Spatio-Temporal Analysis of Urban Data and its Application for Smart Cities

Prakriti Gupta

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Haibo Zeng, Chair
A. Lynn Abbott
Bert Huang

July 7, 2017
Blacksburg, Virginia

Keywords: Traffic Analysis, Electric Vehicle Charging Demand, Quantitative Criminology, Spatio-Temporal Analysis, Data Mining

Spatio-Temporal Analysis of Urban Data and its Application for Smart Cities

Prakriti Gupta

(ABSTRACT)

With the advent of smart sensor devices and Internet of Things (IoT) in the rapid urbanizing cities, data is being generated, collected and analyzed to solve urban problems in the areas of transportation, epidemiology, emergency management, economics, and sustainability etc. The work in this area basically involves analyzing one or more types of data to identify and characterize their impact on other urban phenomena like traffic speed and ride-sharing, spread of diseases, emergency evacuation, share market and electricity demand etc. In this work, we perform spatio-temporal analysis of various urban datasets collected from different urban application areas. We start with presenting a framework for predicting traffic demand around a location of interest and explain how it can be used to analyze other urban activities. We use a similar method to characterize and analyze spatio-temporal criminal activity in an urban city. At the end, we analyze the impact of nearby traffic volume on the electric vehicle charging demand at a charging station.

Spatio-Temporal Analysis of Urban Data and its Application for Smart Cities

Prakriti Gupta

(GENERAL AUDIENCE ABSTRACT)

Because of the ubiquity of the Internet and smart devices, a tremendous amount of data has been collected from multiple sources like vehicles, purchasing details, online searches etc., which is being used to develop innovative applications. These applications aim to improve economic, social and personal lives of people through new start-of-the-art techniques like machine learning and data analytics. With this motivation in mind, we present three applications leveraging the data collected from urban cities to improve the life of people living in such cities. First, we start by using taxi trip data, collected around a given location, and use it to develop a model that can predict taxi demand for next half hour. This model can be used to schedule advertisements or dispatch taxis depending upon the demand. Second, using a similar mathematical approach, we propose a strategy to predict the number of crimes that can happen at a given location on the next day. This helps in maintaining law and order in the city. As our third and last application, we use the traffic and historical charging data to predict electric vehicle charging demand for the next day. Electricity generating power plants can use this model to prepare themselves for the higher demand emerged because of the increasing use of electric vehicles.

Dedication

I dedicate my thesis to my parents, Mrs. Nisharani Gupta and Dr. Prakash Gupta, siblings Jiya and Vijay Jijaji for respecting my decisions, supporting me in my endeavors and constantly motivating me to pursue my goals.

Acknowledgments

I would like to thank my advisor, Dr. Haibo Zeng for giving me an option of problem statement to choose from, which I had an interest on but no experience. Having said that, his constant guidance, support and patience helped me to learn, explore and work on a new subject of data mining. After working with him, I learned the right approach of solving a research problem and I am confident that this will help me to solve a problem more confidently in future. I would like to thank Dr. Bert Huang and Dr. Lynn Abbott for their interest in my work and for being approachable to answer my questions.

I would like to thank EVgo for providing us the necessary data to support our research on the analysis of electric vehicle charging demand.

I thank Aditya and Tomilayo for collaborating with me on the initial work on crime analysis that helped me taking forward the work to the analysis presented in the thesis.

Last but not the least, I would like to thank my family and friends for their continuous support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Purpose and Goals

A staggering 70% of the world population is expected to be living in the cities by 2050 [69]. In order to fulfill the end-meets of this urban crowd, rapid industrial, infrastructure and economic development is taking place around the globe. Green areas are being concretized for residential and commercial use. Roads are being constructed to accommodate the increasing number of vehicles. However the rate of infrastructure development is not same as that of the increasing vehicles. Because of the slow process of large infrastructure development, the cities are not instantly ready to adapt to the rapid urbanization and there are many challenges that urban cities still face. One of the biggest global environmental and economic issues is the use of petroleum and other gasoline resources for conventional fuel vehicles.

To tackle the problems associated with depleting natural resources and climate change, use of alternate energy sources like solar energy, wind power, bio-fuels are being encouraged. The Paris Agreement[1] signed recently by many countries is one of the biggest global actions taken in this regard. In order to curb the use of fossil-fuels and prevent global warming, as an alternative to conventional Internal Combustion Engine (ICE) vehicles, electric vehicles (including plug-in (PEV) and hybrid electric vehicle (PHEV)) have been encouraged[30]. With incentives and programs launched by the government to promote electric vehicles[2], an overall positive trend has been observed (refer to Figure 1.1) year after year against the number of electric vehicles purchased in the US. With the increasing penetration of electric vehicles in the energy market, a higher demand is received by the existing power grid for power generation. This places a significant impact on the grid not only for energy generation but also for operation, maintenance and extension of the infrastructure. Other challenges include problems in domain related to transportation, epidemiology, sustainability

[1] `http://unfccc.int/paris_agreement/items/9485.php`
[2] `https://energy.gov/eere/vehicles/downloads/ev-everywhere-grand-challenge-blueprint`

etc. Statisticians and domain experts have worked toward solving these problems for a long time now. But they were mostly dependent on data collected from surveys.

Because of the recent advancement in smart sensor technologies and cloud storage, data can be collected from different devices like vehicles, phones, buildings, parking stations etc. Use of these data has found to be extremely useful and today, computer scientists, statisticians, domain experts have come together to solve problems using this urban data, which has gathered attention towards the area of urban computing [69].

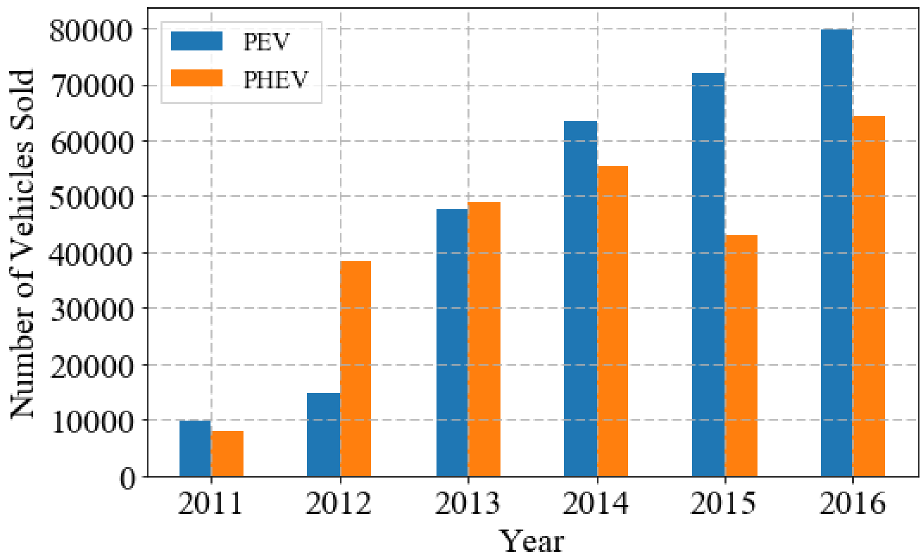


Figure 1.1: Total Plug-in (PEV) and Hybrid (PHEV) Electric Vehicles Sale in the US

The use of big data is beneficial to the smart cities in various ways like better transportation systems, efficient resource utilization [1], remote monitoring and operation etc. It is necessary that while we are pushing the boundaries of the human imagination for the convenience of the people, at the same time we should be vigilant enough for their safety as well. An increase in criminal activities was observed from 2014 to 2016 [22] by collecting the criminal data from 43 large US cities. An increase in 16% in homicide was observed when crime reports were collected from 60 cities in US from 2014 to 2015[3]. Thus, it becomes vital to analyze the crime and predict criminal activities in advance for public safety.

Given the problems associated with smart cities, in this thesis we use various urban data collected from different application-domains and present a framework to analyze them which can be subsequently used to solve urban city problems. Specifically, we concentrate on following three topics:

- Modeling traffic demand near a Location of Interest (LoI)

[3]`https://fivethirtyeight.com/features/scare-headlines-exaggerated-the-u-s-crime-wave/`

- Characterizing crime in an urban city
- Predicting electric vehicle charging demand for a charging station

An introduction to these topics is presented in detail in next section.

1.2 Modeling traffic demand near a Location of Interest (LoI)

One of the important areas of research in the context of smart cities is transportation. With the advanced technologies like GPS enabled vehicles, real-time updates and easy accessibility in form of mobile application, it has become easy to record and track the human mobility. Modeling the traffic means presenting a systematic framework where we can use data to analyze traffic and solve urban transportation problems. It includes identifying traffic speed and cause of traffic congestion [58], better scheduling of road development and maintenance to avoid traffic congestion [28], identifying shortest and fastest routes [65], predict taxi demand for economical ride-sharing activities [31] etc. Studying these problems are regarded as one of the key factors for policy and decision making by local government. In this work, our focus is on building a spatio-temporal model for analyzing taxi demand near a location of interest.

1.3 Characterizing Crime

Predictive policing [18] is the term given to practice of predicting crime beforehand using data mining, statistics and domain knowledge. It plays an important role in taking actions in order to prevent crime. In this work, we use the similar framework presented in the previous section to characterize and predict different criminal activity in an urban city using historical crime data. Along with the historical crime data, we also include the effect of weather conditions, presence of local businesses and police stations into consideration for analyzing crime.

1.4 Predicting electric vehicle charging demand

The charging station networks, that provide charging outlets in residential and commercial areas for the drivers to charge their EVs, have to purchase electricity in advance in today's electricity market, called as the day-ahead market. In this setting, they have to optimally bid [23] for the purchase of electricity from the power-generation plants. In order to reduce the overall operating cost and adapt to the increasing use of EVs, it becomes necessary to forecast charging demand in an immediate, near and long-term time frame. In this work, we analyze the impact of nearby traffic on the charging demand of commercial charging stations in an urban city and predict demand for the next day.

1.5 Data Mining and Framework

Data Mining is the computational process of extracting knowledge from data that otherwise is not apparent by just looking at the data. Figure 1.2 describes the detailed steps of the data mining framework used in this research. We describe them in detail in the next sub-sections. Python's sklearn library[4] was used at each of these steps for implementation and the code is publicly available at `https://github.com/prakritigupta/thesis`.

1.5.1 Data Preprocessing and Feature Extraction

As the name suggests, the first step is to preprocess the raw data so that it can fit to our model analysis. This involves cleaning irrelevant and incorrect data. Since we use different types of data in one model, preprocessing also involves merging heterogeneous data. After cleaning and merging, we extract the important features from the raw data that are most relevant for our model. This is one of the most important steps in the process and extracting correct features have a significant impact on model performance[45, 44]. Since machine learning algorithms do not converge properly if different features are at different scales, it is necessary to perform feature scaling on the data before the model is trained [29]. In this work, we have applied min-max scaling [51] wherever applicable as it gives equal importance to each feature. It is given by the equation:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1.1}$$

where x_{min} and x_{max} are the minimum and maximum value of the respective feature in the training dataset respectively, x is the value before scaling, and x' is the value after scaling.

Once the features are extracted and transformed data is ready, we split the complete data into train, validation (used during model tuning) and test dataset (used for model evaluation). This completes the data-preprocessing and feature extraction step.

1.5.2 Regression Algorithms

Before going into details of model development and tuning - which involves developing a model using a machine learning algorithm, we will discuss the five naive regression algorithms

[4] `http://scikit-learn.org/stable/index.html`

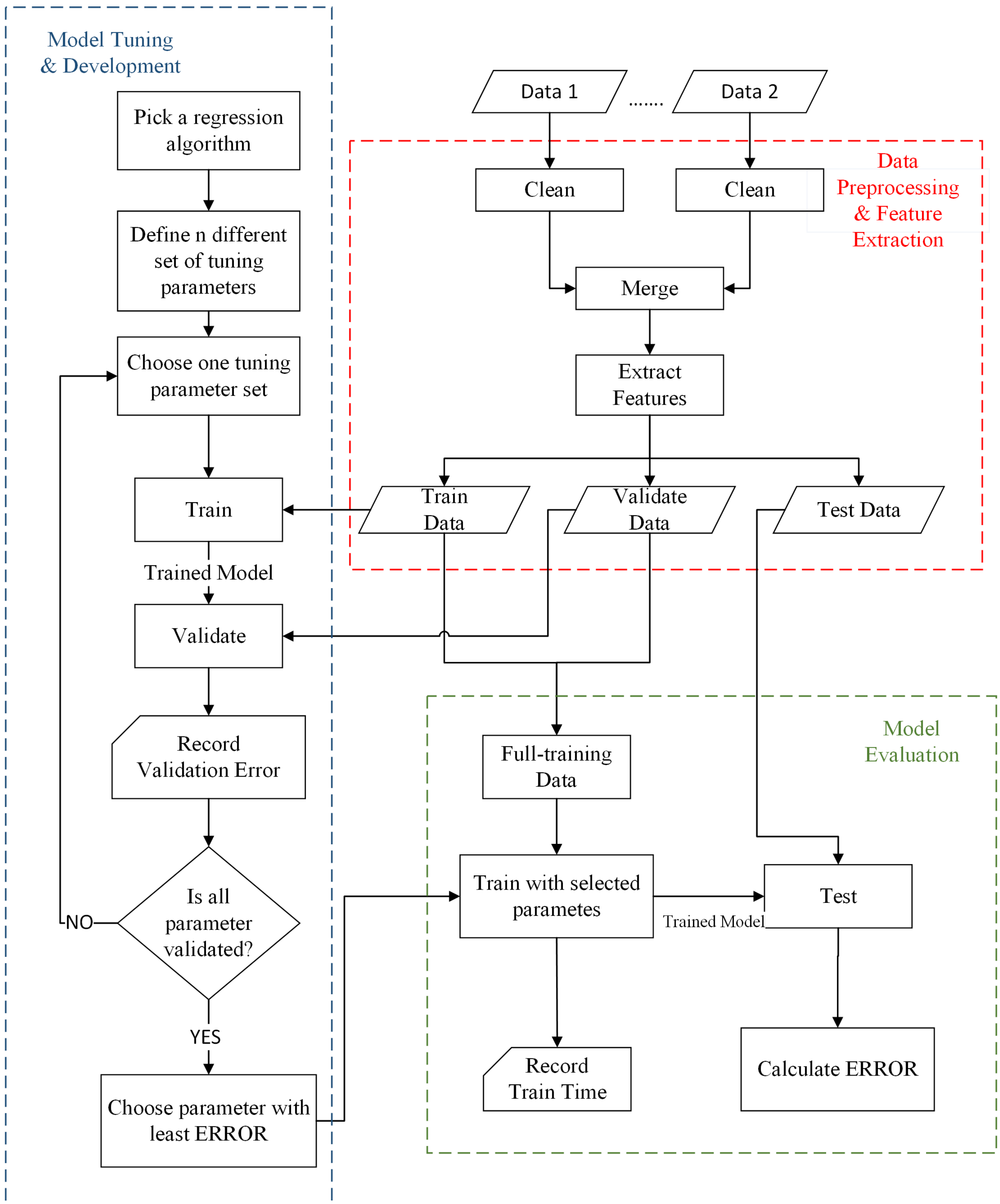


Figure 1.2: Detailed steps of data mining framework used in this research

that has been used throughout this work. A very brief introduction of each of these supervised machine learning algorithms is described below:

1.5.1.1 Linear Regression

Linear Regression is one of the simplest and oldest methods for solving regression problems. It aims to find a linear relationship between its input and output variables and is given by:

$$Y = w_1X_1 + w_2X_2 + \cdots + w_nX_n + \beta \tag{1.2}$$

where X_i is the value of i^{th} dimension of a n dimensional input vector X, Y is the output variable, which is to be predicted, w_i is the weight of feature X_i and β is the intercept term. Higher the weight, higher is the effect of X_i on Y.

It finds the relationship by trying to achieve a goal, known as objective function. The objective function generally is to keep the squared difference between the true and predicted value as least as possible. There are numerous ways to estimate this relationship; we have used two of them here - Ordinary Least Square and Stochastic Gradient Descent [68] referred as LR and LR_SGD respectively. In absence of any constraints put to the model, values can be negative. For simplicity, in our analysis we forced the value to 0, whenever a negative value was predicted.

1.5.1.2 k-Nearest Neighbor

k-Nearest Neighbor (kNN) [3] is based on the idea that the output value of an input is similar to its neighbor. Particularly, weighted kNN says that higher a training point is closer to a given test point, higher is its effect on the test-point output. The closeness is calculated by a distance measure and is given by:

$$d(m, p, q) = \left(\sum_{i}^{n}(|p_i - q_i|)^m\right)^{\frac{1}{m}} \tag{1.3}$$

where p and q are two data points and $d(m, p, q)$ is the distance between them calculated by taking an order $m \in \mathbb{R}_{>0}$ and n is the dimension of a data point.

The output Y_T for test point T can then be given by equation:

$$Y_T = \frac{1}{k}\sum_{i=1}^{k} W_i Y_{X_i} \tag{1.4}$$

where W_i is the weight associated with each neighbor and is inversely proportional to the distance d, Y_{X_i} is the output of a data-point X_i and is weighted average of values of k closest neighbor, and T is a test-point. The performance and complexity of the model depends upon the value k and has to be tuned meticulously as per the given data to get better performance.

1.5.1.3 Decision Tree

Decision Tree consists of building up an empirical tree containing rules, qualified by training data [33]. The non-leaf nodes (condition nodes) contains conditions on data features. For a new test point, the model traverses down the tree to reach the leaf-node. The terminal node contains the class of the test data (classification) or the output value corresponding to it (regression). The criterion we choose for solving regression problem in our analysis is RMSE and is discussed in Section 1.5.4.

1.5.1.4 Support Vector Regression

Support Vector Regression (SVR) [52], a type of Support Vector Machine model [10], is a supervised machine learning algorithm which aims in finding a function f that can give the output values that are deviated from its true value by at most ξ. The function f is similar to Equation 1.2. Without going into too much detail, the objective function to estimate f, by optimizing $\{w_1, w_2, \ldots, w_n\}$ as explained in [52], contains two parameters C and ξ. These parameters describe the penalty parameters for the model and have to be tuned for achieving better performance. Similar to the distance measure we saw in kNN, SVR also requires a measure that can tell how similar two points are. This is known as kernel, and has to be set depending upon the data. In our analysis, we tried linear, rbf, polynominal and sigmoid kernels[5] on our training data and choose the one that performs the best on the validation set.

1.5.1.5 Multi Layer Perceptron

Multilayer Perceptron (MLP) [48] is an artificial neural network technique, that maps inputs to a corresponding output value. It consists of three layers: input, hidden and output layers. Each layer is weighted sum of the previous layer. The final output, in case of a regression function, is given by:

$$y = f(O_h)$$

where f is the activation function. For our analysis, we use a Rectified Linear Unit (RLU) and is given by:

$$f(x) = max(0, x) \tag{1.5}$$

The reason for choosing this activation function is because it keeps the output value always ≥ 0, that goes along with the predicted quantities in our analysis. O_i represents the output of hidden layer i and h represents the total number of hidden layers, and it has to be tuned accordingly. Higher the number of hidden layers, more complex is the model.

[5] `http://scikit-learn.org/stable/modules/svm.html#kernel-functions`

1.5.3 Model Tuning and Development

Like discussed in previous section, most of the supervised data mining models we deal with in this work have parameters that must be tuned to the given data. For this purpose, after picking an algorithm, we define a finite set of parameters as required for that algorithm. We then train the model with every possible combinations of the parameters, and record their performance on the validation dataset. The error can be measured by one of the evaluation metrics discussed in the next section. We then finally select the parameter value for the final model that gives the lowest validation error. Selecting a model with parameters that performs the best on validation set gives a better estimation of how well the model can perform on test dataset[6].

1.5.4 Model Evaluation

This is the last step of the data mining process, where we evaluate the performance of one model over another. There are standard metrics that are being used for evaluating regression models. For our regression analysis, we use Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and SMAPE (Symmetrical Mean Absolute Percentage Error) defined as:

$$\begin{aligned} RMSE &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - o_i)^2} \\ MAE &= \frac{1}{N}\sum_{i=1}^{i=N}|y_i - o_i| \\ SMAPE &= \frac{1}{N}\sum_{i=1}^{i=N}\frac{|y_i - o_i|}{y_i + o_i} * 100 \end{aligned} \tag{1.6}$$

where y_i and o_i represents actual and observed output of i^{th} test point respectively and N is the number of instances in the test data.

Where RMSE and MAE are on absolute scale, SMAPE gives the percentage error. One of the important criteria for selecting SMAPE is because it penalizes more for under-forecasting than over-forecasting[7], which fits to our demand forecasting scenario where supply can be more than demand but not vice-versa. It has already been used in EV charging demand forecasting in [35].

In order to show the robustness of our model across data from different seasons and months of year, we perform a k-fold time-series cross-validated (CV) approach, one described in [7] as *blockedCV*. The k months of data is divided into k sets in an incremented fashion, of size

[6] `https://en.wikipedia.org/wiki/Cross-validation_(statistics)`

[7] `https://en.wikipedia.org/wiki/Symmetric_mean_absolute_percentage_error`

1 month each, such that first s months of data is used for training the model, $(s+1)^{th}$ for parameter selection. By choosing a parameter that performs the best throughout the year helps in reducing the bias in our model. Once the parameters are selected, first s months of data is used for training while $(s+1)^{th}$ month for testing the model performance. This process is repeated for $s = 1, 2, \ldots, (k-1)$. For example: If the overall data contains months from January to December, a training size of 3 means that the model is trained on January, February, March and tested on April after the model parameters are set.

Until this section, we gave the background information about the terminologies that will be used for the rest of the thesis. We discussed the generic steps of model framework and the algorithms we used for our work in Chapters 3, 4 and 5.

1.6 Contribution

In this work, we make three contributions in the field of spatio-temporal analysis of urban data. First, we present a framework for analyzing and predicting nearby traffic demand around a specific location of interest in a spatio-temporal fashion. For second, an earlier work done during a graduate-level course project[8] was adapted to characterize crime. We use the same data exploration and spatial feature extraction methodology as discussed in the course project. The project was based on a classifier, however we present a detailed regression analysis of identifying the intensity of various hotspots throughout the city. We do so by predicting overall and a particular crime type in an urban city using the similar grid-based framework used for analyzing traffic. More information about the contribution can be found in Section 4.1. Lastly, we analyze the impact of nearby traffic on the electric vehicle charging demand recorded at different charging stations in an urban city. The analysis presented here can be used as a starting point for building applications like advertisement scheduling, hotspot detection, EV charging scheduling, day-ahead and real-time electricity purchase etc.

1.7 Organization of Thesis

Until now, we gave the motivation for our problem statement, and introduced the topics that are relevant for understanding this thesis. We briefly explained the data mining steps, that will be used in following chapters and enlisted our contributions we make with this work. Rest of the thesis is organized as follows. In the next chapter, Chapter 2, we discuss some related work on the existing frameworks that are being used for modeling problems of traffic demand, crime and charging prediction. This chapter gives a brief overview of the existing work, but each chapter separately reviews the related work in depth under sub-

[8] `https://github.com/adbharadwaj/cs-5984-urban-computing-project`

section *Literature Review* individually. We then explain the proposed framework for traffic analysis along with the results and observations in Chapter 3. In Chapter 4, we describe how we use the similar framework as described in previous chapter to analyze and predict crime. In Chapter 5, we analyze the impact of nearby traffic on charging demand. We also present an application of the framework to predict future traffic and subsequently use it to predict future charging demand. We summarize our experimental results and observations at the end of each chapter. Finally in Chapter 6, we summarize the conclusions of our work and provide suggestions for future work.

Chapter 2

Related Work

Spatio-temporal data corresponds to a dataset describing events which vary with time and location of the event occurrence. The real world data is composed of information from both of these dimensions, and thus, analyzing spatio-temporal data [4] is considered important in order to understand an underlying urban problem. In this section, we will discuss various frameworks that have been used in the areas of traffic analysis, crime analysis and EV charging demand analysis - areas relevant to our work in this thesis.

Traffic demand modeling has been an established problem statement in the area of Intelligent Transportation System (ITS). One of the most widely adopted techniques is use of statistical properties of the traffic. For example; authors in [60] presented a theoretical model of traffic analysis by treating traffic as a time-series data. A hybrid approach using ARIMA and historical mean is proposed in [43] for traffic speed forecasting. The work in [57] uses covariance matrix calculated using historical traffic data which is referred as spectral analysis, to forecast traffic flow in real-time. The another framework that is quite widely used is using a graph based approach by topologically treating a city as a road network, with roads as edges and intersections as nodes. For instance [66] presents a multi-view mobility graph for the city of Shenzhen, China using heterogeneous data to model the human mobility. Spatio-temporal traffic pattern for New York City (NYC) is characterized using graph-mining techniques in [38]. The spatial distribution of traffic demand in the NYC has been studied in [47]. Another method of studying spatial distribution of traffic demand is a grid based approach, such as [37], where the city is divided into 16 administrative blocks. This is similar to what we have proposed, however at a much smaller spatial scale and is explained in detail in next chapter. Newer methods use modern data-mining techniques which utilizes different forms of human mobility data to solve various urban computing problems [21, 27]. For example, these data can include GPS traces [70], Call Data Records [14], WLAN hotspot connections [53] and recently social networking data [25, 41]. Since we focus on models based on data mining as well, further related works that uses data mining models are discussed in Chapter 3.

The second part of our work is on crime analysis and prediction. Most of the time, criminology research is focused on a people-centric or a place-centric as explained by Bogomolov *et al.* in [9]. For example, [59] presents an algorithm to find a pattern in crimes committed by a same person or a group. The authors in [17] presents an algorithm to detect hotspots in a city using a grid-based approach by using likelihood ratio. We perform a similar grid based approach, but instead of explicitly finding hotspots, we use features to implicitly generate unique hotspots that are more prone to crime and utilize it further to predict future crime. Chainey *et al.* have compared four different frameworks for mapping hotspots and its accuracy in [11]. The four framework includes spatial ellipses, spatial division using administrative zones, grid wise spatial division and Kernel Density Estimation. A completely different approach of looking at the crime problem is through network analysis. One of the very first works done by Sparrow [54] in 1991 aims to analyze criminal activities using network characteristics like centralities, equivalence etc. Following the fact that network analysis can be helpful in understanding and disrupting criminal activities, the authors in [15] further used network analysis to study the effectiveness, or rather ineffectiveness of a criminal network disruption. More recent works have focused on analyzing past criminal data and extracting hidden knowledge about the criminal activities. Work done in [12] describes that the data-mining frameworks used in crime analysis typically includes classification, association analysis, clustering and outlier detection. Other works on crime prediction using data mining sources and how our work is different from them is explained in Chapter 4.

The problem of electric vehicle charging demand has already been identified and had been tackled in the past. A stochastic model of plug-in electric vehicle is presented in [2] for short-term real-time demand forecasting. A demand model based on user preferences and market research is proposed in [20]. A probabilistic power flow model is presented in [32] that considers uncertainty of charging events like battery state and capacity, and takes charging duration into account for plug-in hybrid electric vehicle demand prediction on the simulated data. By considering charging start-time and initial State-of-Charge (SoC), the impact of EV charging on power system load profile is studied in [46]. Chapter 5 further highlights related work that uses data mining techniques to predict electric vehicle charging demand and explain how our analysis is different from the existing work.

In summary, we have discussed various methodologies and frameworks that are currently employed to analyze spatio-temporal data of traffic, crime and EV charging. Individual chapter on these analysis will further discuss existing work on data mining techniques that are being used. The next chapter will explain the framework for the analysis of nearby traffic data.

Chapter 3

Spatio-Temporal Analysis of Nearby Traffic

3.1 Overview

As discussed in the introduction, analyzing and predicting traffic is useful in solving various urban transportation problems. While a city-level traffic prediction is useful on one hand, on another, traffic demand prediction near a specific Location of Interest (LoI) is significant for ITS as well. A LoI can be a building, parking lot, gas station, restaurant etc. Traffic analyzed near a LoI gives an idea of the immediate surrounding and dynamics of human mobility in a smaller geographical region. In this chapter, we present compare and contrast two slight variations of a model framework by applying standard algorithms, enlisted in Section 1.5.2, to analyze and predict traffic demand near a location of interest. This chapter is organized as follows: We start with giving literature overview of related work on traffic modeling, and further explain the need for a nearby traffic demand. Next, we describe the traffic data we use and formulate the problem statement for the model discussing about our framework. After providing the steps for model development, we present results and observation from our analysis. Finally, we end the chapter by providing a summary for the chapter.

3.2 Literature Review

Traffic demand modeling as seen in the earliest works, has been involved around treating it as a temporal data. For example: a time-series problem is formulated in [60] for short-term traffic prediction. The importance of incorporating spatial information is studied in [43] where adding information about accidents helped improving the model performance. With respect to the topology, the problem of traffic modeling is formulated comparing road networks as graphs and applying graph-mining techniques. For example: [66] represents the city of Shenzhen, China as a network, where nodes represents road intersections or street blocks and edges represents traffic volume flowing from one node to another. Similarly, [38] analyze the taxi flow using graph techniques and identify outliers in the flow. Another topological approach to analyze traffic is to divide the city into geographical grids that helps in differentiating major areas of a city like business, residential etc and characterizing them depending upon the geographical usage. As it is significant to follow a particular topology for the analysis, we selected the grid-based approach.

However, the works discussed above focus on traffic throughout the city. On one hand where such models efficiently describe the human dynamics at a city level, they provide very little or no information about traffic flow around a location within a city. The work in [67] has used individual road segments for predicting traffic speed. However the authors only present a time-series analysis of the problem thereby losing spatial relationships between the adjacent road segments. In our work, we take into account traffic of neighboring or adjacent area for our traffic demand analysis. Traffic speed modeling on individual road segments has been studied in [58] using a modified recurrent neural network approach but the analysis is only limited to one-month data, thereby making a room for a spatio-temporal traffic analysis on a smaller region for an extended time period. The work presented here is different from prior works that we present a detailed analysis of traffic near a point of location for an extended time-period (52 weeks). By near, we mean that we analyze traffic around a location within some x miles only; ($x = 1, 2, 3 \dots$). The advantage of picturing the dynamics at this scale is that it can be used to make certain decisions by local authorities and businesses like personalized advertisement scheduling, dispatching nearby taxis etc.

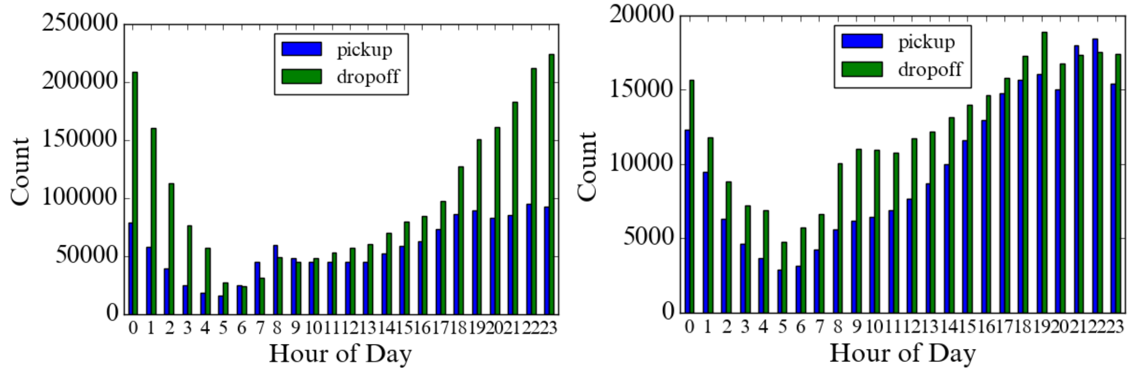
3.3 Data Description

In order to understand the idea behind the framework, we analyze the nearby traffic data recorded around two locations of New York City. The NYC-TLC[1] website provides taxi trip data recorded in the NYC. The main attributes of the data includes a trip's pickup and drop-off location and time. We analyze the data for 52 weeks, from 29th June 2015 (Week 27, Day 1) to 26th June 2016 (Week 25, Day 7) for two locations, one in Brooklyn coordinates given as $L1$ = (40.656737, -73.977436) and another in Flushing, L_2 = (40.764895, -73.819011). The two locations are different with respect to the analysis as they have different volume of traffic nearby. Table 3.1 describes the total taxi pick-ups and drop-offs made within a 1 mile^2 block of each location. The given number of trips was obtained after we cleaned the data that had corrupted latitude and longitude, falls outside the 1 mile^2 block as a part of data preprocessing. We also merged the data of two taxis running in the NYC, Green and Yellow taxi.

[1] `http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml`

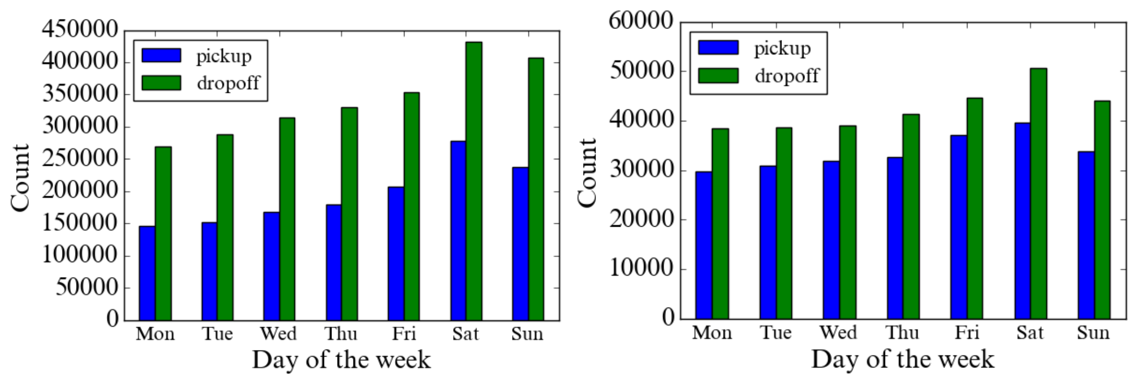
Table 3.1: Total pick-ups and drop-offs recorded between 29th June 2015 to 26th June 2016 within 1 mile^2 block

	Pick-up Count	**Drop-off Count**
Brooklyn	1366022	2394114
Flushing	235612	296557



(a) Location L_1 (b) Location L_2

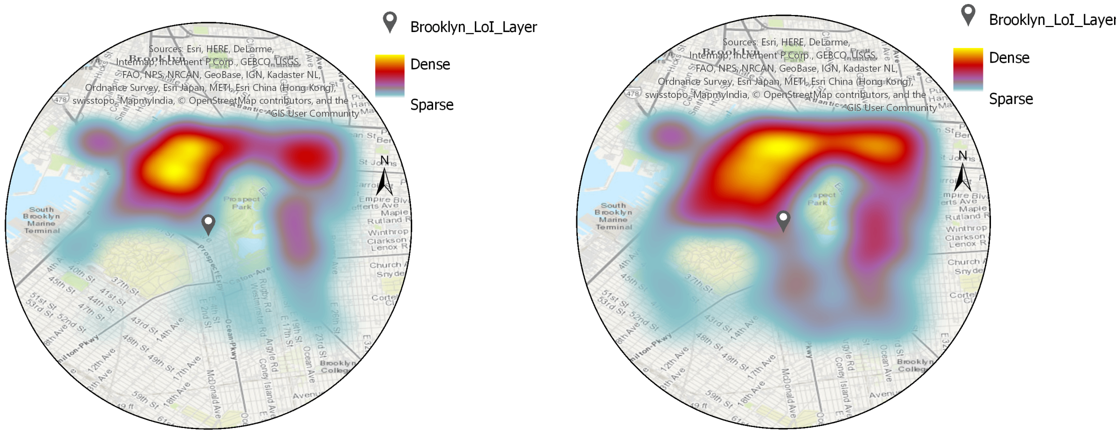
Figure 3.1: Distribution of pick-ups and drop-offs recorded around L_1 and L_2 within 1 mile^2 block depending upon the hour of day



(a) Location L_1 (b) Location L_2

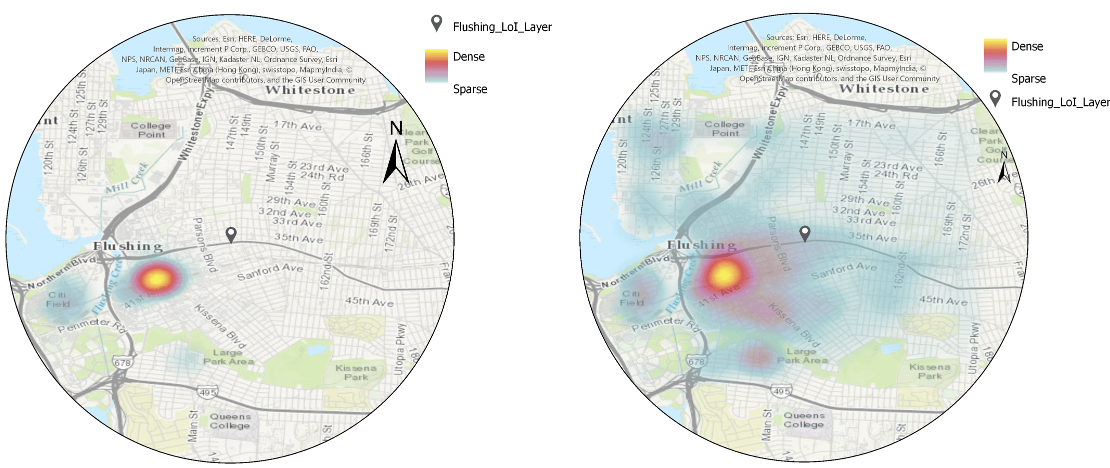
Figure 3.2: Distribution of pick-ups and drop-offs recorded around L_1 and L_2 within 1 mile^2 block depending upon the day of the week

Figure 3.1 and 3.2 shows the hourly and weekly distribution of the traffic for both the locations respectively. Even with two different locations, we still see a similar trend that the traffic decreases towards the early hour and increases throughout the day. Similarly, the traffic in the week is almost same throughout the week, with demand on Saturday reaching a peak, because of the late night drop-offs following a Friday night. Thus, we can see a temporal component attached to the traffic analysis.



(a) Pickup (b) Dropoff

Figure 3.3: Heatmap of pick-ups and drop-offs recorded between 29th June 2015 to 26th June 2016 within 1 mile^2 block of Location L_1



(a) Pickup (b) Dropoff

Figure 3.4: Heatmap of pick-ups and drop-offs recorded between 29th June 2015 to 26th June 2016 within 1 mile^2 block of Location L_2

Since the geographical location of an area has a high impact on the local traffic, it is interesting to observe the distribution of total traffic data around these location for 52 weeks. As seen from the Figure 3.3 and 3.4, we see that location L_1 is busier than L_2. Also, there are certain hotspots where the traffic count is higher than the surroundings, and the effect gradually decreases around it. No traffic around the location L_1 constitutes to the presence of park and cemetery. Where the traffic around L_1 is pretty distributed, traffic around L_2 is higher in one particular location than the rest, which can be contributed to the presence of the Flushing park and Queens museum as compared to the other side. Thus, we see that depending upon the location and other geographical features, hotspots around a location can be varying. This reinstates the spatial part of our analysis. After discussing the space and time dimension of the nearby traffic, in the next section, we formulate the problem statement of analyzing and predicting it in a spatio-temporal setting.

3.4 Problem Statement

3.4.1 Preliminary

We begin by introducing the spatial and temporal unit which will be used throughout the chapter and describes the basis for our model analysis. Non-scaler variables will be represented in capital letters. A subscript and a superscript represents a unit in temporal domain and spatial domain respectively.

Time-stamp

A time-stamp T is a vectorized time unit for our model specified by a week, a day and a half-hour quantity and represented by $T = \{w, d, hh\}$. For better precision, we predict traffic demand at every half hour of the day, but the model can be used for any prediction horizon like hourly and daily. Overall, we divide each day into 48 half-hours instead of 24 hours, as used in [63].

- w: week number of the year that can have values:$\{1, 2, \cdots, 52\}$
- d: day number of the week that can have values: {1 (Mon), 2 (Tues), $\cdots$, 7 (Sun)}
- hh: half-hour number of the day that can have values: {1: [00:00-00:30), $\cdots$, 48: [23:30-00:00)}

Example: 23rd December 2015, 11:49 PM is represented as $T = \{52, 3, 48\}$. With the given T, the next two time-stamps will be $T+1 = \{52, 4, 1\}$ and $T+2 = \{52, 4, 2\}$. Since there are only 2 years and doesn't change for 50% of the data, it has been omitted from time-stamp.

Grid

A grid g, is the spatial unit represented by a square area of around a LoI. The m-miles region near a LoI is divided into $G = g * g$ grids as shown in Figure 3.5 with $m = 1$ and $G = 9$. Each grid is considered as a separate unit and traffic count is predicted for each grid in a G grid-setup.

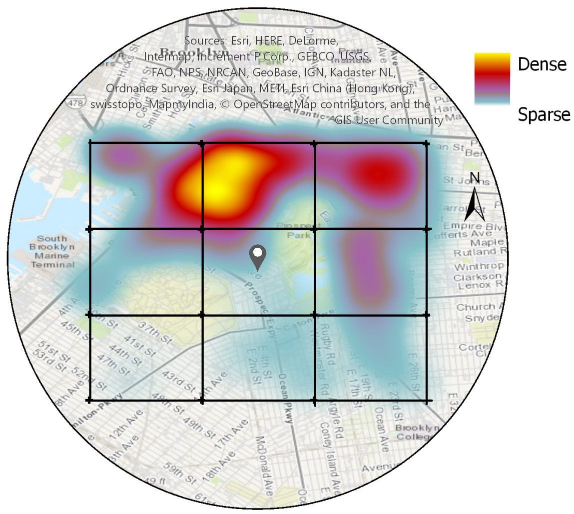


Figure 3.5: Spatial distribution of traffic in 9 Grids for Location L_1 within 1 mile^2

Trip

A taxi trip is represented by a quadruple quantity given by $\{(x_p, y_p), (x_d, y_d), (T_p, T_p), (T_d, T_d)\}$ where subscripts p and d stands for pick-up and drop-off respectively. x and y represents the corresponding longitude and latitude coordinates of the pick-up and drop-off locations. T represent time-stamp recorded with the given pick-up and drop-off event.

Traffic Count

The basic quantity of our traffic prediction model is the traffic count, c. Here, we measure traffic count as number of pick-ups and drop-offs observed at time-stamp T and grid g, given by p_T^g and d_T^g respectively, interchangeably called as c_T^g. In order to calculate traffic count at grid g at time-stamp T from a given trip data, we do the following:

- For a given trip, if the pick-up location (x_p, y_p) falls within the grid g, then $p_{T_p}^g$ += 1
- For a given trip, if the drop-off location (x_d, y_d) falls within the grid g, then $d_{T_d}^g$ += 1

3.4.2 Problem Formulation

The problem of predicting localized traffic demand prediction problem is explained as follows:

Given historical set of pick-up and drop-off count set S_p and S_d, alternatively represented as S_c, at all G grids and past N time-stamps:

$$S_c = \begin{Bmatrix} c_T^1, & c_{T-1}^1, & c_{T-2}^1, & \cdots & c_{T-N}^1, \\ c_T^2, & c_{T-1}^2, & c_{T-2}^2, & \cdots & c_{T-N}^2, \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_T^G, & c_{T-1}^G, & c_{T-2}^G, & \cdots & c_{T-N}^G \end{Bmatrix} \tag{3.1}$$

Using historical pick-up and drop-off data S_c, identify spatio-temporal features $X = \{X_1, X_2, \cdots, X_f\}$, where f = number of features, that can predict traffic with maximum precision for the next time-stamp $T + 1$ for all the G grids. The prediction set S'_c is given as:

$$S'_c = \begin{Bmatrix} c_{T+1}^1, \\ c_{T+1}^2, \\ \vdots \\ c_{T+1}^G \end{Bmatrix} \tag{3.2}$$

3.5 Model Development and Analysis

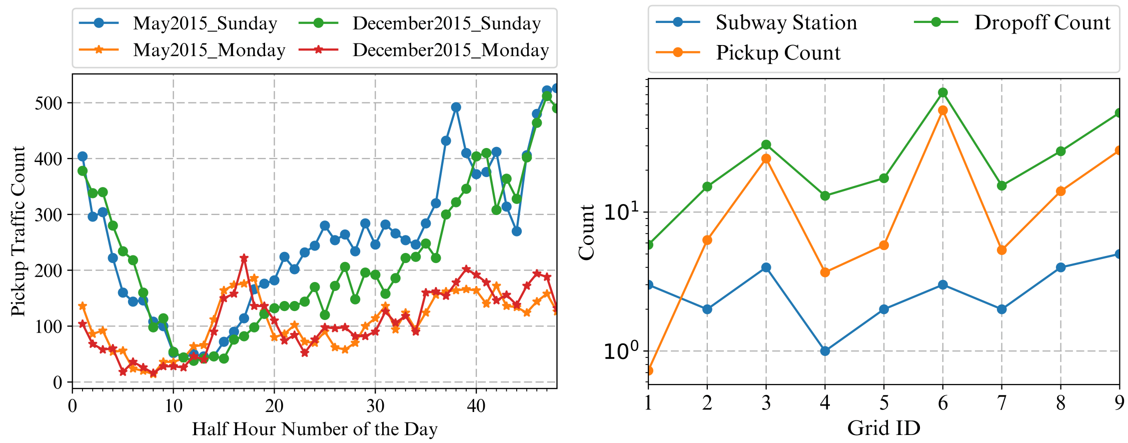
In this section, we will discuss the steps followed for model development.

3.5.1 Feature Extraction

As discussed in Section 1.5.1, identifying the right features for a model is one of the most important steps in the process of knowledge discovery. Keeping that in mind, in this section we analyze some of the insightful details about the data and introduce the spatial and temporal features identified for our model, inspired from [63].

Temporal Features

A typical time-series data tends to follow a pattern. We observe similar patterns on weekdays, which is different from weekend. Similarly, there is also a seasonal impact on traffic demand. Figure 3.6a represents daily pickup traffic made around LoI_1 on Monday and Saturday from Week 20 (May 11-17, 2015) and Week 50 (Dec 7-13, 2015). Irrespective of the season, traffic count on a weekday and a weekend vary considerably. There is a similar trend of traffic throughout a weekday be it summer or winter. However for a weekend, there is a significant



(a) Temporal feature variation (b) Spatial feature variation

Figure 3.6: Traffic variation depicting the importance of temporal and spatial features for location L_1

variation in traffic demand in daytime (between $hh = 20$ to 40) depending upon the season. This knowledge of hour of the day, day of the week and week of the year have to be preserved by the model. Here-forth, we introduce our first set of temporal features, $\{w, d, hh\}$, which is similar to time-stamp T discussed in section 3.4.1.

Spatial Features

In order to incorporate geographical features, we add information like number of subway stations around a location. If an area has more subway stations, relatively higher number of taxi demand is observed because of the last mile problem [55]. We take this into account by specifying number of subway stations located at a grid g in our model. From Figure 3.6b, we see a similar trend in traffic counts and subway station count at various grids. This helps our model to identify grids that are busier than others and accordingly tune parameters for that grid. Also, as we see from Figure 3.5, nearby grids can help establish traffic density around a location. Thus, we also include traffic demand of the neighboring grids for the previous half-hour. Apart from this, since accidents and motor collisions disrupt normal traffic flow, we add accident information for the past half hour in order to take care of such abnormalities. The advantages of having accident information in traffic modeling has already been identified and has been used to improve performance in [43].

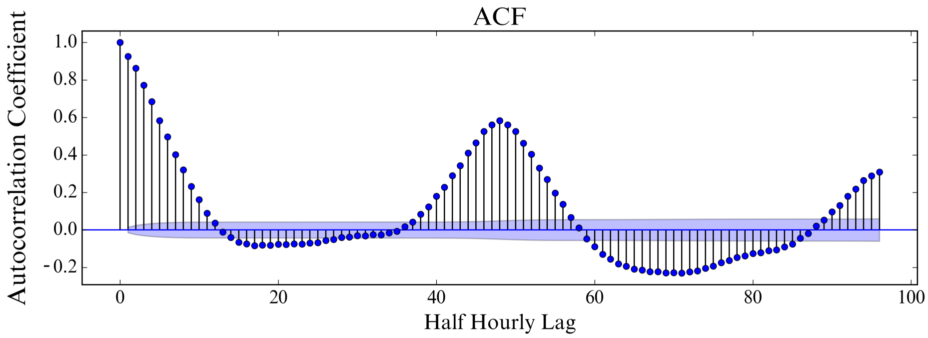


Figure 3.7: Autocorrelation plot of half-hourly lag for pickup count at Location L_1 within 1 mile^2 depicting feature importance of previous hour, day and weekly traffic count

Spatio-Temporal Features

A time-series data such as traffic count tends to follow a trend in a sense that it has a high correlation with respect to the historical traffic. Particularly, the previous hour's and day's traffic at a location helps in understanding the traffic at next instant. In order to quickly show its importance, we analyze the autocorrelation of the traffic count over several historical time instances. Autocorrelation function (ACF) basically gives the correlation between two time-series data x_t and x_{t-h} ($h = 1,2,3 \ldots$) and has been used widely for traffic flow analysis[26]. Mathematically, it is given by:

$$ACF = \frac{covariance(x_t, x_{t-h})}{variance(x_h)} \tag{3.3}$$

Figure 3.7 depicts the autocorrelation between the total traffic observed around the L_1 over the 52 weeks. Each instant on X-axis depicts the relation between the traffic count c_T over historical count c_{T-h}. The previous traffic c_{T-1} is highly correlation and it gradually decreases as we look past hours. However, as we reach the same half-hour of the previous day, we see an increase in the coefficient. This accounts to adding the features that represents the traffic on the previous day at the same half-hour of the day into the model. Similarly, we also add an average of the traffic counts recorded in previous three and four weeks at each grid g.

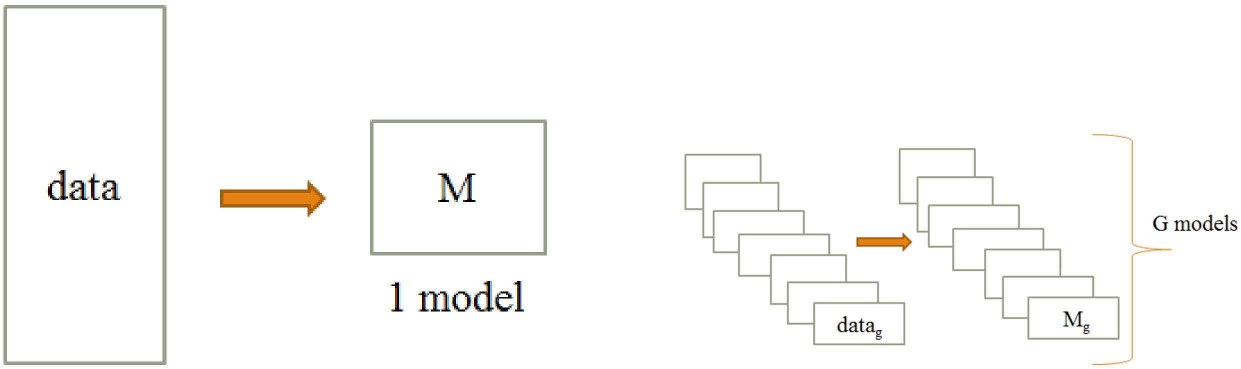
3.5.2 Model Development

We train and develop two different types of model as depicted in Figure 3.8:

- One_Model: A single model is trained using the final data with extracted features from all the G grids.

- Block_Model: A single model is trained using the data from one grid only. A total of G models are trained.

Having two different variations of a framework helps in understanding the nature of the grids and its importance in our model framework.



(a) Number of Models = 1 (One_Model) (b) Number of models = G (Block_Model)

Figure 3.8: Traffic variation depicting the importance of temporal and spatial features for location L_1

3.5.3 Parameter Selection

We ran our model using standard data mining algorithms, namely LR, MLP, and Decision Tree, as described in Section 1.5.2. We skip kNN and SVR as it took a long training time, as compared to the other three. Since MLP and Decision Tree algorithms have parameters which have to be appropriately set to tune the model, we use the cross-validation technique, described in Section 1.5.4. We start with week 27 of the year 2015 (i.e 29^{th} June 2015) and go till 26^{th} June 2016 thereby understanding the model performance with seasonal effects for year around. For example, Figure 3.9 shows the validation error for MLP w.r.t the number of hidden layers.

3.6 Results and Observation

Given the features described in the last section, we trained the regression model on LR, MLP and Decision Tree.

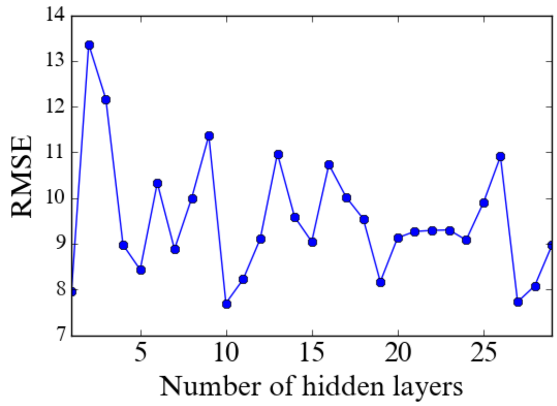


Figure 3.9: Selecting number of hidden layers for MLP for drop-off count at Location L_1 within 1 mile^2, selected parameter (lowest RMSE) = 10 hidden layers

3.6.1 Algorithm wise performance

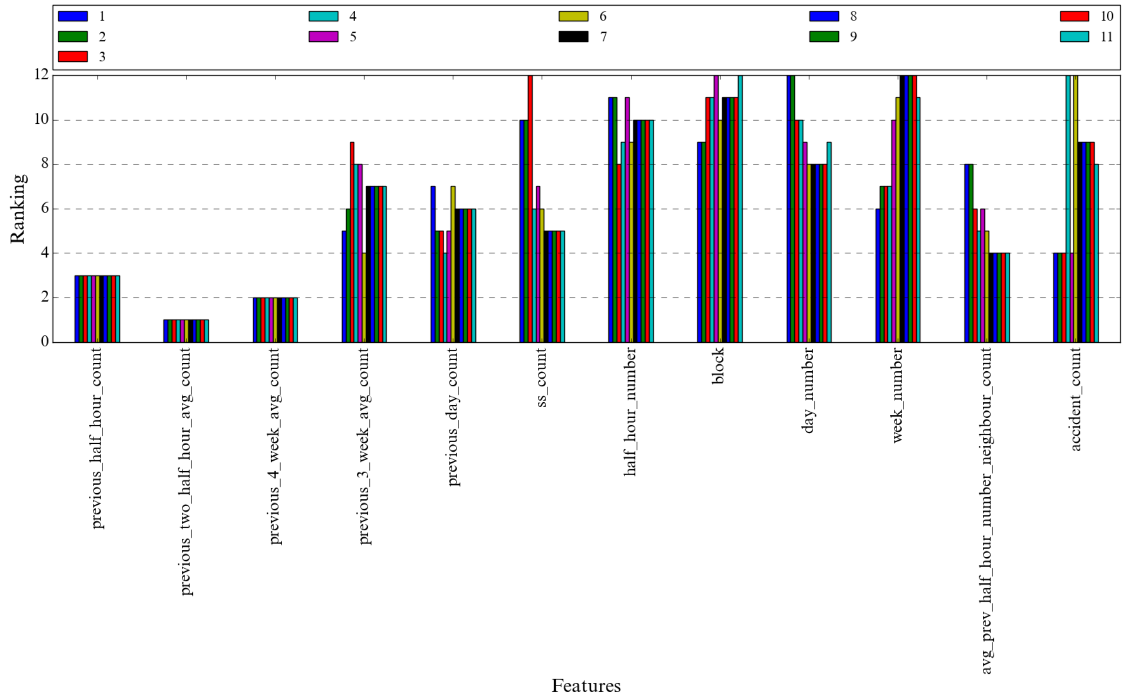
We present our initial results with $G = 9$ grids in order to better understand the features as our data exploration in previous section was also performed for $G = 9$. For a holistic analysis we study the impact of grid-size on model performance at the end of this section.

Table 3.2: Average MAE, RMSE, SMAPE AND Training Time for different algorithms across 11-fold cross-validation

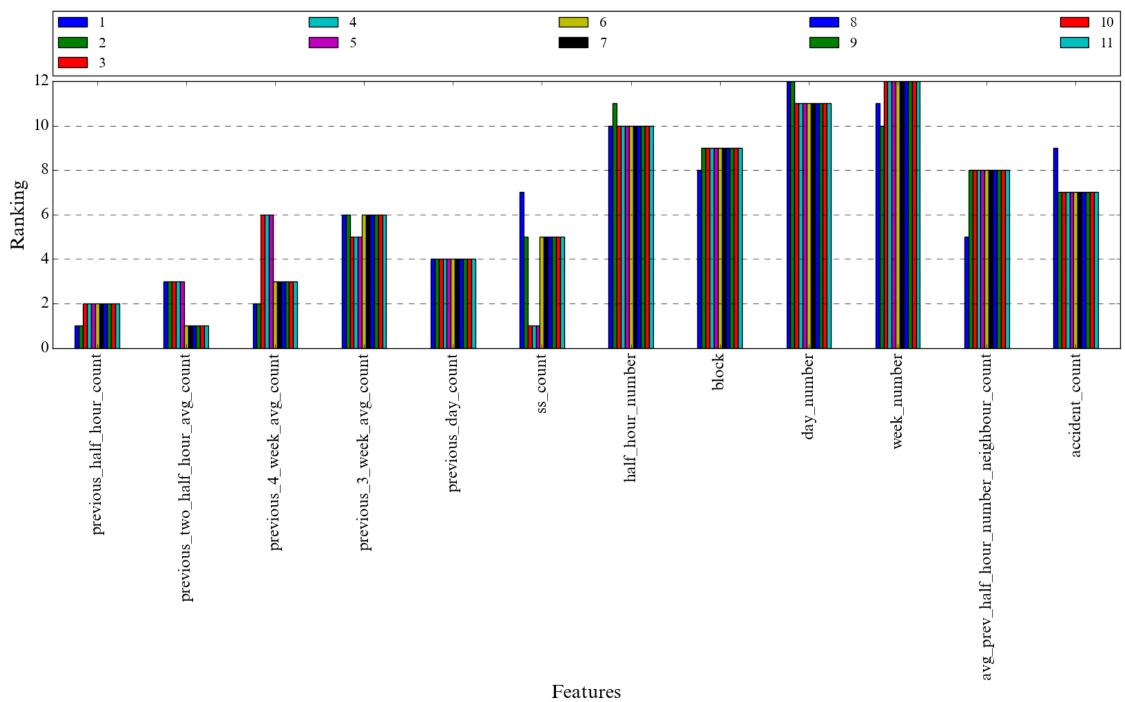
		Pick-up				Drop-off			
Location	**Model**	**Avg. MAE**	**Avg. RMSE**	**Avg. SMAPE**	**Avg. Training Time (in sec)**	**Avg. MAE**	**Avg. RMSE**	**Avg. SMAPE**	**Avg. Training Time (in sec)**
L1	**LR**	2.5	3.52	32.45	0.47	3.72	5.29	22.2	0.44
	MLP	3.22	4.41	35.42	32.29	5.1	7.02	29.4	73.84
	Decision Tree	4.51	6.47	40.32	0.05	6.81	10.22	28.17	0.05
L2	**LR**	0.65	1.05	59.21	0.44	1.07	1.42	51.94	0.5
	MLP	1.03	1.59	52.28	212.67	1.22	1.61	51.49	185.33
	Decision Tree	1.25	1.95	85.05	0.08	1.23	1.65	55.49	0.09

Overall Performance

The average performance of the algorithms over 11-fold Cross-Validation (CV) is summarized in Table 3.2. The numbers in red highlights the one with smallest SMAPE error. As we can see, in general LR performs the best as compared to MLP and Decision Tree for both the locations, and for either type of count, i.e pick-up and drop-off. In case of pick-up prediction for L_2, though MLP improves the performance of LR by a 6.9% over LR, but the average training time for the MLP to learn the model was a whooping 483 times that of LR. Thus, it can be hypothesized that for a spatio-temporal like traffic which has a high correlation with its temporal features, an algorithm as simple as LR can give higher accuracy with a low



(a) Location L_1



(b) Location L_2

Figure 3.10: Feature Ranking for pickup-count prediction across 11-fold CV for both the location for 3×3 grids

training time. To further analyze exactly which features were most important, we performed a recursive feature elimination[2] based on the coefficients assigned by the LR.

Figure 3.10a and 3.10b gives the feature ranking for locations L_1 and L_2 respectively for the pick-up count prediction. The lower the rank, the higher is the weight of that feature in the LR model, and higher is its importance. For Location L_1, top 3 features across all the cross-validation is fixed: *previous_half_hour_count, previous_two_half_hour_avg_count, previous_4_week_avg_count*, thereby confirming from our exploration that temporal features that have a high correlation analyze the traffic accurately for the next instant. Spatial features *accident_count* and *avg_prev_half_hour_number_neighbour_count* effectively rank 4^{th} thereby establishing the fact that traffic at a block is effected by the nearby blocks. On the other hand, in case of location L_2, there looks no consensus among the feature rankings. This could be accounted for the relatively higher percentage error of LR in L_2 than MLP. Also, we notice that features like time-stamp doesn't necessary have a huge importance when it comes to feature importance in a linear model. Thus, the only way LR achieves better accuracy across all the grids is by identifying pattern in the historical counts in its own as well as neighboring grids.

Seasonal effect and time to train

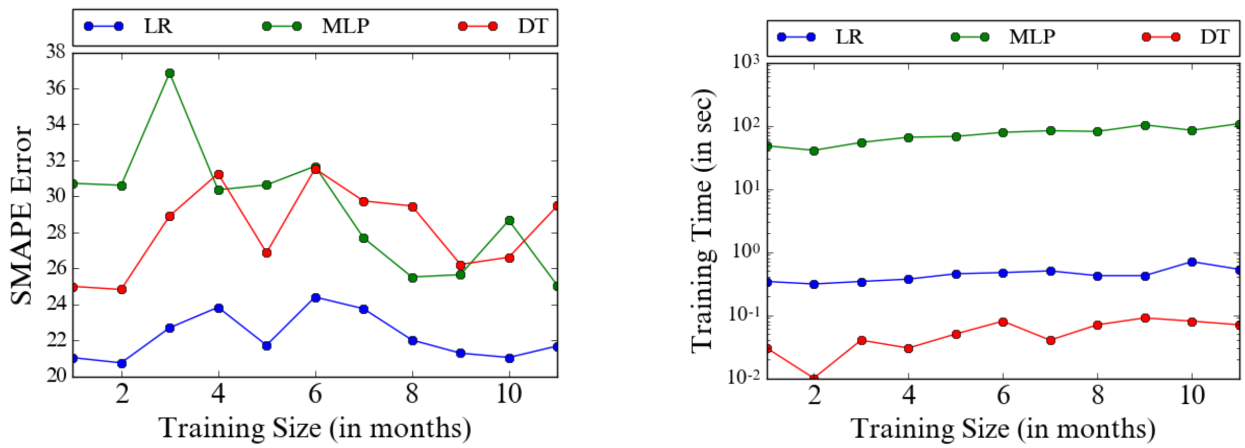
In order to understand how training period affects the performance of the model, we shows the percentage error across all models with different training sizes in Figure 3.11a. On average, LR provides a check on the seasonal effect on the model, specially when the training size is 4, i.e data is trained from July-Oct and tested on November.

Figure 3.11b shows the impact of training size on the time it takes for the model to train. Even though Decision Tree's train is lesser than LR, it gives a high error of 28.17% as compared to 22.2% of LR as recorded in Table 3.2, averaged over 11-fold CV. Thus, not only can LR be used for offline prediction, it is also feasible to implement an online algorithm if the data is available in a streaming fashion.

3.6.2 Feature wise performance

In order to get an idea of the importance of various spatio-temporal features used in the model, we trained the model by keeping one and removing other and compared it with the model with all the features. As it can be seen from Figure 3.12a, when we take only spatial features which includes neighboring blocks's count, accidents and subway station information, the error is as high as 42.79% on average. If we take only temporal features which includes previous hour's, day's and week's information, we see the error decreases to 33.01%. Adding spatial features to the temporal model improves the performance with

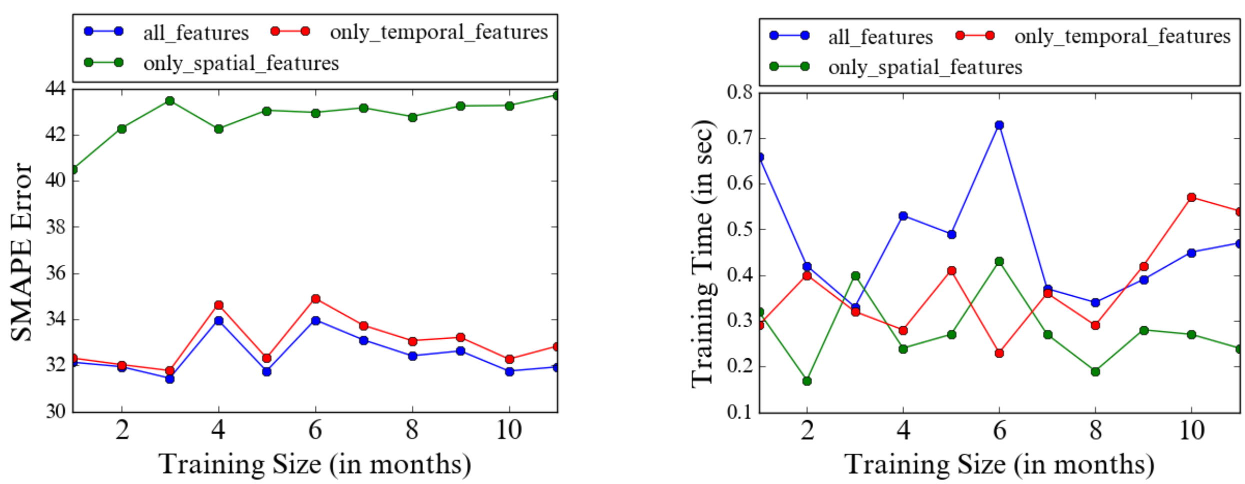
[2] `http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html`



(a) Seasonal Effect (b) Model Training Time

Figure 3.11: Seasonal effect and training time for drop-off count prediction at location L_1 on $G = 9$ grids

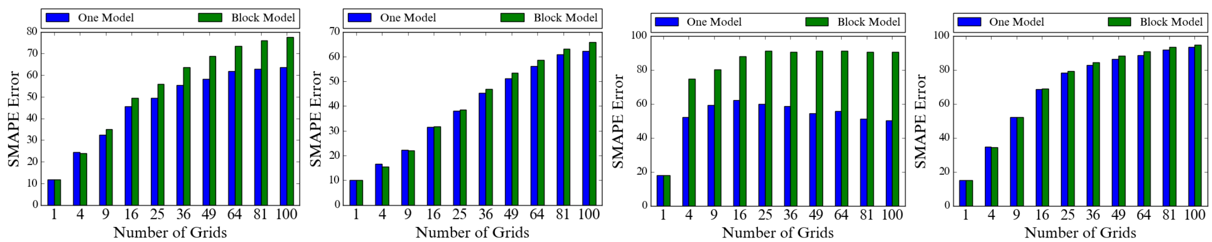
an average error of 32.45%. Even though the improvement is less, but it highlights the importance of spatial features we have used in our model. Since number of features increases the complexity of the model, the training time is higher when we use all the features (Figure 3.12b), but it is still less than 0.8 seconds which has almost negligible effect if we perform an offline analysis.



(a) SMAPE Error (b) Model Training Time

Figure 3.12: Model performance with and without spatial and temporal features for pick-up count prediction at location L_1 on $G = 9$ grids

3.6.3 Effect of grid size and model type on performance



(a) Location L_1 (pickup) (b) Location L_1 (dropoff) (c) Location L_2 (pickup) (d) Location L_2 (dropoff)

Figure 3.13: Impact of number of grids G on model performance at both the locations

The analysis presented so far was done by keeping the total number of grids, $G = 9$ and by considering only One_Model type. In Section 3.5.2, we introduced two variations of our grid-based model framework. In order to compare and contrast these two approaches, we ran our experiments for Block_Model as well. We also analyze the impact of spatial granularity on model performance by using LR on both One_Model and Block_Model type framework by changing number of grids from 1×1 to 10×10. As it can be seen from Figure 3.13, error increases when we increase the number of grids because of the higher precision. But, for both the locations and for both pick-up and drop-off counts, One_Model performs the best in general. However, the difference is most significant in the region that has least pickup counts (Figure 3.13c).

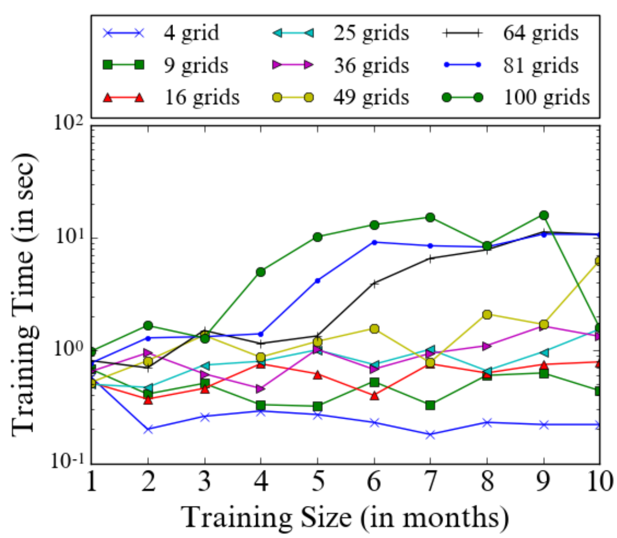


Figure 3.14: Impact of number of grids on training time for drop-off count prediction at Location L_2

Figure 3.14 shows the training time for drop-off count prediction at location L_2. In terms of training time, when training size is small, training time for each grid size is comparable and as small as 1 sec. However, since number of data-points increases with number of grids in One_Model, training time increases with the increasing training size and number of grids.

3.7 Summary

In this chapter, we presented a framework for analyzing nearby traffic data. We started with exploring traffic around two locations in the NYC within 1 miles^2, and identified various spatio-temporal features that are correlated with the future traffic and can be potentially used for traffic forecasting. We formulated our problem statement and introduced two variations of the model. Next, we compared the performance of LR, MLP and Decision Tree across both the locations, and found that in general, LR performs best giving least percentage error and effectively tunes the model to consider seasonal effect, while keeping the training time low. Thus, it can be used for both offline and online analysis in presence of streaming traffic data. From our analysis, we found that the spatio-temporal features like historical traffic, neighboring grid's traffic and information of recent accidents play an important role in traffic prediction. On the other hand, features like week of the year, day of the week has least importance. We also compared performance of the two variations of the framework with different spatial granularity and found that One_Model gives better performance in general. In order to present an application of our framework, we have used this model for predicting future traffic to analyze electric vehicle charging demand in Chapter 5.

Chapter 4

Spatio-Temporal Analysis of Crime

4.1 Overview

In this chapter, we perform a spatio-temporal analysis of the crime data. A part of this work has been adapted from a course project[1] where we perform a similar analysis. Specifically, Section 4.3 and spatial extraction of Section 4.4.1 is adapted from the project (Figure 4.1). Spatio-temporal and temporal features of Section 4.4.1 were my contribution in the project and has been used in this thesis. In the previous course work, a classifier was built to identify whether a given location is a crime hotspot or not at a given time. We extended the work to characterize the intensity of the hotspots by performing a detailed regression analysis. We also compared the performance of the analysis for the total crime vs. individual crimes. In doing so, we use a similar grid-type framework as explained in the previous chapter. This chapter is organized as follows: we start with discussing the related work. Next, we explore the data and formulate the problem statement for the model. After a brief description on feature extraction and problem statement, we present results and our observation from the analysis. Finally, we provide the summary for the chapter.

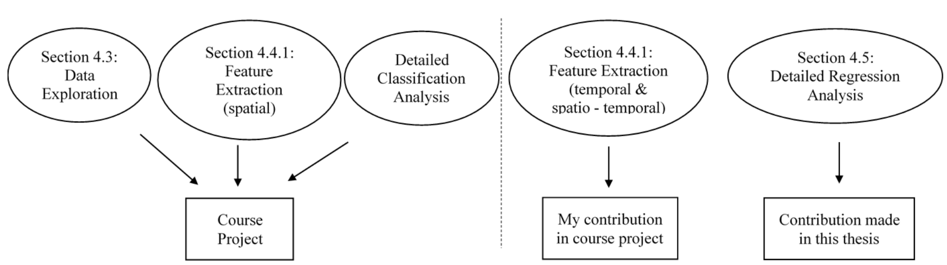


Figure 4.1: Graphical overview of the chapter explaining the sections adapted from the course project and new work done in this thesis

[1] https://github.com/adbharadwaj/cs-5984-urban-computing-project

4.2 Literature Review

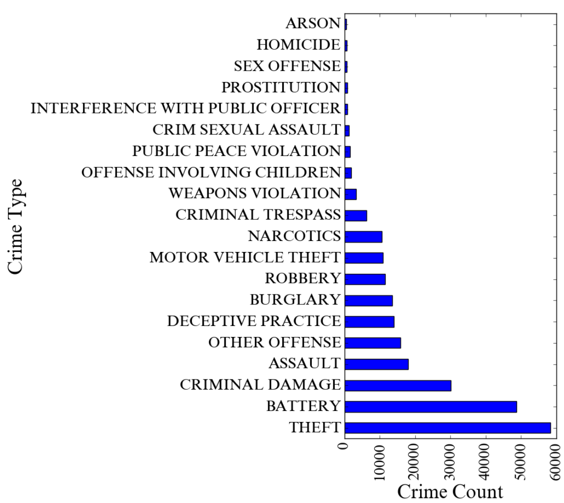
The two major outlooks pertaining to the quantitative criminology research, as explained by the authors in [9] are described as the approaches that analyze crime by targeting the people (criminal individual or the organization) or the place (location of the crime). The work done in [59] presents a framework for identifying a pattern in crimes committed by a certain individual or organization using a pattern sequence algorithm. Four major topological frameworks that are frequently used for analyzing crime have been discussed in [11]. This includes spatial ellipses, spatial division using administrative zones, grid wise spatial division and Kernel Density Estimation (KDE), based on the criminal activities observed. Since these are established approaches, we also apply a grid based approach. However, instead of using criminal activities as a basis of spatial exploration, we use local businesses and presence of police stations to characterize each location. Another interesting approach being used to analyze crime is to look at criminal organizations and their activities as a graph network. One of the very first works done in [54] aims to analyze criminal activities using network characteristics like centralities, equivalence etc. The authors in [15] later uses this approach to study the strength of this network when police intervention occurs to disrupt it.

Because of the recent development in machine learning techniques, [12] highlights the use of data mining techniques such as classification, clustering and anomaly detection for crime analysis. Ballesteros *et al*. in [6] uses historical crime, and census data from Miami and review data collected from Yelp to compute real-time safety information for people and location using ARIMA (Auto Regressive Integrated Moving Average), Linear Exponential Smoothing (LES) and Artificial Neural Network (ANN). A classification based crime prediction is presented in [24], that uses crime and census data using following classifiers: NaiveBayes, OneR, JRip, BayesNet, Decision Table, and J48. A relatively modern approach in [19] used crime data as KDE, Twitter's tweet analysis for Chicago and presented a classification problem. Another classification problem that involves studying sentiment analysis and weather data is presented in [13] which uses Logistic Regression. A Poisson based regression model was proposed in [42], however it analyzes only one type of crime, i.e., juvenile robbery arrest. To the best of our knowledge, no other prior work has collectively used grid-based features, yelp data, weather data and features based on KDE for predicting and analyzing crime. It is important to note that we perform KDE on the spatial data and only autocorrelation for temporal data, as will be explained in next section.

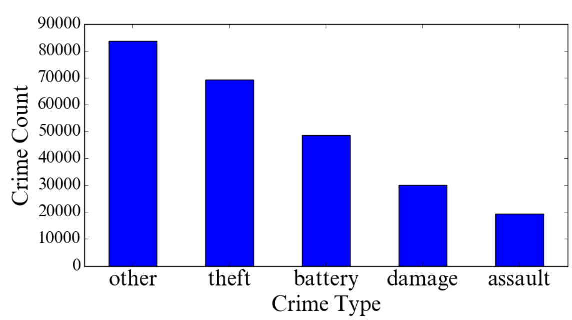
4.3 Data Description

4.3.1 Crime Data

For our analysis we choose Chicago, as it is the third largest city [50] in the United States and has been widely studied for its criminal activities before [39, 49]. We collect crime data for the year 2016, available by the Chicago City Administrative Department[2] website. A total of 251043 crimes were reported in 366 days. The frequency of different types of crimes reported is depicted in Figure 4.2a. However, for our analysis, we only consider the top 4 crime types, i.e., *Theft, Battery, Damage and Assault* and the rest of the types as *Other*. The 5 sub-categories for our analysis is depicted in Figure 4.2b. The dataset available contains temporal information like crime date and time, spatial information like the location where it was committed, and other characteristic information such as crime type, whether an arrest occurred or not etc.



(a) All Crime Types



(b) Aggregated Crime Types

Figure 4.2: Distribution of different type of criminal activities recorded for the year 2016 in Chicago city

4.3.2 Weather Data

In order to incorporate the temporal component in our analysis, we add daily weather information for the year 2016. The weather data was obtained from NOAA's (National Oceanic and Atmospheric Administration) RESTful web services[3]. The data collected was recorded

[2] `http://https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2`
[3] `https://www.ncdc.noaa.gov/cdo-web/webservices/v2#dataCategories`

by the weather monitoring station located at Chicago O'Hare International Airport, IL, US. Since the weather data is collected on a daily basis, our epoch parameter for temporal analysis is one day or 24 hours. These data include precipitation, snow and minimum and maximum temperature recorded for each day. This information helps in identifying seasonal variation in crime.

4.3.3 Police Data

Since criminal activities are influenced by geographical location, spatial information is added to the model by specifying location of police locations throughout the city. It only has spatial information and contains location in latitudes and longitude. The data was obtained from the same website as the crime data. A total of 23 police stations were obtained, plotted in Figure 4.3.

4.3.4 Yelp Data

Yelp[4] is a search and review based company, that offers web and mobile based applications where customers can search nearby local businesses like restaurants. The data containing locations of all unique business as registered by Yelp were downloaded using their web services[5]. These data indicate which parts of the city are more crowded during the day time, and can help in analyzing crime hotspots in a city. After final data gathering and cleaning, we obtained a total of 12944 unique locations.

4.3.5 Data Exploration

Even though a crime is an unfortunate event that can happen anytime and anywhere, usually there are certain places in the city (hotspots) where crime is more likely to occur [56]. ArcGIS Pro[6] software was used for spatial data exploration.
Figure 4.3 shows the heat-map of all the recorded crimes for the year 2016 in the city of Chicago. It also locates the presence of police stations in the city. As we can see, there are three major crime hot-spots, west-side, shoreline and the southern part of the city. Particularly, the downtown area observes a higher intensity of crime, which seems obvious because of the higher number of activities in that area. In order to get knowledge about the geographical area, we also analyze the number of businesses in the neighborhood. We analyzed the effect of day of week on the overall crime of the city, however there was no significant variation and hence, is not added in the exploration.

[4] `https://www.yelp.com`
[5] `https://www.yelp.com/developers/documentation/v2/overview`
[6] `https://pro.arcgis.com/en/pro-app/`

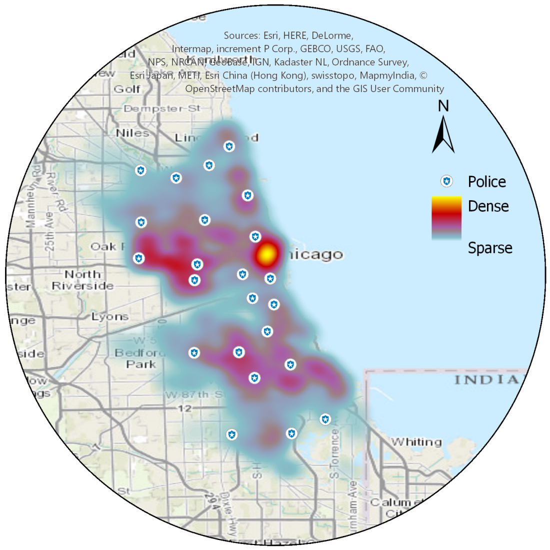
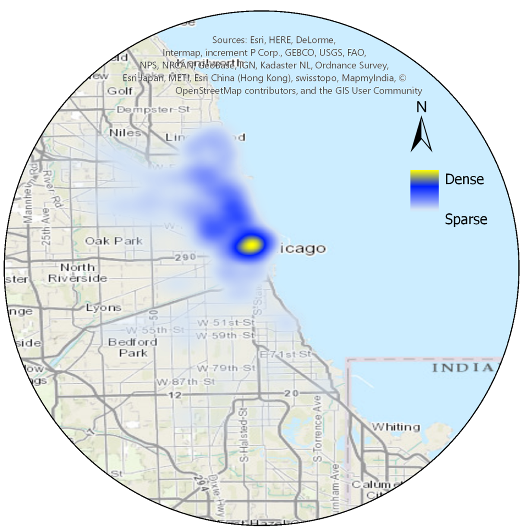
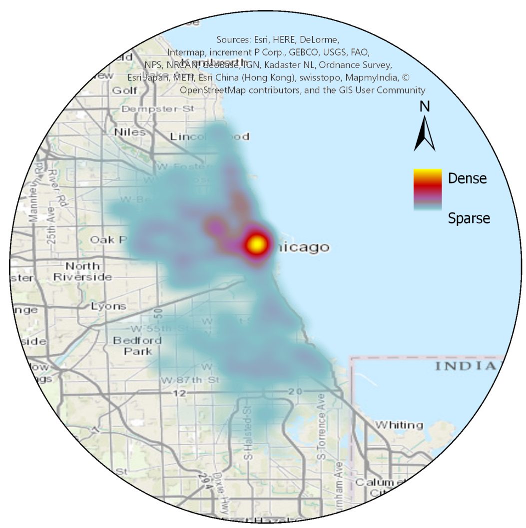


Figure 4.3: Heat-map of all types of criminal activities recorded in Chicago for the year 2016



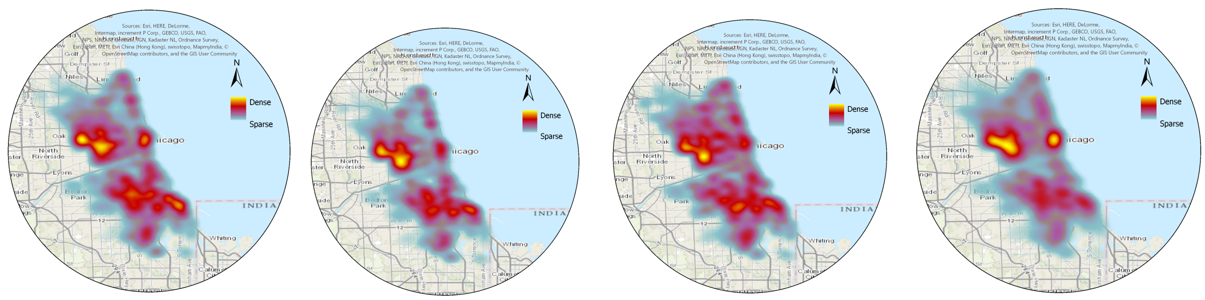
(a) Heat-map of local businesses



(b) Heat-map of theft

Figure 4.4: Heat-map of presence of local businesses and recorded theft activities in Chicago for the year 2016

In order to understand how presence of local businesses impact the crime, we visualize different types of crime and see their distribution with respect to the business. In particular, as can be seen from Figure 4.4, it is much likely that an area with higher local businesses has an increased chance of theft activities. The spatial distribution of other types of crime is presented in Figure 4.5. The another interesting observation is that a higher number of assault, battery and damage incidents was recorded in areas away from businesses because these crimes are usually committed in less busy areas.



(a) Assault (b) Battery (c) Damage (d) Others

Figure 4.5: Heat-map of assault, battery, damage and other activities reported in Chicago for the year 2016

4.4 Model Development and Analysis

From our previous discussion we can see that the location of a city is an important criteria in understanding when and where a crime can occur. Thus, in order to analyze the spatio-temporal crime, we use a similar framework we used for traffic prediction as explained in Section 3.4. We divided the entire city in $g \times g$ grids where each grid g_i is denoted by the location of lower left and upper points of the boundary box of that grid.

4.4.1 Feature Extraction

For each grid g_i and day d_j in the raw data we calculated the following features:

1. **Temporal Features**
   This includes the features that vary with respect to time, and is same for each spatial grid.

   (a) *Precipitation*: Precipitation on the day d_j.

   (b) *Snow*: Snow on the day d_j.

   (c) *Tmax*: Maximum temperature on the day d_j.

   (d) *Tmin*: Minimum temperature on the day d_j.

   (e) *Tavg*: Average temperature on the day d_j.

Since the weather attributes are recorded for each day, our prediction scale is also on a daily basis.

2. **Spatial Features**
   This includes the features that vary with respect to grid, and is same for each time instant.

   (a) *yelp_factor*: Kernel density estimate of neighboring local businesses from the center of grid g_i, which is same across the entire year

   (b) *police_factor*: Kernel density estimate of neighboring police location from the center of grid g_i, which is same across the entire year.

   The spatial features, viz. yelp_factor and police_factor were calculated using Gaussian Kernel Density (KDE)[7] estimator. Mathematically, it is given by:

$$\rho_k(g_i) = \sum_{i=1}^{k \times k} K(dist(g_i, x_i), h) \tag{4.1}$$

   where, K = kernel density function,
   h = bandwidth parameter and taken as 0.0008.

3. **Spatio-Temporal Features**
   This includes the features that vary with respect to each time instant and for each spatial grid.

   As we have previously seen, time-series data has a strong correlation with its historical value, we add similar features in this model.

   (a) *previous_day_crime_freq*: Total number of crimes that recorded on d_{j-1} on grid g_i.

   (b) *previous_week_crime_freq*: Average number of crimes recorded in past week of d_j on grid g_i.

4.4.2 Problem Statement

We formulate our problem statement as a regression problem given below:

Given a set of training data of the form $\{X_d, y_d\}$ till day d, such that: $m = 9$ (total number of features explained in last section) X_i = vector of m spatial and temporal features,
N = total number of training instances and

[7] http://scikit-learn.org/stable/modules/density.html#kernel-density-estimation

y_d is the number of crimes occurred at day d and grid g.

Build a model that can predict y_{d+1} i.e number of crimes at grid g_i on the following day such that:

$$y_{d+1} = f(X_d, y_d, g_i) \tag{4.2}$$

The above equation is a classic problem of regression and the function f is learned using different supervised machine learning algorithms explained briefly in Section 1.5.2.

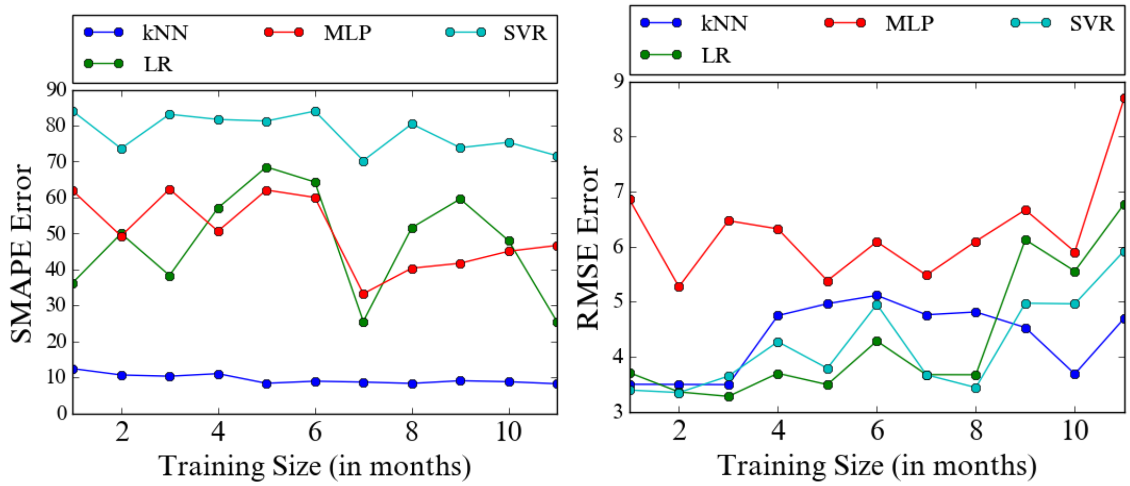
4.5 Results and Observation

Given the features described in the last section, we trained the regression model on LR_SGD, MLP, kNN and SVR. Like the earlier model, we perform cross-validation based evaluation on our dataset by setting aside one month of data for validation. We use first i months of data to train, $(i+1)^{th}$ month to validate and tune the model with parameter that has least RMSE. Once the parameter is tuned for the entire year, we incrementally set the training size to be first i months and test it on $(i+1)^{th}$ month. By having this type of approach, we test every month of data except January. It is important to mention that since the number of test data points is high (eg: $10 \times 10 \times 30$ grids $\times$ days), the absolute error is averaged out and and holds a numerically small value.

4.5.1 Algorithm wise model performance

We start our analysis for using 10×10 grids and by predicting aggregated criminal activity across each grid. This means each grid is about 41 $miles^2$ in area. Figure 4.6a provides the error and training time associated with each algorithm and with specified features. This helps us in understanding which algorithms work well in criminal spatio-temporal data.

On average, kNN gives the least SMAPE of 9.49% averaged over 11 months of cross-validated test data, as can be seen from Figure 4.6a. As the RMSE of LR is less than that of kNN for multiple months, we further went ahead in analyzing why the percentage error for LR is so high. Figure 4.7 shows the actual and predicted value of crime aggregated across all the 10 $\times$ 10 grids for LR, MLP and kNN. The error for SVR is so high, that we dropped it from our further analysis. The high RMSE and SMAPE for MLP is very apparent as MLP fails to correctly predict the overall crime count. The LR does a good job of finding the total crime in the city across all locations, as compared to kNN, however it fails to identify the locations associated with the crime. To further corroborate the fact, we performed a recursive feature elimination to identify the feature importance for LR. Our analysis showed that the spatial features had a lower ranking than temporal features, and thus, LR was successful in achieving good prediction on overall crime for a day when considered only temporally, but



(a) SMAPE (b) RMSE

Figure 4.6: SMAPE and RMSE of all algorithm in 10×10 grids

not spatially. Also a poor seasonal performance can be seen in Figure 4.6a by models like MLP and LR, as it cannot handle the change in the season, and ended up giving high error during the transition from winter to summer and from fall to winter.

4.5.2 Effect of grid size on model performance

After identifying that kNN performs the best algorithm for our data, we evaluate the impact of grid size on the performance of kNN. The 100 grids corresponds to an area of 41.44 miles^2. Next, we take 20×20 grids with each grid having an area of around 10.36 miles^2. As shown in Figure 4.8, we see a slight increase in average SMAPE error across 11 months, from 9.49% to 11.98%, but the training time is increased almost six times from 0.60374 sec to 3.87 sec. One of the reasons for the decrease in model performance is because we only use 12944 unique businesses. As we increase the grids, the distribution of businesses per grid tends to be similar for all grids, thereby making it difficult for the model to identify certain hot-spots. Thus, we see a trade-off in terms the geographical scale we are making our prediction and with what accuracy.

4.5.3 Aggregated Crime vs Crime Type

We extend the proposed model to see its performance in identifying specific crimes. Here, we take the historical data of the crime type for which we are predicting. As can be seen in Figure 4.9, learning individual crime type increases the percentage error but remains less than 15% for all the test months. Thus we can use the model with the suggested features

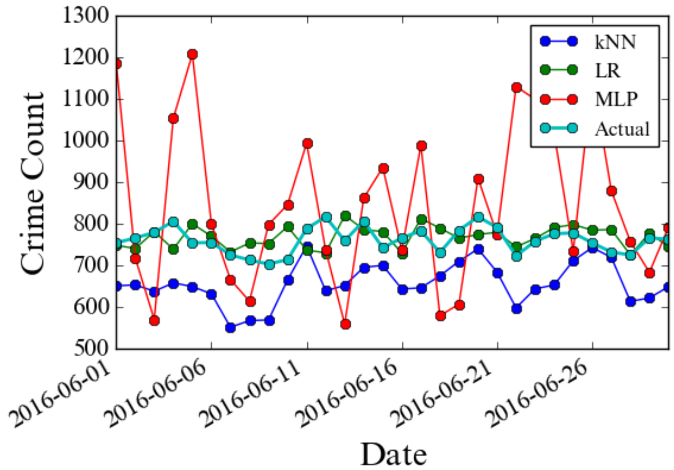
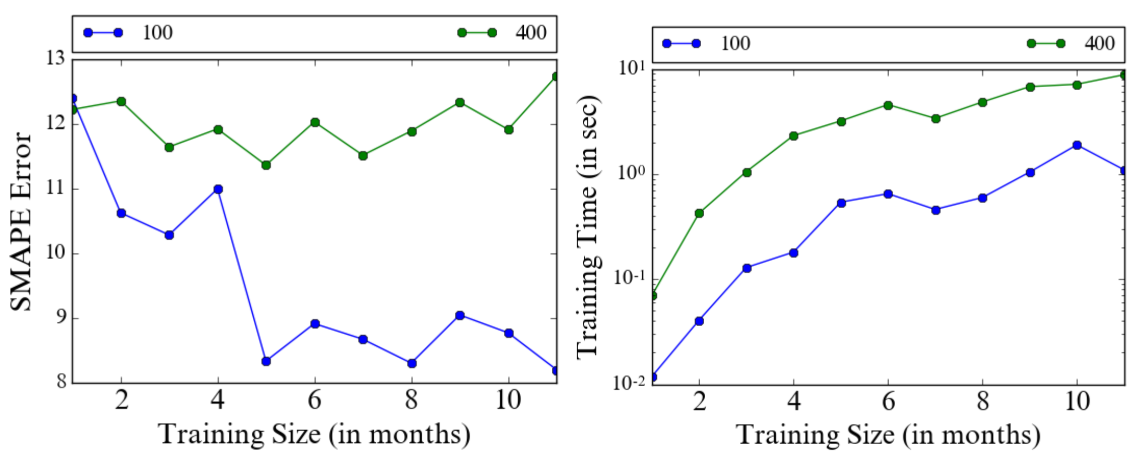


Figure 4.7: Crime count across all locations for the month of June with the highest seasonal effect



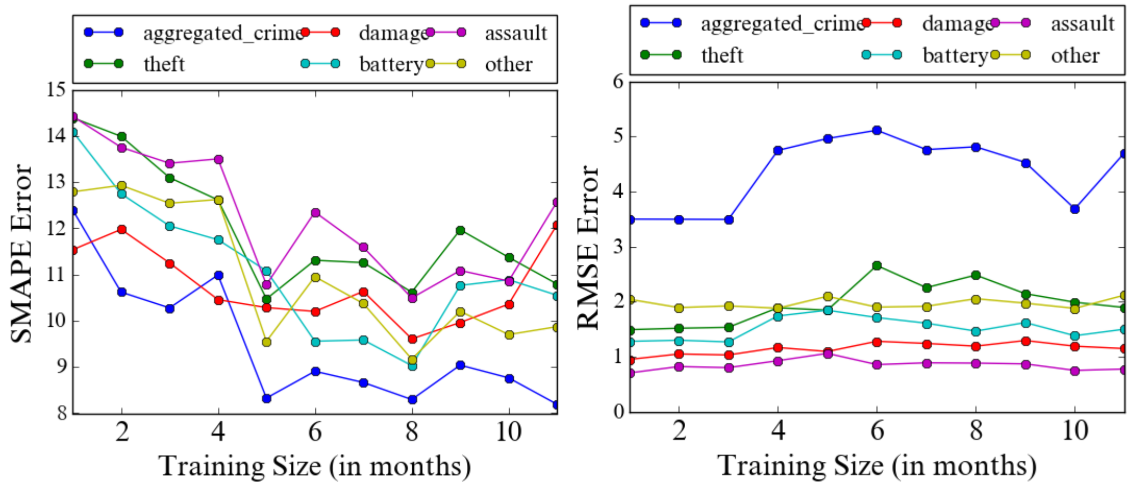
(a) SMAPE (b) Train Time

Figure 4.8: SMAPE and training time of kNN w.r.t to number of grids

for predicting not only overall crime, but also a specific type of crime with only a slight performance trade-off.

4.6 Summary

In this chapter, we used a spatio-temporal approach to solve the problem of predicting crime in Chicago for the year 2016. Particularly, this study advances research in the area of location-centric criminology by considering variables which have not typically been included



(a) SMAPE (b) RMSE

Figure 4.9: SMAPE and RMSE of kNN in 10 × 10 grids when predicting specific crimes and aggregated crime

before, such as police locations and business addresses. We introduce our set of features that captures the temporal and spatial distribution of crime. We compare different regression techniques viz., k-Nearest Neighbor (kNN), Linear Regression (LR), Multi-Layer Perceptron (MLP) and Support Vector Regression (SVR). From our experiments, we found that kNN achieves the best performance (avg. 9.49% error) in predicting overall crime count throughout the city. We also found that with only a slight performance trade-off (max. 15% error), we can predict a specific crime type using the similar features, which illustrates the genericness of our framework to adapt to specific requirements.

In the next chapter, we analyze the impact of nearby traffic on electric vehicle charging demand.

Chapter 5

Spatio-Temporal Analysis of Electric Vehicle Charging Demand

5.1 Overview

As we saw in Figure 1.1, more and more number of people are using electric vehicles every year. Increased EV means increasing charging demand and increased stress on the power grid. EV charging vary in accordance with how long they charge their vehicles on a discharged battery. Thus, it becomes vital to analyze the properties that affect their charging behavior, which in turn is decided by their travel behavior. That being said, analyzing how human mobility impacts EV charging demand becomes an interesting problem. In this chapter, we analyze the impact of historical traffic volume recorded around two charging stations and identify their impact on the charging demand. We start by discussing existing work on forecasting charging demand. We continue analyzing the charging and traffic data, define the problem statement, and explain the analysis steps. We illustrate the application of our traffic model explained in Chapter 3 to predict charging demand. We also analyze the impact of traffic aggregated around four charging locations on the overall demand. This helps in comparing and contrasting the impact of traffic nearby and in general for the entire city. We conclude this chapter by providing the summary of our observations.

5.2 Literature Review

One of the most common problems in power systems, predicting charging demand has been extensively studied before. One of the first questions that is usually asked is what type of prediction? Is it real-time (for next hour or so), short-term (daily or weekly) or long-term (monthly or yearly). Using probability theory, a stochastic model is presented in [2] for a real-time demand forecasting. Surveys based on user preferences and market research has been utilized in [20] to build a demand model. Quantities related to vehicles like battery state and capacity, and related to charging events like charging start-time and duration along with initial State-of-Charge (SoC) has been utilized in [32, 46]. However, characteristics like SoC is not always available and so, often times, the models have to rely on simulated data.

The area of EV charging demand is not untouched from the supervised machine learning techniques for modeling the prediction model. The work proposed in [63] uses Support Vector Machine (SVM) against the Monte Carlo method on one year of simulated historical charging data to improve the forecast for a demand for next day. Artificial Neural Network (ANN) is used in [40] to predict EV charging demand for efficient building power management. Even though the advanced data mining techniques are used in these works, these models simulate the charging event data including charging start time, demand and initial SoC. However, it is difficult to generalize synthetic data for the real-world scenario, especially with the increasing demand that can follow a trend compared to a simple probabilistic distribution.

With the advancement of sensor technologies and Internet of Things, detailed data can be collected from EVs and charging stations. The real-world data then collected can be suitably analyzed to capture and learn the power consumption characteristics and can be later used to predict the demand. A modified pattern sequence algorithm is presented in [34] that uses k-Nearest Neighbor (kNN) technique to predict charging demand for the next 24 hours by matching against a similar historical day. The authors in [62] have compared four data mining algorithms viz Decision Table, Decision Tree (DT), ANN and SVM. Predicted demand can be later used to schedule economical purchase of electricity in the day-ahead market [64]. Apart from this, historical charging demand data can be analyzed to monitor real-time demand and provide a mobile application [36] to EV users to schedule their EV charging conveniently.

Interestingly, only historic charging data along with other supporting data like weather [61] has been used to predict the charging demand. The authors in [5] use historical traffic distribution data in South Korea to cluster and find traffic similarities, use weather data to identify influential factor using relational analysis and build a forecasting model for demand prediction. However, most of the data used in this study is simulated and couldn't be generalized with real-data with utmost confidence.

The contribution of this work is different from other EV charging demand forecasting models by analyzing following different scenarios: 1) We use an end-to-end real world charging and traffic data collected from different locations in the New York City. 2) Not only do we analyze the impact of near-by traffic on the charging demand on multiple locations (spatio-temporal) individually, but also study the impact of the nearby traffic volume on the aggregated charging demand (only temporal) for the entire city. With the availability of real-world taxi volume and charging event data, gives us an exciting research area to explore.

5.3 Data Description

Charging Data

We collected several months of charging data from two EVgo's[1] DC fast charging stations each located in Brooklyn and Flushing and referred as S_1 and S_2 hereafter, respectively. These are the same locations we analyzed the traffic around in Chapter 3. Table 5.1 enlists some of the properties of the data. One charging event corresponds to a session when an EV is connected to one of the charging outlets. Six months of data was collected for station S_1, while only three months of data was collected for station S_2, as it was effectively active for less time period. The basic unit for our forecasting is kWh (kilo Watt hour). Since the average number of events is less than 3 per day, we limit our analysis on a daily basis.

Table 5.1: EV Charging Data Description

	Station S_1	**Station S_2**
Date	07/01/16 - 12/31/16	10/01/16 - 12/31/16
Charging Events	300 (263 weekdays)	131 (112 on weekdays)
% Weekday Events	87.66%	85.49%
#Days with zero demand	43 (23.37%)	21 (22.83%)
Avg. consumption per day (in kWh)	17.35	14.87
Avg. number of events per day	2.13	1.84

Traffic Data

For traffic data, we use the same dataset as discussed in Chapter 3. Specifically, we calculate traffic volume, i.e number of trips recorded per day nearby. Nearby traffic means the trips that are made in the same zone as that of the charging station. The zones are enlisted on the NYC-TLC website[2]. For considering the trips that were made within the city and for a reasonable distance only, we removed the trips that had total trip travel-time more than 24 hours and less than 1 minute. Table 5.2 enlists some of the properties of the data.

Table 5.2: Traffic Data Description

	Station S_1	**Station S_2**
Date	07/01/16 - 12/31/16	07/01/16 - 12/31/16
Total Trips	61713 (52477 weekdays)	8889(1431 on weekdays)
% weekday trips	85.03%	83.90%
Avg. trips per day	333.58	95.58

[1] `https://www.evgo.com/about/`

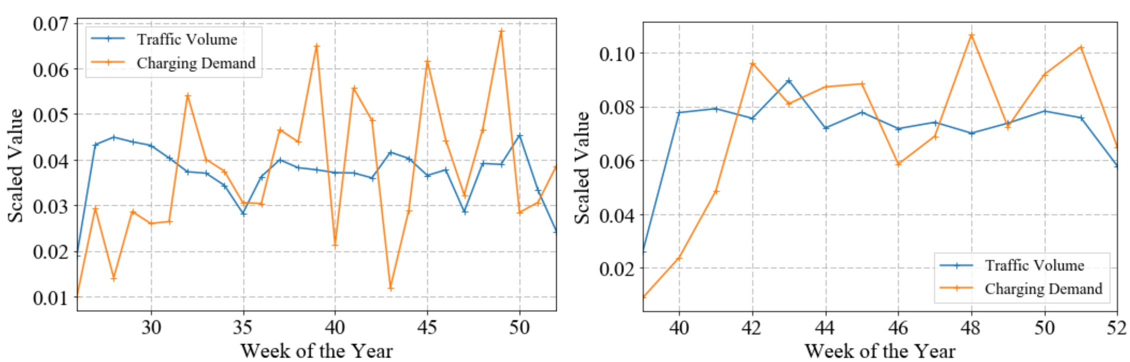
[2] `http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml`

5.3.1 Data Preprocessing

We clean the traffic data with invalid pick-up and drop-off locations. We also filter out trips as described in traffic data description. The days for which no charging events occurred is set to have a demand of 0 kWh. After cleaning, we merge both of these datasets. After the merged data is ready, we separate last 4 weeks, i.e December month data for testing. From the remaining 5 months of data for station S_1, last month (November) is reserved for validation, while another 4 months of data (July-October) is used for training. Similarly, for station S_2, November is reserved for validation and October for training. After separating the training and test data, we extract the temporal features using training data and analyze them further described in detail in next section. This completes the preprocessing step for our modeling process.

5.3.2 Data Exploration

Figure 5.1 shows the variation of the scaled value of charging demand and traffic volume for multiple weeks for both the stations. Scaling the values helps in analyzing both charging and traffic demand on the same plot. Though the demand does not significantly vary with the traffic, there is a noticeable similarity in the trend for the week 32 to 35 for station S_1. For station S_2, we see a similar increasing trend in both types of demand during the initial three weeks. Since most of the data covered in training set is in winter season, we do not study seasonality effect for this data.



(a) Station S_1 (b) Station S_2

Figure 5.1: Fraction Demand and Traffic Volume with respect to week of the year

Figure 5.2 shows the daily charging demand and traffic volume recored as a fraction of total demand and volume. On analyzing the plot, we observe that the traffic volume helps capturing the charging demand for both weekdays and weekends for station S_1. On the contrary, the trend for charging demand and traffic volume for station S_2 is contrasting.

Particularly, the very peak of traffic demand against the low charging demand on Saturdays can be pointed back to the point that station S_2 has tourist attraction nearby, as discussed in Section 3.3, which doesn't necessarily apply that people will charge their vehicles when on a recreational visit to the park or museum. Thus, it can be hypothesized that the nearby traffic volume have an impact on charging demand, but the exact relation (positive or negative) depends upon the geographical location. Therefore, we start with similar set of features for both the locations but later use tailored set of features for both locations individually.

(a) Station S_1 (b) Station S_2

Figure 5.2: Fraction Demand and Traffic Volume with respect to day of the week

With this exploration in mind, in the next section we formulate a simple spatio-temporal time series forecasting problem.

5.4 Problem Statement

Let e_d represents total daily charging demand (in kWh) at day d such that $e_d \in \mathbb{R}_{\geq 0}$ and $d \in \mathbb{Z}_{>0}$. Similarly, t_d represents total traffic volume at day d near the station such that $t_d \in \mathbb{Z}_{\geq 0}$. By collecting the charging demand and taxi volume recoded till day d, a set of discrete time-series data points is obtained which can be represented as:

$$E_d = \{e_1, e_2, e_3, \ldots, e_d\} \tag{5.1}$$

$$T_d = \{t_1, t_2, t_3, \ldots, t_d\} \tag{5.2}$$

where E_d and T_d represents set of daily historical charging demand and historical taxi trips made per day till day d respectively. Since it is assumed that bidding decisions for the next day in a day-ahead market begins at around 12 noon [16], we exclude the charging demand e_d and trips t_d for predicting demand at day $d + 1$.

The problem of daily electric vehicle charging demand in a day-ahead market can be formulated as a typical time-series data-mining problem as follows: Given set of historical charging demand E_{d-1} (in kWh) and taxi trips T_{d-1}, predict the relationship f, such that:

$$e_{d+1} = f(E_{d-1}, T_{d-1}) \tag{5.3}$$

The relationship f is studied separately for both the stations, making it a spatio-temporal forecasting problem. Towards the end of this paper (Section 5.8), we also perform a temporal analysis of the charging demand by aggregating the charging demand and nearby traffic across all the stations.

5.5 Model Development and Analysis

In this chapter, we explain the model we use to analyze the impact of nearby historical traffic on electric vehicle charging demand.

5.5.1 Parameter Tuning

We use SVM, LR_SGD and MLP as our basic algorithms for regression analysis. Since these methods require parameter tuning, we defined multiple sets of tuning parameters for each of these models. We calculated Root Mean Squared Error (RMSE) for each set of parameters, and picked the one that had the lowest RMSE. For our experimental purpose, days in December is used for testing, so $N = 31$ in Equation 1.6.

5.5.2 Feature Extraction and Elimination

Similar to features discussed in previous chapters, we start with extracting various temporal features described as follows:

- *day_d*: day of the week (int: 0-Monday, 6-Sunday)
- *consumption_d-2*: total consumption (in kWh) at day $d-2$
- *events_d-2*: number of charging events at day $d-2$
- *traffic_volume_d-2*: traffic volume at day $d-2$
- *n_week_avg_consumption*: average of total consumption in previous n weeks on the same day of the week as d (n=1, 2, 3, 4)

- *n_week_avg_traffic_volume*: average traffic volume recorded in previous n weeks on the same day of the week as d (n=1, 2, 3, 4)

These were our initial set of features we used for further performance analysis. However, because of the limited data points, it becomes necessary to avoid over-fitting as a consequence of using more features. Thus, we check the correlation of these features with the demand for $d+1$ calculated on the full-training data (including training+validation data). The criteria used for correlation is Pearson Correlation Coefficient, r and is given by:

$$r = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})^2}} \tag{5.4}$$

where $\bar{x}$ and $\bar{y}$ are mean value of x and y respectively.

We then select half of the features that had higher coefficients than the others. The final list of features after first set of feature elimination is listed in Table 5.3. In order to further select the most relevant features, we use one of the common feature elimination techniques as described in [8]. Given the features (Table 5.3) and their correlation coefficient with the forecasted demand, we filter the top k correlated features for our model. The value of k is selected by evaluating the k-featured-model performance on the validation dataset.

Table 5.3: Features selected for the model

Feature	**Station S_1**	**Station S_2**
4_week_avg_consumption	*	*
events_d-2	*	*
4_week_avg_traffic_volume	*	
day_d	*	
traffic_volume_d-2	*	
consumption_d-2	*	*
3_week_avg_consumption		*
2_week_avg_traffic_volume		*
1_week_avg_traffic_volume		*

5.6 Results and Observations

In order to understand the impact of nearby traffic volume on the charging demand, we compare the model with features discussed in previous section with the following two baseline models (BMs) that uses historical charging data only. Out of 4 data mining algorithms presented in [62], SVM and MLP performs the best, so we choose them with the features described in the paper. We refer it as C_SVM, C_MLP, where C represents that the models that use historical Charging data only. Similarly, CT_X represents the model X that we have proposed and contains both Charging and Traffic related features.

Next, we use kNN and Modified Pattern Sequence-based Forecasting (MPSF) described in [34]. It forecasts the demand on an hourly basis. However, the dataset used in our experiment is very sparse with less than 3 charging events per day for both the stations. Therefore, we predict demand on a daily basis rather than hourly for fair evaluation. As discussed in last section, we filter top k features for selecting the most relevant features. Tuning parameters and top k features are selected based on validation set performance. The test data is fixed to 31 days of December 2016.

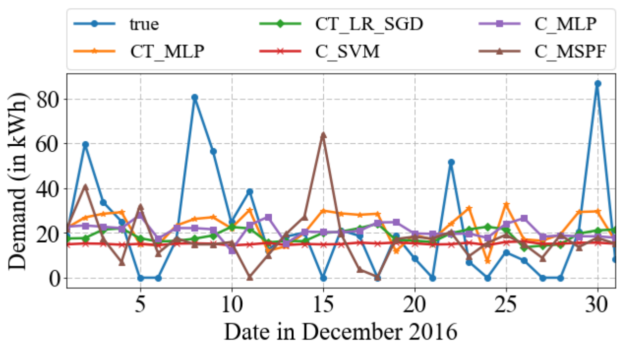
Table 5.4: Error calculated for models with and without features extracted from nearby traffic data (top k features for S_1 and Table 5.3 for S_2)

	Station S_1			**Station S_2**		
Model	**MAE**	**RMSE**	**SMAPE**	**MAE**	**RMSE**	**SMAPE**
C_SVR	16.49	23.85	48.19	11.68	14.6	36.31
C_MLP	17.42	23.32	46.77	10.55	12.79	37.44
C_kNN	17.19	23.69	47.77	11.25	13.56	40.59
C_MSPF	17.7269	26.09	49.49	12.53	16.83	44.29
CT_SVR	17.37	24.05	53.35	11.49	13.96	37.19
CT_MLP	15.82	21.11	44.43	12.99	15.19	42.34
CT_LR_SGD	15.79	22.93	44.19	11.09	13.8	35.96

Table 5.4 shows the error calculated using the models with (CT_X) and without (C_X) traffic features. The lowest error in each column is marked as red. CT_MLP outperforms C_MLP by 2.34% for S_1, suggesting that the nearby traffic plays a significant role in improving the demand forecast for a particular charging station.

For station S_1, an overall improvement of 2.58% is observed by LR_SGD when the spatio-temporal traffic features are added to the model against C_MLP that performs the best when only historical charging data is considered. It selects top 5 features from the features available (Table 5.3) and assigned average weights as follows: *4_week_avg_consumption* (0.23), *events_d-2* (0.04), *4_week_avg_traffic_volume* (0.25), *day_d* (0.24) and *trip_d-2* (0.24), the highest weight given to the average traffic volume made on the same day for one month, which corroborates the fact that nearby traffic has an impact on charging demand.

However for station S_2 (Figure 5.3b), we see that the model LR_SGD that performs the best has a constant output using the features given in Table 5.3. C_MLP and CT_MLP do try to learn the variation in form of a periodical low and peak demand, but there seems to be no general periodicity in the data, so they fail. Even the model that performed best with only historical data (C_SVM) has a constant output. The presence of traffic features helped the model (CT_LR_SGD) to converge better and gave a slight improvement on performance by a 0.35%, which may not be considered too significant. It is important to highlight that we reported the model with all the features for station S_2 than top k features like we did for station S_1. This is because when we selected top k for station S_2, the performance of CT_MLP and CT_SVM wasn't better than the models with historical charging demand only. Moreover, CT_LR_SGD gives a very high error of 52.22% by selecting only *events_d-2* when we tried top-k features, which clearly wasn't sufficient for an improved forecast.



(a) Station S_1 (b) Station S_2

Figure 5.3: Model Output on Test Data for models with and without features extracted from nearby traffic data

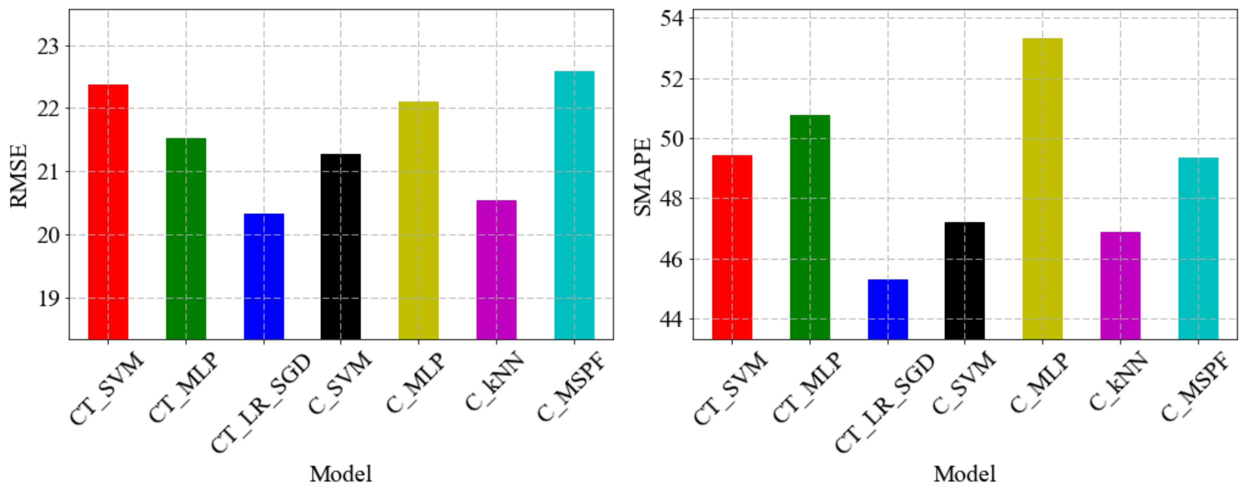
By looking at the results, we say that the nearby traffic has more impact on a station that is more active based on number of charging events made per day. This is because a station with less number of non-zero event days is more reflective of the non-zero traffic volume per day. However, different charging locations that have different demand pattern, can lead to a model with different spatio-temporal features. Thus, if the demand is not as dense as traffic volume, it has a small impact on the forecasting performance.

5.7 Application of Traffic Model Framework

After establishing the fact that nearby traffic can help improving the forecasting accuracy for electric vehicle charging demand, we present one of the ways by which we use the traffic model discussed in Chapter 3. We introduce a term called Impact Factor, IF for a day $d+1$, given by:

$$IF_{d+1} = \sum_{g=1, g \neq g_0}^{G} \frac{T_{d+1}^{g}}{(D_{g_0}^{g})^2} + T_{d+1}^{g_0} \tag{5.5}$$

where
T_{d+1}^{g} represents the traffic volume for the next day for which we have to predict charging demand, g_0 represents the grid which has the station, and $D_{g_0}^{g}$ represents distance of grid g from the grid g_0 and G is the total number of grids.



(a) RMSE Error (b) SMAPE Error

Figure 5.4: SMAPE and RMSE Error for models with historical charging data vs models with futuristic traffic feature as Impact Factor for Station S_1

We took the traffic data from July till December 2016 and used the model to predict the future traffic count from January to March 2017 for station S_1. We didn't further analyze station S_2 because of the limited charging data for station S_2. It is important to highlight that the traffic data for the year 2017 is not published on the website till the date of writing this thesis. We used the predicted traffic T^g_{d+1} at each grid and calculated the IF as defined in Eq. 5.5.

Figure 5.4 shows the RMSE and SMAPE errors recorded for models with IF as features against the baseline models, that contain historical charging data only. RMSE improves by 0.5% and SMAPE decreases from 46.9% to 45.48%.

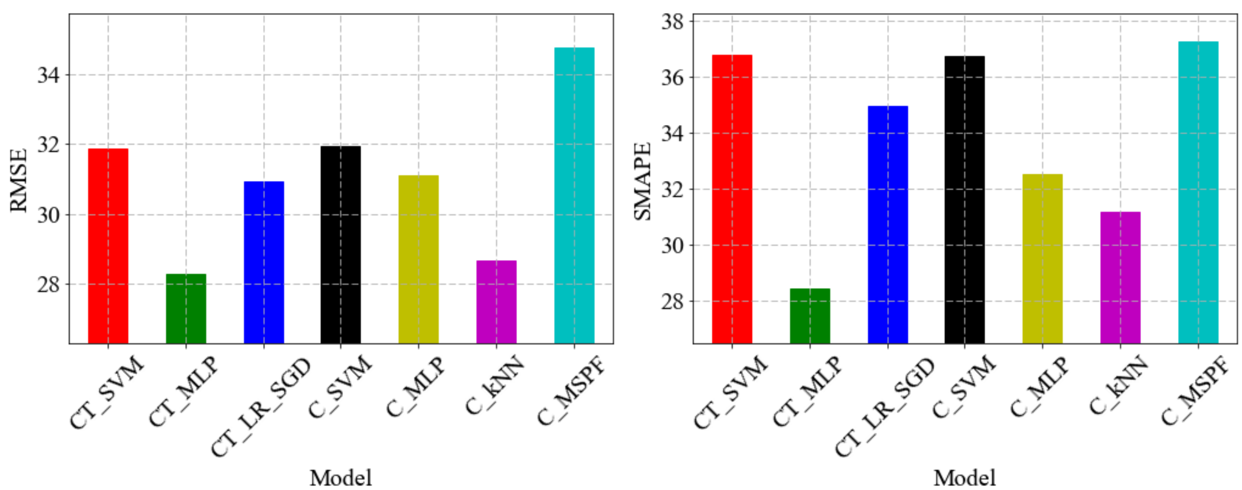
5.8 Aggregated Traffic and Charging Data

Until now, we discussed the spatio-temporal analysis of traffic and charging data at two different locations. In order to understand the impact of the overall traffic on the charging demand throughout the city, we aggregated the nearby traffic volume recorded and the charging data recorded from 4 different charging locations (two from previously discussed and two additional) situated in the NYC on a daily basis. A general statistics of the aggregated data is described in Table 5.5.

We perform a similar analysis as explain in Section 5.5.2. CT_MLP was successful in finding some non-linear relationship from traffic features like *traffic_volume_d-2*, *n_week_traffic_volume (n=4, 3, 2)* and improved the performance by 2.74% against C_kNN which performed best

Table 5.5: Aggregated charging data attributes collected from 4 charging locations in the NYC

	4 NYC Stations
Date	07/01/16 - 12/31/16
Charging Events	584 (493 weekdays)
% Weekday Events	84.42%
#Days with zero demand	10 (5.43%)
Avg. consumption per day (in kWh)	34.18
Avg. number of events per day	3.17



(a) RMSE Error (b) SMAPE Error

Figure 5.5: Error for aggregated models with and without features extracted from traffic data

amongst the baseline models (Figure 5.5). Thus, we see that there is a significant impact of the near-by traffic on the overall charging demand encountered by a fast charging station network operated in a urban city.

5.9 Summary

This chapter investigated the impact of nearby traffic on the EV charging demand. We used taxi trip data to calculate traffic volume around two charging stations in NYC. It was observed that if the charging station is active on a daily basis, having information of nearby traffic data can be useful in improving charging demand forecast by 2.58%. However, if the charging station is less active on a daily basis, (unlike traffic data), having information of nearby traffic information has a very slight affect on the charging demand forecast. We also presented an application of our traffic model to improve charging demand prediction using the predicted traffic volume. Additionally, while considering the charging demand at multiple charging locations throughout the city, demand forecasting was improved by 2.74% using the features extracted from nearby traffic. In our last chapter we will conclude our findings and offer suggestions for future work.

Chapter 6

Conclusion and Future Work

We conclude this thesis by providing the observations from our analysis, highlight any shortcomings and discuss ideas for future extension to this work.

6.1 Conclusion

In this thesis, we performed spatio-temporal analysis of different urban datasets and presented a framework on how it can be used for certain applications for smart cities. The main datasets that were studied include: traffic data, crime data and electric vehicle charging data. We started with describing a framework to analyze nearby traffic around two locations in NYC. We evaluated the performance of basic regression algorithms like LR, Decision Tree and MLP. The results showed that the LR achieves best performance while keeping the training time low and a higher resistance to seasonal effect. In order to explain a genericness of our spatio-temporal analysis, we used a similar approach to analyze crime in the city of Chicago and used the model to predict overall and individual types of crime. After evaluating the performance of kNN, MLP, LR and SVR, the analysis showed that kNN achieves best results by correctly predicting the spatial distribution of the crime. We observed that with only a small performance trade-off, we can also predict count for a particular crime type as compared to predicting overall crime. At the end, we used the nearby traffic to analyze its impact on the charging demand recorded at a charging station. Our results showed that for a station that is active and regularly used, forecasting it's demand for the next day can be improved if we have information about the traffic near the charging station. Conclusively, we demonstrated that how different spatio-temporal data can be useful in analyzing many urban phenomena and can be used in various applications.

Next we will discuss some limitations and suggestions for further work.

6.2 Future Work

The richness of the data is that is contains many hidden information. We presented some of the spatial and temporal features in our analysis, but there are many other features that can be added to further tune the model. For example, in order to learn anomaly in the data we added road accident information only. Other features like information nearby sports and cultural events can be added. Real-time weather data can be added to perform if prediction is performed in an online fashion. We performed our traffic analysis on a half-hourly basis but used it in analyzing charging demand on a daily basis because of the sparseness of our charging data. Thus, given a charging station which is very active, we can further analyze it on a hourly basis with respect to nearby traffic. An analysis on a smaller precision scale would be interesting to check if it can further improve the performance or can predict sudden peak demand. With the massive use of social network, we can further analyze Twitter data, similar to Yelp data to calculate high intensity zones that are prone to crime. Further analysis can be done that are more tuned depending upon the severity of the crime.

Bibliography

[1] E. Al Nuaimi, H. Al Neyadi, N. Mohamed, and J. Al-Jaroodi. Applications of big data to smart cities. *Journal of Internet Services and Applications*, 6(1):25, 2015.

[2] M. Alizadeh, A. Scaglione, J. Davies, and K. S. Kurani. A scalable stochastic model for the electricity demand of electric and plug-in hybrid vehicles. *IEEE Transactions on Smart Grid*, 5(2):848–860, 2014.

[3] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[4] N. Andrienko and G. Andrienko. A visual analytics framework for spatio-temporal analysis and modelling. *Data Mining and Knowledge Discovery*, 27(1):55–83, 2013.

[5] M. B. Arias and S. Bae. Electric vehicle charging demand forecasting model based on big data technologies. *Applied Energy*, 183:327–339, 2016.

[6] J. Ballesteros, B. Carbunar, M. Rahman, N. Rishe, and S. Iyengar. Towards safe cities: A mobile and social networking approach. *IEEE Transactions on Parallel and Distributed Systems*, 25(9):2451–2462, 2014.

[7] C. Bergmeir and J. M. Bentez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192 – 213, 2012. Data Mining for Software Trustworthiness.

[8] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1):245 – 271, 1997.

[9] A. Bogomolov, B. Lepri, J. Staiano, N. Oliver, F. Pianesi, and A. Pentland. Once upon a crime: Towards crime prediction from demographics and mobile data. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, pages 427–434, New York, NY, USA, 2014. ACM.

[10] B. E. Boser and et al. A training algorithm for optimal margin classifiers. In *PROCEEDINGS OF THE 5TH ANNUAL ACM WORKSHOP ON COMPUTATIONAL LEARNING THEORY*, pages 144–152. ACM Press, 1992.

[11] S. Chainey, L. Tompson, and S. Uhlig. The utility of hotspot mapping for predicting spatial patterns of crime. *Security Journal*, 21(1):4–28, 2008.

[12] H. Chen, W. Chung, J. J. Xu, G. Wang, Y. Qin, and M. Chau. Crime data mining: a general framework and some examples. *Computer*, 37(4):50–56, April 2004.

[13] X. Chen, Y. Cho, and S. Y. Jang. Crime prediction using twitter sentiment and weather. In *2015 Systems and Information Engineering Design Symposium*, pages 63–68, April 2015.

[14] B. C. Csáji, A. Browet, V. A. Traag, J.-C. Delvenne, E. Huens, P. Van Dooren, Z. Smoreda, and V. D. Blondel. Exploring the mobility of mobile phone users. *Physica A: statistical mechanics and its applications*, 392(6):1459–1473, 2013.

[15] P. A. C. Duijn, V. Kashirin, and P. M. A. Sloot. The relative ineffectiveness of criminal network disruption. 4:4238 EP –, Feb 2014. Article.

[16] V. Durvasulu and T. M. Hansen. Classifying day-ahead electricity markets using pattern recognition for demand response. In *2016 North American Power Symposium (NAPS)*, pages 1–6, Sept 2016.

[17] E. Eftelioglu, S. Shekhar, D. Oliver, X. Zhou, M. R. Evans, Y. Xie, J. M. Kang, R. Laubscher, and C. Farah. Ring-shaped hotspot detection: A summary of results. In *2014 IEEE International Conference on Data Mining*, pages 815–820, Dec 2014.

[18] A. G. Ferguson. Predictive policing and reasonable suspicion. *62 Emory Law Journal 259 (2012)*.

[19] M. S. Gerber. Predicting crime using twitter and kernel density estimation. *Decision Support Systems*, 61:115 – 125, 2014.

[20] A. Glerum, L. Stankovikj, M. Thémans, and M. Bierlaire. Forecasting the demand for electric vehicles: accounting for attitudes and perceptions. *Transportation Science*, 48(4):483–499, 2013.

[21] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[22] N. Gross and M. Mann. Is there a ferguson effect? google searches, concern about police violence, and crime in u.s. cities, 20142016. *Socius*, 3:2378023117703122, 2017.

[23] Y. Guo, W. Liu, F. Wen, A. Salam, J. Mao, and L. Li. Bidding strategy for aggregators of electric vehicles in day-ahead electricity markets. *Energies*, 10(1):144, 2017.

[24] A. Gupta, A. Mohammad, A. Syed, and M. N. Halgamuge. A comparative study of classification algorithms using data mining: Crime and accidents in denver city the usa. *Education*, 7(7), 2016.

[25] J. He, W. Shen, P. Divakaruni, L. Wynter, and R. Lawrence. Improving traffic prediction with tweet semantics. In *IJCAI*, 2013.

[26] B. M. W. L. A. Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of Transportation Engineering*, 129(6):664–672, Issue: object: doi:10.1061/jtpedi.2003.129.issue-6, revision: rev:1479468037994-30303:doi:10.1061/jtpedi.2003.129.issue-6, .

[27] K. K. Jahromi, M. Zignani, S. Gaito, and G. P. Rossi. Simulating human mobility patterns in urban areas. *Simulation Modelling Practice and Theory*, 62:137–156, 2016.

[28] N. A. Kallioras, K. Kepaptsoglou, and N. D. Lagaros. Transit stop inspection and maintenance scheduling: A gpu accelerated metaheuristics approach. *Transportation Research Part C: Emerging Technologies*, 55:246 – 260, 2015. Engineering and Applied Sciences Optimization (OPT-i) - Professor Matthew G. Karlaftis Memorial Issue.

[29] M. Kantardzic. *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

[30] E. C. Kara, J. S. Macdonald, D. Black, M. Brges, G. Hug, and S. Kiliccote. Estimating the benefits of electric vehicle smart charging at non-residential locations: A data-driven approach. *Applied Energy*, 155:515 – 525, 2015.

[31] R. Katzev. Car sharing: A new approach to urban transportation problems. *Analyses of Social Issues and Public Policy*, 3(1):65–86, 2003.

[32] G. Li and X. P. Zhang. Modeling of plug-in hybrid electric vehicle charging demand in probabilistic power flow calculations. *IEEE Transactions on Smart Grid*, 3(1):492–499, March 2012.

[33] W.-Y. Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.

[34] M. Majidpour, C. Qiu, P. Chu, R. Gadh, and H. R. Pota. Modified pattern sequence-based forecasting for electric vehicle charging stations. In *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 710–715, Nov 2014.

[35] M. Majidpour, C. Qiu, P. Chu, H. R. Pota, and R. Gadh. Forecasting the ev charging load based on customer profile or station measurement? *Applied Energy*, 163:134 – 141, 2016.

[36] M. Majidpour, C. Qiu, C.-Y. Chung, P. Chu, R. Gadh, and H. R. Pota. Fast demand forecast of electric vehicle charging stations for cell phone application. In *PES General Meeting— Conference & Exposition, 2014 IEEE*, pages 1–5. IEEE, 2014.

[37] F. Miao, S. Lin, S. Munir, J. A. Stankovic, H. Huang, D. Zhang, T. He, and G. J. Pappas. Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, ICCPS '15, pages 100–109, New York, NY, USA, 2015. ACM.

[38] M. Momtazpour and N. Ramakrishnan. Characterizing taxi flows in new york city. In *Proceedings of the International Workshop on Urban Computing (UrbComp)*. Citeseer, 2015.

[39] J. D. Morenoff, R. J. Sampson, and S. Raudenbush. Neighborhood inequality, collective efficacy, and the spatial dynamics of urban violence. *Criminology*, 39:517–560, 2001.

[40] K. Nandha, P. H. Cheah, B. Sivaneasan, P. L. So, and D. Z. W. Wang. Electric vehicle charging profile prediction for efficient energy management in buildings. In *2012 10th International Power Energy Conference (IPEC)*, pages 480–485, Dec 2012.

[41] H. Nguyen, W. Liu, P. Rivera, and F. Chen. *TrafficWatch: Real-Time Traffic Incident Detection and Monitoring Using Social Media*, pages 540–551. Springer International Publishing, Cham, 2016.

[42] D. W. Osgood. Poisson-based regression analysis of aggregate crime rates. *Journal of Quantitative Criminology*, 16(1):21–43, 2000.

[43] B. Pan, U. Demiryurek, and C. Shahabi. Utilizing real-world transportation data for accurate traffic prediction. In *2012 IEEE 12th International Conference on Data Mining*, pages 595–604, Dec 2012.

[44] M. Pechenizkiy. The impact of feature extraction on the performance of a classifier: knn, naïve bayes and c4. 5. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 268–279. Springer, 2005.

[45] M. Pechenizkiy, S. Puuronen, and A. Tsymbal. Feature extraction for classification in the data mining process. 2003.

[46] K. Qian, C. Zhou, M. Allan, and Y. Yuan. Load model for prediction of electric vehicle charging demand. In *2010 International Conference on Power System Technology*, pages 1–6, Oct 2010.

[47] X. Qian and S. V. Ukkusuri. Spatial variation of the urban taxi ridership using gps data. *Applied Geography*, 59:31–42, 2015.

[48] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.

[49] R. J. Sampson, S. W. Raudenbush, and F. Earls. Neighborhoods and violent crime: A multilevel study of collective efficacy. *Science*, 277(5328):918–924, 1997.

[50] C. Schnell, A. A. Braga, and E. L. Piza. The influence of community areas, neighborhood clusters, and street segments on the spatial variability of violent crime in chicago. *Journal of Quantitative Criminology*, pages 1–28, 2016.

[51] L. A. Shalabi and Z. Shaaban. Normalization as a preprocessing engine for data mining and the approach of preference matrix. In *2006 International Conference on Dependability of Computer Systems*, pages 207–214, May 2006.

[52] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[53] L. Song, D. Kotz, R. Jain, and X. He. Evaluating location predictors with extensive wi-fi mobility data. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1414–1424. IEEE, 2004.

[54] M. K. Sparrow. The application of network analysis to criminal intelligence: An assessment of the prospects. *Social Networks*, 13(3):251 – 274, 1991.

[55] K. K. Tan, K. K. K. Htet, and A. S. Narayanan. Mitigation of vehicle distribution in an ev sharing scheme for last mile transportation. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2631–2641, 2015.

[56] M. A. Tayebi, R. Frank, and U. Glässer. Understanding the link between social and spatial distance in the crime world. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '12, pages 550–553, New York, NY, USA, 2012. ACM.

[57] T. T. Tchrakian, B. Basu, and M. O'Mahony. Real-time traffic flow forecasting using spectral analysis. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):519–526, June 2012.

[58] J. Wang, Q. Gu, J. Wu, G. Liu, and Z. Xiong. Traffic speed prediction and congestion source exploration: A deep learning method. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 499–508. IEEE, 2016.

[59] T. Wang, C. Rudin, D. Wagner, and R. Sevieri. *Learning to Detect Patterns of Crime*, pages 515–530. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[60] B. M. Williams and L. A. Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6):664–672, 2003.

[61] E. Xydas, C. Marmaras, L. M. Cipcigan, N. Jenkins, S. Carroll, and M. Barker. A data-driven approach for characterising the charging demand of electric vehicles: A uk case study. *Applied Energy*, 162:763–771, 2016.

[62] E. S. Xydas, C. E. Marmaras, L. M. Cipcigan, A. S. Hassan, and N. Jenkins. Electric vehicle load forecasting using data mining methods. In *IET Hybrid and Electric Vehicles Conference 2013 (HEVC 2013)*, pages 1–6, Nov 2013.

[63] E. S. Xydas, C. E. Marmaras, L. M. Cipcigan, A. S. Hassan, and N. Jenkins. Forecasting electric vehicle charging demand using support vector machines. In *Power Engineering Conference (UPEC), 2013 48th International Universities'*, pages 1–6, Sept 2013.

[64] L. Yao, W. H. Lim, and T. S. Tsai. A real-time charging scheme for demand response in electric vehicle parking station. *IEEE Transactions on Smart Grid*, 8(1):52–62, Jan 2017.

[65] D. Zhang, J. Huang, Y. Li, F. Zhang, C. Xu, and T. He. Exploring human mobility with multi-source data at extremely large metropolitan scales. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 201–212, New York, NY, USA, 2014. ACM.

[66] D. Zhang, J. Zhao, F. Zhang, and T. He. comobile: Real-time human mobility modeling at urban scale using multi-view learning. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '15, pages 40:1–40:10, New York, NY, USA, 2015. ACM.

[67] R. Zhang, Y. Shu, Z. Yang, P. Cheng, and J. Chen. Hybrid traffic speed modeling and prediction using real-world data. In *Big Data (BigData Congress), 2015 IEEE International Congress on*, pages 230–237. IEEE, 2015.

[68] T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML 2004: PROCEEDINGS OF THE TWENTY-FIRST INTERNATIONAL CONFERENCE ON MACHINE LEARNING. OMNIPRESS*, pages 919–926, 2004.

[69] Y. Zheng, L. Capra, O. Wolfson, and H. Yang. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):38, 2014.

[70] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma. Understanding mobility based on gps data. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 312–321. ACM, 2008.