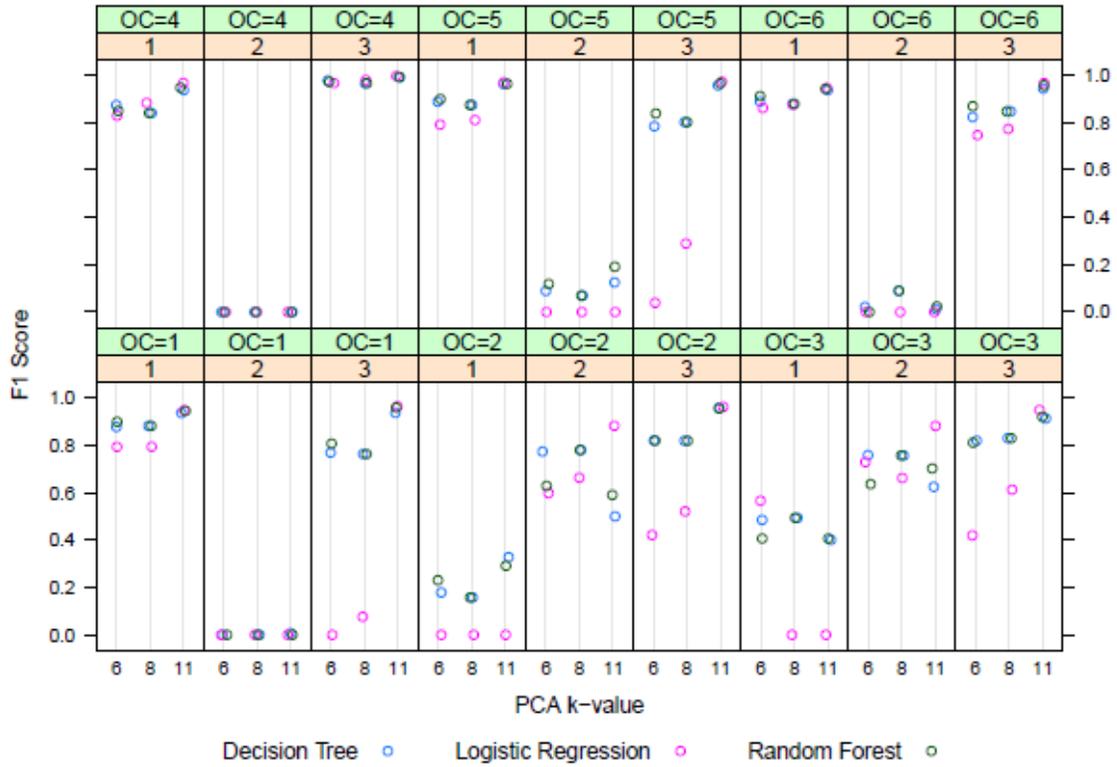


Figure 7-3: Trellis plot of F1 scores for each prediction model by operational condition, traffic regime index and PCA k-value for Cellular-20%

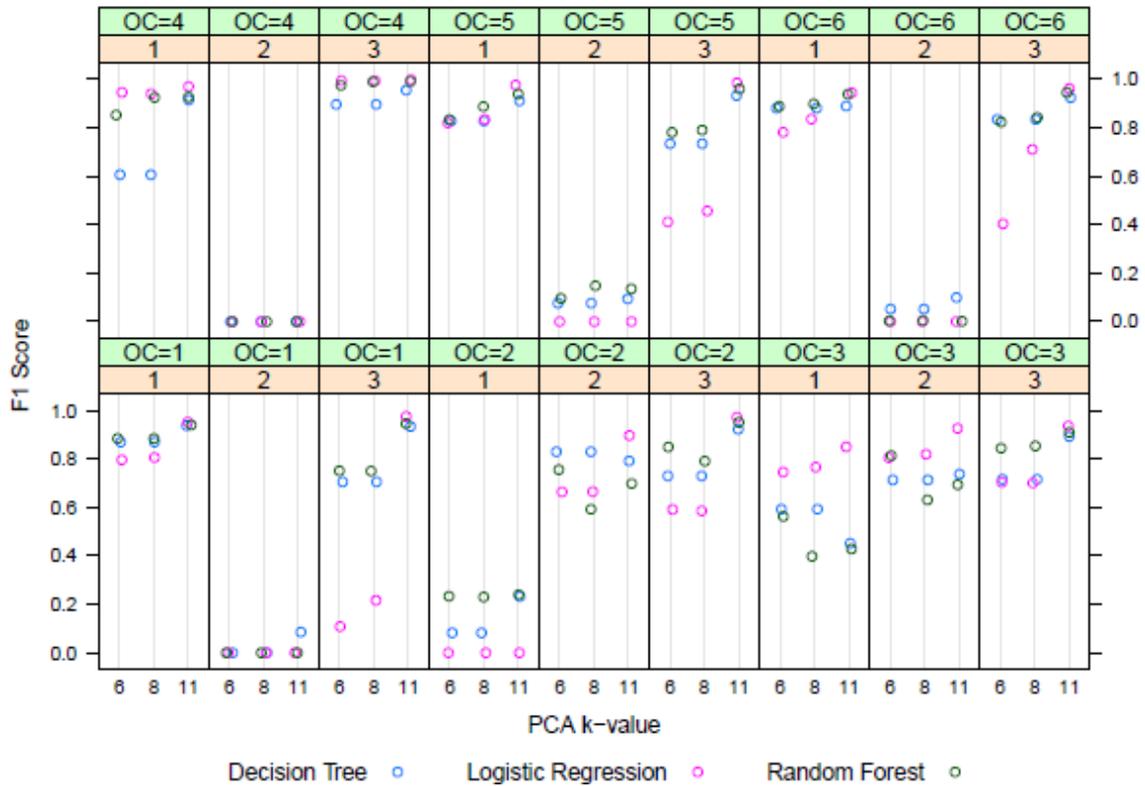


Figure 7-4: Trellis plot of F1 scores for each prediction model by operational condition, traffic regime index and PCA k-value for Cellular-75%

Table 7-1 presents the overall F1 scores of the three prediction models by communication mode and market penetration. The results show that across four scenarios, tree-based prediction models were consistently superior to Logistic Regression-based model. There wasn't a significant difference in the accuracy with increase in market penetration or change in communication model, implying that the prediction models are robust.

Prediction models were also developed using the Decision Trees and Random Forest algorithms with the Gini impurity metric. For Random Forest, two additional tree configurations (250, 1000 trees) were also examined. These validation results are included in Appendix A.

Table 7-1: Comparing F1 Scores of Prediction Models by Communication Mode and Market Penetration for Validation Data

Prediction Model	DSRC-20%	DSRC-75%	Cellular-20%	Cellular-75%
Logistic Regression (11 PC)	0.579	0.596	0.531	0.594
Decision Trees (11 PC, Entropy)	0.670	0.656	0.638	0.650
Random Forest (11 PC, Entropy, 10 Trees)	0.675	0.666	0.654	0.646

7.1.2 Summary Prediction Results Using BSM

The overall results for each model were compared for only 11 principal components since each model had the best performance for 11 principal components. Figure 7-5 to Figure 7-8 show Trellis plots of precision, recall, and F1 scores for three prediction models ((i) Logistic Regression, (ii) Decision Trees with entropy node impurity metric, and (iii) Random Forest with entropy node impurity metric and 10 trees), by traffic regime index for DSRC-20%, DSRC-75%, Cellular-20%, and Cellular-75% scenarios, respectively.

Precision and recall for both Decision Tree and Random Forest is above 0.8 for the free flow and congested (index 3) traffic regimes for both DSRC scenarios (Figure 7-5 and Figure 7-6) and the Cellular, 75% scenario (Figure 7-8). For these scenarios, the tree-based models had nearly similar F1 scores. In general, Decision Tree and Random Forest results were substantially better than Logistic Regression for all three metrics for free flow (index 1) and at capacity traffic regimes (index 2).

The results were, however, much different for the Cellular-20% scenario (Figure 7-7), where Logistic Regression was the best performing model by all three metrics. In this case, Logistic Regression had F1 scores an average of 0.068 points higher than Random Forest and Decision Tree across traffic regime indices. The Logistic Regression model also had a substantially higher precision for the free flow traffic regime and higher recall for the at capacity traffic regime. In this case Random Forest still outperformed Decision Tree, with F1 scores an average of 0.033 points greater. Despite the Cellular-20% exception, Random Forest still had the best results across scenarios for all three metrics. Across all experimental scenarios Random Forest had on average a precision 0.097 points, a recall 0.064 percentage points and an F1 score 0.082 points higher than Logistic Regression. While Random Forest and Decision Tree performed similarly, Random Forest was better able minimize the poor results for the Cellular-20% scenario.

Validation data results can be summarized as follows:

- Results using 11 principal components were substantially better across communication strategies, market penetrations and operational conditions.
- Overall, Random Forest had the best results across scenarios for all three metrics; Random Forest also minimized poor results when it wasn't the best metric.
- More representative data led to better predictions.
 - Across all scenarios, the F1 score was 0.67 for the at capacity traffic regime (22% of total BSMs generated) prediction compared to 0.87 for free flow (42% of BSMs) and 0.95 for congested (36% of BSM) regimes.
 - For operational conditions with more BSMs for the at capacity traffic regime (Low Visibility and Weather + Incidents), F1 scores were about 0.6 points better.

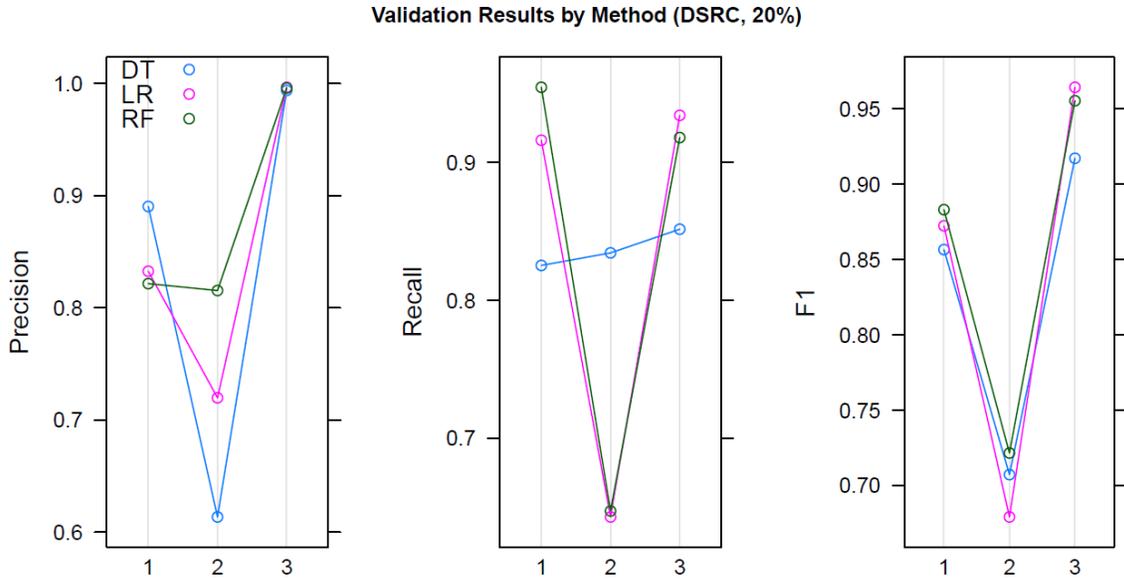


Figure 7-5: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined DSRC-20% validation data using 11 principal components

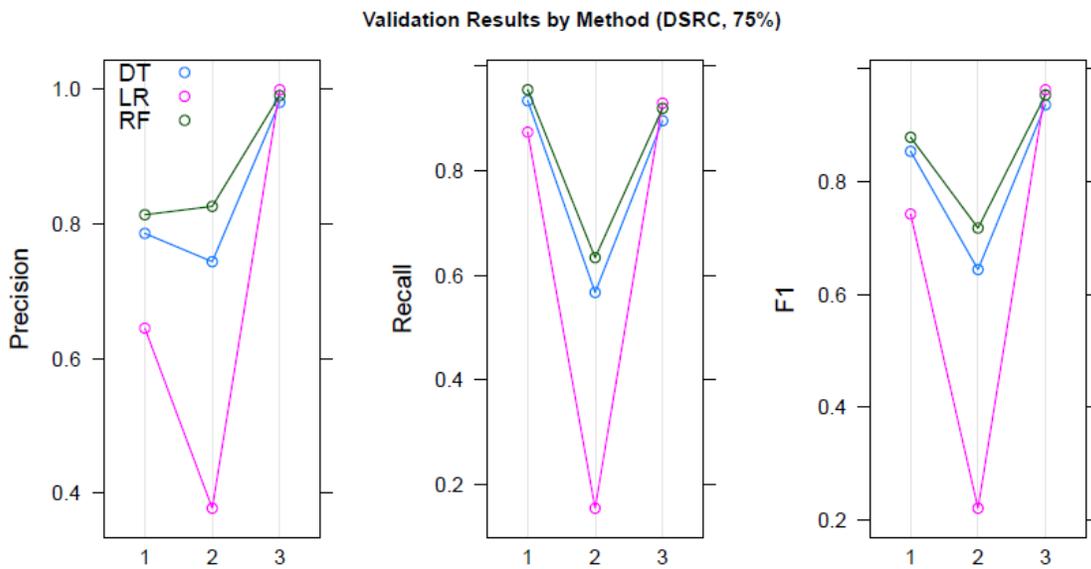


Figure 7-6: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined DSRC-75% validation data using 11 principal components

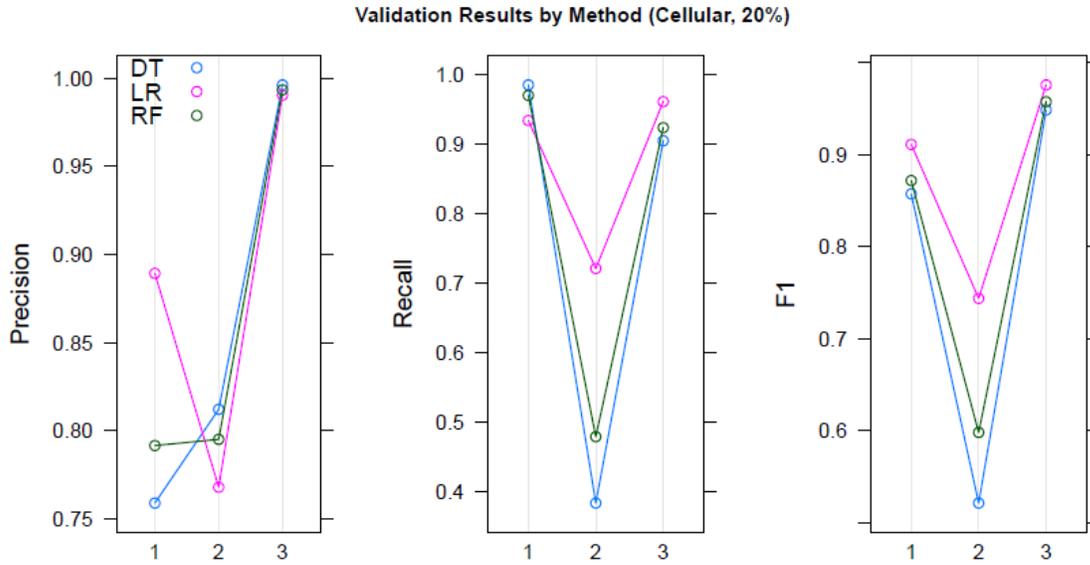


Figure 7-7: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined Cellular-20% validation data using 11 principal components

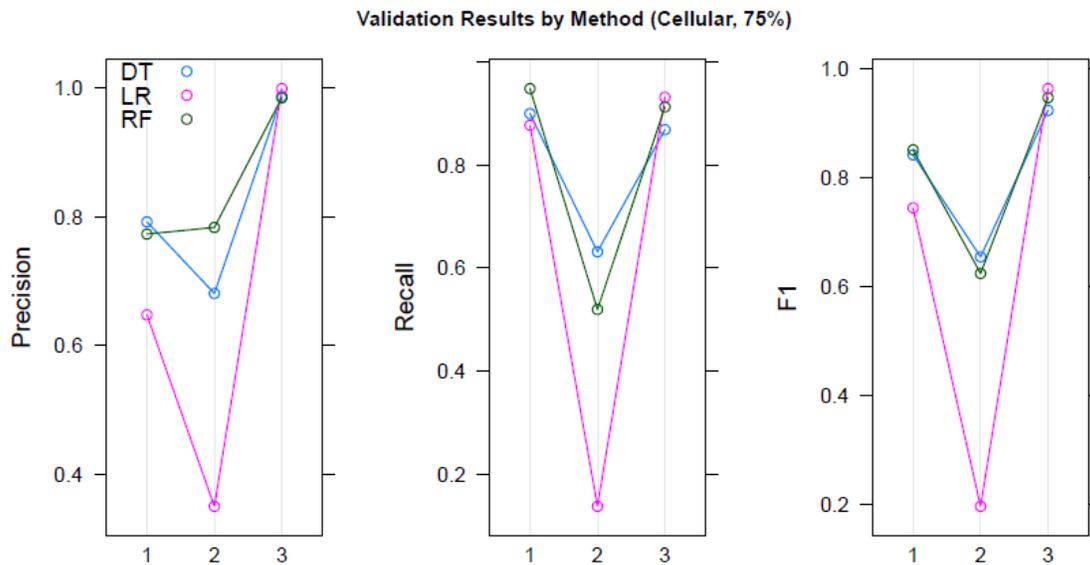


Figure 7-8: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined Cellular-75% validation data using 11 principal components

7.1.3 Summary Prediction Results Using BMM

This section discusses the prediction results for the validation data comprising BMMs. DIDC Controlled BMMs were generated and transmitted according to rules that were intended to increase the representation of messages at the Speed at Capacity traffic regime (index 2), since they were found to be underrepresented in the BSM data (Section 6). It was expected that more balanced

representation of all three traffic regimes would lead to improved results for the Speed at Capacity regime, bringing it closer to the Free Flow and Extreme Congestion Regimes results found using BSMs. A prediction model was developed using the Random Forest algorithm with 11 principal components, 10 trees and entropy metric. Table 7-2 shows the prediction results using BMMs transmitted over a Cellular communication network at 20 percent market penetration, averaged across all 6 operational conditions. The average F1 score for the Free Flow regime (0.93) was better than when using BSMs (0.87); however the average F1 score for the Congested regime (0.77) was substantially worse than when using BSMs (0.96) and the average F1 score for the Speed at Capacity regime was also worse (0.53) than when using BSMs (0.60). The results show that additional messages within Speed at Capacity traffic regime range did not help balance results and overall, had worse results than BSMs.

Table 7-2: Prediction Results Using Random Forest for Validation Data Comprising Basic Mobility Messages (BMM) for Cellular-20% Scenario

Traffic Regime	Precision	Recall	F1
Free Flow	0.99	0.87	0.93
Speed at Capacity	0.69	0.43	0.53
Congested	0.67	0.92	0.77

7.2 Prediction Results Using Test Data

7.2.1 Summary Prediction Results Using BSM

The performance of the three models using 11 principal components were evaluated using the test data set. Results for the test data mirrored the overall validation data results, with Random Forest being the best performing model overall, while Logistic Regression performed best for the Cellular-20% experimental scenario. Figure 7-9 to Figure 7-12 show for the test data Trellis plots of precision, recall, and F1 scores for the evaluated prediction models ((i) Logistic Regression (LR), (ii) Decision Trees (DT) with both entropy and Gini node impurity metrics, and (iii) Random Forest (RF) with both entropy and Gini node impurity metrics and ensembles of 10, 250, and 1000 trees), by traffic regime index for DSRC-20%, DSRC-75%, Cellular-20%, and Cellular-75% scenarios, respectively.

As shown in Figure 7-9, for the DSRC 20% scenario each model had the best precision for a single traffic regime index, with all three models having a precision above 0.8 for the free flow regime and above 0.9 for the congested regime. Decision Tree had the highest recall for the at capacity regime (index 2), but lowest for other regimes implying that it was able to predict over 80% of at capacity states. However, by F1 score, Random Forest was the best performing model for two of the three traffic regime indices and was second best for the congested traffic regime (index 3). This pattern held for the DSRC 75% (Figure 7-10) and Cellular 75% (Figure 7-12) experimental scenarios, where Random Forest performed substantially better for the free flow and at capacity traffic regimes and slightly worse for the congested traffic regime. Figure 7-11 shows the only exception to this pattern. As observed in the overall validation results, Logistic Regression had a much higher precision for the free flow traffic regime and much higher recall for the at capacity regime leading to Logistic Regression having the highest F1 scores for all three traffic regimes. In this case there was a difference of 0.05 points for the free flow traffic regime between Logistic Regression and Random Forest.

As seen for the validation data, Random Forest had the best overall results, with an F1 score on average 0.09 points higher than Logistic Regression and 0.03 points higher than Decision Tree. Across experimental scenarios, Random Forest had the best results with an average F1 score of 0.87 for free flow, 0.67 for at-capacity and 0.95 for congested traffic regimes (Table 7-3). Gini impurity also performed slightly better for both Decision Tree and Random Forest, generally achieving 0.01 higher F1 scores. The model with 1000-tree ensemble generally provided no additional performance over 250-tree ensembles, but required longer training time (average of 13.5 minutes versus 20.3 minutes). Therefore, Random Forest using 11 principal components, Gini impurity, and a 250-tree ensemble is the best model for predicting traffic regime indices on the I-405 network. Additional test results are included in Appendix B.

Test Results by Method (DSRC, 20%)

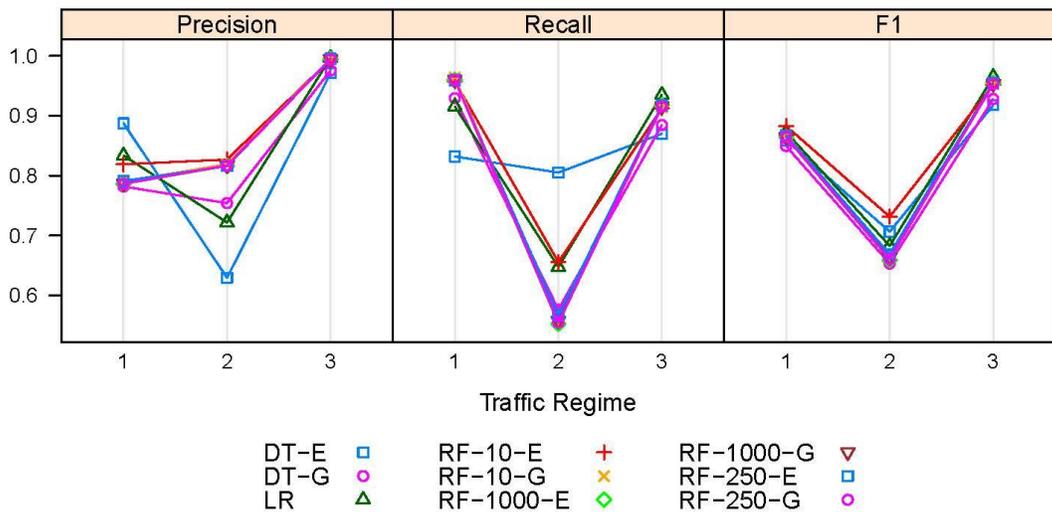


Figure 7-9: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined DSRC-20% test data using 11 principal components

Test Results by Method (DSRC, 75%)

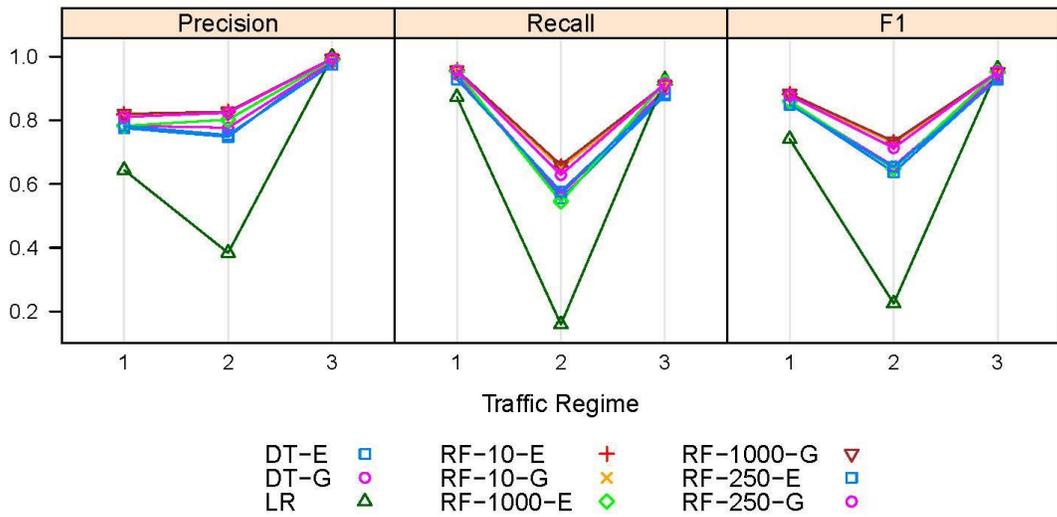


Figure 7-10: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined DSRC-75% test data using 11 principal components

Test Results by Method (Cellular, 20%)

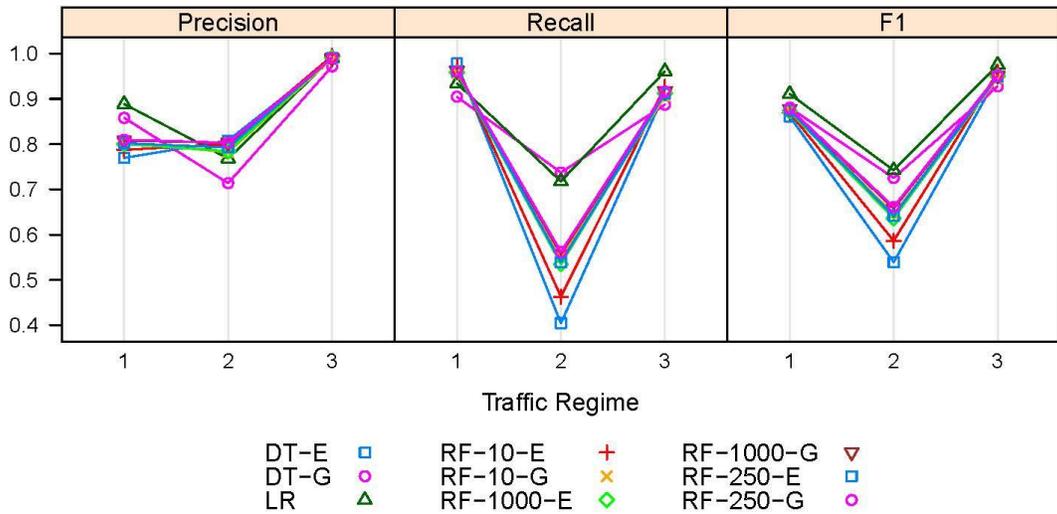


Figure 7-11: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined Cellular-20% test data using 11 principal components

Test Results by Method (Cellular, 75%)

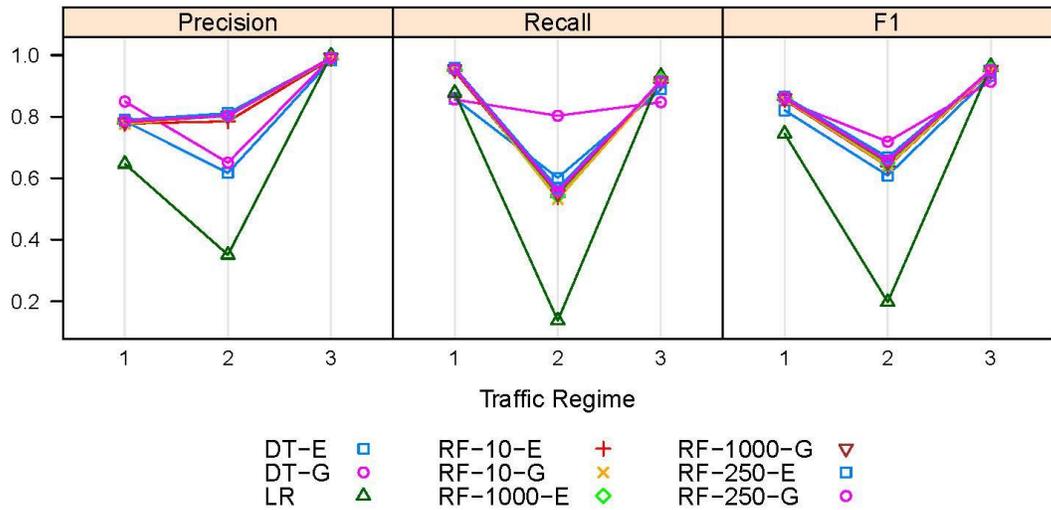


Figure 7-12: Trellis plot of precision, recall and F1 scores for each prediction model for each traffic regime for combined Cellular-75% test data using 11 principal components

Table 7-3: Average Prediction Results Using Random Forest for Test Data Comprising BSM Across All Scenarios

Index	Precision	Recall	F1 Score
1	0.80	0.96	0.87
2	0.81	0.58	0.67
3	0.99	0.92	0.95

Test data results can be summarized as follows:

- Overall, model that used the Random Forest algorithm with 250-tree ensemble and Gini impurity had the best results across scenarios for all three metrics.
- Gini impurity produced slightly better results than entropy information gain metric for Decision Trees and Random Forest.
- More data (i.e., increase in market penetration) didn't necessarily translate into better predictions; however, more representative data did produce higher F1 scores as is evidenced by the higher F1 scores for free flow and congested regimes than for the speed at capacity regime.

7.2.2 Summary Prediction Results Using BMM

This section discusses the prediction results for the test data comprising BMMs. Table 7-4 shows the prediction results using BMMs transmitted over a Cellular communication network at 20 percent market penetration, across all 6 operational conditions. The results were nearly the same as that seen for the validation data.

Table 7-4: Prediction Results Using Random Forest for Test Data Comprising Basic Mobility Messages (BMM) for Cellular-20% Scenario

Traffic Regime	Precision	Recall	F1
Free Flow	0.99	0.87	0.93
Speed at Capacity	0.69	0.43	0.53
Congested	0.67	0.93	0.78

The worse prediction performance using BMMs compared to BSMs is likely due to the use of burst messaging. DIDC uses event-based burst messaging to receive extra messages from around a vehicle when an event is triggered for the vehicle to determine whether the event was vehicle-specific or network induced. In this case, when a vehicle entered the Speed at Capacity traffic regime it triggered burst messages from any vehicle within 100 feet for the next 2-12 seconds. However, vehicles within the burst zone weren't necessarily all transmitting within the Speed at Capacity range. If one vehicle just drops below the divider between the Speed at Capacity regime, the surrounding vehicles are all likely on the border as well. If more fall on the Free Flow regime side, then the burst messages would increase representation of Free Flow regime messages relative to Speed at Capacity and Congested regimes, leading to better results for Free Flow, but worse for Speed at Capacity and Congested.

There are numerous DIDC parameters as well as ranges of possible optimal values for those parameters that can be set using a DIDC Controller. For this study only a single set of parameters and one value was tested. Thus there is potential for substantial improvement on prediction accuracy through the use of either different control parameters, different control values or both.

7.3 Tradeoffs between Information Insight and Cost/Timing

It is expected that connected vehicle data can be processed rapidly using advanced analytics, such as Apache Spark data processing package and machine learning libraries, and high performance computing, such as Microsoft Azure environment, to create precise predictions of traffic flow regimes, prior to the deterioration of roadway conditions. For an agency to consider advanced analytics in predicting traffic flow regimes, predictions should be made in a reasonable time to allow public agency staff to take the necessary congestion mitigating actions. Secondly, advanced analytics shouldn't be so expensive that it is not a viable option. Thus, it is essential to examine the tradeoffs between information insight and the cost and timing of data processing and prediction.

Figure 7-13 shows on the primary Y-axis, the time taken to train the models for the study, and on the secondary Y-axis, the time taken for prediction. The time taken for training the model includes the time taken for the following activities:

1. assign connected vehicle messages (BSM, BMM) to the more than 1 million 100' x 100' boxes;
2. calculate the average speeds for each box at 5-minute intervals;
3. combine data for all features for the previous 1 hour;
4. normalize the data;
5. perform principal component analysis to transform the normalized data; and
6. train the data.

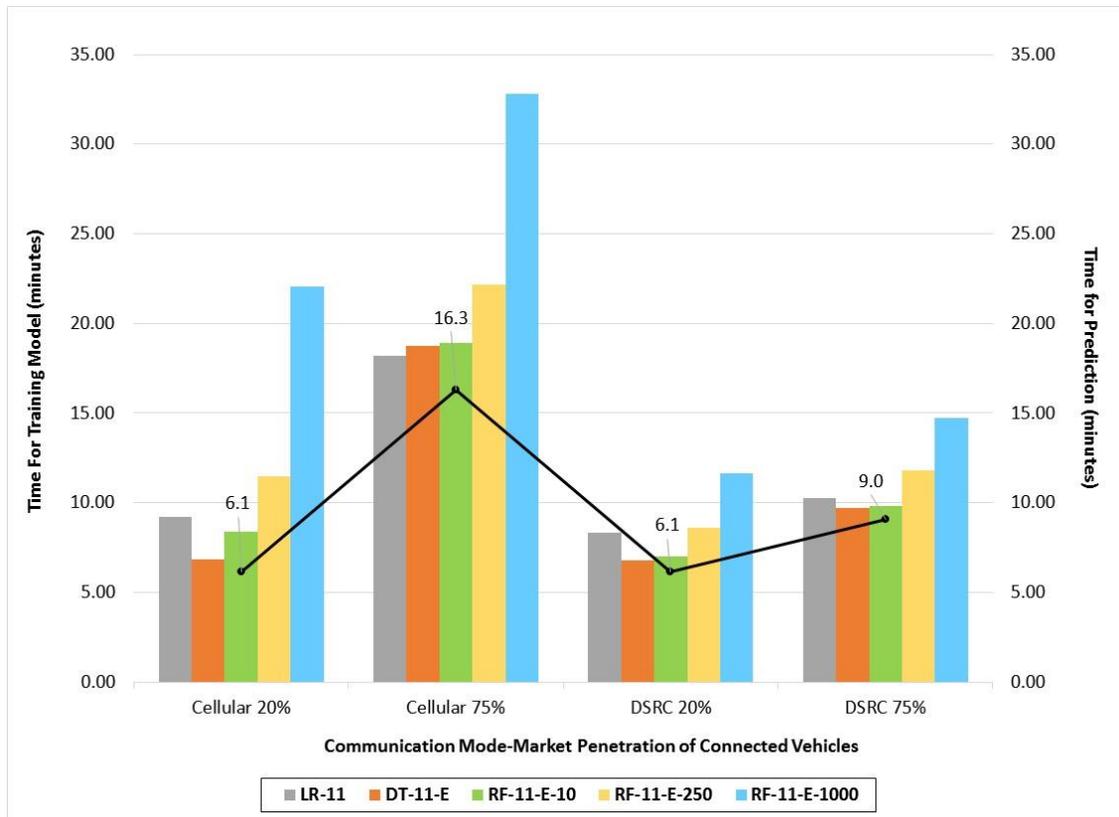


Figure 7-13: Comparing Time taken for Model Training and Prediction

It should be pointed out that the time taken for training does not include the time taken for defining a grid network and processing demand, incident, and weather data, as these are a factor of the type of data that are available to the agency and the geographic scope of the network.

The time taken for prediction includes the time taken for predicting the traffic regimes using the trained model. The prediction time also includes the time taken for items 1 to 5 above, since the user will need to process the data for the previous 1 hour.

The average training time was 13.4 minutes. The model that used the Decision Trees algorithm took the least time for training (6.8 minutes) for the scenario where messages were transmitted from 20% of the vehicles via DSRC every 10th of a second. The model that used the Random Forest Algorithm with 1000 trees took the most time for training (32.8 minutes) for the scenario where messages were transmitted from 75% of the vehicles via Cellular every 10th of a second. In general, the differences between the models are negligible, except for the model that used Random Forest with 1000 trees.

As was seen in Section 6, the DSRC-20% scenario had the least amount of data (~156 million BSMs), followed by the DSRC-75% scenario (~586 million BSMs), then the Cellular-20% scenario (~1.3 billion BSMs), and finally the Cellular-75% scenario with approximately 4.9 billion BSMs. With increase in data, the training time increased on average across models from 8.5 minutes for DSRC-20% scenario to 22.2 minutes for Cellular-75% scenario, but, there wasn't a corresponding increase in accuracy as seen in Section 7.2. This may have been a factor of the measure chosen (i.e., traffic flow regimes) and/or the network type (freeway).

The average prediction time was 9.4 minutes, and ranged from 6.1 to 16.3 minutes. The prediction times varied only by a few seconds across the models developed in this study since the differences in the models are in the training algorithms. Once the models are trained, the process is nearly identical.

In summary, for a network the size of I-405 corridor, developing and training prediction models may take less than an hour depending on the number of connected vehicles in the network. Once, the models are trained, an agency will be able to make predictions in less than 15-30 minutes, depending on the number of connected vehicles. The monthly cost of setting up the Azure environment, as defined in Section 4.2, and for processing and management of data for a network the size of the I-405 corridor was \$4,500.

8 Conclusions

In a connected vehicle environment, wireless sub-second data exchange connects vehicles, the infrastructure, and travelers' mobile devices. These data have the promise to transform the geographic scope, precision, and latency of transportation system control, thereby resulting in significant safety, mobility, and environmental benefits. However, the new data influx also has the potential to over-burden legacy computational and communication systems. Although connected vehicle technology can facilitate ubiquitous system coverage, existing prediction methods, computational platforms, and data management methods are insufficient to process the data within a reasonable timeframe for real-time predictions. With increased market adoption of connected vehicle technology, this data explosion is imminent, thereby necessitating big data solutions to fully exploit connected vehicle data for transformational improvements to the transportation system operations and management.

The focus of this analysis was to develop and test analytic tools that can handle data that is of such volume, velocity, and variety that it cannot be processed or managed using traditional tools (e.g., relational database management systems), and requires technologies that support big data. A secondary motivation for this study was to provide a practical example of how connected vehicle (CV) data can improve transportation operations with the intent of motivating others to investigate potential applied uses of CV data.

The study presented a technical approach that combined Apache Spark's open source data analytics and machine learning techniques to predict traffic flow regimes using simulated connected vehicle messages. The computational resources and analytic environment used for this work were provisioned via the Microsoft Azure cloud environment. Predictions were made for the following hour at 5-minute intervals for 100' x 100' boxes in less than 20 minutes. The study demonstrated that connected vehicle data can be processed rapidly using advanced ("big data") analytics and high performance computing to create precise predictions of traffic flow regimes, prior to the deterioration of roadway conditions. Public agency staff will be able to improve travel within these corridors by assessing the predicted congestion levels and undertaking suitable congestion mitigating actions.

Data used for this study along with data from a number of other open CV data sets can be found on the USDOT's RDE (<https://www.its-rde.net>) and other sources. The resulting code and documentation from this study is non-proprietary and will be posted on the USDOT's OSADP (itsforge.net) alongside many other existing Open Source CV applications. Researchers and application developers are encouraged to use the code and data to further research and development in this area, and share their results via sites such as the RDE and the OSADP.

8.1 Key Findings

The study validated the two hypotheses.

- *Hypothesis #1: Proposed approach that makes use of high-volume connected vehicle data, advanced analytics, and cloud computing will meet computational speed requirements for a real-time decision support system*

- The best performing model (Random Forest with 250-tree ensemble) was able to fully process an hour's worth of BSMs into the 100' x 100' grid boxes, calculate the average speed for each box, direction and 5-minute interval, find average speeds in upstream and downstream boxes, join environmental features, perform normalization and Principal Components Analysis, and make a prediction for the following hour, at 5-minute intervals for each of the 100' x 100' boxes in 6 to 16 minutes.
- *Hypothesis #2: Proposed approach will be able to predict traffic flow regimes with high temporal and geographic precision and accuracy for higher market penetration of messages*
 - Across all experimental scenarios, the model that used the Random Forest algorithm with 11 principal components, 250-tree ensemble and the Gini node impurity metric, had the best results with an average F1 score of 0.83. The F1 scores were 0.87 for free flow, 0.67 for at capacity and 0.95 for congested traffic regimes. Predictions were made for 100' x 100' boxes nearly an hour in advance at 5-minute intervals.
 - More data (i.e., increase in market penetration) didn't necessarily translate into better predictions; however, more *representative* data did produce higher F1 scores as is evidenced by the higher F1 scores for free flow and congested regimes than for the speed at capacity regime.

8.2 Future Research

The study showed that the model that used the Random Forest algorithm was the best overall, with an average F1 score of 0.83. While the overall score is good, for the at capacity regime it was only 0.67. Overall, at capacity BSMs were about 22% of the total BSMs – which is approximately half of what was generated for the other two regimes. This shows that the data was imbalanced. In our study, due to schedule and budget constraints, we examined the use of TCA-DIDC to oversample speed at capacity regimes using a single set of parameters. There are numerous DIDC parameters as well as ranges of possible optimal values for those parameters that can be set using a DIDC Controller. Thus there is potential for substantial improvement on prediction accuracy through the use of either different control parameters, different control values or both. Future research should focus on developing prediction models by either undersampling the majority classes (i.e., free flow and congested traffic regimes) or oversampling the minority class (i.e., speed at capacity regime) using TCA-DIDC and/or statistical techniques.

Another potential research could focus on predicting other traffic phenomena, such as queue lengths at signalized intersections, conflicts, etc.

References

1. Connected Vehicle Pilot Deployment Program, United States Department of Transportation. www.its.dot.gov/pilots/index.htm. Accessed November 8, 2016.
2. Sicular, S. Gartner's Big Data Definition Consists of Three Parts, Not to Be Confused with Three "V"s. <blogs.gartner.com/svetlana-sicular/gartners-big-data-definition-consists-of-three-parts-not-to-be-confused-with-three-vs/>. Accessed November 8, 2016.
3. Society of Automotive Engineers (SAE) standard J2735, Dedicated Short Range Communications (DSRC) Message Set Dictionary, November 2009.
4. Vasudevan, M., Negron, D., Feltz, M., Mallette, J., and Wunderlich, K. Predicting Congestion States from Basic Safety Messages Using Big Data Graph Analytics, *Transportation Research Record*, Vol. 2500, pp. 59—66, 2015.
5. Trajectory Conversion Algorithm (TCA) Software, Version 2.3.3. <https://itsforge.net/index.php/community/explore-applications#/38/67>, Accessed November 17, 2016.
6. Trajectory Converter Analysis Dynamic Interrogative Data Capture (TCA-DIDC), Version 2.4. <https://itsforge.net/index.php/community/explore-applications#/38/104>, Accessed November 17, 2016.
7. Hunter, T., Das, T., Zaharia, M., Abbeel, P., and Bayen, A. Large-Scale Estimation in Cyberphysical Systems Using Streaming Data: A Case Study With Arterial Traffic Estimation, *IEEE Transactions on Automation Science and Engineering*, Vol. 10, No. 4, October 2013.
8. Work, D. Real-time estimation of distributed parameters systems: Application to traffic monitoring, PhD thesis, University of California, Berkeley, 2010.
9. Hofleitner, A., Herring, R., Abbeel, P., and Bayen, A. Learning the Dynamics of Arterial Travel From Probe Data Using a Dynamic Bayesian Network, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 13, No. 4, December 2012.
10. Hofleitner, A., Herring, R., and Bayen, A. Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning, *Transportation Research: An International Journal, Part B: Methodological*, Vol. 46B, Issue 1, January 2012.
11. Herring, R. Real-Time Traffic Modeling and Estimation with Streaming Probe Data using Machine Learning, PhD thesis, University of California, Berkeley, 2010.
12. Elhenawy, M., Rakha, H., and Chen, H. An Automatic Traffic Congestion Identification Algorithm based on Mixture of Linear Regressions, Virginia Tech Transportation Institute, 3500 Transportation Research Plaza, Blacksburg, VA 2406.
13. Wunderlich, K., Vasudevan, M., and Wang, P. TAT Volume III: Guidelines for Applying Traffic Microsimulation Modeling Software (March 2016 Update), Report# FHWA-HOP-16-070, Prepared by Noblis for FHWA, August 2016.

14. Deurbrouck, T., Larkin, J., and K. Wunderlich. Trajectory Conversion Algorithm (TCA) Software, Version 2: Concept of Operations, Prepared by Noblis for U.S.DOT, November 2013.
15. Microsoft Azure: Cloud Computing Platform and Services. <https://azure.microsoft.com/en-us/>, Accessed November 17, 2016.
16. HDInsight – Hadoop, Spark and R Solutions for the Cloud, MS Azure. <https://azure.microsoft.com/en-us/services/hdinsight/>, Accessed November 17, 2016.
17. Apache Hadoop Software, Version 2.7.1. <http://hadoop.apache.org/docs/r2.7.1/>, Accessed November 17, 2016.
18. Apache Spark Software, Version 1.6.1. <http://spark.apache.org/releases/spark-release-1-6-1.html>, Accessed November 17, 2016.
19. Raiche, G. nFactors: an R package for parallel analysis and nongraphical solutions to the Cattell scree test. R package version 2.3.3., 2010.
20. Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. A Limited Memory Algorithm for Bound Constrained Optimization. SIAM Journal on Scientific Computing, 16(5), 1190-1208, 1995.
21. Sarkar, Deepayan. Lattice: Multivariate Data Visualization with R. Springer, New York. ISBN 978-0-387-75968-5, 2008.

APPENDIX A: Prediction Results – Validation Data

The tables in Appendix A displays the prediction results using validation data by operational condition, market penetration, and communication strategy. Precision, recall, and F1 score are reported for each prediction algorithm by traffic regime index ('Index').

Table A-1: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Demand Operational Condition (OC #1) and DSRC-20 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.6751	0.9960	0.8047
	2	0	0	0
	3	0.5385	0.0134	0.0262
Decision Trees (6 PCA, Entropy)	1	0.7747	0.9900	0.8692
	2	0	0	0
	3	0.9433	0.4981	0.6520
Random Forest (6 PCA, Entropy, 10 trees)	1	0.7740	0.9956	0.8710
	2	0	0	0
	3	0.9744	0.4889	0.6512
Logistic Regression (8 PCA)	1	0.6766	0.9944	0.8052
	2	0	0	0
	3	0.5597	0.0425	0.0425
Decision Trees (8 PCA, Entropy)	1	0.7747	0.9900	0.8692
	2	0	0	0
	3	0.9433	0.4981	0.6520
Random Forest (8 PCA, Entropy, 10 trees)	1	0.7844	0.9944	0.8770
	2	0	0	0
	3	0.9701	0.5300	0.6855
Logistic Regression (11 PCA)	1	0.8929	0.9988	0.9429
	2	0	0	0
	3	0.9940	0.9281	0.9599
Decision Trees (11 PCA, Entropy)	1	0.8913	0.9830	0.9349
	2	0.1776	0.0658	0.0960
	3	0.9975	0.8841	0.9374
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8882	0.9998	0.9407
	2	0	0	0
	3	0.9983	0.9084	0.9512
Random Forest (11 PCA, Gini, 10 trees)	1	0.8838	0.9995	0.9381
	2	0	0	0
	3	0.9982	0.9010	0.9471
Random Forest (11 PCA, Gini, 250 trees)	1	0.8838	0.9995	0.9381
	2	0	0	0
	3	0.9982	0.9010	0.9471

Prediction Algorithm	Index	Precision	Recall	F1 Score
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8845	0.9999	0.9387
	2	0	0	0
	3	0.9995	0.9030	0.9488

Table A-2: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Demand Operational Condition (OC #1) and DSRC-75 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.6802	0.9960	0.8083
	2	0	0	0
	3	0.5155	0.0132	0.0257
Decision Trees (6 PCA, Entropy)	1	0.7834	0.9920	0.8754
	2	0	0	0
	3	0.9530	0.5222	0.6747
Random Forest (6 PCA, Entropy, 10 trees)	1	0.7889	0.9944	0.8798
	2	0	0	0
	3	0.9656	0.5435	0.6955
Logistic Regression (8 PCA)	1	0.6856	0.9929	0.8111
	2	0	0	0
	3	0.6391	0.0433	0.0811
Decision Trees (8 PCA, Entropy)	1	0.7835	0.9920	0.8755
	2	0	0	0
	3	0.9530	0.5224	0.6748
Random Forest (8 PCA, Entropy, 10 trees)	1	0.7901	0.9943	0.8805
	2	0	0	0
	3	0.9636	0.5481	0.6987
Logistic Regression (11 PCA)	1	0.9061	0.9985	0.9501
	2	0.3333	0.0004	0.0007
	3	0.9862	0.9671	0.9766
Decision Trees (11 PCA, Entropy)	1	0.8998	0.9604	0.9291
	2	0.1565	0.0940	0.1175
	3	0.9865	0.9098	0.9466
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8890	0.9990	0.9408
	2	0	0	0
	3	0.9943	0.9166	0.9539
Random Forest (11 PCA, Gini, 10 trees)	1	0.8867	0.9993	0.9397
	2	0	0	0
	3	0.9951	0.9105	0.9509
Random Forest (11 PCA, Entropy, 250 trees)	1	0.8886	0.9976	0.9400
	2	0	0	0
	3	0.9898	0.9170	0.9520
Random Forest (11 PCA, Gini, 250 trees)	1	0.8901	0.9986	0.9412
	2	0	0	0
	3	0.9926	0.9214	0.9557
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.8887	0.9979	0.9401
	2	0	0	0
	3	0.9901	0.9167	0.9520

Prediction Algorithm	Index	Precision	Recall	F1 Score
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8895	0.9988	0.9410
	2	0	0	0
	3	0.9926	0.9191	0.9544

Table A-3: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Demand Operational Condition (OC #1) and Cellular-20 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.6589	1	0.7944
	2	0	0	0
	3	0	0	0
Decision Trees (6 PCA, Entropy)	1	0.8438	0.9163	0.8786
	2	0	0	0
	3	0.7721	0.7674	0.7697
Random Forest (6 PCA, Entropy, 10 trees)	1	0.8450	0.9634	0.9003
	2	0	0	0
	3	0.8687	0.7557	0.8083
Logistic Regression (8 PCA)	1	0.6647	0.9888	0.7950
	2	0	0	0
	3	0.5887	0.0407	0.0762
Decision Trees (8 PCA, Entropy)	1	0.8247	0.9504	0.8831
	2	0	0	0
	3	0.8359	0.7037	0.7641
Random Forest (8 PCA, Entropy, 10 trees)	1	0.8247	0.9504	0.8831
	2	0	0	0
	3	0.8359	0.7037	0.7641
Logistic Regression (11 PCA)	1	0.9075	0.9947	0.9491
	2	0	0	0
	3	0.9825	0.9501	0.9660
Decision Trees (11 PCA, Entropy)	1	0.8833	0.9993	0.9377
	2	0.4492	0.0033	0.0066
	3	0.9970	0.8859	0.9382
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8994	0.9984	0.9463
	2	0	0	0
	3	0.9917	0.9309	0.9603
Random Forest (11 PCA, Gini, 10 trees)	1	0.8934	0.9959	0.9419
	2	0	0	0
	3	0.9862	0.9151	0.9493
Random Forest (11 PCA, Gini, 250 trees)	1	0.8932	0.9965	0.9420
	2	0	0	0
	3	0.9878	0.9146	0.9498
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8930	0.9967	0.9420
	2	0	0	0
	3	0.9884	0.9140	0.9498

Table A-4: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Demand Operational Condition (OC #1) and Cellular-75 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.6666	0.9871	0.7958
	2	0	0	0
	3	0.6378	0.0591	0.1082
Decision Trees (6 PCA, Entropy)	1	0.8044	0.9450	0.8690
	2	0	0	0
	3	0.7957	0.6329	0.7050
Random Forest (6 PCA, Entropy, 10 trees)	1	0.8004	0.9864	0.8837
	2	0	0	0
	3	0.9397	0.6247	0.7505
Logistic Regression (8 PCA)	1	0.6800	0.9868	0.8052
	2	0	0	0
	3	0.7813	0.1253	0.2160
Decision Trees (8 PCA, Entropy)	1	0.8044	0.9450	0.8690
	2	0	0	0
	3	0.7957	0.6329	0.7050
Random Forest (8 PCA, Entropy, 10 trees)	1	0.8029	0.9824	0.8836
	2	0	0	0
	3	0.9239	0.6302	0.7493
Logistic Regression (11 PCA)	1	0.9093	0.9979	0.9516
	2	0	0	0
	3	0.9879	0.9616	0.9746
Decision Trees (11 PCA, Entropy)	1	0.8951	0.9825	0.9368
	2	0.6641	0.0456	0.0853
	3	0.9527	0.9142	0.9331
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8889	0.9957	0.9393
	2	0	0	0
	3	0.9847	0.9084	0.9450
Random Forest (11 PCA, Gini, 10 trees)	1	0.8912	0.9971	0.9412
	2	0	0	0
	3	0.9890	0.9142	0.9501
Random Forest (11 PCA, Gini, 250 trees)	1	0.8918	0.9975	0.9417
	2	0	0	0
	3	0.9916	0.9173	0.9530
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8916	0.9982	0.9419
	2	0	0	0
	3	0.9934	0.9168	0.9535

Table A-5: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Visibility Operational Condition (OC #2) and DSRC-20 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0	0	0
	2	0.6653	0.8800	0.7577
	3	0.7092	0.5663	0.6297
Decision Trees (6 PCA, Entropy)	1	0.2321	0.0362	0.0626
	2	0.7629	0.9551	0.8482
	3	0.9403	0.8112	0.8710
Random Forest (6 PCA, Entropy, 10 trees)	1	0.1682	0.2345	0.1959
	2	0.7733	0.8036	0.7882
	3	0.9673	0.7853	0.8668
Logistic Regression (8 PCA)	1	0	0	0
	2	0.6451	0.8833	0.7457
	3	0.6992	0.5062	0.5873
Decision Trees (8 PCA, Entropy)	1	0.2321	0.0362	0.0626
	2	0.7629	0.9551	0.8482
	3	0.9403	0.8112	0.8710
Random Forest (8 PCA, Entropy, 10 trees)	1	0.1867	0.2920	0.2278
	2	0.7585	0.7425	0.7504
	3	0.8928	0.7615	0.8220
Logistic Regression (11 PCA)	1	0.0625	0.0003	0.0006
	2	0.8114	0.9990	0.8955
	3	0.9981	0.9419	0.9692
Decision Trees (11 PCA, Entropy)	1	0.4007	0.0629	0.1087
	2	0.7824	0.9790	0.8697
	3	0.9918	0.8565	0.9192
Random Forest (11 PCA, Entropy, 10 trees)	1	0.2220	0.3082	0.2581
	2	0.8276	0.7929	0.8099
	3	0.9935	0.9375	0.9647
Random Forest (11 PCA, Gini, 10 trees)	1	0.1729	0.4122	0.2436
	2	0.8148	0.6246	0.7071
	3	0.9918	0.9297	0.9597
Random Forest (11 PCA, Gini, 250 trees)	1	0.1729	0.4122	0.2436
	2	0.8148	0.6246	0.7071
	3	0.9918	0.9297	0.9597
Random Forest (11 PCA, Gini, 1000 trees)	1	0.1736	0.4141	0.2447
	2	0.8153	0.6232	0.7064
	3	0.9893	0.9304	0.9589

Table A-6: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Visibility Operational Condition (OC #2) and DSRC-75 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0	0	0
	2	0.6601	0.8926	0.7590
	3	0.7078	0.4964	0.5836
Decision Trees (6 PCA, Entropy)	1	0.1044	0.2324	0.1441
	2	0.8029	0.8214	0.8121
	3	0.9632	0.6030	0.7417
Random Forest (6 PCA, Entropy, 10 trees)	1	0.1526	0.2383	0.1861
	2	0.8078	0.8078	0.8078
	3	0.9471	0.8014	0.8682
Logistic Regression (8 PCA)	1	0	0	0
	2	0.6321	0.9656	0.7640
	3	0.8488	0.3530	0.4986
Decision Trees (8 PCA, Entropy)	1	0.1044	0.2324	0.1441
	2	0.8029	0.8214	0.8121
	3	0.9632	0.6030	0.7417
Random Forest (8 PCA, Entropy, 10 trees)	1	0.1626	0.2525	0.1978
	2	0.7716	0.8128	0.7917
	3	0.9589	0.7303	0.8291
Logistic Regression (11 PCA)	1	0	0	0
	2	0.8307	0.9997	0.9074
	3	0.9996	0.9426	0.9703
Decision Trees (11 PCA, Entropy)	1	0.1491	0.4070	0.2182
	2	0.8247	0.6076	0.6997
	3	0.9590	0.9142	0.9361
Random Forest (11 PCA, Entropy, 10 trees)	1	0.2358	0.2878	0.2592
	2	0.8463	0.8434	0.8449
	3	0.9875	0.9321	0.9590
Random Forest (11 PCA, Gini, 10 trees)	1	0.1867	0.3017	0.2306
	2	0.8434	0.7927	0.8173
	3	0.9954	0.9242	0.9585
Random Forest (11 PCA, Entropy, 250 trees)	1	0.1486	0.3973	0.2163
	2	0.8249	0.6247	0.7110
	3	0.9884	0.9246	0.9555
Random Forest (11 PCA, Gini, 250 trees)	1	0.1907	0.3298	0.2416
	2	0.8445	0.7752	0.8083
	3	0.9937	0.9274	0.9594
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.1484	0.3969	0.2161
	2	0.8253	0.6274	0.7129
	3	0.9933	0.9243	0.9576

Table A-7: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Visibility Operational Condition (OC #2) and Cellular-20 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0	0	0
	2	0.5415	0.6700	0.5989
	3	0.4290	0.4152	0.4220
Decision Trees (6 PCA, Entropy)	1	0.1471	0.2284	0.1789
	2	0.7808	0.7681	0.7744
	3	0.8873	0.7642	0.8212
Random Forest (6 PCA, Entropy, 10 trees)	1	0.1594	0.4198	0.2311
	2	0.7914	0.5224	0.6294
	3	0.8258	0.8159	0.8208
Logistic Regression (8 PCA)	1	0	0	0
	2	0.5859	0.7662	0.6640
	3	0.5599	0.4881	0.5215
Decision Trees (8 PCA, Entropy)	1	0.1408	0.1791	0.1577
	2	0.7724	0.7899	0.7811
	3	0.8709	0.7753	0.8203
Random Forest (8 PCA, Entropy, 10 trees)	1	0.1408	0.1791	0.1577
	2	0.7724	0.7899	0.7811
	3	0.8709	0.7753	0.8203
Logistic Regression (11 PCA)	1	0	0	0
	2	0.7939	0.9953	0.8833
	3	0.9912	0.9375	0.9636
Decision Trees (11 PCA, Entropy)	1	0.2057	0.8111	0.3282
	2	0.8464	0.3554	0.5006
	3	0.9935	0.9286	0.9600
Random Forest (11 PCA, Entropy, 10 trees)	1	0.1919	0.6101	0.2919
	2	0.8083	0.4659	0.5911
	3	0.9853	0.9293	0.9564
Random Forest (11 PCA, Gini, 10 trees)	1	0.1828	0.4166	0.2541
	2	0.7917	0.6173	0.6937
	3	0.9851	0.9142	0.9483
Random Forest (11 PCA, Gini, 250 trees)	1	0.1900	0.3458	0.2452
	2	0.7978	0.6953	0.7430
	3	0.9832	0.9207	0.9509
Random Forest (11 PCA, Gini, 1000 trees)	1	0.1893	0.3545	0.2468
	2	0.7981	0.6870	0.7384
	3	0.9832	0.9203	0.9507

Table A-8: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Visibility Operational Condition (OC #2) and Cellular-75 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0	0	0
	2	0.6264	0.7052	0.6635
	3	0.5658	0.6187	0.5911
Decision Trees (6 PCA, Entropy)	1	0.0812	0.0831	0.0821
	2	0.7362	0.9490	0.8292
	3	0.9791	0.5810	0.7292
Random Forest (6 PCA, Entropy, 10 trees)	1	0.1925	0.2950	0.2330
	2	0.7996	0.7152	0.7550
	3	0.8462	0.8506	0.8484
Logistic Regression (8 PCA)	1	0	0	0
	2	0.6193	0.7166	0.6644
	3	0.5713	0.5999	0.5853
Decision Trees (8 PCA, Entropy)	1	0.0812	0.0831	0.0821
	2	0.7362	0.9490	0.8292
	3	0.9791	0.5810	0.7292
Random Forest (8 PCA, Entropy, 10 trees)	1	0.1505	0.4880	0.2300
	2	0.7924	0.4724	0.5919
	3	0.8050	0.7762	0.7904
Logistic Regression (11 PCA)	1	0	0	0
	2	0.8121	0.9989	0.8958
	3	0.9976	0.9464	0.9713
Decision Trees (11 PCA, Entropy)	1	0.1926	0.2933	0.2325
	2	0.7925	0.7896	0.7910
	3	0.9937	0.8595	0.9217
Random Forest (11 PCA, Entropy, 10 trees)	1	0.1676	0.4184	0.2394
	2	0.8083	0.6141	0.6979
	3	0.9824	0.9201	0.9502
Random Forest (11 PCA, Gini, 10 trees)	1	0.1691	0.4138	0.2401
	2	0.8035	0.6200	0.6999
	3	0.9854	0.9155	0.9492
Random Forest (11 PCA, Gini, 250 trees)	1	0.1679	0.4005	0.2366
	2	0.8059	0.6317	0.7082
	3	0.9915	0.9206	0.9547
Random Forest (11 PCA, Gini, 1000 trees)	1	0.1679	0.3990	0.2364
	2	0.8069	0.6324	0.7091
	3	0.9907	0.9224	0.9553

Table A-9: Comparing Precision, Recall, F1 Score of Prediction Models for the Weather + Incidents Operational Condition (OC #3) and DSRC-20 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.6021	0.8918	0.7189
	2	0.7844	0.8807	0.8298
	3	0.8171	0.5294	0.6425
Decision Trees (6 PCA, Entropy)	1	0.5531	0.5629	0.5579
	2	0.8436	0.8083	0.8255
	3	0.6945	0.7372	0.7153
Random Forest (6 PCA, Entropy, 10 trees)	1	0.3799	0.8842	0.5314
	2	0.8609	0.7582	0.8063
	3	0.8745	0.6698	0.7586
Logistic Regression (8 PCA)	1	0.6112	0.8867	0.7236
	2	0.7655	0.8843	0.8206
	3	0.8132	0.4922	0.6132
Decision Trees (8 PCA, Entropy)	1	0.5529	0.5578	0.5554
	2	0.8436	0.8083	0.8255
	3	0.6935	0.7381	0.7151
Random Forest (8 PCA, Entropy, 10 trees)	1	0.2479	0.8986	0.3886
	2	0.8500	0.5406	0.6609
	3	0.9435	0.6939	0.7997
Logistic Regression (11 PCA)	1	0.5533	0.0351	0.0661
	2	0.7883	0.9827	0.8748
	3	0.9755	0.8834	0.9272
Decision Trees (11 PCA, Entropy)	1	0.5779	0.8896	0.7006
	2	0.8663	0.8777	0.8720
	3	0.9868	0.7944	0.8802
Random Forest (11 PCA, Entropy, 10 trees)	1	0.3514	0.8988	0.5053
	2	0.8869	0.6935	0.7784
	3	0.9920	0.8445	0.9123
Random Forest (11 PCA, Gini, 10 trees)	1	0.3580	0.8972	0.5118
	2	0.8860	0.7021	0.7834
	3	0.9924	0.8416	0.9108
Random Forest (11 PCA, Gini, 250 trees)	1	0.3580	0.8972	0.5118
	2	0.8860	0.7021	0.7834
	3	0.9924	0.8416	0.9108
Random Forest (11 PCA, Gini, 1000 trees)	1	0.3113	0.8562	0.4566
	2	0.8738	0.6504	0.7457
	3	0.9948	0.8421	0.9121

Table A-10: Comparing Precision, Recall, F1 Score of Prediction Models for the Weather + Incidents Operational Condition (OC #3) and DSRC-75 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.6262	0.9228	0.7461
	2	0.7950	0.8558	0.8243
	3	0.7965	0.5826	0.6730
Decision Trees (6 PCA, Entropy)	1	0.3048	0.9567	0.4623
	2	0.8774	0.6723	0.7613
	3	0.9859	0.6799	0.8048
Random Forest (6 PCA, Entropy, 10 trees)	1	0.3101	0.8759	0.4580
	2	0.8617	0.6401	0.7346
	3	0.9370	0.7681	0.8442
Logistic Regression (8 PCA)	1	0.6364	0.9044	0.7471
	2	0.7701	0.8910	0.8262
	3	0.8448	0.5259	0.6483
Decision Trees (8 PCA, Entropy)	1	0.3048	0.9567	0.4623
	2	0.8774	0.6723	0.7613
	3	0.9859	0.6799	0.8048
Random Forest (8 PCA, Entropy, 10 trees)	1	0.2888	0.8919	0.4363
	2	0.8451	0.5780	0.6865
	3	0.9120	0.7548	0.8260
Logistic Regression (11 PCA)	1	0.7872	0.9368	0.8555
	2	0.9120	0.9385	0.9251
	3	0.9770	0.8743	0.9228
Decision Trees (11 PCA, Entropy)	1	0.3171	0.8394	0.4603
	2	0.8652	0.6651	0.7521
	3	0.9807	0.8247	0.8959
Random Forest (11 PCA, Entropy, 10 trees)	1	0.3034	0.8669	0.4494
	2	0.8777	0.6278	0.7320
	3	0.9911	0.8450	0.9122
Random Forest (11 PCA, Gini, 10 trees)	1	0.3182	0.8314	0.4603
	2	0.8676	0.6711	0.7568
	3	0.9942	0.8394	0.9103
Random Forest (11 PCA, Entropy, 250 trees)	1	0.3271	0.8613	0.4742
	2	0.8720	0.6716	0.7588
	3	0.9911	0.8355	0.9067
Random Forest (11 PCA, Gini, 250 trees)	1	0.3275	0.8578	0.4740
	2	0.8737	0.6763	0.7624
	3	0.9938	0.8360	0.9081
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.3273	0.8530	0.4731
	2	0.8701	0.6752	0.7604
	3	0.9912	0.8350	0.9064
Random Forest (11 PCA, Gini, 1000 trees)	1	0.3328	0.8659	0.4808
	2	0.8747	0.6800	0.7651
	3	0.9939	0.8364	0.9084

Table A-11: Comparing Precision, Recall, F1 Score of Prediction Models for the Weather + Incidents Operational Condition (OC #3) and Cellular-20 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.5586	0.5748	0.5666
	2	0.6389	0.8510	0.7299
	3	0.6173	0.3186	0.4203
Decision Trees (6 PCA, Entropy)	1	0.3371	0.8681	0.4857
	2	0.8959	0.6588	0.7593
	3	0.8567	0.7867	0.8202
Random Forest (6 PCA, Entropy, 10 trees)	1	0.2694	0.8308	0.4069
	2	0.8800	0.4986	0.6366
	3	0.8077	0.8166	0.8121
Logistic Regression (8 PCA)	1	0	0	0
	2	0.6082	0.7280	0.6627
	3	0.6102	0.6167	0.6135
Decision Trees (8 PCA, Entropy)	1	0.3455	0.8681	0.4943
	2	0.8949	0.6571	0.7578
	3	0.8582	0.8051	0.8308
Random Forest (8 PCA, Entropy, 10 trees)	1	0.3455	0.8681	0.4943
	2	0.8949	0.6571	0.7578
	3	0.8582	0.8051	0.8308
Logistic Regression (11 PCA)	1	0.0020	0.0001	0.0002
	2	0.7923	0.9968	0.8829
	3	0.9981	0.9066	0.9502
Decision Trees (11 PCA, Entropy)	1	0.2558	0.9309	0.4014
	2	0.8621	0.4908	0.6255
	3	0.9981	0.8434	0.9143
Random Forest (11 PCA, Entropy, 10 trees)	1	0.2735	0.7919	0.4065
	2	0.8539	0.5979	0.7033
	3	0.9874	0.8638	0.9215
Random Forest (11 PCA, Gini, 10 trees)	1	0.2790	0.7834	0.4115
	2	0.8537	0.6220	0.7196
	3	0.9861	0.8479	0.9118
Random Forest (11 PCA, Gini, 250 trees)	1	0.2903	0.8103	0.4274
	2	0.8641	0.6301	0.7288
	3	0.9893	0.8545	0.9170
Random Forest (11 PCA, Gini, 1000 trees)	1	0.2912	0.8137	0.4289
	2	0.8642	0.6301	0.7288
	3	0.9896	0.8541	0.9169

Table A-12: Comparing Precision, Recall, F1 Score of Prediction Models for the Weather + Incidents Operational Condition (OC #3) and Cellular-75 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.6342	0.9028	0.7450
	2	0.7721	0.8395	0.8044
	3	0.8138	0.6189	0.7031
Decision Trees (6 PCA, Entropy)	1	0.4492	0.8692	0.5923
	2	0.8279	0.6260	0.7129
	3	0.6904	0.7420	0.7153
Random Forest (6 PCA, Entropy, 10 trees)	1	0.4326	0.8028	0.5622
	2	0.8604	0.7703	0.8128
	3	0.8876	0.8038	0.8436
Logistic Regression (8 PCA)	1	0.6448	0.9421	0.7656
	2	0.7817	0.8598	0.8189
	3	0.8278	0.6064	0.7000
Decision Trees (8 PCA, Entropy)	1	0.4492	0.8692	0.5923
	2	0.8279	0.6260	0.7129
	3	0.6904	0.7420	0.7153
Random Forest (8 PCA, Entropy, 10 trees)	1	0.2546	0.9094	0.3979
	2	0.9030	0.4843	0.6305
	3	0.8895	0.8190	0.8528
Logistic Regression (11 PCA)	1	0.7849	0.9251	0.8493
	2	0.9113	0.9402	0.9255
	3	0.9839	0.8915	0.9354
Decision Trees (11 PCA, Entropy)	1	0.3033	0.8828	0.4514
	2	0.8629	0.6430	0.7369
	3	0.9926	0.8100	0.8920
Random Forest (11 PCA, Entropy, 10 trees)	1	0.2815	0.8943	0.4282
	2	0.8862	0.5691	0.6931
	3	0.9677	0.8572	0.9091
Random Forest (11 PCA, Gini, 10 trees)	1	0.2789	0.8240	0.4168
	2	0.8637	0.6028	0.7100
	3	0.9817	0.8535	0.9131
Random Forest (11 PCA, Gini, 250 trees)	1	0.3014	0.8868	0.4499
	2	0.8827	0.6189	0.7276
	3	0.9857	0.8556	0.9161
Random Forest (11 PCA, Gini, 1000 trees)	1	0.3020	0.8780	0.4495
	2	0.8821	0.6245	0.7313
	3	0.9853	0.8558	0.9160

Table A-13: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #4) and DSRC-20 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.8688	0.9777	0.9201
	2	0	0	0
	3	0.9953	0.9810	0.9881
Decision Trees (6 PCA, Entropy)	1	0.8079	0.9880	0.8889
	2	0	0	0
	3	0.9976	0.9664	0.9817
Random Forest (6 PCA, Entropy, 10 trees)	1	0.8230	0.9803	0.8948
	2	0	0	0
	3	0.9963	0.9694	0.9827
Logistic Regression (8 PCA)	1	0.8536	0.9801	0.9125
	2	0	0	0
	3	0.9957	0.9775	0.9865
Decision Trees (8 PCA, Entropy)	1	0.8079	0.9880	0.8889
	2	0	0	0
	3	0.9976	0.9664	0.9817
Random Forest (8 PCA, Entropy, 10 trees)	1	0.8362	0.9798	0.9023
	2	0	0	0
	3	0.9962	0.9736	0.9848
Logistic Regression (11 PCA)	1	0.9468	0.9976	0.9715
	2	0	0	0
	3	0.9996	0.9977	0.9986
Decision Trees (11 PCA, Entropy)	1	0.9041	0.9994	0.9494
	2	0	0	0
	3	1	0.8672	0.9289
Random Forest (11 PCA, Entropy, 10 trees)	1	0.9245	1	0.9608
	2	0	0	0
	3	1	0.9915	0.9957
Random Forest (11 PCA, Gini, 10 trees)	1	0.9210	0.9837	0.9514
	2	0	0	0
	3	0.9969	0.9926	0.9947
Random Forest (11 PCA, Gini, 250 trees)	1	0.9210	0.9837	0.9514
	2	0	0	0
	3	0.9969	0.9926	0.9947
Random Forest (11 PCA, Gini, 1000 trees)	1	0.9205	0.9837	0.9511
	2	0	0	0
	3	0.9969	0.9925	0.9947

Table A-14: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #4) and DSRC-75 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.8113	0.9230	0.8636
	2	0	0	0
	3	0.9859	0.9696	0.9777
Decision Trees (6 PCA, Entropy)	1	0.6988	0.9908	0.8195
	2	0	0	0
	3	0.9981	0.9337	0.9648
Random Forest (6 PCA, Entropy, 10 trees)	1	0.8300	0.9247	0.8748
	2	0	0	0
	3	0.9861	0.9732	0.9796
Logistic Regression (8 PCA)	1	0.8051	0.9225	0.8598
	2	0	0	0
	3	0.9858	0.9681	0.9769
Decision Trees (8 PCA, Entropy)	1	0.6988	0.9908	0.8195
	2	0	0	0
	3	0.9981	0.9337	0.9648
Random Forest (8 PCA, Entropy, 10 trees)	1	0.8691	0.9115	0.8898
	2	0	0	0
	3	0.9836	0.9817	0.9826
Logistic Regression (11 PCA)	1	0.9372	0.9946	0.9650
	2	0	0	0
	3	0.9990	0.9955	0.9973
Decision Trees (11 PCA, Entropy)	1	0.7524	1	0.8587
	2	0.0135	0.0263	0.0179
	3	1	0.9369	0.9674
Random Forest (11 PCA, Entropy, 10 trees)	1	0.9077	0.9356	0.9215
	2	0	0	0
	3	0.9880	0.9853	0.9867
Random Forest (11 PCA, Gini, 10 trees)	1	0.8833	0.9355	0.9087
	2	0	0	0
	3	0.9884	0.9853	0.9869
Random Forest (11 PCA, Entropy, 250 trees)	1	0.8943	0.9355	0.9144
	2	0	0	0
	3	0.9885	0.9874	0.9879
Random Forest (11 PCA, Gini, 250 trees)	1	0.8962	0.9355	0.9154
	2	0	0	0
	3	0.9885	0.9880	0.9882
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.8957	0.9355	0.9152
	2	0	0	0
	3	0.9885	0.9874	0.9879
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8999	0.9355	0.9174
	2	0	0	0
	3	0.9885	0.9886	0.9885

Table A-15: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #4) and Cellular-20 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.7250	0.9706	0.8300
	2	0	0	0
	3	0.9945	0.9413	0.9672
Decision Trees (6 PCA, Entropy)	1	0.7950	0.9715	0.8745
	2	0	0	0
	3	0.9946	0.9624	0.9782
Random Forest (6 PCA, Entropy, 10 trees)	1	0.7454	0.9887	0.8500
	2	0	0	0
	3	0.9979	0.9480	0.9723
Logistic Regression (8 PCA)	1	0.8173	0.9617	0.8836
	2	0	0	0
	3	0.9930	0.9679	0.9803
Decision Trees (8 PCA, Entropy)	1	0.7308	0.9902	0.8409
	2	0	0	0
	3	0.9980	0.9374	0.9667
Random Forest (8 PCA, Entropy, 10 trees)	1	0.7308	0.9902	0.8409
	2	0	0	0
	3	0.9980	0.9374	0.9667
Logistic Regression (11 PCA)	1	0.9416	0.9962	0.9681
	2	0	0	0
	3	0.9992	0.9947	0.9970
Decision Trees (11 PCA, Entropy)	1	0.8832	1	0.9380
	2	0	0	0
	3	1	0.9832	0.9915
Random Forest (11 PCA, Entropy, 10 trees)	1	0.9014	1	0.9481
	2	0	0	0
	3	1	0.9869	0.9934
Random Forest (11 PCA, Gini, 10 trees)	1	0.7239	0.9782	0.8321
	2	0	0	0
	3	0.9959	0.9418	0.9681
Random Forest (11 PCA, Gini, 250 trees)	1	0.8668	0.9783	0.9192
	2	0	0	0
	3	0.9961	0.9799	0.9879
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8199	0.9789	0.8924
	2	0	0	0
	3	0.9961	0.9689	0.9823

Table A-16: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #4) and Cellular-75 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.9113	0.9814	0.9451
	2	0	0	0
	3	0.9967	0.9908	0.9937
Decision Trees (6 PCA, Entropy)	1	0.4484	0.9324	0.6056
	2	0	0	0
	3	0.9838	0.8213	0.8952
Random Forest (6 PCA, Entropy, 10 trees)	1	0.7436	0.9969	0.8518
	2	0	0	0
	3	0.9994	0.9504	0.9743
Logistic Regression (8 PCA)	1	0.9033	0.9781	0.9392
	2	0	0	0
	3	0.9961	0.9893	0.9927
Decision Trees (8 PCA, Entropy)	1	0.4484	0.9324	0.6056
	2	0	0	0
	3	0.9838	0.8213	0.8952
Random Forest (8 PCA, Entropy, 10 trees)	1	0.8870	0.9644	0.9241
	2	0	0	0
	3	0.9932	0.9855	0.9893
Logistic Regression (11 PCA)	1	0.9437	0.9962	0.9692
	2	0	0	0
	3	0.9993	0.9968	0.9981
Decision Trees (11 PCA, Entropy)	1	0.8420	1	0.9142
	2	0	0	0
	3	1	0.9127	0.9543
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8623	1	0.9260
	2	0	0	0
	3	1	0.9807	0.9902
Random Forest (11 PCA, Gini, 10 trees)	1	0.8356	1	0.9104
	2	0	0	0
	3	1	0.9744	0.9871
Random Forest (11 PCA, Gini, 250 trees)	1	0.8654	1	0.9278
	2	0	0	0
	3	1	0.9816	0.9907
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8660	1	0.9282
	2	0	0	0
	3	1	0.9818	0.9908

Table A-17: Comparing Precision, Recall, F1 Score of Prediction Models for the Bottleneck Trouble Operational Condition (OC #5) and DSRC-20 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.7039	0.9904	0.8229
	2	0	0	0
	3	0.8030	0.1109	0.1949
Decision Trees (6 PCA, Entropy)	1	0.8766	0.7010	0.7790
	2	0.0920	0.2916	0.1399
	3	0.5646	0.7042	0.6267
Random Forest (6 PCA, Entropy, 10 trees)	1	0.8518	0.9227	0.8858
	2	0.1173	0.1709	0.1391
	3	0.8426	0.6321	0.7223
Logistic Regression (8 PCA)	1	0.7212	0.9893	0.8343
	2	0	0	0
	3	0.8640	0.1918	0.3139
Decision Trees (8 PCA, Entropy)	1	0.8980	0.7010	0.7874
	2	0.0920	0.2916	0.1399
	3	0.5805	0.7503	0.6546
Random Forest (8 PCA, Entropy, 10 trees)	1	0.8667	0.9169	0.8911
	2	0.1089	0.1940	0.1395
	3	0.8560	0.6661	0.7492
Logistic Regression (11 PCA)	1	0.9519	0.9984	0.9746
	2	0	0	0
	3	0.9869	0.9765	0.9817
Decision Trees (11 PCA, Entropy)	1	0.9637	0.8066	0.8782
	2	0.0852	0.4873	0.1451
	3	0.9961	0.8863	0.9380
Random Forest (11 PCA, Entropy, 10 trees)	1	0.9567	0.9602	0.9584
	2	0.1716	0.2488	0.2031
	3	0.9962	0.9393	0.9669
Random Forest (11 PCA, Gini, 10 trees)	1	0.9532	0.9666	0.9599
	2	0.1924	0.2371	0.2124
	3	0.9943	0.9369	0.9648
Random Forest (11 PCA, Gini, 250 trees)	1	0.9532	0.9666	0.9599
	2	0.1924	0.2371	0.2124
	3	0.9943	0.9369	0.9648
Random Forest (11 PCA, Gini, 1000 trees)	1	0.9527	0.9694	0.9610
	2	0.2042	0.2402	0.2207
	3	0.9967	0.9368	0.9658

Table A-18: Comparing Precision, Recall, F1 Score of Prediction Models for the Bottleneck Trouble Operational Condition (OC #5) and DSRC-75 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.7029	0.9861	0.8208
	2	0	0	0
	3	0.7774	0.1284	0.2205
Decision Trees (6 PCA, Entropy)	1	0.8086	0.9215	0.8614
	2	0.0891	0.1707	0.1171
	3	0.9077	0.5269	0.6667
Random Forest (6 PCA, Entropy, 10 trees)	1	0.8492	0.8329	0.8410
	2	0.1462	0.2596	0.1870
	3	0.6735	0.6509	0.6620
Logistic Regression (8 PCA)	1	0.7317	0.9888	0.8410
	2	0	0	0
	3	0.8925	0.2545	0.3960
Decision Trees (8 PCA, Entropy)	1	0.8086	0.9215	0.8614
	2	0.0891	0.1707	0.1171
	3	0.9077	0.5269	0.6667
Random Forest (8 PCA, Entropy, 10 trees)	1	0.8739	0.8444	0.8589
	2	0.1253	0.2854	0.1742
	3	0.7209	0.6845	0.7022
Logistic Regression (11 PCA)	1	0.9503	0.9987	0.9739
	2	0	0	0
	3	0.9927	0.9772	0.9849
Decision Trees (11 PCA, Entropy)	1	0.9575	0.9279	0.9424
	2	0.1407	0.3414	0.1993
	3	0.9848	0.9111	0.9465
Random Forest (11 PCA, Entropy, 10 trees)	1	0.9534	0.9584	0.9559
	2	0.1794	0.2485	0.2084
	3	0.9898	0.9395	0.9640
Random Forest (11 PCA, Gini, 10 trees)	1	0.9547	0.9436	0.9491
	2	0.1528	0.2796	0.1976
	3	0.9945	0.9366	0.9647
Random Forest (11 PCA, Entropy, 250 trees)	1	0.9542	0.9598	0.9570
	2	0.1756	0.2664	0.2117
	3	0.9921	0.9258	0.9578
Random Forest (11 PCA, Gini, 250 trees)	1	0.9534	0.9608	0.9571
	2	0.1809	0.2656	0.2152
	3	0.9956	0.9297	0.9615
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.9541	0.9586	0.9563
	2	0.1709	0.2686	0.2089
	3	0.9920	0.9229	0.9562
Random Forest (11 PCA, Gini, 1000 trees)	1	0.9533	0.9647	0.9590
	2	0.1839	0.2539	0.2133
	3	0.9955	0.9289	0.9610

Table A-19: Comparing Precision, Recall, F1 Score of Prediction Models for the Bottleneck Trouble Operational Condition (OC #5) and Cellular-20 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.6614	0.9867	0.7919
	2	0	0	0
	3	0.4112	0.0204	0.0388
Decision Trees (6 PCA, Entropy)	1	0.8820	0.8955	0.8887
	2	0.0915	0.0862	0.0888
	3	0.7960	0.7743	0.7850
Random Forest (6 PCA, Entropy, 10 trees)	1	0.9213	0.8794	0.8999
	2	0.0951	0.1586	0.1189
	3	0.8230	0.8556	0.8390
Logistic Regression (8 PCA)	1	0.6946	0.9748	0.8112
	2	0	0	0
	3	0.7596	0.1787	0.2894
Decision Trees (8 PCA, Entropy)	1	0.8895	0.8611	0.8751
	2	0.0516	0.1058	0.0693
	3	0.8107	0.7941	0.8023
Random Forest (8 PCA, Entropy, 10 trees)	1	0.8895	0.8611	0.8751
	2	0.0516	0.1058	0.0693
	3	0.8107	0.7941	0.8023
Logistic Regression (11 PCA)	1	0.9468	0.9953	0.9705
	2	0	0	0
	3	0.9860	0.9613	0.9735
Decision Trees (11 PCA, Entropy)	1	0.9469	0.9806	0.9635
	2	0.1251	0.1252	0.1252
	3	0.9963	0.9219	0.9576
Random Forest (11 PCA, Entropy, 10 trees)	1	0.9558	0.9734	0.9645
	2	0.1706	0.2175	0.1912
	3	0.9977	0.9365	0.9661
Random Forest (11 PCA, Gini, 10 trees)	1	0.9498	0.9617	0.9557
	2	0.1323	0.2031	0.1602
	3	0.9937	0.9234	0.9573
Random Forest (11 PCA, Gini, 250 trees)	1	0.9546	0.9556	0.9551
	2	0.1537	0.2630	0.1940
	3	0.9944	0.9335	0.9630
Random Forest (11 PCA, Gini, 1000 trees)	1	0.9534	0.9601	0.9568
	2	0.1512	0.2357	0.1843
	3	0.9943	0.9335	0.9629

Table A-20: Comparing Precision, Recall, F1 Score of Prediction Models for the Bottleneck Trouble Operational Condition (OC #5) and Cellular-75 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.7125	0.9624	0.8188
	2	0	0	0
	3	0.7708	0.2799	0.4107
Decision Trees (6 PCA, Entropy)	1	0.8265	0.8275	0.8270
	2	0.0500	0.1576	0.0759
	3	0.8136	0.6677	0.7335
Random Forest (6 PCA, Entropy, 10 trees)	1	0.8711	0.7972	0.8325
	2	0.0595	0.2428	0.0956
	3	0.8116	0.7512	0.7802
Logistic Regression (8 PCA)	1	0.7238	0.9816	0.8332
	2	0	0	0
	3	0.8754	0.3082	0.4559
Decision Trees (8 PCA, Entropy)	1	0.8265	0.8275	0.8270
	2	0.0500	0.1576	0.0759
	3	0.8136	0.6677	0.7335
Random Forest (8 PCA, Entropy, 10 trees)	1	0.8766	0.8973	0.8868
	2	0.1292	0.1726	0.1478
	3	0.8219	0.7598	0.7896
Logistic Regression (11 PCA)	1	0.9539	0.9980	0.9755
	2	0	0	0
	3	0.9905	0.9783	0.9844
Decision Trees (11 PCA, Entropy)	1	0.9315	0.8898	0.9101
	2	0.0602	0.2140	0.0939
	3	0.9947	0.8768	0.9321
Random Forest (11 PCA, Entropy, 10 trees)	1	0.9484	0.9282	0.9382
	2	0.0969	0.2226	0.1350
	3	0.9906	0.9304	0.9596
Random Forest (11 PCA, Gini, 10 trees)	1	0.9493	0.9315	0.9403
	2	0.1066	0.2334	0.1463
	3	0.9901	0.9312	0.9598
Random Forest (11 PCA, Gini, 250 trees)	1	0.9479	0.9540	0.9510
	2	0.1269	0.1959	0.1541
	3	0.9914	0.9339	0.9618
Random Forest (11 PCA, Gini, 1000 trees)	1	0.9483	0.9475	0.9479
	2	0.1210	0.2071	0.1527
	3	0.9907	0.9343	0.9617

Table A-21: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #6) and DSRC-20 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.8112	0.9226	0.8633
	2	0	0	0
	3	0.7865	0.6995	0.7404
Decision Trees (6 PCA, Entropy)	1	0.8423	0.6864	0.7564
	2	0.0757	0.1335	0.0966
	3	0.6395	0.8075	0.7137
Random Forest (6 PCA, Entropy, 10 trees)	1	0.8565	0.9603	0.9054
	2	0.1333	0.0013	0.0026
	3	0.8781	0.8122	0.8439
Logistic Regression (8 PCA)	1	0.8393	0.9339	0.8841
	2	0	0	0
	3	0.8093	0.7638	0.7859
Decision Trees (8 PCA, Entropy)	1	0.8423	0.6864	0.7564
	2	0.0757	0.1335	0.0966
	3	0.6395	0.8075	0.7137
Random Forest (8 PCA, Entropy, 10 trees)	1	0.8576	0.8944	0.8756
	2	0.2784	0.0160	0.0302
	3	0.7595	0.8244	0.7906
Logistic Regression (11 PCA)	1	0.8913	0.9985	0.9419
	2	0	0	0
	3	0.9944	0.9221	0.9569
Decision Trees (11 PCA, Entropy)	1	0.9194	0.7536	0.8283
	2	0.0999	0.3872	0.1588
	3	0.9931	0.8357	0.9076
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8949	0.9940	0.9418
	2	0.1474	0.0241	0.0414
	3	0.9974	0.9158	0.9548
Random Forest (11 PCA, Gini, 10 trees)	1	0.8896	0.9964	0.9400
	2	0.1438	0.0095	0.0178
	3	0.9949	0.9102	0.9507
Random Forest (11 PCA, Gini, 250 trees)	1	0.8896	0.9964	0.9400
	2	0.1438	0.0095	0.0178
	3	0.9949	0.9102	0.9507
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8884	0.9987	0.9404
	2	0	0	0
	3	0.9967	0.9135	0.9533

Table A-22: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #6) and DSRC-75 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.8011	0.8982	0.8469
	2	0	0	0
	3	0.7340	0.6900	0.7113
Decision Trees (6 PCA, Entropy)	1	0.8534	0.8511	0.8522
	2	0.0978	0.2146	0.1344
	3	0.9769	0.7368	0.8400
Random Forest (6 PCA, Entropy, 10 trees)	1	0.8927	0.9561	0.9233
	2	0	0	0
	3	0.8612	0.8965	0.8785
Logistic Regression (8 PCA)	1	0.8439	0.9442	0.8912
	2	0	0	0
	3	0.8257	0.7804	0.8024
Decision Trees (8 PCA, Entropy)	1	0.8534	0.8511	0.8522
	2	0.0978	0.2146	0.1344
	3	0.9769	0.7368	0.8400
Random Forest (8 PCA, Entropy, 10 trees)	1	0.8891	0.8800	0.8845
	2	0	0	0
	3	0.7372	0.9014	0.8111
Logistic Regression (11 PCA)	1	0.8874	0.9993	0.9401
	2	0	0	0
	3	0.9981	0.9273	0.9614
Decision Trees (11 PCA, Entropy)	1	0.8831	0.9850	0.9313
	2	0.1824	0.0305	0.0522
	3	0.9771	0.8969	0.9353
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8887	0.9986	0.9404
	2	0	0	0
	3	0.9951	0.9181	0.9551
Random Forest (11 PCA, Gini, 10 trees)	1	0.8845	0.9989	0.9382
	2	0	0	0
	3	0.9966	0.9185	0.9559
Random Forest (11 PCA, Entropy, 250 trees)	1	0.8859	0.9970	0.9382
	2	0	0	0
	3	0.9893	0.9205	0.9537
Random Forest (11 PCA, Gini, 250 trees)	1	0.8843	0.9985	0.9379
	2	0	0	0
	3	0.9947	0.9175	0.9545
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.8864	0.9972	0.9385
	2	0	0	0
	3	0.9897	0.9219	0.9546
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8847	0.9987	0.9382
	2	0	0	0
	3	0.9957	0.9188	0.9557

Table A-23: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #6) and Cellular-20 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.8033	0.9303	0.8622
	2	0	0	0
	3	0.8159	0.6895	0.7474
Decision Trees (6 PCA, Entropy)	1	0.8640	0.9165	0.8895
	2	0.0890	0.0121	0.0212
	3	0.8149	0.8329	0.8238
Random Forest (6 PCA, Entropy, 10 trees)	1	0.8648	0.9656	0.9124
	2	0	0	0
	3	0.9026	0.8392	0.8697
Logistic Regression (8 PCA)	1	0.8229	0.9356	0.8756
	2	0	0	0
	3	0.8219	0.7305	0.7735
Decision Trees (8 PCA, Entropy)	1	0.8796	0.8798	0.8797
	2	0.0866	0.0928	0.0896
	3	0.8532	0.8424	0.8478
Random Forest (8 PCA, Entropy, 10 trees)	1	0.8796	0.8798	0.8797
	2	0.0866	0.0928	0.0896
	3	0.8532	0.8424	0.8478
Logistic Regression (11 PCA)	1	0.9046	0.9941	0.9472
	2	0	0	0
	3	0.9836	0.9513	0.9672
Decision Trees (11 PCA, Entropy)	1	0.8857	0.9972	0.9382
	2	0.6331	0.0061	0.0122
	3	0.9917	0.9012	0.9443
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8941	0.9962	0.9424
	2	0.1851	0.0127	0.0238
	3	0.9976	0.9207	0.9576
Random Forest (11 PCA, Gini, 10 trees)	1	0.8888	0.9963	0.9395
	2	0	0	0
	3	0.9904	0.9117	0.9494
Random Forest (11 PCA, Gini, 250 trees)	1	0.8907	0.9965	0.9406
	2	0.0952	0.0001	0.0003
	3	0.9911	0.9165	0.9524
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8903	0.9967	0.9405
	2	0	0	0
	3	0.9914	0.9156	0.9520

Table A-24: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #6) and Cellular-75 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.6861	0.9056	0.7807
	2	0	0	0
	3	0.5915	0.3057	0.4031
Decision Trees (6 PCA, Entropy)	1	0.8702	0.8906	0.8803
	2	0.0747	0.0393	0.0515
	3	0.8207	0.8477	0.8340
Random Forest (6 PCA, Entropy, 10 trees)	1	0.8703	0.9054	0.8875
	2	0.1522	0.0014	0.0027
	3	0.7899	0.8582	0.8227
Logistic Regression (8 PCA)	1	0.8109	0.8605	0.8349
	2	0	0	0
	3	0.6945	0.7261	0.7099
Decision Trees (8 PCA, Entropy)	1	0.8702	0.8906	0.8803
	2	0.0747	0.0393	0.0515
	3	0.8207	0.8477	0.8340
Random Forest (8 PCA, Entropy, 10 trees)	1	0.8754	0.9213	0.8978
	2	0.2874	0.0022	0.0044
	3	0.8172	0.8675	0.8416
Logistic Regression (11 PCA)	1	0.8948	0.9985	0.9438
	2	0	0	0
	3	0.9951	0.9290	0.9609
Decision Trees (11 PCA, Entropy)	1	0.8794	0.8982	0.8887
	2	0.0825	0.1243	0.0992
	3	0.9924	0.8633	0.9234
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8856	0.9951	0.9372
	2	0.1538	0.0006	0.0012
	3	0.9871	0.9058	0.9447
Random Forest (11 PCA, Gini, 10 trees)	1	0.8871	0.9950	0.9380
	2	0.0213	0.0001	0.0001
	3	0.9873	0.9096	0.9469
Random Forest (11 PCA, Gini, 250 trees)	1	0.8876	0.9966	0.9389
	2	0	0	0
	3	0.9906	0.9108	0.9490
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8877	0.9960	0.9388
	2	0	0	0
	3	0.9893	0.9113	0.9487

Table A-25: Comparing Precision, Recall, F1 Score of Prediction Models Overall (All Operational Conditions) and DSRC-20 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.5546	0.7208	0.6268
	2	0.4501	0.1260	0.1969
	3	0.3897	0.4213	0.4049
Decision Trees (6 PCA, Entropy)	1	0.8035	0.7591	0.7807
	2	0.6909	0.7704	0.7285
	3	0.7496	0.7473	0.7485
Random Forest (6 PCA, Entropy, 10 trees)	1	0.7549	0.9285	0.8328
	2	0.7857	0.6791	0.7285
	3	0.9165	0.7173	0.8048
Logistic Regression (8 PCA)	1	0.5444	0.6946	0.6104
	2	0.3584	0.1206	0.1804
	3	0.4137	0.4448	0.4287
Decision Trees (8 PCA, Entropy)	1	0.8086	0.7588	0.7830
	2	0.6909	0.7704	0.7285
	3	0.7515	0.7556	0.7535
Random Forest (8 PCA, Entropy, 10 trees)	1	0.7200	0.9106	0.8042
	2	0.7520	0.9330	0.8328
	3	0.8914	0.7329	0.8044
Logistic Regression (11 PCA)	1	0.8327	0.9162	0.8725
	2	0.7198	0.6428	0.6791
	3	0.9967	0.9343	0.9645
Decision Trees (11 PCA, Entropy)	1	0.8904	0.8254	0.8567
	2	0.6137	0.8344	0.7073
	3	0.9939	0.8516	0.9173
Decision Trees (11 PCA, Gini)	1	0.7822	0.9294	0.8495
	2	0.7539	0.5751	0.6525
	3	0.9751	0.8842	0.9274
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8217	0.9547	0.8832
	2	0.8155	0.6471	0.7216
	3	0.9961	0.9181	0.9555
Random Forest (11 PCA, Gini, 10 trees)	1	0.7979	0.9610	0.8719
	2	0.8203	0.5847	0.6828
	3	0.9946	0.9138	0.9525
Random Forest (11 PCA, Entropy, 250 trees)	1	0.7904	0.9594	0.8667
	2	0.8156	0.5627	0.6659
	3	0.9959	0.9177	0.9552
Random Forest (11 PCA, Gini, 250 trees)	1	0.7979	0.9610	0.8719
	2	0.8203	0.5847	0.6828
	3	0.9946	0.9138	0.9525
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.7854	0.9595	0.8637
	2	0.8145	0.5481	0.6553
	3	0.9957	0.9184	0.9555
Random Forest (11 PCA, Gini, 1000 trees)	1	0.7885	0.9607	0.8661
	2	0.8165	0.5590	0.6636
	3	0.9956	0.9149	0.9536

Table A-26: Comparing Precision, Recall, F1 Score of Prediction Models Overall (All Operational Conditions) and DSRC-75 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.5559	0.7161	0.6259
	2	0.4901	0.1080	0.1770
	3	0.4005	0.4574	0.4271
Decision Trees (6 PCA, Entropy)	1	0.7072	0.9034	0.7934
	2	0.7052	0.6534	0.6783
	3	0.9676	0.9676	0.7865
Random Forest (6 PCA, Entropy, 10 trees)	1	0.7483	0.9005	0.8173
	2	0.7913	0.6260	0.6990
	3	0.8822	0.8822	0.8218
Logistic Regression (8 PCA)	1	0.5443	0.6879	0.6078
	2	0.3963	0.1269	0.1922
	3	0.4336	0.4801	0.4557
Decision Trees (8 PCA, Entropy)	1	0.7072	0.9034	0.7934
	2	0.7052	0.6534	0.6783
	3	0.9676	0.6625	0.7865
Random Forest (8 PCA, Entropy, 10 trees)	1	0.7447	0.8837	0.8083
	2	0.7510	0.5993	0.6667
	3	0.8590	0.7637	0.8086
Logistic Regression (11 PCA)	1	0.6454	0.8734	0.7423
	2	0.3779	0.1554	0.2202
	3	0.9996	0.9288	0.9629
Decision Trees (11 PCA, Entropy)	1	0.7862	0.9335	0.8536
	2	0.7442	0.5674	0.6439
	3	0.9810	0.8953	0.9362
Decision Trees (11 PCA, Gini)	1	0.7844	0.9420	0.8560
	2	0.7732	0.5672	0.6544
	3	0.9825	0.8978	0.9383
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8139	0.9544	0.8786
	2	0.8264	0.6338	0.7174
	3	0.9910	0.9188	0.9535
Random Forest (11 PCA, Gini, 10 trees)	1	0.8101	0.9491	0.8741
	2	0.8107	0.6335	0.7112
	3	0.9941	0.9150	0.9529
Random Forest (11 PCA, Entropy, 250 trees)	1	0.7911	0.9578	0.8665
	2	0.8110	0.5681	0.6681
	3	0.9899	0.9140	0.9504
Random Forest (11 PCA, Gini, 250 trees)	1	0.8106	0.9564	0.8775
	2	0.8237	0.6288	0.7132
	3	0.9933	0.9155	0.9528
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.7917	0.9571	0.8666
	2	0.8088	0.5709	0.6693
	3	0.9909	0.9135	0.9506
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8011	0.9603	0.8735
	2	0.8234	0.5990	0.6935
	3	0.9936	0.9151	0.9527

Table A-27: Comparing Precision, Recall, F1 Score of Prediction Models Overall (All Operational Conditions) and Cellular-20 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.5540	0.7445	0.6353
	2	0.2132	0.0236	0.0424
	3	0.4391	0.4591	0.4489
Decision Trees (6 PCA, Entropy)	1	0.7778	0.8904	0.8302
	2	0.8131	0.5976	0.6889
	3	0.8443	0.8103	0.8270
Random Forest (6 PCA, Entropy, 10 trees)	1	0.7557	0.9173	0.8287
	2	0.7816	0.4374	0.5609
	3	0.8636	0.8352	0.8492
Logistic Regression (8 PCA)	1	0.5321	0.8551	0.6560
	2	0.4830	0.0262	0.0496
	3	0.5101	0.3784	0.4345
Decision Trees (8 PCA, Entropy)	1	0.7809	0.8788	0.8270
	2	0.7320	0.6079	0.6642
	3	0.8649	0.8066	0.8347
Random Forest (8 PCA, Entropy, 10 trees)	1	0.7362	0.9178	0.8170
	2	0.8014	0.3791	0.5147
	3	0.8434	0.8201	0.8316
Logistic Regression (11 PCA)	1	0.8893	0.9334	0.9108
	2	0.7679	0.7211	0.7437
	3	0.9904	0.9606	0.9753
Decision Trees (11 PCA, Entropy)	1	0.7589	0.9845	0.8571
	2	0.8121	0.3843	0.5217
	3	0.9962	0.9044	0.9481
Decision Trees (11 PCA, Gini)	1	0.8007	0.9589	0.8727
	2	0.7845	0.5350	0.6362
	3	0.9911	0.9122	0.9500
Random Forest (11 PCA, Entropy, 10 trees)	1	0.7915	0.9696	0.8715
	2	0.7950	0.4796	0.5983
	3	0.9935	0.9233	0.9571
Random Forest (11 PCA, Gini, 10 trees)	1	0.7975	0.9593	0.8709
	2	0.7858	0.5359	0.6372
	3	0.9896	0.9052	0.9455
Random Forest (11 PCA, Entropy, 250 trees)	1	0.8027	0.9618	0.8751
	2	0.7953	0.5426	0.6451
	3	0.9913	0.9125	0.9502
Random Forest (11 PCA, Gini, 250 trees)	1	0.8097	0.9570	0.8772
	2	0.7910	0.5648	0.6591
	3	0.9906	0.9152	0.9514
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.8010	0.9592	0.8730
	2	0.7867	0.5381	0.6390
	3	0.9914	0.9126	0.9504
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8080	0.9590	0.8770
	2	0.7951	0.5616	0.6582
	3	0.9908	0.9134	0.9505

Table A-28: Comparing Precision, Recall, F1 Score of Prediction Models Overall (All Operational Conditions) and Cellular-75 Scenario for Validation Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.5785	0.5756	0.5770
	2	0.4112	0.0828	0.1379
	3	0.4452	0.6549	0.5301
Decision Trees (6 PCA, Entropy)	1	0.7619	0.8576	0.8069
	2	0.6972	0.6778	0.6873
	3	0.8217	0.7128	0.7634
Random Forest (6 PCA, Entropy, 10 trees)	1	0.7756	0.8710	0.8205
	2	0.6439	0.6564	0.6974
	3	0.8692	0.8023	0.8344
Logistic Regression (8 PCA)	1	0.5957	0.6839	0.6368
	2	0.3785	0.0956	0.1527
	3	0.4890	0.6160	0.5452
Decision Trees (8 PCA, Entropy)	1	0.7619	0.8576	0.8069
	2	0.6972	0.6778	0.6873
	3	0.8217	0.7128	0.7634
Random Forest (8 PCA, Entropy, 10 trees)	1	0.9233	0.7243	0.8118
	2	0.8044	0.4223	0.5539
	3	0.8683	0.8016	0.8336
Logistic Regression (11 PCA)	1	0.6475	0.8776	0.7452
	2	0.3501	0.1370	0.1969
	3	0.9989	0.9320	0.9643
Decision Trees (11 PCA, Entropy)	1	0.7920	0.9001	0.8426
	2	0.6808	0.6317	0.6553
	3	0.9873	0.8689	0.9243
Decision Trees (11 PCA, Gini)	1	0.8505	0.8559	0.8532
	2	0.6498	0.8042	0.7188
	3	0.9920	0.8474	0.9140
Random Forest (11 PCA, Entropy, 10 trees)	1	0.7730	0.9486	0.8519
	2	0.7833	0.5195	0.6247
	3	0.9846	0.9131	0.9475
Random Forest (11 PCA, Gini, 10 trees)	1	0.7784	0.9463	0.8542
	2	0.7767	0.5382	0.6358
	3	0.9885	0.9124	0.9489
Random Forest (11 PCA, Entropy, 250 trees)	1	0.7834	0.9546	0.8606
	2	0.8042	0.5464	0.6507
	3	0.9893	0.9177	0.9522
Random Forest (11 PCA, Gini, 250 trees)	1	0.7838	0.9564	0.8615
	2	0.8051	0.5496	0.6532
	3	0.9914	0.9158	0.9521
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.7917	0.9571	0.8666
	2	0.8088	0.5709	0.6693
	3	0.9909	0.9135	0.9506
Random Forest (11 PCA, Gini, 1000 trees)	1	0.7845	0.9539	0.8610
	2	0.8010	0.5529	0.6542
	3	0.9912	0.9162	0.9522

APPENDIX B: Prediction Results – Test Data

The tables in Appendix B displays the prediction results using test data by operational condition, market penetration, and communication strategy. Precision, recall, and F1 score are reported for each prediction algorithm by traffic regime index ('Index').

Table B-1: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Demand Operational Condition (OC #1) and DSRC-20 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.8941	0.9816	0.9358
	2	0.1888	0.0635	0.0950
	3	0.9841	0.8893	0.9343
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8868	0.9987	0.9393
	2	0	0	0
	3	0.9923	0.9158	0.9525
Random Forest (11 PCA, Gini, 10 trees)	1	0.8883	0.9995	0.9406
	2	0	0	0
	3	0.9984	0.9109	0.9526
Random Forest (11 PCA, Gini, 250 trees)	1	0.8861	0.9996	0.9394
	2	0	0	0
	3	0.9987	0.9040	0.9490
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8862	0.9996	0.9395
	2	0	0	0
	3	0.9987	0.9045	0.9493

Table B-2: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Demand Operational Condition (OC #1) and DSRC-75 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.8977	0.9574	0.9266
	2	0.1643	0.1015	0.1255
	3	0.9861	0.9105	0.9468
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8868	0.9983	0.9393
	2	0	0	0
	3	0.9923	0.9158	0.9525
Random Forest (11 PCA, Gini, 10 trees)	1	0.8856	0.9979	0.9384
	2	0	0	0
	3	0.9911	0.9117	0.9497
Random Forest (11 PCA, Entropy, 250 trees)	1	0.8872	0.9977	0.9393
	2	0	0	0
	3	0.9906	0.9172	0.9525

Prediction Algorithm	Index	Precision	Recall	F1 Score
Random Forest (11 PCA, Gini, 250 trees)	1	0.8901	0.9986	0.9412
	2	0	0	0
	3	0.9926	0.9214	0.9557
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.8866	0.9981	0.9391
	2	0	0	0
	3	0.9915	0.9151	0.9518
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8861	0.9988	0.9391
	2	0	0	0
	3	0.9939	0.9136	0.9520

Table B-3: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Demand Operational Condition (OC #1) and Cellular-20 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.8941	0.9980	0.9432
	2	0.3231	0.0130	0.0250
	3	0.9973	0.9512	0.9512
Random Forest (11 PCA, Entropy, 10 trees)	1	0.9018	0.9985	0.9477
	2	20.3571	0.0004	0.0008
	3	0.9927	0.9328	0.9618
Random Forest (11 PCA, Gini, 10 trees)	1	0.8951	0.9959	0.9428
	2	0	0	0
	3	0.9861	0.9144	0.9489
Random Forest (11 PCA, Gini, 250 trees)	1	0.8954	0.9965	0.9433
	2	0	0	0
	3	0.9878	0.9154	0.9502
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8951	0.9960	0.9428
	2	0	0	0
	3	0.9863	0.9143	0.9489

Table B-4: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Demand Operational Condition (OC #1) and Cellular-75 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.8983	0.9841	0.9392
	2	0.6269	0.0331	0.0630
	3	0.9526	0.9176	0.9348
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8902	0.9963	0.9403
	2	0	0	0
	3	0.9876	0.9079	0.9461
Random Forest (11 PCA, Gini, 10 trees)	1	0.8924	0.9917	0.9394
	2	0	0	0
	3	0.9755	0.9144	0.9440
Random Forest (11 PCA, Gini, 250 trees)	1	0.8924	0.9972	0.9419
	2	0	0	0
	3	0.9901	0.9141	0.9506

Prediction Algorithm	Index	Precision	Recall	F1 Score
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8919	0.9973	0.9417
	2	0	0	0
	3	0.9907	0.9129	0.9502

Table B-5: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Visibility Operational Condition (OC #2) and DSRC-20 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.3481	0.0543	0.0941
	2	0.7917	0.9422	0.8604
	3	0.9240	0.8815	0.9022
Random Forest (11 PCA, Entropy, 10 trees)	1	0.2306	0.3171	0.2670
	2	0.8524	0.8370	0.8446
	3	0.9939	0.9245	0.9580
Random Forest (11 PCA, Gini, 10 trees)	1	0.1674	0.4129	0.2382
	2	0.8134	0.6197	0.7035
	3	0.9919	0.9252	0.9574
Random Forest (11 PCA, Gini, 250 trees)	1	0.1654	0.4008	0.2342
	2	0.8167	0.6245	0.7078
	3	0.9863	0.9288	0.9567
Random Forest (11 PCA, Gini, 1000 trees)	1	0.1649	0.3977	0.2331
	2	0.8166	0.6272	0.7095
	3	0.9887	0.9289	0.9579

Table B-6: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Visibility Operational Condition (OC #2) and DSRC-75 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.1440	0.4248	0.2150
	2	0.8126	0.6017	0.6915
	3	0.9838	0.8919	0.9356
Random Forest (11 PCA, Entropy, 10 trees)	1	0.2306	0.3171	0.2670
	2	0.8524	0.8370	0.8446
	3	0.9939	0.9245	0.9580
Random Forest (11 PCA, Gini, 10 trees)	1	0.2068	0.3102	0.2482
	2	0.8527	0.8176	0.8348
	3	0.9925	0.9277	0.9590
Random Forest (11 PCA, Entropy, 250 trees)	1	0.1833	0.3460	0.2397
	2	0.8471	0.7586	0.8004
	3	0.9887	0.9249	0.9558
Random Forest (11 PCA, Gini, 250 trees)	1	0.1907	0.3298	0.2416
	2	0.8445	0.7752	0.8083
	3	0.9937	0.9274	0.9594
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.1705	0.3936	0.2379
	2	0.8438	0.6999	0.7651
	3	0.9923	0.9245	0.9572

Prediction Algorithm	Index	Precision	Recall	F1 Score
Random Forest (11 PCA, Gini, 1000 trees)	1	0.2061	0.3102	0.2476
	2	0.8514	0.8163	0.8335
	3	0.9941	0.9280	0.9599

Table B-7: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Visibility Operational Condition (OC #2) and Cellular-20 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.2044	0.7688	0.3229
	2	0.8484	0.3488	0.4944
	3	0.9351	0.9276	0.9313
Random Forest (11 PCA, Entropy, 10 trees)	1	0.1985	0.6010	0.2984
	2	0.8137	0.4916	0.6129
	3	0.9897	0.9347	0.9614
Random Forest (11 PCA, Gini, 10 trees)	1	0.1862	0.3570	0.2447
	2	0.7886	0.6730	0.7262
	3	0.9804	0.9125	0.9452
Random Forest (11 PCA, Gini, 250 trees)	1	0.1903	0.3441	0.2451
	2	0.7953	0.6928	0.7405
	3	0.9838	0.9233	0.9526
Random Forest (11 PCA, Gini, 1000 trees)	1	0.1878	0.3595	0.2467
	2	0.7946	0.6756	0.7303
	3	0.9841	0.9213	0.9517

Table B-8: Comparing Precision, Recall, F1 Score of Prediction Models for the Low Visibility Operational Condition (OC #2) and Cellular-75 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.1597	0.4032	0.2288
	2	0.7945	0.6082	0.6890
	3	0.9865	0.9033	0.9431
Random Forest (11 PCA, Entropy, 10 trees)	1	0.1674	0.4027	0.2365
	2	0.8036	0.6211	0.7007
	3	0.9792	0.9178	0.9475
Random Forest (11 PCA, Gini, 10 trees)	1	0.1710	0.4471	0.2474
	2	0.8081	0.5960	0.6860
	3	0.9881	0.9191	0.9523
Random Forest (11 PCA, Gini, 250 trees)	1	0.1675	0.4005	0.2362
	2	0.8066	0.6268	0.7054
	3	0.9883	0.9243	0.9552
Random Forest (11 PCA, Gini, 1000 trees)	1	0.1675	0.3998	0.2361
	2	0.8068	0.6300	0.7075
	3	0.9919	0.9238	0.9566

Table B-9: Comparing Precision, Recall, F1 Score of Prediction Models for the Weather + Incidents Operational Condition (OC #3) and DSRC-20 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.5679	0.8892	0.6931
	2	0.8819	0.8549	0.8682
	3	0.9484	0.8227	0.8811
Random Forest (11 PCA, Entropy, 10 trees)	1	0.3335	0.8948	0.4859
	2	0.8778	0.6796	0.7661
	3	0.9973	0.8294	0.9056
Random Forest (11 PCA, Gini, 10 trees)	1	0.3290	0.9027	0.4823
	2	0.8816	0.6603	0.7550
	3	0.9943	0.8388	0.9099
Random Forest (11 PCA, Gini, 250 trees)	1	0.3082	0.8627	0.4541
	2	0.8742	0.6424	0.7406
	3	0.9924	0.8419	0.9110
Random Forest (11 PCA, Gini, 1000 trees)	1	0.3101	0.8598	0.4558
	2	0.8737	0.6471	0.7435
	3	0.9929	0.8417	0.9111

Table B-10: Comparing Precision, Recall, F1 Score of Prediction Models for the Weather + Incidents Operational Condition (OC #3) and DSRC-75 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.3131	0.9117	0.4661
	2	0.8669	0.6488	0.7422
	3	0.9914	0.7945	0.8821
Random Forest (11 PCA, Entropy, 10 trees)	1	0.3335	0.8948	0.4859
	2	0.8778	0.6796	0.7661
	3	0.9973	0.8294	0.9056
Random Forest (11 PCA, Gini, 10 trees)	1	0.3375	0.9079	0.4920
	2	0.8850	0.6856	0.7726
	3	0.9947	0.8244	0.9016
Random Forest (11 PCA, Entropy, 250 trees)	1	0.3555	0.8785	0.5062
	2	0.8698	0.7111	0.7825
	3	0.9944	0.8254	0.9020
Random Forest (11 PCA, Gini, 250 trees)	1	0.3275	0.8578	0.4740
	2	0.8737	0.6763	0.7624
	3	0.9938	0.8360	0.9081
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.3500	0.8793	0.5007
	2	0.8731	0.7060	0.7807
	3	0.9944	0.8266	0.9028
Random Forest (11 PCA, Gini, 1000 trees)	1	0.3519	0.8946	0.5051
	2	0.8775	0.7043	0.7814
	3	0.9954	0.8273	0.9036

Table B-11: Comparing Precision, Recall, F1 Score of Prediction Models for the Weather + Incidents Operational Condition (OC #3) and Cellular-20 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.2739	0.9285	0.4230
	2	0.8824	0.5245	0.6580
	3	0.9899	0.8638	0.9226
Random Forest (11 PCA, Entropy, 10 trees)	1	0.2574	0.7939	0.3888
	2	0.8478	0.5570	0.6723
	3	0.9914	0.8661	0.9246
Random Forest (11 PCA, Gini, 10 trees)	1	0.2860	0.8057	0.4222
	2	0.8528	0.6186	0.7171
	3	0.9860	0.8428	0.9088
Random Forest (11 PCA, Gini, 250 trees)	1	0.2997	0.8361	0.4412
	2	0.8689	0.6302	0.7305
	3	0.9888	0.8532	0.9160
Random Forest (11 PCA, Gini, 1000 trees)	1	0.3008	0.8400	0.4429
	2	0.8692	0.6300	0.7305
	3	0.9894	0.8535	0.9164

Table B-12: Comparing Precision, Recall, F1 Score of Prediction Models for the Weather + Incidents Operational Condition (OC #3) and Cellular-75 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.2852	0.5986	0.3863
	2	0.8157	0.7249	0.7677
	3	0.9876	0.8356	0.9053
Random Forest (11 PCA, Entropy, 10 trees)	1	0.2841	0.8536	0.4263
	2	0.8727	0.6039	0.7139
	3	0.9852	0.8516	0.9135
Random Forest (11 PCA, Gini, 10 trees)	1	0.2914	0.8735	0.4370
	2	0.8774	0.6116	0.7207
	3	0.9904	0.8513	0.9156
Random Forest (11 PCA, Gini, 250 trees)	1	0.3055	0.8629	0.4512
	2	0.8762	0.6382	0.7385
	3	0.9893	0.8552	0.9174
Random Forest (11 PCA, Gini, 1000 trees)	1	0.2945	0.8599	0.4387
	2	0.8778	0.6214	0.7277
	3	0.9893	0.8568	0.9183

Table B-13: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #4) and DSRC-20 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.9184	0.9952	0.9553
	2	0	0	0
	3	0.9990	0.8691	0.9295

Prediction Algorithm	Index	Precision	Recall	F1 Score
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8898	0.9466	0.9173
	2	0	0	0
	3	0.9898	0.9849	0.9873
Random Forest (11 PCA, Gini, 10 trees)	1	0.9285	0.9779	0.9526
	2	0	0	0
	3	0.9959	0.9916	0.9938
Random Forest (11 PCA, Gini, 250 trees)	1	0.9396	0.9779	0.9584
	2	0	0	0
	3	0.9959	0.9942	0.9950
Random Forest (11 PCA, Gini, 1000 trees)	1	0.9390	0.9779	0.9581
	2	0	0	0
	3	0.9959	0.9940	0.9950

Table B-14: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #4) and DSRC-75 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.7538	0.9994	0.8594
	2	0	0	0
	3	1	0.9357	0.9668
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8898	0.9466	0.9173
	2	0	0	0
	3	0.9898	0.9849	0.9873
Random Forest (11 PCA, Gini, 10 trees)	1	0.7999	0.9472	0.8673
	2	0	0	0
	3	0.9898	0.9637	0.9766
Random Forest (11 PCA, Entropy, 250 trees)	1	0.8997	0.9472	0.9228
	2	0	0	0
	3	0.9900	0.9878	0.9889
Random Forest (11 PCA, Gini, 250 trees)	1	0.8962	0.9355	0.9154
	2	0	0	0
	3	0.9885	0.9880	0.9882
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.8942	0.9472	0.9199
	2	0.0000	0.0000	0.0000
	3	0.9900	0.9867	0.9883
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8964	0.9472	0.9211
	2	0	0	0
	3	0.9900	0.9872	0.9886

Table B-15: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #4) and Cellular-20 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.8947	1	0.9444
	2	0	0	0
	3	1	0.9857	0.9928

Prediction Algorithm	Index	Precision	Recall	F1 Score
Random Forest (11 PCA, Entropy, 10 trees)	1	0.9092	1	0.9525
	2	0	0	0
	3	1	0.9888	0.9944
Random Forest (11 PCA, Gini, 10 trees)	1	0.8269	0.9779	0.8960
	2	0	0	0
	3	0.9959	0.9707	0.9831
Random Forest (11 PCA, Gini, 250 trees)	1	0.8836	0.9778	0.9283
	2	0	0	0
	3	0.9960	0.9837	0.9898
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8773	0.9777	0.9247
	2	0	0	0
	3	0.9960	0.9824	0.9891

Table B-16: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #4) and Cellular-75 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.8737	1	0.9326
	2	0	0	0
	3	1	0.8806	0.9365
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8296	1	0.9069
	2	0	0	0
	3	1	0.9724	0.9860
Random Forest (11 PCA, Gini, 10 trees)	1	0.8308	1	0.9076
	2	0	0	0
	3	1	0.9727	0.9861
Random Forest (11 PCA, Gini, 250 trees)	1	0.8596	1	0.9245
	2	0	0	0
	3	1	0.9798	0.9898
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8548	1	0.9217
	2	0	0	0
	3	1	0.9787	0.9892

Table B-17: Comparing Precision, Recall, F1 Score of Prediction Models for the Bottleneck Trouble Operational Condition (OC #5) and DSRC-20 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.9608	0.8079	0.8777
	2	0.0886	0.4518	0.1482
	3	0.9883	0.9198	0.9528
Random Forest (11 PCA, Entropy, 10 trees)	1	0.9534	0.9572	0.9553
	2	0.1854	0.2765	0.2219
	3	0.9932	0.9320	0.9617
Random Forest (11 PCA, Gini, 10 trees)	1	0.9501	0.9721	0.9609
	2	0.2063	0.2098	0.2081
	3	0.9955	0.9401	0.9670

Prediction Algorithm	Index	Precision	Recall	F1 Score
Random Forest (11 PCA, Gini, 250 trees)	1	0.9510	0.9666	0.9588
	2	0.2130	0.2509	0.2304
	3	0.9965	0.9395	0.9671
Random Forest (11 PCA, Gini, 1000 trees)	1	0.9512	0.9671	0.9591
	2	0.2129	0.2509	0.2303
	3	0.9967	0.9389	0.9669

Table B-18: Comparing Precision, Recall, F1 Score of Prediction Models for the Bottleneck Trouble Operational Condition (OC #5) and DSRC-75 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.9363	0.9573	0.9467
	2	0.1077	0.1724	0.1325
	3	0.9968	0.8813	0.9355
Random Forest (11 PCA, Entropy, 10 trees)	1	0.9534	0.9572	0.9553
	2	0.1854	0.2765	0.2219
	3	0.9932	0.9320	0.9617
Random Forest (11 PCA, Gini, 10 trees)	1	0.9521	0.9545	0.9533
	2	0.1734	0.2671	0.2103
	3	0.9929	0.9300	0.9604
Random Forest (11 PCA, Entropy, 250 trees)	1	0.9558	0.9528	0.9543
	2	0.1771	0.2986	0.2224
	3	0.9896	0.9244	0.9559
Random Forest (11 PCA, Gini, 250 trees)	1	0.9534	0.9608	0.9571
	2	0.1809	0.2656	0.2152
	3	0.9956	0.9297	0.9615
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.9558	0.9549	0.9553
	2	0.1823	0.2910	0.2241
	3	0.9887	0.9279	0.9573
Random Forest (11 PCA, Gini, 1000 trees)	1	0.9552	0.9565	0.9558
	2	0.1884	0.2918	0.2290
	3	0.9906	0.9296	0.9591

Table B-19: Comparing Precision, Recall, F1 Score of Prediction Models for the Bottleneck Trouble Operational Condition (OC #5) and Cellular-20 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.9528	0.9744	0.9634
	2	0.1452	0.1654	0.1546
	3	0.9975	0.9384	0.9671
Random Forest (11 PCA, Entropy, 10 trees)	1	0.9530	0.9768	0.9648
	2	0.1665	0.1745	0.1704
	3	0.9974	0.9411	0.9685
Random Forest (11 PCA, Gini, 10 trees)	1	0.9437	0.9590	0.9513
	2	0.1053	0.1613	0.1274
	3	0.9931	0.9151	0.9525

Prediction Algorithm	Index	Precision	Recall	F1 Score
Random Forest (11 PCA, Gini, 250 trees)	1	0.9478	0.9639	0.9558
	2	0.1357	0.1742	0.1526
	3	0.9936	0.9345	0.9631
Random Forest (11 PCA, Gini, 1000 trees)	1	0.9472	0.9646	0.9558
	2	0.1325	0.1695	0.1487
	3	0.9947	0.9329	0.9628

Table B-20: Comparing Precision, Recall, F1 Score of Prediction Models for the Bottleneck Trouble Operational Condition (OC #5) and Cellular-75 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.9456	0.7565	0.8406
	2	0.0577	0.3904	0.1006
	3	0.9912	0.9108	0.9493
Random Forest (11 PCA, Entropy, 10 trees)	1	0.9471	0.9375	0.9423
	2	0.1179	0.2339	0.1567
	3	0.9896	0.9273	0.9574
Random Forest (11 PCA, Gini, 10 trees)	1	0.9458	0.9596	0.9527
	2	0.1361	0.1950	0.1603
	3	0.9939	0.9277	0.9597
Random Forest (11 PCA, Gini, 250 trees)	1	0.9465	0.9529	0.9497
	2	0.1318	0.2020	0.1595
	3	0.9925	0.9339	0.9623
Random Forest (11 PCA, Gini, 1000 trees)	1	0.9466	0.9529	0.9497
	2	0.1339	0.2047	0.1619
	3	0.9925	0.9345	0.9626

Table B-21: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #6) and DSRC-20 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.9095	0.7743	0.8364
	2	0.1034	0.3443	0.1591
	3	0.9930	0.8446	0.9128
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8880	0.9980	0.9398
	2	0.1467	0.0053	0.0102
	3	0.9975	0.9168	0.9555
Random Forest (11 PCA, Gini, 10 trees)	1	0.8871	0.9972	0.9389
	2	0.0643	0.0054	0.0100
	3	0.9950	0.9095	0.9504
Random Forest (11 PCA, Gini, 250 trees)	1	0.8830	0.9981	0.9370
	2	0	0	0
	3	0.9951	0.9127	0.9521
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8833	0.9980	0.9372
	2	1	0.0005	0.0010
	3	0.9950	0.9137	0.9526

Table B-22: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #6) and DSRC-75 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.8854	0.9840	0.9321
	2	0.1705	0.0537	0.0816
	3	0.9961	0.8865	0.9381
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8880	0.9980	0.9398
	2	0.1467	0.0053	0.0102
	3	0.9975	0.9168	0.9555
Random Forest (11 PCA, Gini, 10 trees)	1	0.8841	0.9961	0.9368
	2	0.0920	0.0038	0.0074
	3	0.9965	0.9083	0.9504
Random Forest (11 PCA, Entropy, 250 trees)	1	0.8879	0.9953	0.9385
	2	0.1667	0.0010	0.0019
	3	0.9868	0.9192	0.9518
Random Forest (11 PCA, Gini, 250 trees)	1	0.8843	0.9985	0.9379
	2	0	0	0
	3	0.9947	0.9175	0.9545
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.8888	0.9958	0.9392
	2	0.1538	0.0010	0.0019
	3	0.9881	0.9216	0.9537
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8873	0.9979	0.9394
	2	0.1818	0.0010	0.0019
	3	0.9946	0.9185	0.9551

Table B-23: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #6) and Cellular-20 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.8813	0.9879	0.9316
	2	0.1313	0.0229	0.0390
	3	0.9980	0.8896	0.9407
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8941	0.9990	0.9436
	2	0.3129	0.0073	0.0142
	3	0.9973	0.9226	0.9585
Random Forest (11 PCA, Gini, 10 trees)	1	0.8877	0.9962	0.9388
	2	0.0833	0.0001	0.0002
	3	0.9900	0.9091	0.9479
Random Forest (11 PCA, Gini, 250 trees)	1	0.8905	0.9962	0.9404
	2	0.1429	0.0001	0.0002
	3	0.9903	0.9166	0.9520
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8896	0.9967	0.9401
	2	0	0	0
	3	0.9913	0.9144	0.9513

Table B-24: Comparing Precision, Recall, F1 Score of Prediction Models for the Many Incidents Operational Condition (OC #6) and Cellular-75 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Decision Trees (11 PCA, Entropy)	1	0.8867	0.9321	0.9088
	2	0.0744	0.0695	0.0719
	3	0.9855	0.8940	0.9375
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8849	0.9933	0.9360
	2	0.0630	0.0006	0.0012
	3	0.9842	0.9093	0.9452
Random Forest (11 PCA, Gini, 10 trees)	1	0.88607	0.99713	0.93833
	2	0.04098	0.00038	0.00076
	3	0.99282	0.91169	0.95053
Random Forest (11 PCA, Gini, 250 trees)	1	0.8866	0.9963	0.9382
	2	0	0	0
	3	0.9900	0.9140	0.9505
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8863	0.9964	0.9381
	2	0	0	0
	3	0.9902	0.9131	0.9501

Table B-25: Comparing Precision, Recall, F1 Score of Prediction Models Overall (All Operational Conditions) and DSRC-20 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.5564	0.7228	0.6288
	2	0.4532	0.1262	0.1975
	3	0.3918	0.4253	0.4078
Logistic Regression (8 PCA)	1	0.5469	0.6945	0.6119
	2	0.3571	0.1211	0.1808
	3	0.4137	0.4483	0.4303
Logistic Regression (11 PCA)	1	0.8336	0.9155	0.8726
	2	0.7218	0.6475	0.6826
	3	0.9967	0.9349	0.9648
Decision Trees (11 PCA, Entropy)	1	0.8873	0.8319	0.8587
	2	0.6294	0.8051	0.7065
	3	0.9710	0.8699	0.9177
Decision Trees (11 PCA, Gini)	1	0.7819	0.9293	0.8492
	2	0.7540	0.5763	0.6533
	3	0.9756	0.8845	0.9278
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8188	0.9574	0.8827
	2	0.8269	0.6558	0.7314
	3	0.9941	0.9124	0.9515
Random Forest (11 PCA, Gini, 10 trees)	1	0.7902	0.9638	0.8684
	2	0.8190	0.5605	0.6655
	3	0.9951	0.9145	0.9531
Random Forest (11 PCA, Entropy, 250 trees)	1	0.7910	0.9594	0.8671
	2	0.8160	0.5664	0.6687
	3	0.9953	0.9171	0.9546

Prediction Algorithm	Index	Precision	Recall	F1 Score
Random Forest (11 PCA, Gini, 250 trees)	1	0.7862	0.9600	0.8645
	2	0.8170	0.5546	0.6607
	3	0.9941	0.9154	0.9531
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.7860	0.9596	0.8642
	2	0.8163	0.5527	0.6591
	3	0.9952	0.9178	0.9549
Random Forest (11 PCA, Gini, 1000 trees)	1	0.7873	0.9598	0.8650
	2	0.8169	0.5579	0.6630
	3	0.9946	0.9154	0.9534

Table B-26: Comparing Precision, Recall, F1 Score of Prediction Models Overall (All Operational Conditions) and DSRC-75 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.5539	0.7183	0.6255
	2	0.5060	0.1112	0.1823
	3	0.4047	0.4589	0.4301
Logistic Regression (8 PCA)	1	0.5433	0.6878	0.6071
	2	0.4026	0.1309	0.1976
	3	0.4371	0.4824	0.4586
Logistic Regression (11 PCA)	1	0.6439	0.8728	0.7411
	2	0.3838	0.1591	0.2250
	3	0.9995	0.9273	0.9621
Decision Trees (11 PCA, Entropy)	1	0.7754	0.9463	0.8524
	2	0.7478	0.5549	0.6371
	3	0.9923	0.8784	0.9319
Decision Trees (11 PCA, Gini)	1	0.7827	0.9436	0.8556
	2	0.7767	0.5676	0.6559
	3	0.9831	0.8966	0.9379
Random Forest (11 PCA, Entropy, 10 trees)	1	0.8188	0.9574	0.8827
	2	0.8269	0.6558	0.7314
	3	0.9941	0.9124	0.9515
Random Forest (11 PCA, Gini, 10 trees)	1	0.8128	0.9563	0.8787
	2	0.8279	0.6507	0.7287
	3	0.9930	0.9065	0.9478
Random Forest (11 PCA, Entropy, 250 trees)	1	0.7819	0.9293	0.8492
	2	0.7540	0.5763	0.6533
	3	0.9756	0.8845	0.9278
Random Forest (11 PCA, Gini, 250 trees)	1	0.8106	0.9564	0.8775
	2	0.8237	0.6288	0.7132
	3	0.9933	0.9155	0.9528
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.7827	0.9535	0.8597
	2	0.8022	0.5445	0.6487
	3	0.9894	0.9190	0.9529
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8206	0.9571	0.8836
	2	0.8273	0.6596	0.7340
	3	0.9931	0.9123	0.9510

Table B-27: Comparing Precision, Recall, F1 Score of Prediction Models Overall (All Operational Conditions) and Cellular-20 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.5551	0.7445	0.6360
	2	0.2220	0.0248	0.0446
	3	0.4392	0.4597	0.4492
Logistic Regression (8 PCA)	1	0.5333	0.8554	0.6570
	2	0.4748	0.0260	0.0493
	3	0.5100	0.3784	0.4345
Logistic Regression (11 PCA)	1	0.8887	0.9342	0.9109
	2	0.7684	0.7179	0.7423
	3	0.9902	0.9608	0.9753
Decision Trees (11 PCA, Entropy)	1	0.7694	0.9783	0.8614
	2	0.8082	0.4046	0.5392
	3	0.9887	0.9141	0.9499
Decision Trees (11 PCA, Gini)	1	0.8580	0.9047	0.8808
	2	0.7138	0.7373	0.7254
	3	0.9719	0.8874	0.9277
Random Forest (11 PCA, Entropy, 10 trees)	1	0.7874	0.9711	0.8697
	2	0.8002	0.4629	0.5865
	3	0.9949	0.9262	0.9593
Random Forest (11 PCA, Gini, 10 trees)	1	0.8031	0.9578	0.8737
	2	0.7825	0.5494	0.6456
	3	0.9889	0.9058	0.9455
Random Forest (11 PCA, Entropy, 250 trees)	1	0.8022	0.9614	0.8746
	2	0.7926	0.5392	0.6418
	3	0.9911	0.9121	0.9499
Random Forest (11 PCA, Gini, 250 trees)	1	0.8098	0.9607	0.8788
	2	0.8031	0.5620	0.6613
	3	0.9903	0.9161	0.9518
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.8007	0.9589	0.8727
	2	0.7845	0.5350	0.6362
	3	0.9911	0.9122	0.9500
Random Forest (11 PCA, Gini, 1000 trees)	1	0.8078	0.9614	0.8780
	2	0.8031	0.5568	0.6576
	3	0.9906	0.9149	0.9512

Table B-28: Comparing Precision, Recall, F1 Score of Prediction Models Overall (All Operational Conditions) and Cellular-75 Scenario for Test Data Set

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (6 PCA)	1	0.5774	0.5757	0.5765
	2	0.4136	0.0838	0.1394
	3	0.4457	0.6544	0.5303
Logistic Regression (8 PCA)	1	0.5958	0.6850	0.6373
	2	0.3833	0.0964	0.1540
	3	0.4900	0.6170	0.5462

Prediction Algorithm	Index	Precision	Recall	F1 Score
Logistic Regression (11 PCA)	1	0.6469	0.8775	0.7448
	2	0.3510	0.1373	0.1974
	3	0.9988	0.9320	0.9643
Decision Trees (11 PCA, Entropy)	1	0.7855	0.8593	0.8208
	2	0.6171	0.6014	0.6091
	3	0.9834	0.8887	0.9337
Decision Trees (11 PCA, Gini)	1	0.8494	0.8557	0.8525
	2	0.6504	0.8031	0.7187
	3	0.9916	0.8472	0.9137
Random Forest (11 PCA, Entropy, 10 trees)	1	0.7774	0.9484	0.8544
	2	0.7847	0.5390	0.6390
	3	0.9873	0.9104	0.9473
Random Forest (11 PCA, Gini, 10 trees)	1	0.7763	0.9575	0.8574
	2	0.8045	0.5318	0.6404
	3	0.9900	0.9121	0.9495
Random Forest (11 PCA, Entropy, 250 trees)	1	0.7892	0.9589	0.8658
	2	0.8113	0.5676	0.6679
	3	0.9902	0.9115	0.9493
Random Forest (11 PCA, Gini, 250 trees)	1	0.7856	0.9547	0.8619
	2	0.8028	0.5568	0.6576
	3	0.9915	0.9161	0.9523
Random Forest (11 PCA, Entropy, 1000 trees)	1	0.7832	0.9537	0.8601
	2	0.8016	0.5447	0.6487
	3	0.9894	0.9186	0.9527
Random Forest (11 PCA, Gini, 1000 trees)	1	0.7824	0.9546	0.8600
	2	0.8029	0.5501	0.6529
	3	0.9922	0.9160	0.9526

U.S. Department of Transportation
ITS Joint Program Office-HOIT
1200 New Jersey Avenue, SE
Washington, DC 20590

Toll-Free "Help Line" 866-367-7487
www.its.dot.gov

FHWA-JPO-XX-XXX



U.S. Department of Transportation