REAL-TIME TRAFFIC CONGESTION PREDICTION USING BIG DATA

A Project report
presented to

The Faculty of the Department of Applied Data
Science

San Jose State
University

in Partial Fulfillment of the Requirements for the Master's
Degree

In

Data
Analytics

B
y

Chaithanya Reddy Bogadi -
013803998

Naga Sindhu Korlapati -
012467832

Rajasree Rajendran –
013774358

Sai Chaitanya Tolem –
013008788

Sindu Ravichandran –
013769821

May
2020

Acknowledgment

We would like to thank all the great people whose encouragement and support helped us in

working on this project. We would like to wholeheartedly thank Professor **Dr. Jerry Gao** and

Professor **Dr. Ronald Mak**, our Technical Advisors, for continually guiding us and motivating us

to do better. We would also like to thank Professor **Dr. Lee Chang** for his valuable suggestions

and
guidance.

Table of Contents

Table of Figures 6 **Table of Tables** 8 **Abstract** 9 **1.1 Project Goals and Background** 10 **1.2 Analysis of Requirements** 12 **1.2.1 Functional Requirements** 13 **1.2.2 Non-Functional Requirements** 14 **1.3 Project Deliverables** 17 **1.4 Technology and Solution Survey** 18 **1.4.1 Autonomous Vehicle Technology** 18 **1.4.2 Adaptive Traffic Signals** 19 **1.4.3 Vehicle to Infrastructure smart corridors** 20 **1.4.4 Real-time traffic feedback** 21 **1.5 Literature Survey of Existing Research** 23 **2.1 Data Exploration Strategy and Planning** 26 **2.2 Data Sources and Dataset Parameters** 29 **2.2.1 Traffic Dataset** 31 **2.2.2 Weather Dataset** 35 **2.3 Data Collection and training datasets** 36 **2.3.1 Data Collection** 36 **2.3.1.1 Traffic Dataset** 36 **2.3.1.2 Meta Dataset** 37 **2.3.1.3 Incident Dataset** 37 **2.3.1.4 Weather Dataset** 38 **2.3.2 Training Datasets** 39 **2.4 Data Cleansing and Validation** 46 **2.4.1 Traffic Data** 46 **2.4.2 Metadata** 47 **2.4.3 Incident Data** 47 **2.4.4 Data Validation** 48 **2.5 Data Transformation and Tools** 52

2.5.1 Tools 54 **2.6 Raw data visualization** 55 **3.1 Project Organization** 64 **3.2 Resource Requirements** 67 **3.3 Project Schedule** 68 **3.3.1 Project Organization** 68 **3.3.2 Project Timeline and Schedule** 70 **4.1 Problem Formulation** 71 **4.2 Foundation of Proposed Solutions** 78 **4.2.1 Proposed Model:** 80 **4.3 Feature Engineering** 81 **4.3.1 Numerical Features** 81 **4.3.2 Feature Selection** 82 **4.4 Select and Justify Solution Model** 88 **4.4.1 Ensemble Learning** 88 **4.4.2 Simple RNN** 90 **4.4.3 GRU** 91 **4.4.4 LSTM** 92 **a) Model 1 – Train a time series ML model with traffic features** 94 **b) Model 2 – Train a time series ML model with traffic and weather features** 95 **c) Model 3 - Train a time series ML model with traffic, weather, incident features** 95 **4.5 Justify Solution Model** 96 **5.1 Model Building and Training** 99 **5.1.1 Train Test Split** 99 **5.1.2 Model Training** 100 **5.2 Model Execution and Evaluation** 104 **5.3 Model Validation** 106 **6. Evaluation and Visualization** 110 **6.1 Analysis of Model Execution/Evaluation Results** 110 **6.1.1 Traffic Speed Prediction** 111 **6.1.2 Model Evaluation** 116 **6.1.3 Incident Classification Models** 118 **6.2 Achievements and Constraints** 122 **6.3 Quality Evaluation of Model Functions and Performance** 125

6.4 Evaluation of Models vs. Requirements 129 **Stage 1 – Initial Model Building and training** 129 **Stage 2 – Retraining model with more data** 130 **Stage 3 – Resolving overfitting** 131 **Stage 4 – Building Additional Models** 131 **Stage 5 – Data Sparsity** 131 **Stage 6 – Location-specific Models** 131 **6.5 Information Visualization** 132 **7.1 Summary** 137 **7.2 Benefits and Shortcoming** 140 **7.3 Potential Model Applications** 141 **7.4 Lessons Learned** 142 **7.5 Recommendations for Future Work** 143 **7.6 Contributions and Impacts on Society** 144 **References** 146

Table of Figures

Figure 1 Super-commuter Boom in the San Francisco Bay Area .... 11 Figure 2 Autonomous and Connected Vehicle Technology. .... 19 Figure 3 Camera recording vehicle count. .... 20 Figure 4 Vehicle to Infrastructure. .... 21 Figure 5 Schematic representation of the data modeling process .... 27 Figure 6 KDD Process .... 28 Figure 7 Datasets from Multiple sources .... 30 Figure 8 ER Diagram for the considered data sources .... 30 Figure 9 Meta Dataset .... 37 Figure 10 Data Collection Flow Diagram .... 38 Figure 11 Screenshot of scaling function... 40 Figure 12 Data after Minmax scaling .... 41 Figure 13 Time Series data to Supervised Learning data .... 42 Figure 14 Train Test Split Process .... 46 Figure 15 Incident data .... 48 Figure 16 Process flow diagram used for data cleaning and validation .... 51 Figure 17 Direction feature Label encoding .... 53 Figure 18 Data distribution of Total flow, Occupancy and Speed .... 53 Figure 19 Heat map of Speed by Freeway vs. Hour of the day .... 55 Figure 20 Plot of Speed by Hour per day of week .... 56 Figure 21 Plot of number of incidents in Bay Area .... 57 Figure 22 Bar plot of most incidents in Bay Area .... 57 Figure 23 Bar plot of most dangerous locations in Bay Area .... 58 Figure 24 Incident Heat map... 59 Figure 25 Weather data pair plot .... 60 Figure 26 Traffic and weather dashboard .... 61 Figure 27 Traffic-Incident dashboard .... 62 Figure 28 Traffic-Weather- Incident dashboard .... 63 Figure 29 Data Science Project Life Cycle .... 64 Figure 30 Work Breakdown Structure .... 69 Figure 31 PERT Chart .... 70 Figure 32 INRIX 2018 Global traffic scorecard .... 72 Figure 33 Annual Hours of Delay in San Francisco in 2017 .... 73 Figure 34 Congestion times .... 73 Figure 35 Congestion Costs .... 74 Figure 36 Cars in American Commute .... 75 Figure 37 How Americans commute to work .... 76 Figure 38 American commute through years... 77 Figure 39 Growing commute times in Bay Area .... 78 Figure 40 Top 10 features in data .... 84 Figure 41 Correlation matrix of features .... 85 Figure 42 Stacking in Machine Learning.. 88 Figure 43 Ensemble modeling 89

Figure 44 Basic architectures of RNN, GRU and LSTM cells .... 90 Figure 45 Simple Single RNN Unit .... 91 Figure 46 GRU

Operation .... 92 Figure 47 LSTM operation .... 93 Figure 48 Data modeling flowchart .... 94 Figure 49 One to One Model .... 96 Figure 50 One to Many Model.. 97 Figure 51 RNN Unrolled Time .... 98 Figure 52 LSTM model architecture.. 101 Figure 53 GRU model architecture .... 103 Figure 54 Predicted result versus Actual for LSTM Model .... 105 Figure 55 Predicted result versus Actual for GRU Model .... 106 Figure 56 Validation Loss Curve of LSTM .... 107 Figure 57 Loss Curve of Gated Recurrent Unit Model... 108 Figure 58 Ensemble Stacking model architecture .... 109 Figure 59 Ensemble model learning curve .... 110 Figure 60 400001 VDS location .... 112 Figure 61 Predicted result versus Actual for LSTM Model .... 113 Figure 62 Predicted result versus Actual Speed for GRU Model .... 113 Figure 63 Actual versus Predicted Speed graphs for three different Models .... 114 Figure 64 Actual versus Predicted Speed graphs for three different Models .... 115 Figure 65 Bay Area Incident Prone Areas .... 116 Figure 66 Metric Errors comparison for various models .... 117 Figure 67 Incidents Confusion Matrix for Freeway 101 .... 119 Figure 68 Classification model performances .... 120 Figure 69 ROC Curve for Incident Model .... 121 Figure 70 RMSE and MSE score comparisons of models .... 122 Figure 71 Training and validation Loss for Logistic Model .... 126 Figure 72 Overfitting Random Forest Model Learning curve .... 127 Figure 73 Random Forest Learning curve after Regularization .... 128 Figure 74 Training and Validation Loss curve for Gradient Boosting .... 129 Figure 75 Loss curves for LSTM .... 130 Figure 76 Raw data dashboard 132

Figure 77 Actual vs Predicted Dashboard for selected date for selected station .... 133 Figure 78 UI using Folium map .... 134 Figure 79 Web interface developed for traffic prediction .... 135 Figure 80 Exploring the web interface - A Glimpse .... 136 Figure 81 Final Work Breakdown Structure 138

Table of Tables

Table 1 Non-Functional requirements .... 15 Table 2 Data Science lifecycle .... 16 Table 3 Technology Solution Survey Table .... 21 Table 4 Traffic congestion literature review .... 25 Table 5 Schema of Traffic dataset .... 31 Table 6 Schema of Incident dataset .... 32 Table 7 Traffic flow - The dataset schema used as input to the time series model .... 34 Table 8 Weather dataset schema (Wang, 2018) .... 35 Table 9 Traffic preprocessed data .... 39 Table 10 Traffic data after feature engineering .... 40 Table 11 Traffic and Weather data .... 43 Table 12 Training data for Traffic – Weather .... 43 Table 13 Combined Incident, Traffic and Weather data .... 44 Table 14 Training dataset for Incident .... 44 Table 15 Key Parameters for incident prediction model .... 45 Table 16 Data Rules and Constraints .... 49 Table 17 Weather data validation .... 52 Table 18 Statistical Summary on traffic features ....

81 Table 19 Statistical Summary on Weather data .... 82 Table 20 Feature Engineered data of Incidents.. 86 Table 21 Incident features derivation .... 86 Table 22 Features Set of Traffic Model .... 94 Table 23 Features set for Traffic and Weather Model .... 95 Table 24 Features set of all three sources Traffic, Weather, Incident data .... 95 Table 25 Various State of the art techniques comparison .... 124 Table 26 Model training in HPC 125

Abstract

The correlation of high population and high use of private vehicles is a known contributor to traffic

congestion, consequently increasing carbon dioxide emissions into the environment, and traffic

incidents. It is essential to manage traffic well to help the urban population in their daily lives and

improve overall transportation efficiency. Accurate traffic information prediction is imperative for

urban planning, intelligent urban traffic management, and determining the risk of road accidents.

Most of the existing work implemented batch methods to process traffic data, which cannot support

real-time prediction. Currently, there is minimal literature available on real-time traffic prediction.

Our project will develop a real-time traffic model by analyzing various

sensor data.

Our real-time traffic prediction system helps to detect and display current traffic conditions on a particular road. The proposed model uses data recorded from road sensors and a variety of other sources. Capturing massive amounts of sensor data and processing it in real-time is a very challenging task. The sensor data is ingested by streaming analytics platforms using big data technologies and processed using various deep learning and machine learning algorithms.

Due to the availability of a wide range of data analytic techniques, it is challenging for businesses and data scientists to choose the right approach to make data-driven traffic decisions. We will provide reliable models that can help predict traffic congestion. Moreover, our models can support the adaptive toll fare and carpool system on congested roads to reduce potential traffic congestion and provide suggestions for urban planners. This project will fill the gap in the data analytics field by contributing a more accurate and reliable model that uses IoT sensor data and other data sources.

Organizations like transportation authorities and public safety services can also benefit from this approach by implementing it in their platforms to make proactive decisions.

1.1 Project Goals and Background

One of the biggest problems faced by big cities is inefficient traffic management due to traffic congestion. The world population is increasing day-by-day, thus causing a spike in the higher use of private vehicles, causing traffic congestion. This can result in a total of $166 billion in congestion costs (cost of extra time and fuel spent) in the United States (Schrank, 2019). This also causes increased traffic incidents and higher carbon dioxide emissions to the environment. Air quality is diminished in areas with heavy traffic, thus affecting overall public health. Another consequence of traffic congestion is economic losses associated with infrastructure costs and waste of productive time spent in traffic jams. Since building new roads is not a viable solution in many areas (Mann, 2014), finding a new way to utilize existing resources effectively is essential. Developing new technological solutions to achieve more efficiency is necessary to reduce traffic congestion and improve public transportation. Since managing traffic congestion is a crucial aspect of effective urban planning, accurate prediction of traffic information is needed. Traffic prediction systems require real-time data and they are designed for large-scale road networks.

Moreover, the transportation sector has progressed these days by the advancements in different technologies such as autonomous vehicles, wireless communication in vehicles, and so on. These modern technologies might change the entire transportation sector, where newer business models will be developed based on 'transport-as-a-service' or TaaS (Asel, 2018). These technologies generate vast amounts of heterogeneous traffic data. Using this data effectively can result in understanding the weaknesses in the transport network to take necessary actions to improve efficiency. From various studies, we can see the boom in Bay Area Super-commuter percentage, which can account for the increased traffic congestion all over the Bay Area. Due to economic

factors, people tend to reside outside the main Metropolitan area and commute to the job location. This causes a massive increase in the daily flow of traffic throughout the area.

Figure 1 Super-commuter Boom in the San Francisco Bay Area

Source: (Apartment List, 2019)

This project aims to implement a real-time traffic prediction model, considering how weather and incidents in a particular area affect the traffic in that area. In our scenario, we are considering the San Francisco Bay Area freeways, specifically Freeways 101, 280, 680, and 880, since these are the most popular freeways used by Bay Area residents. These Freeways contribute to a major part of the Bay Area super-commuter traffic, which is on the rise day by day. We also aim to identify the time taken to resolve the congestion, which can help the end-users understand the expected delay during their commute. Another major issue we intend to tackle is calculating the time to reach the destination, keeping in mind the real-time traffic congestions and incidents. The ultimate goal of our project is to help end-users in obtaining travel time, and current traffic conditions like speed and incident probability, at particular locations at particular times. Our project will also help individuals identify incident prone areas, and times taken to resolve the incident by considering weather and incident data. On achieving the best results with this model, it can lead to better traffic management in urban areas, ultimately reducing carbon-dioxide emissions. It can impact positively on improving air quality in urban areas. It can also be helpful for efficient time management since

the time wasted on stationary traffic could be further utilized for productive activities or spending

time with family. In the future, this project can be used by traffic operators to predict traffic flow

levels and identify potential problem areas. Similarly, transit planners can evaluate delays and

issues in transit services using this model.

1.2 Analysis of Requirements

This project focuses on developing methods that can accurately predict traffic congestion

in a specific road network, using data recorded from road sensors and a variety of other sources.

The results of this project can be used by individuals to identify the best times to travel to avoid

wasting time in congestion. In the future, city planners can use this to establish smart cities,

counties can regulate traffic, and researchers can analyze the traffic situation in a particular city at

a specific time. The analysis of these results can help in identifying the bottlenecks in urban transit

systems. Transportation authorities, city transport departments, and other businesses that manage urban traffic can benefit from an accurate traffic prediction model. To improve the efficiency of our prediction model, we researched and reviewed various relevant literature to identify multiple sources that could potentially affect traffic flow and cause congestion.

1.2.1 Functional Requirements

Real-time traffic prediction will require the following modules for continuous data collection and infrastructure management in the cloud.

- **Data Collection and Management:** For this project, we need continuous retrieval of real-time traffic data captured by the Performance Measurement System (PeMS) from sensors and weather data from Dark Sky API. The data must be cleaned, transformed, and merged before loading into our cloud storage. The data pre-processing and transformation can be done using automated scripts using Python, and all these steps should be automated and monitored. Any failures may result in missing data that must be handled separately. Hence, alert monitoring should be included in the data collection script to alert developers for inconsistency in the process.

- **Report Generation and Analysis:** This project also requires calculating statistical information and comparing that information with real-time traffic volume and weather data. Another critical requirement for this project is to find the correlation between traffic flow volume and other factors using various machine learning algorithms and to predict traffic flow volume based on a trained model. This statistical analysis requires a report and graphs generation to understand more about the data.

- **Automated Job Status Monitoring:** The process of ETL is exceptionally challenging since it requires adopting the best big data technologies, deep learning, high-performance computing (HPC), and in-memory computing. Embracing big data technologies involves batch processing in a distributed environment. Batch processing typically runs with big files and datasets in large volumes for a longer time. We plan to use Hadoop to process our enormous volume of data using spark jobs. Hadoop environment has YARN manager services for job monitoring. We will use YARN manager API to monitor job status and to trigger alerts in case of any failures.

- **User-Friendly Web UI for Business and Users:** Our project will be used by individuals to plan their commute plans. Hence our web UI should be more user-friendly for customers. Our project should have a data dashboard for easy understanding. A search facility should be available for users to make their experience more dynamic.

1.2.2 Non-Functional Requirements

Below are the non -functional requirements for the traffic prediction system.

- **Reliability:** After an end-user performs an action, the server and all the views in the system will be updated according to that action.

- **Usability:** Our system is user-friendly and easy to use; users can participate in the experiment without any advanced knowledge or instructions.

- **Safety & Security:** Our system ensures secure end-user access, and the data is encrypted using a password.

- **Speed & Throughput**: After a user acts, it is updated in his view as soon as possible.

- **Capacity:** Any number of users can access our system.

- **Availability:** The user can access our UI functionality at any time.

- **Portability:** The system will work on any computer or mobile phone.

Table 1 Non-Functional requirements

Requirements	**Characteristics**
Reliability	Valid traffic data information Data Integrity

y

Usability	Easy to use User-Friendly No need for advanced knowledge
Security	Secure User access Encryption of user data
Speed and Throughput	As soon as possible
Capacity	Multiple user access
Availability	24/7 available Distributed Architecture
Portability	Any operating system

To identify the requirements for this project, we followed the data science life cycle model published by (Sapp, 2017). Our project was split into six parts based on the model, and the scope was studied from a data scientist's perspective and business perspective.

Table 2 Data Science lifecycle

Task (Proportion of effort)

Subtasks Business Data Scientist

Problem Understanding (5 - 10%)

a) Determine
objective X X

b) Define success
criteria X X

c) Assess constraints X X

Data Understanding (10 - 25%)

a) Assess data situation
X X

b) Obtain data access X

c) Explore data X X

Data Preparation

a) Filter data X (20 - 40%)

b) Clean data X

c) Feature engineering X X

Modeling

a) Select a model (20 - 30%)
approach
X

b) Build models X

a) Select model X

Evaluation of results (5 - 10%)

b) Validate model X

c) Explain model X X

Deployment

a) Deploy model X (5 - 15%)

b) Monitor and maintain
X X

c) Terminate X X

1.3 Project Deliverables

This project aims to develop a model that can help understand traffic congestion using real-time traffic data, weather data, and incident data, which can help in effective traffic management and planning. We plan to build a data pipeline that supports a web application to predict real-time traffic and help users by giving the best time to travel. We propose to deliver an interface to provide recommendations for the end-users about the time taken to resolve the congestion and time taken to reach their destinations, which will encourage public transport use during peak hours. We are also showing key performance indicators of traffic behavior like predicted traffic volume, speed, and occupancy. This interface will be helpful for individual users to understand the times when traffic speeds and incidents are high and plan their commute accordingly. The City traffic planners can use our interface to identify incident prone areas and figure out better management techniques considering traffic patterns during the day. This proposed project aims at a thorough analysis of understanding factors affecting traffic and identifying underlying causes of congestion.

1.4 Technology and Solution Survey

Traffic congestion has become an enormous problem, especially in highly populated cities, states, and even countries. With the evolution of big data technologies and advancement in the area of artificial intelligence, researchers started exploring and implementing tangible solutions to tackle this issue (Sorta, 2017).

The following were some smart solutions and technologies that use AI to minimize traffic

congestion.

- Autonomous Vehicle Technology
- Adaptive traffic signals
- Vehicle to Infrastructure smart corridors
- Real-time traffic feedback

1.4.1 Autonomous Vehicle Technology

This technology is slowly transforming the entire transportation system. Sensors and AI-enabled systems make vehicles more intelligent and have complete control over the vehicle, thus reducing traffic congestion and accidents (Tom, 2018).

Figure 2 Autonomous and Connected Vehicle Technology.

Source: HDR (HDR, 2019)

1.4.2 Adaptive Traffic Signals

The City of Columbus initiated a project to improve its traffic signal timings based on the idle time of vehicles. This is a part of their smart city pilot project to overcome traffic issues. The following is an image from the City of Grapevine, Texas, in the United States. The cameras detect the presence of traffic in the lane and determine how long the cars have been waiting. The system calculates and determines what steps to take in order to minimize the flow at the intersection (Sorta, 2017)

.

Figure 3 Camera recording vehicle count.

Source: Grapevine Texas government (Govt T., n.d.)

1.4.3 Vehicle to Infrastructure smart corridors

The Vehicle to Infrastructure or V2I is a vehicle communication model. It shares the vehicle information through cameras, sensors, and traffic lights, etc. This is bi-directional and wireless communication with road infrastructures. This real-time communication helps the vehicle to know the road conditions, accidents, traffic congestion, and availability for parking. This bi-directional communication is also helpful in avoiding accidents and reducing traffic. This concept is heavily used in freight transportation to send the weather and accident-related alerts to drivers (Sorta, 2017).

Figure 4 Vehicle to Infrastructure.

Source: National Operations Center of Excellence (NOCE, n.d.)

1.4.4 Real-time traffic feedback

This new project in Kansas, which has a free streetcar that can carry more than six thousand passengers in a day. This car not only carries the passengers but also provides continuous feedback about its surrounding traffic conditions. This vehicle moves downtown to reduce the traffic

congestion issues (Sorta, 2017).

Table 3 Technology Solution Survey Table

Technologies Solutions Sponsor(s) Country Year

Agent-based traffic

management using

Reinforcement Learning

(Benekohal, 2010)

of
Department
of

Research
and
Research
and

Innovativ
e
Innovativ
e

Technolog
y
Technolog
y

sing

zin

USA 2010
USA 2010
USA 2010

ge

Department

Adaptive Traffic Signal Dynamically Department of USA 2013

of
Dept.
of

(s) (Ban, 2013) adjusting signals

based on
traffic

deman
ds

Electronics
and
Electronics
and

Transportati
on

Informatio
n
Informatio
n

(New
York)

Technology
New
Technology
New

Advanced
Traveler

Delh
i

Information
System

India 2014
India 2014
India 2014

(ATIS)
(Sivanandam,

2012
)

ng

tio

atio

Hong Kong
ITS

(Intelligent
Transportation

te

Systems) (Govt,
2010)

travel
time

Dept.

eme

Govt. of
Hong
Govt. of
Hong

Kon
g
Kon
g

China 2010
China 2010
China 2010

Research
and
Research
and

Innovativ
e
Innovativ
e

Technolog
y

USA 2013
USA 2013
USA 2013

Advance
Weather

Responsive
Traffic

Management
(AWRTM)

(Roemer,
2013)

First lane
traffic

prediction using AI
(Delft,

2013
)

ce

eme

Department
of
Department
of

Dutch traffic
and
Dutch traffic
and

transpo
rt
transpo

rt

laboratory
for

Department
student
of
s
Department
of

Dutch 2013
Dutch 2013
Dutch 2013

Research
and
Research
and

Innovativ
e
Innovativ
e

Multi-Dimensional
Model

for traffic congestion
(Al-

Holou,
2012)

Technolo
y

USA 2011
USA 2011
USA 2011

g a

1.5 Literature Survey of Existing Research

A significant amount of work has been done in the area of transportation using machine

learning, deep learning, and big data technologies. Proper prediction of traffic congestion will help

city planners and traffic departments to make proactive decisions, thus mitigating traffic-related issues.

Guilherme (Guerreiro, 2016) proposed a scalable architecture that can support and is capable of handling real-time and historical data. The leading technologies they used in their work are Spark on Hadoop and MongoDB. To harmonize data from traffic management centers, traffic service providers, traffic operators, etc., they used the DATEX-II data model. Their proposed ETL architecture uses CRISP-DM (Cross Industry Standard Process for Data Mining). The DATEX-II data model is adopted in harmonizing traffic data provided by the highway operators. To store both real-time and historical data, the authors used MongoDB. For validation, they used Spark standalone mode that reduced the processing time by nearly 50 percent.

In another work, Fan Hsun (Hsun, 2018) used streaming data to make real-time predictions. To support their real-time processing system, they used Apache Storm spouts and bolts. Here, the spout is used as an interface between the topology and data. Once data gets settled, it is forwarded to bolt for processing. The data calculations were done on the bolt. Two different types

of data were used in making predictions: vehicle data from the national freeway database, which was collected every five minutes and weather data from the Central Weather Bureau in Taiwan. To analyze the traffic data, they ran different types of experiments such as analyzing on rainy and non-rainy days, vehicle speed detection at different periods, etc. The prediction accuracy of the model was determined based on the mean absolute relative error (MARE) and mean squared error (MSE)
.

A hybrid evolutionary model to predict the traffic flow was proposed by Min-Lian Huang (Huang, 2015). This new hybrid algorithm to forecast traffic at intersections combines two different techniques, v-Support Vector Regression(v-SVR) to forecast the traffic flow with a Gaussian loss function for a short term prediction and a new evolutionary model called CCGA (Chaos Map, Cloud Model, Genetic Algorithm) to detect the right parameters from the model (v-SVR).

The work done by Declan McHugh (McHugh, 2014) proposed a new method to analyze traffic.

His approach combined traditional and non-traditional sources such as twitter to make predictions. He used tweets to detect location, but the approach wasn’t much help in getting real-time predictions. To determine the important features, he used three different sources such as weather, spatial, and standard travel time. Due to the lack of data quality, he didn’t choose quarterly or annual trends. He concentrated mostly on weekly and daily trends.

One of the works done by Mohiuddin Chowdhury et al. (Chowdhury, 2018) was to precompute the traffic density function from the node information based on the past traffic data. They used adjacent road node information, which has traffic density from various lanes. An individual dataset was prepared by considering the various intersections in cities. A time series-based algorithm called PROPHET (a forecasting tool made by Facebook) was used in making long time predictions of traffic.

Until now, researchers worked to suggest either the best algorithms that can make the right predictions or focused on the infrastructure that accommodates the problem data. We are proposing

a big data architecture along with a model that supports real-time predictions.

Table 4 Traffic congestion literature review

Author Data Sources Tools and/or Techniques Outcomes

Tsen
g

Traffi

et.al

c

.

port Vector

›s, Fuzzy

Able to improve

Able to improve

prediction accuracy by

prediction accuracy by

iona
l

25.6
%

Sou
r

ce
s

Decla
n

Fan
Hsun

, MongoDB,

, MongoDB,

Time

Aqib

Combination
of

et.al
.

Combination
of
Combination
of

ng, deep

Convolution
Neural

Mc
Hugh

Convolution
Neural

Network achieved
with

Series Models, Ordinary
Least

Network achieved
with

Series Models, Ordinary
Least

MAPE
2.59

Square,
etc.

Mohiudd
i

SARIMA
and

n

SARIMA
and

Chowdh

SARIMA
and

u

Multivariate gave
best

ry
et.al.

Multivariate gave
best

ircular

r Traffic

tion

result
s

Muham
m

CMTF approach
has

ad

CMTF approach
has

given better outcomes

given better outcomes

in making dynamic

in making dynamic

congestion prediction

ter m

to

predictions compared to

the traditional

approach es

Min-

Liang -

Huan g

v-Support vector

Machine) with gaussian loss

function, CGPA algorithm

Genetic Algorithm of

parameters for (v-GSVR)

(Chaos Map, cloud model, and

Shu o

, Distributed

em,

Wan g

Model outperformed in

Model outperformed in

making short term

making short term

predictions compared

LSTM had given better

LSTM had given better

results than CNN for	(Convolutional Neural Networks), LSTM (Long Short-Term Memory Network)	results than CNN for short term range.

2.1 Data Exploration Strategy and Planning

In this section, we describe our data exploration workflow and give a brief overview of each phase required for the project. The first step in the process is to identify the problem correctly, as our problem is to predict the traffic congestion in real-time. Based on that, we need to collect and understand the data that we obtained from multiple sources. The prediction results entirely depend on the input data that we feed into the model for training purposes. After understanding the data, we perform data preprocessing, which includes data cleaning and data transformation. In this phase, we need to remove missing data using different data techniques, perform feature engineering, and standardize the data. After the preprocessing stage, we build a predictive model for training purposes. Once we have trained the data, we validate the model by comparing the predicted results with the original traffic data. For validation, we use root mean square and mean absolute error metrics. Finally, we deploy the model for online traffic prediction service. Below is the step-by-step procedure for the entire data modeling process.

Figure 5 Schematic representation of the data modeling process

KDD is an iterative process in which data can again be transformed, and new datasets and features

can be integrated to obtain accurate results. The below diagram shows the KDD process

implemented for the
project.

Figure 6 KDD Process

Data Cleaning cleans and removes unnecessary or irrelevant data.

- Clean each dataset using imputation techniques for null values or remove the entire record

  from the dataset. For example, in time series traffic flow data, we use forward fill to fill the

  missing records at a particular timestamp value.

- Validate that the data is cleaned. For example, a zip code should be a 5-digit numeric value.

Data Integration combines data from various sources into useful and meaningful information

(Pearlman, 2019).

- Integrate multiple data sources like traffic metadata, traffic incidents and weather

  information to increase model accuracy.

Data Selection identifies and retrieves where relevant features from the dataset.

- The sensor details and speed of the vehicles are the most crucial features.

- Weather measurements, location details

- Traffic incidents with severity information

- Traffic metadata with latitude and longitude details in the station.

Data transformation converts the data into useful formats that can be fed into the model (Oracle,

n.d.).
- Split the timestamp value and explode the rows for every minute, hour, day, week and month on time-series data.
- Make sure all the measures like speed, temperature, precipitation, etc. maintain standard units of measurements and ranges.

Data mining uses well-defined techniques to mine useful information from data.

- Build the traffic flow prediction model.
- Perform hyperparameter tuning to improve the accuracy of the model.

Pattern Evaluation uses different methods to extract data patterns during data mining and

identifies insightful and potentially useful patterns.

Knowledge Presentation presents the observed results.

2.2 Data Sources and Dataset Parameters

We are using multiple data from disparate sources such as PeMS, DarkSky Weather API etc. From our vast research, we observed that in addition to traffic data from road sensors. We could see that congestion rises during inclement weather conditions such as rain, hail and high winds since the traffic demand does not drop and freeway capacity drops. At the same time, severe weather conditions as snow are known to reduce traffic demand, which in turn reduces congestion.

The study of weather data with respect to traffic has a considerable impact on predicting real-time

traffic. The predictions significantly influence travel times by alerting the traveler to any unfavorable conditions. The prediction system also helps to identify the best times to travel and helps to add to IoT data in the long run.

Figure 7 Datasets from Multiple sources

Figure 8 ER Diagram for the considered data sources

2.2.1 Traffic Dataset

We use data from the California Department of Transportation, which is publicly available through the performance measurement system (PeMS). PeMS gets data from sensors like Intelligent Transportation System (ITS), Vehicle Detector Stations (VDS), and traffic counters (PeMS, 2013). Various parameters are identified in PeMS. We prepared dataset schemas for this project to ingest freeway traffic data that is collected on 1-hour time intervals. This data includes the flow of vehicles, vehicle speeds, vehicle occupancy, the identifier of Vehicle Detector Station (VDS), and other information (Aqib, 2019). Five years (2015-2019) hourly Traffic dataset with size 4GB/year was collected for California District 4 and later Bay Area California highways – US 101, I680, I880, I280 was filtered from the dataset (PeMS, n.d.).

Table 5 Schema of Traffic dataset

S. No	Parameter Name

**Descripti
on**

pti

1 ID int An integer value that uniquely identifies the Station

Metadata. Use this value to 'join' other
clearinghouse files that contain Station
Metadata.

2 Freeway int Freeway Number

Identifie
r

4 Freeway
Direction

number that identifies the county that

A string indicating the freeway direction.

contains this census station within
PeMS

5 County

6 City string City

7 Latitude int Latitude

8 Longitude int Longitude

9 Length int Length

10 Type string Type

11 Lanes int Total number of lanes

12 Name string Name

13 User IDs [1-4] string User entered string identifier

**Incident
Data**:

The PeMS dataset contains California Highway Patrol (CHP) incidents from all Caltrans Districts. Each downloadable file includes all incidents that occurred in a day across the state. We downloaded all the data related to the Bay Area in California. It has what type of incident, where it happened, at the time all such information was present in it. The following were the descriptions of the dataset.

Five years (2015-2019) Incident dataset with size 369MB was collected for California District 4 and later Bay Area California highways – US 101, I680, I880, I280 data was filtered. Dataset Source: (PeMS, n.d.)

Table 6 Schema of Incident dataset

S. No	Name	Data type	Comment
1	Incident ID	int	An integer value that uniquely identifies this incident within PeMS.

2	Incident Number	int	An integer incident number
3	Timestamp	Date time	Date and time of the incident with a format of MM/DD/YYYY HH24:MI: SS. For example, 9/3/2013 13:58, indicating 9/3/2013 1:58 PM.
4	Description	string	A textual description of the incident.
5	Location	string	A textual description of the location.
6	Area	string	A textual description of the Area. For example, East Sac.
7	Latitude	float	Latitude
8	Longitude	float	Longitude
9	District	int	The District number
10	Freeway Number	int	Freeway Number
11	Freeway Direction	string	Freeway Direction
12	Severity	string	Severity

13	Duration	int	Duration
14	incident_id	int	incident_id

Table 7 Traffic flow - The dataset schema used as input to the time series model

S.No	Attribute Name	Data type	Description
1	StationId	int	VDS unique identifier
2	dayOfMonth	int	Gregorian calendar day in “dd” format
3	month	int	Gregorian calendar month in “mm” format
4	year	int	Gregorian calendar year in the “yyyy” format.
5	hours	int	Clock hours 0 to 23
6	weekDays	int	Weekdays values
7–18	occupancy	float	Occupancy in 1-hour

intervals

Table 1 gives the schema of the prepared dataset. These chosen parameters collected by the vehicle

detector stations (VDSs) are enough for different analyses and traffic predictions. Along with the

aggregated values, we also take the lane information of the selected corridor (PeMS, 2013).

2.2.2 Weather Dataset

To find the correlation between traffic parameters and weather conditions at different

granularities, we obtained the weather dataset for the chosen corridor district from the DarkSky

API. Each weather station provides information about general weather conditions. The weather

data collected is matched with the timestamp of the data sampled from PeMS (Koesdwiady, 2016).

5 years (2015-2019) hourly Weather dataset with size 30MB/year was collected for California

District 4 and later Bay Area California highways – US 101, I680, I880, I280 was filtered. Dataset

Source: https://darksky.net/dev

Table 8 Weather dataset schema (Wang, 2018)

S.No	Attribute Name	Data Type	Description
1	Temperature	float	Air temperature from 2 meters above ground level
2	Dew point temperature	float	Dew point temperature from 2 meters above ground
3	Wind speed	float	Wind speed ten meters above ground level
4	Direction	char	Wind direction ten meters above ground level
5	Visibility	char	Automated sensor horizontal visibility
6	Road temperature	float	Surface temperatures, in Celsius
7	Snow depth	float	Snowfall depth, in

millimeters
.

8 Precipitation float Precipitation accumulation

at 5-minute intervals,
in

millimeters
.

2.3 Data Collection and training datasets

2.3.1 Data Collection

2.3.1.1 Traffic Dataset

The PeMS website has a complex interface. Manually clicking and exporting the data to Excel is a tedious process, and manually downloading the sheer volume of traffic data will be insurmountable. This project will use a web scraping script that automates the data download from the state of California Clearinghouse repository of Caltrans PeMS website (High, 2016). The clearinghouse HTML web pages were downloaded manually, consisting of the links to download the hourly traffic zip files and station metadata. Web scraping is a technique that extracts data from any web page (lc, n.d.). The script uses beautiful soup python package to scrape the download links in saved HTML files from the PeMS data online and persist.

We plan to use five years of PeMS road traffic data collected from the VDS (Vehicle Detection Sensor) in one of the big corridors in California. We merge all the scraped PeMS files into one file for thorough processing and combine with station metadata using the stationed field. We further analyze the data and derive or transform features that are required for our model from the raw data.

Our project involves a large amount of data and requires an efficient storage mechanism. The file format in which our data is stored should be more efficient in performance and storage. Therefore, we store the final processed traffic dataset into AWS S3.

2.3.1.2 Meta Dataset

The station metadata contains descriptive information of the traffic stations along freeways. The station ID identifies each one-hour reading of traffic. This information is present in another dataset, in the name of metadata files. The reported station metadata is in a tab-delimited format with the field names defined below.

Figure 9 Meta Dataset

2.3.1.3 Incident Dataset

The incident data was downloaded manually from the PeMS website dataset containing all the California Highway Patrol (CHP) incidents from all Caltrans. Each downloadable file includes all incidents that occurred in one day across California.

2.3.1.4 Weather Dataset

The Dark Sky API allows you to look up the weather from anywhere in the world. The list of city names in the Bay Area was taken to find the geolocation attributes of the cities from the Geopy library and used in combining latitude and longitude values from DarkSky weather API. All the relevant weather information is extracted using DarkSky API and persisted in AWS S3.

Figure 10 Data Collection Flow Diagram

2.3.2 Training Datasets

Traffic Dataset

The preprocessed traffic data looks as shown below. For January 2019 month alone, it contains nearly

four lakhs records and 24

columns.

Table 9 Traffic preprocessed data

In our case, our target variable is speed. Among 24 features, we removed the redundant, unnecessary features from our dataset using the feature engineering technique, and we kept the relevant features required for our analysis. So, after removing a few columns, the dataset looks as shown below. The first step we did is to group the rows for each station id.

Table 10 Traffic data after feature engineering

The second step we applied is to normalize the data. For Deep Learning models to converge faster,

it’s required to scale the data. The larger values can likely slow down the learning. We used the

Min-Max scaling technique from Python Sklearn library. This technique helps to scale the data

between 0 and 1. The below formula explains the scaling method applied to the dataset.

Figure 11 Screenshot of scaling function

We used a deep learning multivariate time series model for the prediction. Multivariate time series means there is more than one variable used for each time step. In our case, we used speed and occupancy as variables. Before applying machine learning models, time series forecasting problems must reframe as supervised learning problems from a sequence to pairs of input and output sequences (Brownlee, 2017). In Python Pandas library, shift function helps to transform time series data to supervised learning problems. This function helps to create copies of the column that are pushed backward or forward based on our requirements (Brownlee, 2017)**.**

We defined a function named prepare_data that takes input as a multivariate time series data and gives the output as a supervised learning dataset. Here, in this case, our function takes four input arguments.

- Data: Sequence of observations as a 2D NumPy array

- n_in: Number of lag observations as input (X).

- n_out: Number of observations as output (y).

- dropnan: Boolean whether or not to drop rows with NaN values.

In the end, the function returns the output as a data frame of the series framed from supervised learning. The below screenshot shows the input time-series data, and the function returns as supervised learning data. Here in our case, there are two variables: one is speed, and another one is occupancy. Therefore, it is a multivariate time series model.

Figure 12 Data after Minmax scaling

The above function returned as a supervised learning dataset, shown below. Here in our case, var1 is occupancy, and var2 is Speed. We gave n_in as 12 and n_out as 6. That means we are asking

the function to move 12 steps backward and 6 steps forward to learn the sequence of patterns in

case of time series data. Technically it means the number of previous time steps to use as an input

variable to predict the next time-periods.

Figure 13 Time Series data to Supervised Learning data

Weather Dataset

Similar to traffic data, historical weather data can be used as a training dataset. Historical weather

data can be downloaded from DarkSky API and combined with traffic data. Daily weather may

affect the traffic flow based on the locations. After merging with traffic data, we can identify how

the weather has an impact on congestion, and the target variable can be identified. Then, the calculated historical data can be used to train the models.

First, we combined traffic data station ids with weather data station id’s based on latitude and longitude nearest distance between two tables. Later on, we used merge as of function using Python Pandas library to map both the data from both the tables. The final joined data looks as shown below.

Table 11 Traffic and Weather data

As described before, the traffic data, the same steps we followed for traffic and weather data combination. For this dataset, we have five features, namely speed, occupancy, precipitation, wind speed, and visibility.

Table 12 Training data for Traffic – Weather

Incident Dataset

The preprocessed combined incident, weather, and traffic data look as shown below. The below-preprocessed data covers from 2015 January to 2019 December. The data contains nearly 3000 incident records for 101 freeway itself and nine essential columns.

Table 13 Combined Incident, Traffic and Weather data

We plan to use this combined multi-source data in predicting Incident on a given freeway. The

ideal target variable is Incident, which had derived from the provided incident description feature.

Each incident description was labeled. Incidents occurred due to traffic hazards, and incidents

occurred due to traffic collision.

To support the incident, we engineered new features from day of week, hour of the day from

timestamp which could support the target variable. Now to deal with the categorical data, we

implemented one-hot encoding and Label Encoder to encode categorical into numeric data. In

order to train any machine learning model, the input must be in numerical format.

Below is a small snippet of our encoded dataset. So, there exist a greater number of columns than

what we have shown here.

Table 14 Training dataset for Incident

Once we encoded the categorical data into numerical form, the resultant features were raised from 9 to 14. The training data is highly imbalanced; we balanced the data by applying the SMOTE technique. Thus, balanced data has given us a better view of training the machine learning model.

The key parameters in the model are:

Table 15 Key Parameters for incident prediction model

Key Parameters	**Their definitions**
Occupancy	This gives us information on how many vehicles were occupied on a lane at that station at a particular timestamp.
Speed	The average speed of the vehicles in past hour
day_of_week	This gives us information about what day it is. We Label encoded ranging from 1 to 7.
Hour of the day	Hour of that specific station
Hourly Visibility	Automated Sensor Visibility

Hourly Precipitation	Precipitation accumulation for every 60 minutes, in millimeters .
Station Id’s (400661,401255,401277,401472,401516)	Each Station Id is converted into One Hot encoding
Incident	Incident happened, indicated as 1 else 0

The way the model identifies value for those parameters is through search. There are two types of search, Randomized Search and Grid Search. We employed grid search though it is time-consuming but yields best results than coming up with random values.

The following figure shows the ideal train and validation of results for our model.

Figure 14 Train Test Split Process

2.4 Data Cleansing and Validation

2.4.1 Traffic Data

All data sources potentially include errors and missing values – data cleaning addresses these anomalies. Unclean data can lead to a wide range of problems, including linking errors, model misspecification, errors in parameter estimation, and incorrect analysis leading users to draw false conclusions. For the data cleaning process, we used the Pandas library present in python software.

- **Missing Data**: We did observe a small percentage of missing data in speed and total flow columns. Instead of cleaning the values using mean and mode imputation methods, we removed the entire observation whenever we encountered the missing data.

- **Irrelevant Data**: The traffic data contains more columns giving individual lane information about speed and total flow. We considered the average speed and total flow at the station and dropped the lane specific columns from the dataset.
- **Duplicates data**: We haven't encountered any duplicates present in the data.
- **Data type conversion**: Initially date column is stored as a string data type, we then converted to timestamp data type. We make sure all other data columns are having the right data type.
- **Data Filtering:** We filtered the data for four different freeways, namely 101, 280,680 and 880. In addition to that we filtered the data only for North direction alone.

2.4.2 Metadata

The station metadata contains cleaned data and all the features are in the right format. It

doesn’t require any cleaning process.

2.4.3 Incident Data

- Incident Data is one of the essential features of our dataset. The obtained dataset has

  timestamp in improper format, we converted it into proper format.

- The retrieved incident data from PeMS contains data of all the districts in California. For

  our analysis, we require only data related to the Bay Area. So, we filtered out the data that

  are not relevant to us.

Figure 15 Incident data

- There are some negative values that exist in the feature duration, which is to be avoided.

  Removing the records from the data may incur data loss. We tried replacing those negative

  values with zero.

- There are few changes needed for data type conversion. Such as duration, converted to

  integers.

2.4.4 Data Validation

Based on the literature review, we validated our data by applying unique rules and assumptions on it. The assumptions such as non-negative values of speed, number of cars and occupancy rates. If the data doesn’t comply with the assumptions, we specify that particular value as not available. Based on the constraints mentioned in the reference, we apply these constraints to our dataset for validation purposes.

Data-Type Constraints: Each column values should be in respective data types. For example, Boolean, numeric, datetime, etc.

Range Constraints: Values in a particular column should fall within a certain range. For example, speed values should fall between 2-100 mph.

Unique Constraints: A combination of fields must be unique across a dataset and it should not contain any null values. For example, in the traffic dataset, StationID is unique for the entire dataset.

Table 16 Data Rules and Constraints

Constraints Dataset Variables

Data

type, district are categorical data

.

Metadat
a

ıbers, station id, latitude, and longitude are
ta types. Freeway direction, city, state is
categorical.

Data-type
Constraint
s

Incident
data

Station-id, occupancy, speed, traffic flow should be
numeric data types. The timestamp should be in date-time
format. Direction.

d be numeric. The direction, location should
be the categorical features

nperature, wind speed, and precipitation are numeric
types. Visibility and wind direction are categorical data
types.

nperature, wind speed, and precipitation are numeric
types. Visibility and wind direction are categorical data
types.

Traffic

Range
Constraint
s

S
1

0-

Traffic
vehicl

Traffic
Data

- N or S

Day:
0-23

District -
4

Occupancy:

District - 4
Incident Data
Incident Month: 1-12
Incident Day: 1-31
Weather Data
Wind speed: 0-25 m/s
Wind direction: 0-360 deg
Pressure: 940- 1060 Mb
Temperature: 30 - 100 Fahrenheit Mandatory Constraints Traffic Data Station Id

There are various ways that we can perform data validation. In this project, based on the above unique rules, we used FME software for our validation purpose. The main advantage of this software is that it enables us to customize our own data workflows based on our requirements.

Additionally, we can reuse the above validation workflow any number of times. Below is the process flow diagram used for data cleaning and validation purposes.

Figure 16 Process flow diagram used for data cleaning and validation

Our datasets are in raw format. Based on user-defined schema mentioned in the data source section,

assumptions, and rules, we parse and validate each field present in the datasets. In the above

diagram, we also check the data using summary statistics like mean, median, etc. We visualize the

data using histograms and bar plots to detect outliers.

Weather data Validation:

Following were the validation check points for the weather data. We applied the below important rules in validating data from the meteorological resource center (*The Meteorological Resource Center*).

Table 17 Weather data validation

Feature	**Screening Criteria to flag the data points**
Temperature	• On a monthly basis if the recorded value is less than or greater than local record value. We need to flag the point. • If it is greater than a 5 °C change from the previous hour • If temperature does not vary by more than 0.5 °C form 12 consecutive hours
Dew Point Temperature (DPT)	• If DPT is greater than given ambient temperature in the given time period.
Precipitation	• If precipitation is > 25 mm in one hour • If it is < 100 mm in 24 hours
Wind Speed	• If the value is less than zero or greater than 25 m/s • If it does not vary by more than 0.1 m/s for three consecutive hours • If it does not vary by more than 0.5 m/s for 12 consecutive hours

Visibility • If visibility value is less than 100 meters

2.5 Data Transformation and Tools

There is a saying that data is like crude oil for the machine learning models, which implies it has to be refined into features to train any models (Koehrsen, 2018). The process of transforming raw data into a ready-to-use form is known as data transformation. With plenty of available data, there is a huge potential to find business value from it. However, transforming the data cleverly and orienting the data around the business user needs is a challenging task.

- **Categorical encoding:** This representation of categorical variables to numeric vectors is known as label encoding. Label encoder converts each value in a column to a number. In our traffic dataset district, direction, lane_type features are converted to numeric values by using the label encoding technique. We used Scikit learn label encoder library for converting categorical variables.

Figure 17 Direction feature Label encoding

- **Skewed Data:** Skew is the degree of distortion from a normal distribution. It means that the distribution tends to have a long tail on one side or the other. Below distribution plots consist of skewed data. We applied log transformation and square root transformations on total_flow, occupancy and speed variables to make them symmetrical.

Figure 18 Data distribution of Total flow, Occupancy and Speed

- **Normalization:** Normalization or scaling refers to bringing all the columns into the same range. While implementing the deep learning models, the first step we need to do is to normalize the data. For this, we applied a min-max normalization technique. This technique rescales the data to values between 0 and 1. We used a sci-kit learn min_max scaler library for normalizing the data.

- **Basic Statistics:** We calculate basic aggregations with functions like mean, median, min and max. We visualize them to see the spread of data. In Pandas, the function called describe () describes the numeric type columns in the data frame to provide basic statistical information. Other packages such as Matplotlib and Seaborn need transformed data for more granular and customized visualizations. With our traffic data, we can visualize the flow lane-wise or know the weather information in a certain period.

2.5.1 Tools

For us to work together as a team and to process huge volumes of data, we need to have a cloud platform in place to store all our data files in a common storage location and use the same cloud products (for data ingestion, data processing) to make our code compatible with each other. We plan to use Amazon S3 to store the data files. We will have a separate high-end virtual instance that can host an IPython notebook to write and test our code. For processing big data, we will use the Spark framework. We had a High Processing Computer account, where we can install all the tools needed for our project. We used pre-defined cloud solutions for data storage and HPC machines for data processing. Using cloud products is advantageous since it helps our application to be fault-tolerant, scalable, and distributed in nature.

- **Storage:** Amazon S3

- **Processing/Transformations:** Jupyter Notebook, Apache Spark, Amazon Data pipelines,

EM
R

- **Data Visualization:** Tableau, Apache Superset

- **Libraries**: Seaborn, Matplotlib, Numpy, Pandas, Tensor flow, Keras, Folium.

- **Graphical User Interface:** HTML, Flask, Javascript.

2.6 Raw data visualization

Figure 19 Heat map of Speed by Freeway vs. Hour of the day

The above heat map gives us information about the traffic conditions on different freeways.

Red color indicates the speed is low at that particular time and whereas green is high speed. Low speed means there is high occupancy in that area. It looks like freeways 101, 280, 680 and 880 highways have high traffic between 6 am till 10 am time and between 3 pm till 7 pm time.

Figure 20 Plot of Speed by Hour per day of week

According to the hourly speed it is possible to define which hours can be considered rush hour. In addition to that, we can see that on weekends the speed is comparatively higher than on weekdays.

From the above graph, we can define the rush and no-rush hours.

- Morning rush hour: 06: 00 - 09: 00

- No-rush hour: 09: 00 -12: 00

- Evening rush hour: 14: 00- 18: 00

Figure 21 Plot of number of incidents in Bay Area

The X-axis on the above graph shows the description of the type of incidents, whereas, y-axis

gives the count of the number of incidents that happened in that specific area. Most of the incidents registered in the Bay Area were traffic hazards and traffic collisions. Traffic hazard alone accounts

for nearly 34,071 cases.

Figure 22 Bar plot of most incidents in Bay Area

The above bar graph gives us insight on most incident occurring areas. The x-axis gives the count

of the incidents, whereas, y-axis gives the areas. Within the Bay Area San Jose and Oakland had

more incidents. Predominantly San Jose has registered more incidents in district four than any

other
.

Figure 23 Bar plot of most dangerous locations in Bay Area

Most of the dangerous locations resided near San Francisco. Both 180 E /Treasure Island off-ramp and on-ramp had registered the highest number of cases.

Figure 24 Incident Heat map

The above heat map shows the occurrence of incident time in each day of the week in the year

2019. The above graph clearly shows the evidence that the high number of incidents are occurring

on working days. The majority of them occurred during office hours, leaving time, which is

between 3 PM to 7 PM.

Weather data:

Initially, we tried to understand the behavior of weather using pair plots. Pair plot is a bivariate analysis visualization technique used to study the relation between two variables.

Figure 25 Weather data pair plot

One clear insight we could draw from the above plot is, the only variable temperature has a normal

distribution

.

On analyzing the data, we visualized how weather affects traffic, emphasizing on parameters like wind speed, visibility and traffic speed. The visualizations were plotted on a dashboard, with multiple graphs where the speed and occupancy in each hour of the day, wind speed affecting traffic speed, visibility affecting traffic speed and the analysis of average visibility in each hour of the day are visualized. We have the option to filter data for each day for each freeway in the dashboard. This helps us understand how weather affects traffic for each freeway on each date and hour of the day.

Figure 26 Traffic and weather dashboard

The interactive dashboard shown below helped us to analyze the relation between lane occupancy

and incidents. We have observed a pattern of increase in occupancy after the occurrence of

incidents. Based on this analysis, we have decided to include incident data as a feature for our

model
.

Figure 27 Traffic-Incident dashboard

Figure 28 Traffic-Weather- Incident dashboard

The above dashboard is from three datasets, namely traffic, weather, and incident data. We combined these three tables using metadata files from all three sources. The first left geographic shows the freeway path. The bottom left figure describes the average speed and occupancy at each hour of the day. From the graph, we could infer that the higher the speed, the lesser the occupancy is. The rightmost figures show the number of incidents by County and weather information for

various dates and freeways. We used Tableau software to visualize the dashboard.

3.1 Project Organization

Data science projects are often an iterative process that involves many backs and forth

testing of new ideas, tweaking new features and tuning hyperparameters for our models. Our end

goal is to develop a model that satisfies all our business insights with good prediction accuracy.

Accuracy is hugely task-dependent, and it involves many development and testing phases. This

clearly shows that the traditional software development lifecycle will not be useful for our project.

After several analyses, we followed Gartner's principle and designed our project to have the

following phases (Figure 28). This iterative approach to continual improvement phases will help

us to improve our model.

Figure 29 Data Science Project Life Cycle

Step 1: Requirements Analysis & Literature review

This step is very important to start any software development project. Requirements analysis helps

us to understand more about our business insights and to define our problem statement very clearly.

It helps to identify our functional and nonfunctional requirements for our project. The literature

review helps us to understand the various existing approaches towards solving a problem and provides a foundation of knowledge on a topic. We identified the requirements to carry out an

effective traffic prediction after a thorough literature review.

Step 2: Data Collection and Preprocessing

As we said earlier, data scientists spend 80% of their development time in preparing the data,

making this phase a crucial part of the project. Finding the perfect dataset to work with is vital

since it can have a positive or negative impact on our project goals. We identified relevant traffic

datasets and various other data sources. We need to preprocess the collected data, to make it ready

for the consumption of the machine learning models.

Step 3: Feature Engineering and Data Selection

Selecting the right features from a dataset helps to remarkably improve the performance of a

model. Feature engineering and data selection are very significant steps in using machine learning

algorithms and building predictive models. This phase also involves dimensionality reduction and data normalization. Domain knowledge is helpful in this phase to extract the appropriate features we require.

Step 4: Model Development

This step in the data science project life cycle involves applying various machine learning algorithms on our cleaned data. We need to know about various machine learning libraries and decide the correct algorithm for our data. This is the most exciting phase in our project, and data scientists also tend to tweak different hyperparameters to achieve better efficiency in their developed models.

Step 5: Model Testing and Validation

After model development, we should evaluate our model by calculating metrics like,

1. R2 to measure goodness-of-fit

2. Error scores like MAE (Mean Average Error), or RMSE (Root Mean Square Error) to

measure the distance between the predicted and observed data points.

This phase tests the developed model for accuracy and validates it to ensure that the business goals

are satisfied.

Steps 2 - 5 will be an iterative process that may need to repeat until we achieve better accuracy for

our model.

Step 6: Model Deployment

After development, our model has to be deployed in any cloud platform and made accessible for

the public. We can create web services to access our model or deploy it as a pickle file and allow

users to load and run it.

Step 7: Model Prediction and Decision Making

After finalizing our machine learning model, we can make predictions using our test data.

Identifying insightful and potentially useful patterns from the resulting data will help us to make sound business decisions. One of the essential skills required in this phase is to be able to derive actionable insights from our data.

As our academic project needs a transparent and iterative product management method for a self-organizing team, we decided to follow Agile development. We created project modules for each individual in the team and are working on the assigned tasks continuously. To emphasize more on the work to be done, we decided to use Kanban methodology over Scrum. Based on our requirements, Trello, an online project management and visualization tool developed for Kanban methodology, became more appropriate for our team. We created and monitored all our project modules and tasks using Trello Dashboard. Any changes or development in our modules will be tracked with the Trello dashboard. The source code for the corresponding modules in our project is maintained in GitHub. All the project modules will be reviewed by peers before the final commit.

3.2 Resource Requirements

We will need to analyze large amounts of traffic data and processing it traditionally will not be sufficient. Hence, we decided to use high-performance computing with cloud services.

Cloud services make it easier and more affordable to manage and maintain big data. With Infrastructure as a Service (IaaS) in the cloud, we can use the resources required to analyze our data whenever we need it. IaaS allows us to design the environment we need and to store and process the data. It also helps us to deploy the developed model for online serving.

In this project, we are predicting traffic congestion using four challenging technologies like big data, deep learning, in-memory computing (Spark), and high-performance computing (GPUs). Big data refers to the data with the four V’s: volume, variety, velocity, and veracity. We are collecting large amounts of streaming data which should be processed at high speed. GPUs provide parallel computing to speed up computations. Processing the data with Spark, which deploys in-memory computing, and developing the model to run on GPUs will help us to process real-time data at a faster rate. Deep learning is one of the most leading-edge technologies used for transport-related

predictions. Hence, in this project, we decided to use cloud services that can handle real-time data.

3.3 Project Schedule

3.3.1 Project Organization

We did our project organization with the help of a Work Breakdown Structure (WBS). A WBS

helps in organizing the project into multiple levels based on hierarchy, efficiently accomplishing

the project objectives (Hans, R. T., 2013). Our plan has six main parts for easy organization.

Figure 30 Work Breakdown Structure

3.3.2 Project Timeline and Schedule

Figure 31 PERT Chart

We use the PERT chart to manage and monitor our project from start to end. The project timeline begins on September 1, 2019 and the end date is considered as March 19, 2020. The project schedule was divided into 8 parts, namely:

1. Literature survey

2. Requirement analysis

3. Data collection and preprocessing

4. Environment setup in the

cloud

5. Feature engineering

6. Model development

7. Model validation and testing

8. Model deployment

A systematic schedule was charted out using a PERT chart, which denotes each project schedule step and its respective timeline. PERT (Program Evaluation and Review Technique) chart is a project management tool that helps to schedule, organize and coordinate tasks within a project (Bird, 2019). PERT chart helps to identify the minimum time required to complete the entire project. Using this PERT chart, we have mapped our project schedule and the critical path of our project. The critical path is shown using red arrows and project flow is represented by black arrows in the chart.

4.1 Problem

Formulation

In this project, we aim to address a severe and worsening problem in today's world: traffic congestion. All over the world, traffic congestion has been increasing, despite the countries being developed or developing. It poses a significant threat to urban life quality. Traffic congestion causes a reduction in vehicle speeds, causing increased journey times, fuel consumption, and other operational costs, all eventually adding up to environmental pollution. The root cause of traffic congestion is the elevated use of private automobiles. Private vehicles enable personal flexibility and are a symbol of higher statuses in developing countries. During rush hours, an individual occupant in private vehicles causes more congestion than multiple people on a bus or any other public transport. Hence, they are not efficient modes of transportation. Problematic road designs, inefficient management by authorities, and inaccurate information on traffic conditions can further worsen the situation. Over the last few decades, there has been a rapid hike in the number of vehicles in developing countries due to multiple factors. Some of the major factors that result in this are increased purchasing-power of middle-tier socioeconomic classes, reduction in prices, and

increased use of vehicle availability.

The cost of traffic congestion is very high. According to INRIX 2018 Global traffic scorecard,

Americans lost 97 hours in traffic congestion, costing the country $87 billion in 2018, which is an

average of $1348 per driver. San Francisco is the 65th most congested city in the world, and the

8th most overcrowded in the United States, according to the INRIX 2018 report (INRIX, 2018).

Figure 32 INRIX 2018 Global traffic scorecard

Source: INRIX (INRIX, 2018)

According to the study conducted by Texas A&M University and the Urban Mobility

Report, people in the San Francisco Bay Area lose about 253,838 hours in traffic. On further analysis of the Urban Mobility Report, we can see that congestion happens mostly during the peak hours of traffic, which are early morning hours and evening hours on weekdays. Late nights, very early mornings, and weekends have zero traffic congestion.

Figure 33 Annual Hours of Delay in San Francisco in 2017

Source: Urban Mobility Report (Urban Mobility Report, 2019)

Figure 34 Congestion times

Source: INRIX Scorecard (INRIX, 2018)

Based on the study, congestion costs during the year 2017 are in the table below. The congestion costs are so vast that there is an enormous impact on the economic and social aspects of the individual, and eventually, the Nation too.

Figure 35 Congestion Costs

Source: INRIX San Francisco Score Card (INRIX, 2018)

Cars still are the primary mode of transportation for Americans. Due to the lack of an efficient public transport system, private vehicles are always preferred by Americans for transport. An increased volume of cars on the road is not just the reason for congestion. Other primary reasons are the on-road jams caused due to poor signal timing, bad driving habits, accidents, lane merging, constructions, to quote a few. According to the analysis, up to 40% of congestion is caused by bottlenecks, and accidents cause 25 % of congestion. From the below statistics, we can deduce that only 10% of the population uses public transportation as their primary mode of commute

(Nationwide,

2019).

Figure 36 Cars in American Commute

Source: Statista (Statista, 2019)

The way people commute to work has not changed much over a decade, as we can realize from

the below graph. In 2006 and 2016, the most common means of transport remains to be a personal

car/truck or van.

Figure 37 How Americans commute to work

Source: Statista (Statista, 2017)

In a survey conducted by the Washington Post, the average American commute increased to about

27 minutes in 2018, which is the highest ever, according to data from the U.S. Census Bureau.

This survey proves that an average commuter spends 20 minutes more per week than they did ten

years ago, which adds up to 17 hours more per year (Ingraham, C., 2019).

Figure 38 American commute through years

Source: Washington Post (Ingraham, C., 2019)

The growing commute times in the San Francisco Bay Area is a major issue which needs to be fixed with the best possible methods. There has been a steady increase in the commute times in the Bay Area throughout the years. Predicting traffic congestion considering weather and incident details will be more accurate than predictions with just the traffic data. Considering the major freeways in the Bay Area like Freeways 101, 280, 680 and 880 will help in gaining better focus and identifying the major bottlenecks in the daily commute. The below study shows the growing commute times in major Bay Area cities.

Figure 39 Growing commute times in Bay Area

Source: Commercial Cafe (Baldassari, E. ,2019)

The ways to tackle congestion include improving road quality, correcting the intersections,

improving road signs and markings, changing working hours, synchronizing traffic lights, and

establishing efficient public transport systems.

4.2 Foundation of Proposed Solutions

Traffic congestion has been a severe and worsening problem in much of the world, representing an undoubted menace to the quality of human life. Congestion also results in a reduction in entire traffic speed, eventually increasing travel time, fuel consumption, and air pollution. One of the most efficient solutions for congestion is to improve traffic infrastructure and vehicle management.

There are many disadvantages in our existing urban road systems like an inefficient design for intersections, lack of suitable signs, difficulty in managing the cycles of traffic signals.

Constructing new roads, enhancing carpool lanes can be useful to some extent. However, building many roads and urban expressways may be counterproductive and will remain as a long-term project and may result in making congestion even worse. Implementing smart traffic signals can result in saving a considerable amount of time and fuel consumption. As we know the most common solution to avoid traffic congestion are,

1. Improve road usage using modern technologies like smart signals, centralized monitoring.
2. Expand carpool, HOV lanes, and car-sharing.
3. Utilize alternative approaches for commuting like public transit and bicycling, etc.

All the above-stated options will be most effective only when applied in a group. As mentioned earlier, the US population has increased by 30% in recent years, and the community in the Bay Area tends to grow without any fall. Also, the private car remains the most common way of transportation in the US. According to the current commuter trend in California, even though there are many roads and carpool lanes, additional drivers will tend to occupy the new space. If there is a continuous imbalance of potential drivers and road space prevails, this will make congestion permanent in urban areas.

In recent times traffic experts say, increasing fuel cost and congestion pricing can reduce traffic congestion. Increasing toll lanes and charging for the road based on their demand will not make

everyone happy. All the commuters will not be able to pay more to commute for their daily work.

4.2.1 Proposed Model:

About 87.9 percent of America's daily commuters use private vehicles, and millions are

wanting to move at the same time of the day. The underlying problem of America's road system is

a lack of capacity to handle peak-hour loads. The definition of congestion is waiting in lines, and

it is found in all growing major metropolitan regions (Downs, A. ,2018). Only one perfect solution

to overcome bottlenecks without any contradictions is to make better use of the roadways we

already have. Building an Intelligent transport system in smart cities will help to reduce congestion

eventually. In our model, we propose users with the best time to commute and the best mode to

commute. Our model works on the concept that more time a car spends on the road leads to

congestion. In our proposed solution, we give suggestions to our users to reach their destination in

a short interval of time. Congestion is a warning signal to expand public transit. In our model, we also compare our best route with other modes of public transportation (Greenfield, A. 2014). In

peak hours, we suggest our users take public transit if the time taken for the commute is

comparably lesser than private car transportation.

Our model considers various data sources like weather and incident data to predict traffic

congestion. Traffic congestion statistics states that 40% of congestion is caused by "bottlenecks,"

about 25% by "traffic incidents," and 10% by "work zones" (Traffic Congestion Statistics, n.d).

Hence, it is vital to consider the incident data in our analysis. Our model forecasts traffic

congestion in any particular area using the previous five years of traffic, weather, and incident

data. Hence, we believe that results predicted our model would be more accurate, and it will

immensely help the commuters who need to plan for their works well ahead of time.