Vehicle Future Location Prediction Using Apache Spark for Optimal Performance

By

Muhammad Daud Kamal

(2015.-NUST-MS-GIS-117781)

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Remote Sensing and GIS

Institute of Geographical Information Systems
School of Civil and Environmental Engineering
National University of Sciences & Technology
Islamabad, Pakistan

March 2019

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by Mr. Muhammad Daud Kamal, (Registration No. 117781), of Institute of Geographical Information System has been vetted by undersigned, found complete in all respects as per NUST statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: _____

Name of Supervisor: Dr. M. Ali Tahir

Date: _____

Signature (HOD): _____

Date: _____

Signature(Dean/Principal): _____

Date: _____

CERTIFICATE

It is certified that the content and form of this thesis entitled **"Vehicle Future Location Prediction Using Apache Spark for Optimal Performance"** submitted by **"Muhammad Daud Kamal"** have been found satisfactory for the requirement of the Master of Science degree in Remote Sensing and Geographical Information Systems.

Supervisor: Dr. M. Ali Tahir

Designation: Assistant Professor, IGIS, NUST

Member: Dr. Salman Atif

Designation: Assistant Professor, IGIS, NUST

Member: Khunsa Fatima

Designation: Lecturer, IGIS, NUST

Member: Mr. Asim Mushtaq

Designation: AVP Products Manager, TPL Holdings (Pvt) Limited

External Examiner: Signature _____

Name_____

Designation_____

ACADEMIC THESIS: DECLARATION OF AUTHORSHIP

I, **Muhammad Daud Kamal**, declare that this thesis and the work presented in it are my own and have been generated by me as the result of my own original research.

Vehicle Future Location Prediction Using Apache Spark for Optimal Performance

I confirm that:

1. This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text;
2. Wherever any part of this thesis has previously been submitted for a degree or any other qualification at this or any other institution, it has been clearly stated;
3. I have acknowledged all main sources of help;
4. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
5. None of this work has been published before submission.
6. This work is not plagiarized under the HEC plagiarism policy.

Signed: ……

Date: ……

DEDICATION

“To my loving mother who taught me right and wrong”,

“To my father who taught me that there are no shortcuts in life…” &

“To my brother and sister for their unconditional support. May Allah bless them all.”

ACKNOWLEDGEMENTS

All praises to Allah Almighty Who is the all merciful and the most merciful. To Whom all the knowledge belongs and Who is knower of unknown. All thanks to Allah who blessed me with guidance and knowledge to achieve this. His boundless mercy and blessings are foremost and everything. My all gratitude belongs to Hi, and only to Him.

My thanks to my family who believed in me and prayed for my success especially my father who supported me in every aspect possible. My mother whose prayers took me where I am today. My very special gratitude to my supervisor Dr. Ali Tahir who not only supervised me for the work but also be kind enough to extend his support in every problem that I came across. His farsightedness and visionary judgement made this day possible. Also it was only due to his personal efforts that I was able to submit my paper in journal. May Allah reward him in return. My thanks to Dr. Ejaz and Dr. Javed who allowed me to work in PhD room where I was able to concentrate thoroughly on my work. Sir Junaid who helped me fulfilling the thesis requirements. Their respect, courtesy and availability are highly acknowledged. May Allah shower his blessings upon them.

I am very thankful to my fellows Maryam and Aasia who guided me in important phases of my thesis work. I would also like to acknowledge my closest friends Ehtisham, Abdullah and Rizwan for just being there with me though this. I am also thankful for direct and indirect altruistic support of many whom I haven't mentioned here. I acknowledge all of their sincere efforts.

Muhammad Daud Kamal

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The number of devices equipped with GPS sensors has increased enormously, which are generate a massive amount of data. Analysis can be carried out to use these big data in commercial and security related applications. One such application of these big data are to predict the future location of vehicles based on their previous locations. Using these predictions, different location-aware proactive systems can be develop. There are many models and algorithms which helps predict the future location with high probabilities. What is lacking in this field of study is the need of a system that utilizes minimum computing resources and can predict future locations with higher probability. In this study, we developed an algorithm according to the dataset used that results in lower latency and higher precision. Apache Spark, a big data platform was used for reducing the processing time and computing resources. The algorithm achieved 75% to 85% of accuracy and in some cases where the uses do not change their daily routine frequently to 100% accuracy for all previously visited locations. We compared the prediction results of Apache Spark with simple python algorithm without using Apache Spark and experimentally found that Apache Spark predicts processes up to 300% times faster. This algorithm can help find useful knowledge for commercial, intelligence and other security related application.

Chapter 1

Introduction

Analyzing the movement pattern has always been of keen area of interest, may they be automobile, humans or any other moving object. These movement patterns can help analysts in making decision related to the behavior patterns of an object. For example, the idea of Geo-marketing can be evolved if the pattern of the people who are shopping are observed. Similarly, different location aware applications can help in planning an urban unit by observing the traffic patterns. Approximately 6.8 billion mobile phones were predicted to be used worldwide in 2013 (Mobile, 2013). A mobile user location is better estimated these day by the techniques that are currently being developed and used by the telecommunication providers. Therefore, mobile user's patterns and activities are sensed by using different mobility data records that are saved by telecommunication companies (Giannotti and Pedreschi, 2008).

The objective of observing mobility data is to see why and when the objects move. In order to accomplish this various data sources are used such as Global System for Mobile communications (GSM) or Global Positioning System (GPS) for analyzing and later transforming them into meaningful predicting patterns. The process of predicting pattern is known as Knowledge Discovery (KD) i.e. that is produced from raw data and converted into meaningful knowledge (Renso et al., 2013). With a particular objective to settle on an informed decision concerning which advancements to understand, information was assembled from a few previous literature studies. The survey presented in this study aims at assisting researchers in choosing algorithm for next location prediction. In the existing surveys, however, there are no proper

guidelines for algorithm selection as per the use case. One of such survey papers is (Petzold et al., 2005), where the authors used five algorithms for four users that had different patterns. In most of the studies, few aspects of trajectory prediction are discussed. For example, there are studies which focus on indoor and outdoor navigation. In many studies, the authors have validated the accuracy of these algorithms on given datasets. In our review, we investigated 28 algorithms mostly used in trajectory prediction. We presented a comprehensive survey of several algorithms. The survey starts with discussion on public and proprietary datasets.

For this research we have proposed a methodology for ranking of these algorithms based on number of citations, type of dataset and accuracy. After the ranking is done, we have categorized the paper based on machine learning approaches. For example, for each approach we have investigated the type of algorithm, accuracy, dataset, indoor or outdoor and number of citations. A detailed discussion is also presented, and the top three algorithms are proposed which are based on Markov Model (Ashbrook and Starner, 2003a), NextPlace (Scellato et al., 2011) and Transition Matrix (Yavaş et al., 2005).

One of the aim of this study is to describe the selected studies and discuss the results of these trajectory prediction algorithms highlighted in literature. This information will give the readers an indication of the characteristics of these algorithms. The outcomes feature the qualities and shortcomings of the algorithms best studied and will fill in as a helpful guide to the research community and the business in understanding which techniques to utilize. Thus, developers never again need to depend on numerous studies accessible on the web.

The main purpose of this research is to analyze the spatial-temporal mobility patterns of GPS data using new technologies for big data i.e. Apache Spark in order to reduce the time taken per job for discovering useful information, which can help assist decision making for real-world scenarios.

1.1. Background Information

There are many trajectory prediction algorithms exist in the literature. Over the years various researchers have proposed novel algorithms which cater their need. The focus of our study is to survey the available algorithms and identify the top three algorithms in terms of efficiency, performance and ease of use. Broadly, trajectory prediction algorithm are derived from machine learning approaches such as Bayesian networks (Jensen, 1996), Hidden Markov Models (Rabiner and Juang, 1986), Decision Trees (Quinlan, 1986), Neural Networks (Specht, 1990) and State Predictor Methods (Garnier et al., 1978). This section describes existing work in this field while commenting on above mentioned parameters.

Research in mobility data is not that new. However, in the last years it has gained popularity for data mining and artificial intelligence. Substantial amounts of information are produced by GPS and telecommunication technologies advancement. In a survey paper where (Petzold et al., 2004), used five algorithms for four users that had different patterns.

New innovations and advancement are giving hints of producing pervasive computing for mobility data which helps predict the accuracy. The trajectories that are stored for semantics of mobility data are aiding in finding useful information about the movements of the objects (Vieira et al., 2009). In an investigation (Marketos et al., 2013) the perspectives identified with data

accumulation and taking care of trajectories that are feeding to the databases with proper data. The trajectories recreation for producing meaningful trajectories includes procedures for gathering movement data and cleansing the data gathered, compression of data and map coordinating to deliver noise-free trajectories. For production of semantically compliant trajectories raw spatial data from common repository need to be recovered using different remaking tasks along with semantic trajectories. Moreover, (Spaccapietra et al., 2008) defined the concept behind management of trajectory and their representations. The main focus of the research was analysis on an extensive scale for phenomena related to mobility with more focus on semantic behavior of the data. The main goal of analyzing the behavior is indicating which behavior defines which moving object.

Unprecedented amount of geo-spatial data gathered from moving objects defies human capability to analyses it. A study by (Morzy, 2006) found new methods for processing and mining of moving objects. For modeling and representing trajectories, (Güting et al., 2010b) discuss the problem in the context of database systems. Moving objects databases represent a set of moving objects using abstract data types and maintains complete histories of movement.

An open source software Secondo has frame work for big trajectory data whose data model is not fixed. This DBMS prototype can be used for different data models. "*WhereNext"* where previously visited trajectory patterns were extracted (Monreale et al., 2009) that uses previously extracted trajectory patterns. In most of the studies, few aspects of trajectory prediction are discussed. For example, there are studies which focus on indoor and outdoor navigation. Similarly the other studies highlights public and private datasets. In many studies, the authors have validated the accuracy of these algorithms on given datasets. In

our research, we proposed a novel algorithm using Apache Spark in order to minimize the query and processing time for GPS big data. The reason because Apache Spark is becoming de facto for processing big data in the computing world. We used it to predict future locations of vehicles GPS data.

In a survey carried by (Cheng et al., 2003) who emphasized on the importance of mobile wireless systems in location prediction. The authors briefly described different types of location prediction and analysis algorithms and categorized them into two types based on the approach i.e. domain independent algorithms and domain specific algorithms rather than conducting a detailed survey of all applications and methodologies involved in location prediction. In a different study, the authors focused on the improvement of safety on the road by performing a study on ways and methods by which different dangerous accidents can be avoided by predicting such situation in advance in order to practice the precaution (Lefèvre et al., 2014). The authors based the study on the models that describe motion and risk and also described how the selection of the estimated method is effected by the single out motion model. They used different simulation models such as Monto Carlo simulation, Physics based motion models, Dynamic models, Kinematic models, Gaussian noise simulation, Maneuver-based motion models and Interaction-aware motion models which is built on Dynamic Bayesian networks. In a research by (Lin and Hsu, 2014), issues regarding mobility of different individuals have been studied such as transportation mode, patterns of trajectory, significance of location and other location based models. The authors performed a detailed review of many different algorithms and techniques and compared the obtained results regarding the issues of individual's mobility. Two types of graph approaches were used for mining trajectories from raw traces. First

being the transitions among important scenarios while second the trajectories are changed to spatio-temporal sequence.

An investigation was carried out by (Petzold et al., 2005) for the accuracy of different prediction methods and also studied different location trajectories that were most frequently visited. The author used Augsburg indoor location tracking benchmarks as predictor loads and also used various techniques to model activities. The scenarios involving visiting different offices in the building over some interval of time. These techniques that were studied include Bayesian networks, Neural networks, State and Markov predictors. The author achieved variations among accuracy of different predictive models. Focus on developing a probabilistic model based on the generations of trajectories by tracking different objects over time was carried by (Morris and Trivedi, 2008). The survey also emphasizes on the use of topographical map to specify the points of interests which are in turn connected by activity paths which then explain the way the motion of the objects takes place. The main aim of the model is to concentrate on the events of interest. This way the surveillance systems automatically focus on events such as abnormality detection, prediction of activities, and interaction of objects, online activity analysis, and classification of path, speed profiling and virtual fencing. The existing solutions for geolocation prediction (GP) and divided geolocation prediction into two primary parts were reviewed by (Xu et al., 2017). The initial step proposed to manufacture a geolocation expectation show is mining popular geolocation region (MPGR), and second is mining personal trajectory (MPT). The results described the basic concepts of GP, the characteristics of MPGR and MPT. They also discussed the limitations, openings, and the future geolocation prediction analytical trends for mobility big data.

A new method was proposed for in-door next location prediction method that uses sequences of previously visited locations. Indoor locations reside inside a building that could be an office or an institute. Augsburg indoor location tracking benchmarks (Petzold, 2005) are applied. Several prediction algorithms techniques on the same data and set-up were evaluated. The authors found different number of accuracies for different types of algorithms. The prediction accuracies of 70% to 90% were achieved. A framework for predicting human mobility behavior was proposed by (Do and Gatica-Perez, 2012). They proposed a method that combines a set of models which learn various mobility patterns from past observations. Considering a difficult real-life data set, they demonstrated the potential of their approach on predicting human mobility in real world conditions. One issue that they encountered is the low rate of trusted observations; this was solved by improving the sensing technique and by exploiting the missing observation in the learning framework. They considered the prediction of user mobility when they arrive to or leave a place and considered only discrete contextual variables to simplify the estimation of basic models. In an experiment carried by (Froehlich and Krumm, 2008) showed how the regularity of a driver's traveling behavior could be exploited to predict the end-to-end route for their current trip. The authors made three primary contributions. First, authors provided a methodology for automatically extracting routes from raw GPS data without knowledge of the underlying road structure. Secondly, authors presented a detailed discussion and analysis of repeat trip behavior from a real world dataset of 14,468 trips from 252 drivers. Finally, authors developed and evaluated two algorithms that used a driver's trip history to make route predictions of their current trip.

Multi-depot vehicle routing problem (MDVRP) was proposed by (Ho et al., 2008). The number of depots is not limited to one in many real-world situations. Similarly, (Guessoum et al., 2016) proposed a methodology for the prediction of a user's outdoor location derived from contextual data (current location, day of the week, time and speed), which were collected with a GPS device and with a smartphone. This methodology was based on spatial clustering of data and on time segmentation to find points of interest that the user visits every day and every hour. With these points of interest, data considered noise were used to improve the prediction. The results of this classification/prediction, which was based on two user profiles in the data set, demonstrated the relevance and simplicity of the methodology.

A model based on spatio-temporal context of user visiting history was proposed by (Gao et al., 2012) for location prediction, which models a user's visiting behavior in the context of spatial historical trajectories and the temporal periodic patterns. By applying smoothing techniques to the model training, this model obtained significant improvement compared to many state-of-the-art approaches. To predict the next location of a mobile user in support of various location-based services a study was carried out by (Ying et al., 2011) who explored the semantic trajectories of mobile users. The core of their framework was a prediction strategy which evaluated the score of next stay location for a given mobile user. In the SemanPredict framework, a cluster-based prediction technique to predict the next location of a mobile user was proposed. Through a series of experiments, they validated this proposal and show that the proposed location prediction framework has excellent performance under various conditions.

In a research by (Ashbrook and Starner, 2003b) where they collected four months of data from a single user and then developed algorithms to extract places and locations from that data. These locations were then used to form a predictive model of the user's movements. These preliminary results suggested that their method may be able to find locations that are semantically meaningful to the user. The algorithm seemed to give consistent results across subjects. Bayes based Predictors were used in order to add to the performance of their prediction for leveraging big data (Matekenya et al., 2016). They studied a large CDR Dataset. First the authors explored the dataset and find that they can use call activity to generate prior probabilities for use in a Bayes predictor. With this reasoning, the authors developed an enhanced Bayes predictor which uses a distance threshold and the users' regular location to improve generation of prior probabilities. Experimental results show that the enhancements they proposed increase accuracy of the Bayes based predictor by 17 percentage points. In the end the authors concluded that it is feasible to leverage big cellular data to enhance location predictors without relying on external data. A visit history-based activity prediction system was proposed to enable activity-aware mobile services in smart cities (Kim et al., 2015).

For this purpose, the system adopted a causality based activity prediction model using Bayesian networks. It used location-based activity causality. The causalities are inherent in common daily activities that usually disclose an activity sequence pattern. The model effectively captures such activity causalities as well as activity sequence patterns from so-called influential contexts such as visit place, visit time, duration, and transportation mode. Authors develop a hierarchical Bayesian Networks to efficiently tackle the complexity of learning

activity causalities between many diverse influential contexts. The authors evaluated the prediction model using the American Time-Use Survey (ATUS) dataset which included more than 10,000 people's location and activity history. The evaluation results showed that it can predict users' potential activities with up to 90% of accuracy for top 3 activities, more than 80% for top 2 activities, and about 65% for top 1 activity while considering a relatively large number of daily activities defined in the ATUS dataset.

A hybrid system for the location recognition and prediction which addressed key issues of location-based services, such as location recognition and prediction proposed by (Cho, 2016). The system used a hybrid method combining k-Nearest neighbor (kNN) and decision tree to effectively recognize the locations not only in outdoor environment but also in indoor environment.

For the location prediction part of the proposed system, Hidden Markov Model (HMM) is used to identify user's next location using the location sequence together with other context information. To improve the performance, the prediction part of the proposed system automatically extracts the meaningful intermediate locations by exploiting G-means clustering algorithm. To further reduce the complexity of constructing probabilistic model and shorten the running time, the proposed system performs G-means algorithm only with the stay points. Experiments were performed to evaluate the accuracy of prediction model on mobile devices. Authors achieved the average prediction accuracy higher than 90% through the experiments. A research by (Simmons et al., 2006) presented an approach to predict driver intent using Hidden Markov Models. The results demonstrate an accuracy of over 98% accuracy of prediction in most cases. Much

of this accuracy is due to the fact that there tend to be very few places where choices have to be made while most of the predictions are forced. For places where choices are available, the use of additional factors (time-of-day, speed, day-of-week) often improve prediction accuracy immeasurably. Similarly, (Gambs et al., 2012) presented an algorithm for next place prediction based on a mobility model of an individual called a multi-order Markov Chain (n-MMC) that keeps track of the n previous locations visited. Experiments on three different datasets showed that the accuracy of this prediction algorithm ranges from 70% to 95%.

An approach was presented by (Jeung et al., 2008) which forecasted an object's future locations in a hybrid manner utilizing not only motion function but also objects' movement patterns. The trajectory patterns of objects were defined and discovered to evaluate spatio-temporal predictive queries. The trajectory pattern tree indexing the patterns discovered for efficient query processing. The authors also proposed the hybrid prediction algorithm which could provide accurate results for both distant time queries and non-distant time queries. The findings of these comprehensive experiments demonstrated that their techniques were more accurate and efficient than the existing predictive methods. A probabilistic approach was considered by (Krumm and Horvitz, 2006). Authors introduced an open-world model of destinations that helped the algorithm work well in spite of a lack of training data at the beginning of the training period.

The applications of pre-destination include proactively delivering information about upcoming points of interest and traffic problems. This reduced cognitive load on the driver by eliminating information about places that he or she is unlikely to encounter. Based on GPS data from 100 drivers (Krumm, 2008)

showed that a Markov model is a simple, effective way to predict near-term, future road segments. Looking at the most recent 10 segments into the past, they predicted the next segment with about 90% accuracy. This study could be used to be used to warn drivers of upcoming traffic disruptions, provide anticipatory information, and trigger automatic vehicle behaviors. A technique was developed by (Karbassi and Barth, 2003), which is an estimation that handles the prediction problem, consisting of first predicting vehicle routes (for station-to-station trips), followed by estimating vehicle arrival times. The authors achieved fairly good results, given the temporal sparseness of the location data (every 30 seconds). Authors found that much better prediction can be achieved with Kalman filtering techniques; however the location data on the vehicles must be acquired at higher frequencies. Further, what was carried out here was short-term prediction (5 - 20 minutes in advance) based on trips in progress. The prediction techniques developed here were limited to direct station-testation trips.

Confidence estimation was introduced by (Petzold et al., 2004) into the State Predictor Method. Their motivation was that in many cases it is better to make no prediction instead of a wrong prediction to avoid frustrations by too many miss predictions. Authors proposed three methods for confidence estimation. For all three methods the prediction accuracy increases with confidence estimation. The best gain of 1.95 was reached for the Confidence Counter method. In that case the prediction accuracy without confidence estimation was 44.5%, and the prediction accuracy with confidence estimation reached 86.6%. But the quantity decreases with the estimation which meant the system delivers prediction results less often. The best result was reached again with the Confidence Counter method. A prediction accuracy of about 90% was achieved. However in that case the quantity

was about 33%. A mining algorithm on the mobility patterns of users, forming mobility rules from these patterns, and predicting mobile user next movements by using the mobility rules was proposed by (Yavaş et al., 2005). The authors evaluated the performance of this algorithm using simulation and compared the obtained results with the performance of two other prediction methods, Mobility Prediction based on Transition Matrix (TM) and Ignorant Prediction. When compared to the performance of the baseline method, which is Ignorant Prediction, this method provides a very good performance in terms of precision and recall.

Travel-speed prediction was examined by (Xu and Wolfson, 2003) for query processing. They proposed three methods of applying travel-speed prediction, namely Speed Update Triggered Revision (SUTR), Query Triggered Revision (QTR), and QTR with Query Relaxation QTR+QR, and analyzed them theoretically and experimentally. For the short term future (15 minutes or less), query processing with travel-speed prediction provides more accurate answers than without travel-speed prediction. QTR+QR is equivalent to QTR (i.e. it provides the same answer set), but with a much lower number of trajectory revisions and QTR+QR and SUTR are suitable for different situations depending on the query rate. A new data mining model was introduced by (Morzy, 2007) for aiming at the efficient prediction of unknown location of moving objects based on movement patterns discovered from raw data.. The main agenda of the paper proved by the experiments conducted is that data mining techniques can be successfully employed for real-time location prediction in mobile environments. While most expensive and burdensome computations (e.g. the discovery of frequent trajectories) can be performed offline and periodically, the online

matching of partial trajectories with the database of movement rules is executed very fast but still a more efficient matching strategies for even better accuracy could be developed. Similarly (Morzy, 2006) presented a new model of movement rules discovered from moving object data. Movement rules provide a simplification and generalization of a large set of moving objects and allow for predicting the location of a moving object.

A technique was introduced to predict next location of a moving object by (Monreale et al., 2009). Their definition of a future location prediction for a moving object was based on the previous movements of all moving objects in a certain area without considering any information about the user. The previously proposed methods used temporal information only to order events; they used T-patterns which are intrinsically equipped with temporal information. Furthermore, they defined an evaluation function for a T-pattern set in order to choose the best one for the construction of a good T-pattern Tree. Shortcomings of linear prediction with an architecture was overcame by (Tao et al., 2004) that supported arbitrary motion patterns, not necessarily known in advance. They proposed the concept of recursive motion functions and proved both theoretically and empirically, that it can accurately express a large number of movements. Finally, they develop a spatio-temporal access method that generalizes the current state-of-the-art indexes.

NextPlace was presented by (Scellato et al., 2011), an approach for spatio-temporal user location prediction based on nonlinear analysis of the time series of start times and duration times of visits to significant locations. This approach not only allows forecasting the next location of a user, but also his/her arrival and

residence time, i.e., the interval of time spent in that location. Moreover, existing models were not able to extend predictions further in the future, since the main focus is on the next movement of a user. The authors evaluated *NextPlace* comparing it with a version based on a linear predictor and a probabilistic technique based on spatio-temporal Markov predictors over four different datasets. Authors reported an overall prediction precision up to 90% and a performance increment of at least 50% over the state of the art methods. The authors showed how the adoption of a nonlinear prediction framework can improve prediction precision with respect to other techniques even for long-term predictions.

A spatial temporal prediction method was proposed by (Liu et al., 2016) which is called as Spatial Temporal Recurrent Neural Networks (ST-RNN). The experimental results on real datasets showed that ST-RNN outperformed the state-of-the-art methods and can well model the spatial and temporal contexts. A classification was presented by (Tran et al., 2012) for approaching decision trees to predict the next place of mobile users. The authors implemented an optimizer to find the best parameter combination for each user, since users had widely varying behavior. Finally, the performance of the approach was demonstrated by the results of the experiments on the real life dataset of 80 mobile users provided by Nokia.

Based on the literature review above, we found out that there are not many studies done regarding optimization of the algorithm as the data for these algorithms are usually very large. Therefore, we are going to develop a trajectory

prediction algorithm that can predict a vehicle future location effectively and efficiently with lowest latency rate.

1.2. Rationale

Huge amount of data is generated by daily used automated machines such as mobile phones, vehicle trackers and other electronics which goes to waste. There are some research work done to use these data for commercial and security purposes by assessing user's movement pattern and their social behaviors. These big data required big data platform to analyze the data quickly and efficiently. There are many algorithms to predict user's future algorithm but the time taken and latency for the job done and queries is very high. We needed to develop a system that can communicate with the big data in a more efficient manner. For that we need to use these predictive algorithms for big data platforms. In this research we are using such a platform to analyze and predict movement patterns and future locations from GPS data.

1.3. Objectives

To identify the top three trajectory prediction algorithms based on efficiency, performance and ease of use.

To develop an algorithm for performing spatial-temporal analysis on raw GPS data collected through trackers installed in vehicles in order to discover functional knowledge by prediction future locations suited for real time applications/ decision making.

To run algorithms on big data platform for reducing processing time and making computing effective and efficient.

1.4. Study Area

We are using GPS tracker data from a Trakker company stationed in Pakistan whose vehicles are distributed all over the country. Mainly data records are from the city of Lahore, Karachi and Islamabad.

1.5. Scope of the Study

The work displayed in this research shows a proof of idea where GPS information examination can yield results that assistance numerous areas of making decisions. It shows the utilization of GPS data produced by two thousand two hundred and sixty-one users. This information gathered can be taken into account to classify user behavior and describe interests of users on different levels. This knowledge can be used to personalize road safety, anticipation and security related issues to improve the user experience. It elaborates procedures adopted for processing the raw GPS data from collection till analysis. Some records can be created to generalize the process for use with any kind of mobility data. The GPS mobility data collected is constrained free and movement data is based on the behaviors of the user.

Chapter 2

Materials and Methods

2.1 Data Sources, Quality and Limitation

For this study we studied the literature and the research arenas available online for recent technologies that are in use. Upon reviewing the literature and some local research works carried out in our academic institute the main big data technology that were in use were Hadoop(Hadoop, 2018), Secondo (Güting et al., 2010a), Parallel Secondo (Lu and Güting, 2012) and Hermes (Hermes, 2009). All of these deals with spatio-temporal databases. However, these software technologies had some limitations with respect to the time taken per process. Or main objective in this study is to find and use those technologies that can achieve the lowest latency rate for spatial objects and big data. We search the recent trends used by the researchers in research arenas on the internet. A lot of new technologies came up like Apache Hive, Apache Flink, and Apache Spark etc. that were relatively compatible then the ones in the literature review. In order to choose the best suited technology, we observed our skills and put forward the objectives and selected Apache Spark. Apache Spark is being used widely in recent times for Big Data and machine learning and is said to be the faster compare to other technologies.

There were some limitations in the data, the size of the data was too huge to be processed on common analytical or GIS software's. Not even a single table of the database can be accessed or opened in excel. In order to get the most important data for our analysis on the trajectories we decreased the number of columns from thirty three to five columns. The data and time column was in datetime format.

We needed it change and make separate columns for date and time and make a new column in the data for the Day of the week e.g. Monday. We needed to develop the data in such format so that we can easy run our queries and so that we do not have to do it on run time to avoid delay in the processes. We exported the data from Microsoft SQL Server Management Studio to open source CSV format. It helped us to reduce the 30 GBs database to a 10 GBs CSV file. In order to export such huge data the processing power and memory of the system in use were increased.

2.2 Materials – Choice of Technologies and Software's

Apache Spark developed in year 2009 at the Berley's lab ("Apache Spark History," 2018) said to have achieved the lowest latency rate in comparison with Secondo and Parallel Secondo. It is freely available for number of operating systems such as Windows, Linux, and Mac OS. Apache Spark is a Unified Analytics Engine for Big Data processing and management, that supports streaming data, batched data, SQL, Graph, and also machine learning processes. In general Apache Spark software is used for clustering of systems for very fast query response. It provides executable environment for all the Spark applications in the Kernel of Spark core. The actual advantage of Apache Spark while comparing with other technologies like Hadoop and MapReduce, which only uses disk for memory while Apache Spark uses both memory and can also make use of the disk for the processes. Apache Spark is versatile unlike Hadoop ecosystem, as it does not have its own distributed file system but can make use of Hadoop Distributed File System (HDFS). Figure 2.1 illustrates a comparison between Hadoop and Apache Spark. Apache Spark is a standalone software which does not uses any resource manager. But if we want to use it for more than one node

environment setup we can use YARN or MESOS for resource management, along with a distributed file system such as HDFS or S3 in that case. We can safely say that Apache Spark is an independent computing framework.

- Apache Spark is designed to be made in use with a lot of programming languages and many architectures. It is Apache's largest ongoing project with over 200 organizations and a double of that contributors making frameworks of the current software release.
- It was import that SPARK should be developed in languages that the developers are already familiar with. The programming languages that Apache Spark supports are:

O Java
O Python
O Scala
O SQL
O R

See Figure 2.2 for a complete architecture of Apache Spark working environment.

- There are many methods to store Spark's data. It is often linked with Hadoop Distributed File System (HDFS). Other open sources projects are as follow:

O MapR
O Amazon S3
O Google Cloud
O Apache Cassandra
O Apache HPFS (Hadoop)
O Apache HBase
O Apache Hive
O Berkeley's Tachyon Project

However, it is recommended to choose the data storage environment that researchers are already using.

2.3 Overview of Apache Spark Components

Spark stack consists of four optimized libraries that are used over Spark Core for managing four different use cases. One application typically requires at least one library with Spark Core. But Spark's versatility and flexibility become most useful when we are requiring two or more than two libraries at the same time over Spark Core. As illustrated bellow in Figure 2.3. Spark Core is also known as the heart of Spark and is responsible for task scheduling and management functionalities. Apache Spark has a build-in library for processing of structured data, the Spark SQL. Which can be used for complicated SQL database queries and algorithm based analytics. Spark SQL supports HIVE, SQL-like HiveQL query, JDBC and ODBC. This can also enable some degree of connections with existing databases. Warehouses and business intelligence environments. Spark Streaming is used for integration of sources like Flume (optimized for data logs) and Kafka (optimized for distributed messaging) for data streams and supporting scalability and fault-tolerance. MLib module is used for scalable Machine Learning and statistical algorithms such as correlations, hypothesis, a classification, regression, clustering, and principal component analysis. GraphX supports a large number of graph algorithms such as "page Rank" for instance. GraphX has been recently introduced to Apache Spark libraries for supporting analysis and computation of graphs of data e.g. Pregel API. It was originally developed as a separate UC Berkeley research project (Gonzalez et al., 2014). Spark R is used for providing lightweight mechanism of data sciences and statistical usage of R. it was introduced to Apache Spark 1.4 release version. We are currently working in 2.0 version of Apache Spark

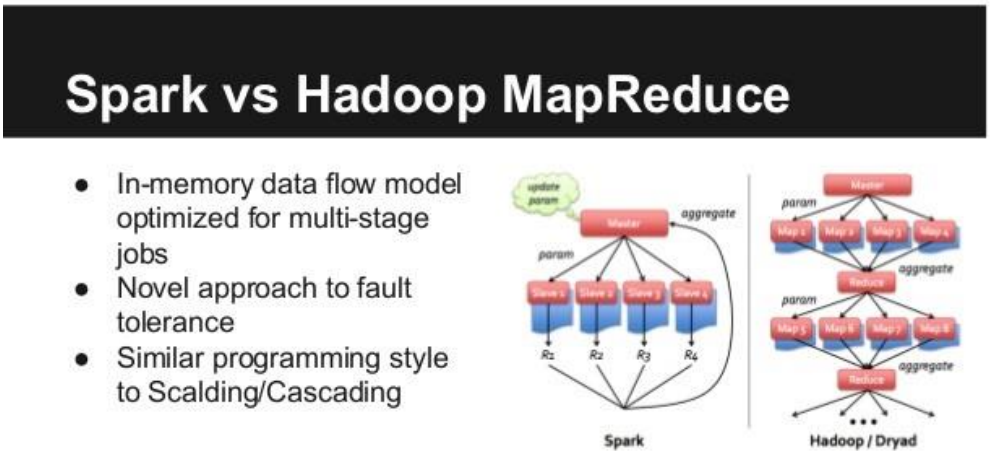

Figure 2.1 Comparison between Hadoop and Apache Spark (Mapreduce, 2019)

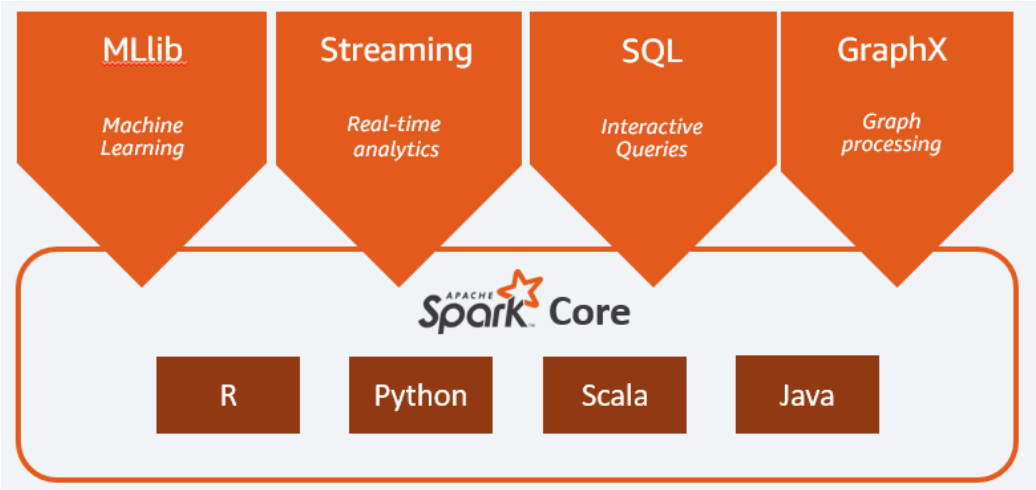

Figure 2.2 Architecture of Apache Spark working environment (Spark, 2019)

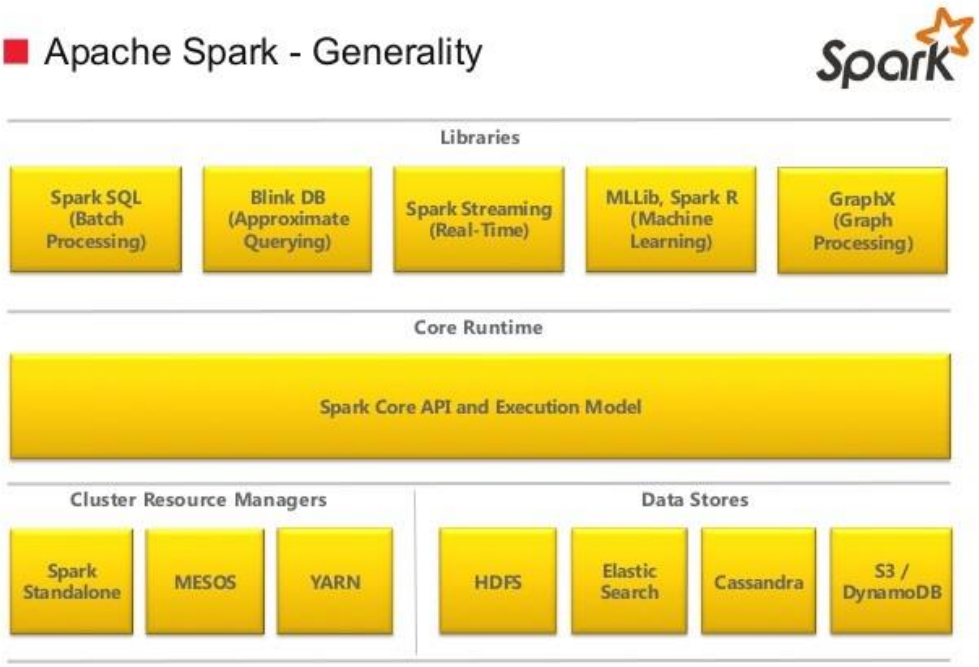

Figure 2.3 Spark Core API and Execution Model (Spark, 2018)

Data Pipelining in Apache Spark

Advantage of Apache Spark is that it combines different techniques and merge them in a single framework. Usually tasks like selection of data, transferring of data in different ways and then analyzing that resultant data will require a series of different frameworks. For instance, Apache Oozis, Hive, Hadoop, SECONDO etc. Apache Spark does this by combining all these tasks together and using both batched and streamed data and achieving more productive workflow in result. Another advantage of Apache Spark is its processing in in-memory and only utilizes disk if the memory limitations has exceeded.

Deployment of Apache Spark

We already developed our algorithm for predicting the future location of vehicles using Python programming language so we selected the PySpark utility for *Windows 7*. We installed *PySpark* by using *pip* installation command. A versatile library used to download and install python packages and software's. We used the following command:

pip install pyspark

PySpark was locally installed in the system. After that we added its path to the environment variable to make its usage available out of the root/installation folder.

2.4 Methodology

In this section we describe the methodology adopted for this research work. The study is divided into different stepwise components in lined with the objectives. The steps are data collection, data pre-processing according to the

requirements of the algorithm we developed. Then creation of spatio-temporal database in Apache Spark, creation of locations from user's data and then the queries we designed for the spatio-temporal data. A flowchart of the research work carried is displayed in Figure 2.4. The subsequent sections are described one by one in detail for each step with relevant figures and tables.

A brief introduction to each section is as follows:

- Raw GPS data collection - we requested real-time data from a vehicle GPS tracker company which resides in Pakistan for GPS sensors data. We received the raw GPS data of GPS receivers installed in a number of vehicles. Detailed discussion is available in Section 2.4.1.
- Data preprocessing – Data preprocessing involved cleansing of data, removal of unwanted data fields and removal of missing fields to avoid null data in columns and adding new columns to the data for reducing run time conversion of data on each query. Other format changes along with necessary modifications that the data required.
- Creation of spatio-temporal database in Spark – When the data was prepared and cleansed we exported the spatio-temporal data which was in csv format for necessary actions to Spark. We created a new database and stored the spatio-temporal data into SparkSQL table. We used PySpark syntax for storing the data in Spark database.
- Design of Spatio-temporal Queries – Investigation of the data requires many queries that inquires the data for highlighting spatio-temporal analysis. The results of these queries were stored as separate records for future location prediction and to carryout analysis on them. Complex algorithm design and development are presented here.

- Creation of locations from Users data – The results from the algorithm were visualized on web using special geospatial visualization libraries of python. We extracted the result for web maps on runtime to avoid any delay in the result generation and later result visualization.

Raw GPS Data Collection

To develop an effective big data algorithm for future trajectory prediction we need real time data that has been collected over extensive amount of time. For that purpose we requested real time GPS data from a GPS provider company that keeps track of all its GPS devices in a database server. The GPS tracking company provided us vehicles GPS data that started from *1st January 2016* and ended on *31st December 2016*. A total of *2261* vehicles contributed to the data. The data was received in a Microsoft SQL Server Management Studio (MSSQL). Each month has its own table. The number of rows per table were roughly between 7 million to 10 million. Each table had thirty three columns which consisted of location Names, vehicle status, vehicle latitude/longitude, GPS time, drivers ID and many more other columns. The data acquired was in a backup MSSQL file of about 3 GBs. After restoring the backup in MSSQL. The size of the data was increased to 30 GBs. The preview of the received data in MSSQL is shown in Table 2.1.

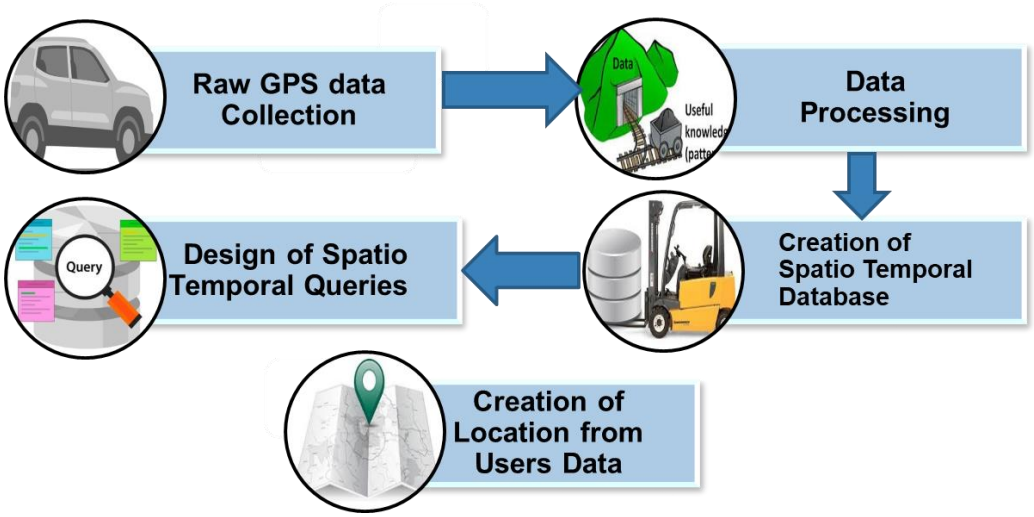

Figure 2.4 Methodology Process Flow Diagram

Table 2.1 Column names of the data received in MSSQL

Id	ReportGroupDate	Ordered
Cellnumber	vehicleReg	Assembletime
DriverId	Vehiclestatus	MobileSpeed
Mobileodo	Location	Skillset
C2	C3	C4
C5	Gpstime	locationName
locationdistance	Locationdirection	Latitude
Longitude	C6	C7
C8	Distance	Direction
Zipcode	Country	Areagroup
Locationtolerance	Province	C9

Data Preprocessing

Data preprocessing stage is an important step in carrying out a research study. In this step raw GPS data was converted to a usable format and unwanted rows and columns were removed from the database. The date obtained from the vehicle GPS tracker company contained a lot of columns. Many of them we not useful for the program we were going to design and the study we were going to carry out. Those attributes of the data were removed and only required attributes of the data were retained.

- In order to extract only the five required columns that were: datetime, vehicle registration number, location name, latitude, and longitude. We extracted these columns to comma separated value (csv) file, where each csv has as many tables as we asked to be extracted. For extracting only one table from the MSSQL database to csv we used the following commands in windows command line:

bcp "SELECT ReportGroupDate, vehicleReg, locationName, latitude, longitude FROM dbo.Analytics_Set_4 queryout E:\tables_from_sql\table1.csv -t, -c -S . -d bacha –T

- The output of this command generated a csv file in the desired directory for the 1st month, as each month has a separate database table. Upon displaying the initial rows of the data using another command, the data did not had any column names. The command we used was:

more /e E:\table1.csv p 10

This command displayed the 1st ten rows of the data which did not had the data column names as in the database of MSSQL. Before assigning column headings to the data we extracted the number of tables

we needed from the database. We extracted all twelve tables from the database MSSQL in order to use it for our prediction algorithm.

- This command merged all twelve tables of each month of the year to a single csv file. This function was performed by the command:

bcp "SELECT ReportGroupDate, vehicleReg, locationName, latitude, longitude FROM dbo.Analytics_Set_1
union SELECT ReportGroupDate, vehicleReg, locationName, latitude, longitude FROM dbo.Analytics_Set_2
union SELECT ReportGroupDate, vehicleReg, locationName, latitude, longitude FROM dbo.Analytics_Set_3
union SELECT ReportGroupDate, vehicleReg, locationName, latitude, longitude FROM dbo.Analytics_Set_4
union SELECT ReportGroupDate, vehicleReg, locationName, latitude, longitude FROM dbo.Analytics_Set_5
union SELECT ReportGroupDate, vehicleReg, locationName, latitude, longitude FROM dbo.Analytics_Set_6
union SELECT ReportGroupDate, vehicleReg, locationName, latitude, longitude FROM dbo.Analytics_Set_7
union SELECT ReportGroupDate, vehicleReg, locationName, latitude, longitude FROM dbo.Analytics_Set_8
union SELECT ReportGroupDate, vehicleReg, locationName, latitude, longitude FROM dbo.Analytics_Set_9
union SELECT ReportGroupDate, vehicleReg, locationName, latitude, longitude FROM dbo.Analytics_Set_10
union SELECT ReportGroupDate, vehicleReg, locationName, latitude, longitude FROM dbo.Analytics_Set_11
union SELECT ReportGroupDate, vehicleReg, locationName, latitude, longitude FROM dbo.Analytics_Set_12"
queryout E:\tables_from_sql\table_12.csv -t, -c -S . -d bacha –T

The result of this command was stored in table_12.csv. Which had more than 100 million rows of data for all the month of the year.

- To assign column header to the csv files extracted, we developed another csv file which only had the headers, as we could not open the extracted csv file because of their enormous size. We used another command to add the headers csv file to table_12 csv file. The command we used was:

copy /A E:\tables_from_sql\headers.csv + E:\tables_from_sql\table_12.csv E:\tables_from_sql\ headers_table_12.csv

The output of this command was stored in another csv file which had now both the headers and the data of all the twelve months extracted from MSSQL database.

This csv file was then used for the main future location prediction algorithm. Other files for three months and six months were also created to check the accuracy of the developed algorithm. The working of the algorithm will be discussed in detail in the coming sections of methodology.

- In the next step of data preprocessing we needed to change the single column of data and time to separate columns for each date and time in order to minimize the time taken in run time conversion with respect to the queries if the algorithm. We created new columns for each date and time and removed the previous datetime format column from the data. We also added another column of "Day" which was generated from the date. We used the following pandas code a library of python in order to achieve the desired format.

```
nfp = read_csv("tables_headers_12.csv", low_memory=False,
parse_dates=False, na_values=[ '', '\x1a"], infer_datetime_format=True,
error_bad_lines=False)
nfp.dropna()
#assign the datetime column to temp
nfp['ReportGroupDate'] = pd.to_datetime(nfp['ReportGroupDate'],
errors='coerce')
nfp.dropna()
temp = pd.DatetimeIndex(nfp['ReportGroupDate'])
nfp['Date'] = temp.date
nfp['Day'] = nfp['ReportGroupDate'].dt.weekday_name
nfp['Time'] = temp.time
del nfp['ReportGroupDate']
nfp.to_csv('tables_headers_12_no-errors.csv',index=False)
with open('tables_headers_12_no-errors.csv', 'r') as infile,
open('tables_headers_12_daytime.csv', 'a') as outfile:
fieldnames=['Date','Day','Time','vehicleReg','locationName','latitude','longitude'
]
writer = csv.DictWriter(outfile, fieldnames=fieldnames)
# reorder the header first
writer.writeheader()
for row in csv.DictReader(infile):
```

 # writes the reordered rows to the new file
writer.writerow(row)

- Finally the last output is named table_header_12_daytime.csv. The initial size of the file in database was 30GBs before the preprocessing. That size was reduced to a 10 GB csv file. There are a total of 100 million records to be used in Spark database as explained earlier. Data within the csv is arranged in comma separated format in the following sequence:

O Date
O Time
O Day
O VehicleReg
O LocationName
O Latitude
O Longitude

Creation of spatial-temporal Database in Apache Spark

The Apache Spark utility that we used was command line, without any Graphical User Interface (GUI) in python called as PySpark. Before creation of the database in PySpark, we used spark utility that make use of as many cores of the system as we want to use in a standalone environment. Initially we used only one core, but the processing power was not as efficient. That why we used all the available cores on our system by using the '*' sign for using all cores of the system and created spark context.

We used Spark Context to define the cores that are to be used by the local system. The python command we used for this purpose was:

sc = SparkContext ("local[*]","Daud")
*spark = SparkSession.builder *
*.master("local") *
*.appName ("Data cleaning") *
.getOrCreate ()

The reason because Spark Context is the entry point to the Spark Core. A Spark Context is assumed as an application of Spark. In this case our application name is “data cleansing” as mention in the lines of code. Once a Spark Context is created we can use it for making Dataframes, creating RDDs, broadcast variables, access spark services until the Spark Context is stopped. A Spark Context govern all the steps inside the Spark application.

Figure 2.5 shows the functionality of a Spark Context with graphical representations. A Spark Context offers the following functionalities:

- Spark Env
- Spark Conf
- Deployment environment (a master URL)
- Application name
- Unique identifier of execution
- Deploy mode
- Default level of parallelism without specifying the number explicitly by a user
- Spark user
- Time when Spark Context was created in milliseconds
- Spark Version
- Storage status

We created a SQL Context and assigned the already created Spark Context to it using the following command

SqlContext = SQLContext (sc)

In order to use the csv file in our study, Apache Spark provides us three different utilities to accomplish this step. We used one of them looking at our requirements.

The three APIs are:

- RDD (Resilient Distributed Dataset)
- Dataframes
- Datasets

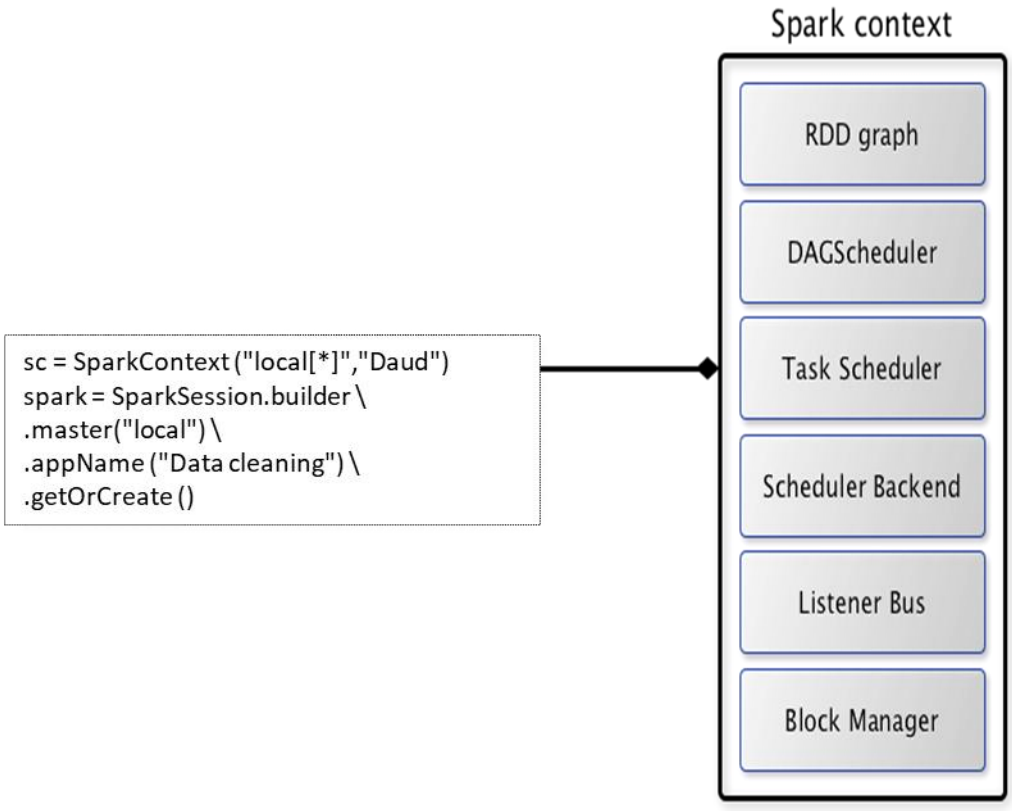

Figure 2.5 Spark Context Services

Resilient Distributed Dataset (RDD)

From the inception of Spark, RDD are used for interaction with the core level of Spark. As RDD are involved at the core level of Spark applications. RDD are immutable elements of the data that are partitioned across nodes in the Spark cluster which can be operated in parallel along a low-level API. RDD is used when we are working with an unstructured data or stream of data or when imposing the schema, such as columnar format is not important and calling data by using the columns name is not necessary. In our study we are going to use data attributes with their column names. So we cannot use RDDs explicitly for this study.

Data Frames

DataFrames are also an immutable API of Spark distributed data collection. Advantage of a DataFrame over RDD is that in DataFrame the data is organized in tabular format and named columns just like tables in any relational database. DataFrames are used on higher-level abstraction which gives the edge to programmers to interact with Spark core in different languages other than the specific languages that API are actually programmed.

Dataset

A Dataset in Spark 2.0 version has characteristics of two API at once. One is strongly-typed API and other an un-typed API. For Scala and Java we have strongly-typed JVM objects while for Python and R we only have un-typed APIs as they do not have any compile-time type-safety. But that is alright as we are using an un-typed table for our research work. Table 2.2 and Figure 2.6 illustrates an overview of the APIs.

Table 2.2 Typed and Un-typed APIs

Language	**Main Abstraction**
Python	DataFrame
Java	Dataset
Scala	Dataset & DataFrame (Dataset[Row])
R	DataFrame

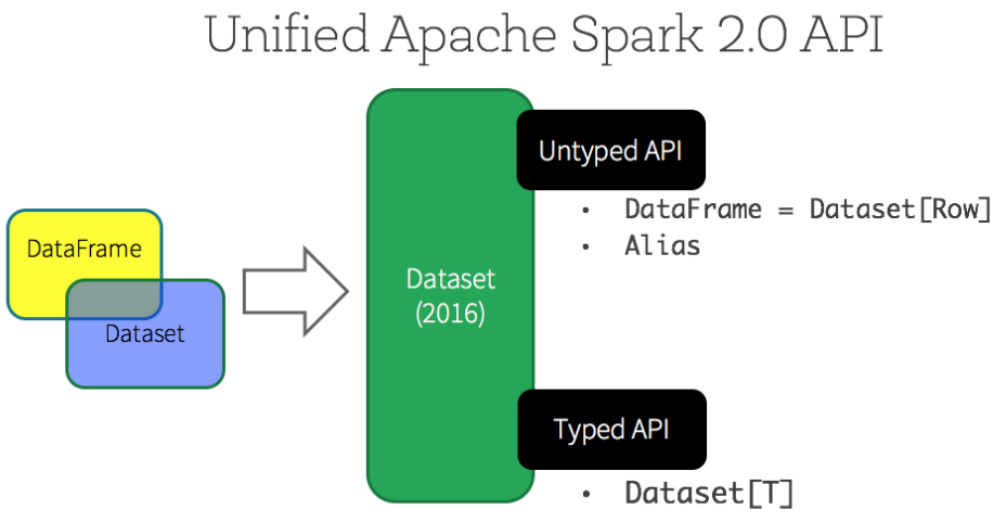

Figure 2.6 Apache Spark APIs (APIs, 2019)

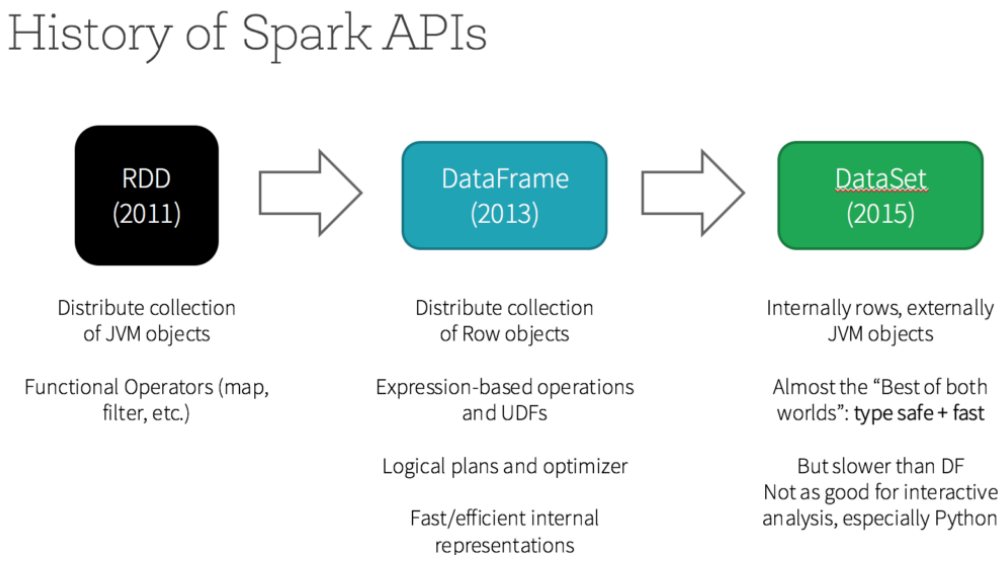

Figure 2.7 History of Spark APIs (DataFrame, 2019)

We used datasets in Spark 2.0 which are called DataFrames. As in Spark 2.0 the DataFrame APIs are merged with Dataset API for unifying data processing capabilities. Figure 2.7 shows the history of Spark APIs over the years.

We used DataFrames in Apache Spark version 2.0 API for managing our data. We assigned the final csv file generated for all the twelve months of the year and has been preprocessed to a DataFrame using the following lines of code:

```
dfz = spark.read.format ("csv").option ("header","true").option
("mode","DROPMALFORMED").load
("C:\Users\DaudKamal\Desktop\dauddata/tables_headers_12_daytime.csv")
```

Here, dfz is the DataFrame we are using and Spark's API to read the CSV file after loading it. We created the spatio-temporal database by using the ***createOrReplaceTempView*** library of Apache Spark. We stored the DataFrame 'dfz' in our SparkSQL table and used this table for our algorithm.

```
dfz.createOrReplaceTempView ("table1z")
```

The command ***createOrReplaceTempView*** is used in Apache version 2.0 or above, the previous versions used 'registerTempTable' for this purpose.

Design of Spatial-temporal Queries

As we explained in our objectives section that we are going to predict the future location of a user vehicle using Apache Spark. The design of the query included the spatio-temporal aspect of the data i.e. where and when. We queried for where a user vehicle will be at a given time and on a given day. For examples what is the location of a particular user for instance say 'A' on 'Monday' between '9Am' to '9:30Am'. We asked the user to input a valid user vehicle number, the day they want to inquire, and the time between which they want to predict the vehicle location. After the user has input these four parameters, we store them in

variables and use them in a spatio-temporal query to get the necessary results for the algorithm. The SparkSQL query is as follows:

*dfz = spark.sql("SELECT * from table1z where vehicleReg = '%s' AND Day = '%s' AND (Time = '%s' or Time = '%s')" % (vehicleRegz ,Dayz, Timez, Timez2))*

Now, the DataFrame *dfz* has an updated result stored from the query. It has the rows for the vehicle, day, time 1 and time 2 that the user queried. The next is the algorithm that we designed to get the top three probable locations for the user query. Few of the variables used are as follows:

alltableList ← *all locations from the query*
mylistz ← *all unique locations from the query*
bc ← *count of all the locations*
first probability ← *- sys.maxsize*
second probability ← *- sys.maxsize*
third probability ← *- sys.maxsize*

The following is the pseudo code of the algorithm we developed:

for** i **to** unique locations **do

total count: = 0;

for** j **to** all locations **do

if** j==i **then

total count : = add 1 to total count;

current probability: = total count divide by count of all the locations and multiply by hundred

end

if** current probability is greater than first probability **than

third probability : = second probability;

second probability : = first probability;

first probability : = current probability;

first location : = i;

elif (**current probability is greater than second probability **than

third probability : = second probability;

```
        second probability : = current probability;
        second location : = i;
        elif  current probability is greater than third probability than
        third probability : = current probability;
        third location : = I;
        if third probability != -sys.maxsize than
        third location : = I;
        end
        end
    end
end
```

At the end of this program we get the location name of the top three locations and probability of their occurrences.

Creation of Locations from Users' Data

When we get the top three locations names from the program in the three variables for first, second and third locations. We run another code to get the location's latitude and longitudes from the table. The following pseudo code explains the steps involved.

```
if third probability != -sys.maxsize than
        dataframe = Select location name, Latitude , Longitude from table
where locationName is equal to third location;
        third latitude = dataframe.first().Latitude;
        third longitude = dataframe.first().Longitude;
        print("Probability for Third location is '%s' is %s and Latitude: %s
Longitude: %s  "  % (third location, third probability , third latitude, third
longitude))
end
if second probability != -sys.maxsize than
         dataframe = Select location name, Latitude , Longitude from table
where locationName = second location;
second latitude = dataframe.first().Latitude;
second longitude = dataframe.first().Longitude;
        print("Probability for second location is '%s' is %s and Latitude: %s
Longitude: %s  "  % (second location, second probability , second latitude,
second longitude))
end
```

```
if first probability != -sys.maxsize than
        dataframe = Select location name, Latitude , Longitude from table2z
where locationName = first location;
first latitude = dataframe.first().Latitude;
first longitude = dataframe.first().Longitude
        print("Probability for first location is '%s' is %s and Latitude: %s
Longitude: %s  "  % (first location, first probability , first latitude, first
longitude))
else:
        print ("No Records for the above query!")
end
```

We have now all the latitude and longitudes for the top three locations and their names and probabilities. We used folium library of python to generate a web map and marked these three locations for each query developed. The pseudo code is as follows:

```
m=folium.Map([df['colLat'].mean(),df['colLong'].mean()],tiles="openstreetmap
", zoom_start=10);
for i in range(0,len(df)) do
        folium.Marker([df.iloc[i]['colLat'],df.iloc[i]['colLong']],popup=df.iloc[i
]['colLoc']).add_to(m)
end
m.save('index.html');
```

Chapter 3

Results and Discussions

This chapter presents the results and discussion obtained by following the methodology outlined in the previous chapters. The implementation of spatio-temporal queries on GPS data are presented. Design of queries, their implementation and the results obtained from the algorithm developed are discussed in detail. The discussion primarily involves the visual interpretation of each query, the numeric results of queries, and the insight into particular user locations where they commute to. These queries are categorized by the time taken on producing the outputs and their analytics are discussed at length in subsequent sections.

3.1 Time Taken by Spatio-Temporal Queries

3.1.1 Spatio-Temporal queries for a Single User

Our main objective was to reduce the time taken for each query while it predicts the future location of a user. We predicted the user's vehicle locations based on the day and the time between which they want to know the locations. To get to know how the time taken changes with respect to the size of the data we started from a single hour window on a single day of the week. We considered 'Monday' for being having the highest activities as it is the start of a week for single vehicle. There are actual registration numbers for the vehicles but we want the vehicles registration numbers to be anonymous in this study so we are calling the users alphabetically with user A, user B, and so on. We are starting the first

window from one hour for user A. The following Figure 3.1 to Figure 3.4 describes the time taken for one query for user A on different time window throughout a day.

3.1.2 Single Query for Multiple Users

Here, we are going to run one query for multiple users to see the difference between the times taken by each user vehicle for the same query. As the amount of data size may not be the same for each user for similar time frame and for same day. Therefore the time taken for each user vehicle future location prediction will be variable. The query we are going to run is for ‘Monday, and the time window we selected is from ’04:30 PM’ to ’05:30 PM’. The query is as follows in Table 3.1. Time taken for each user is displayed in Figure 3.5.

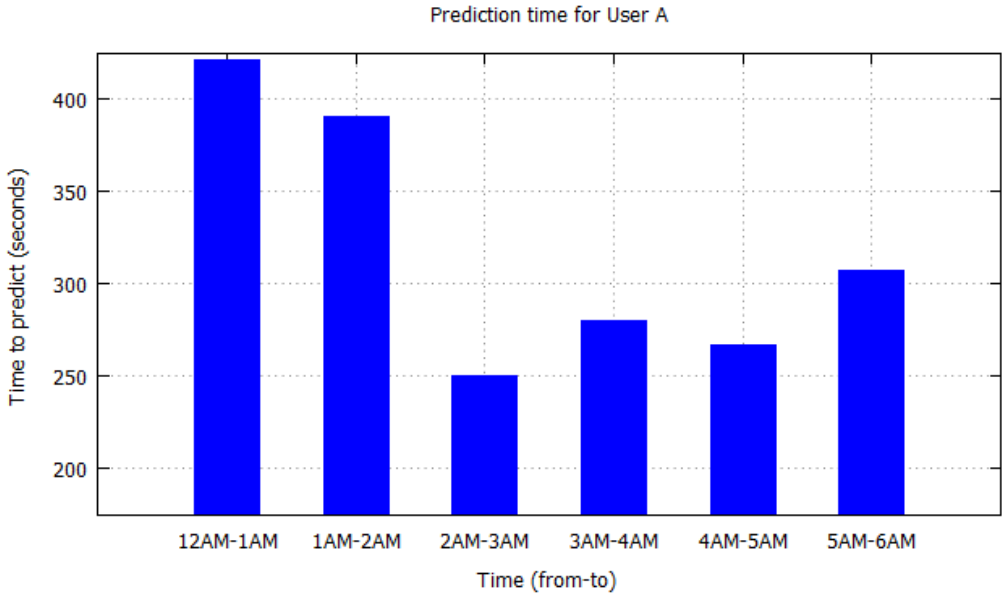

Figure 3.1 Time taken for User A (12AM-06PM)

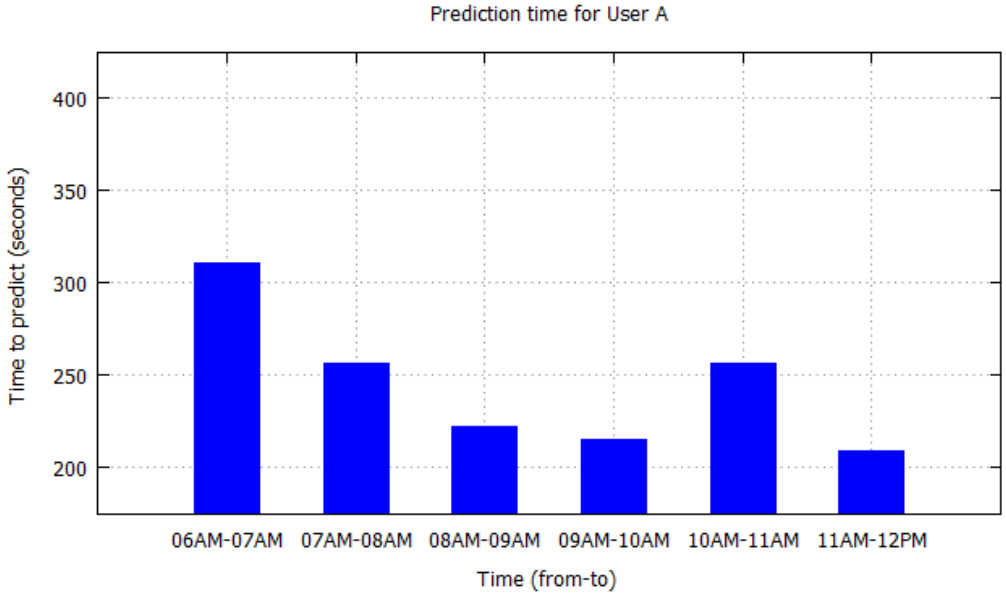

Figure 3.2 Time taken for User A (06AM-12PM)

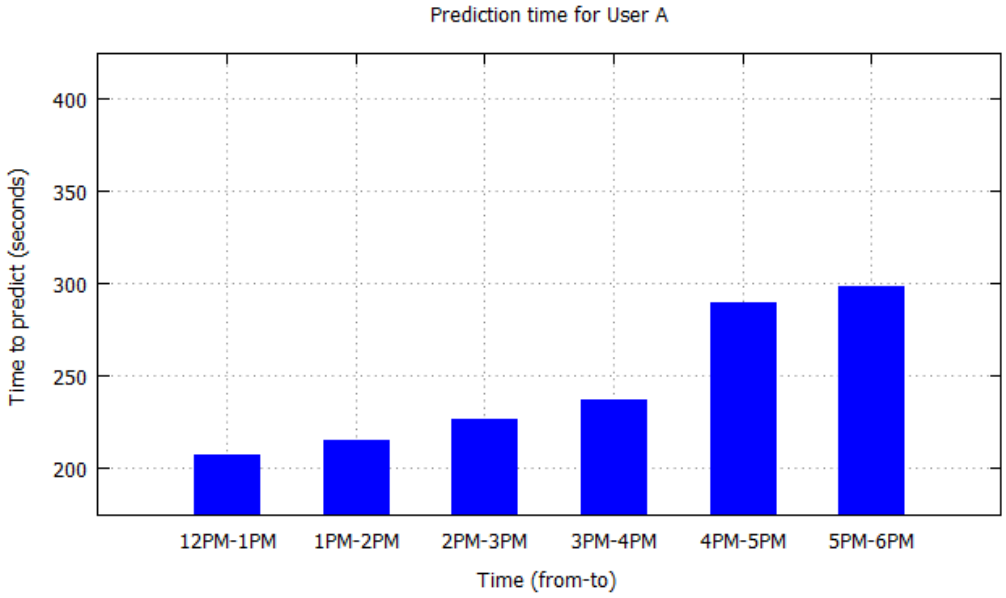

Figure 3.3 Time taken for User A (12PM-06PM)

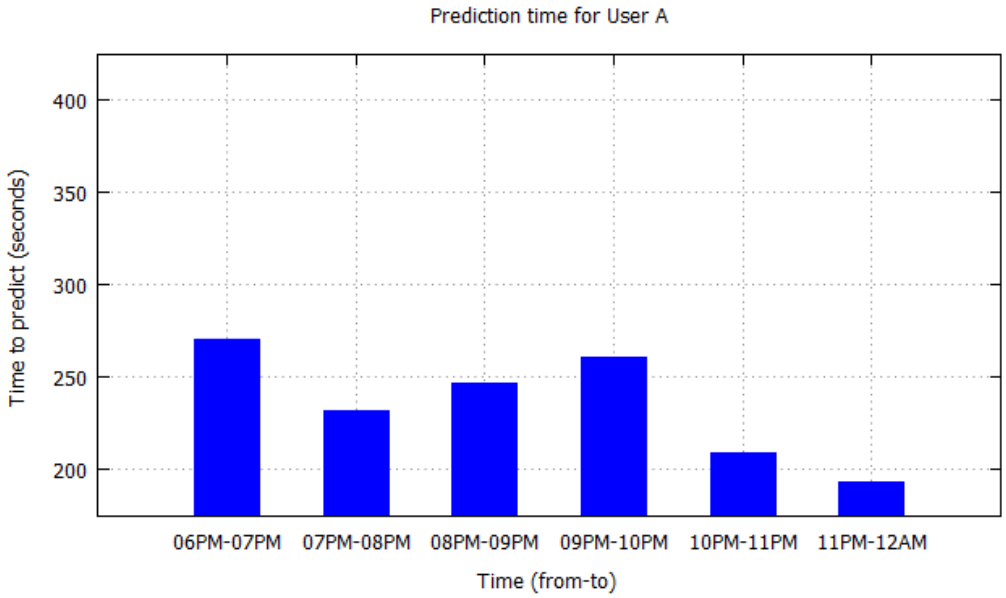

Figure 3.4 Time taken for User A (06PM-12AM)

Table 3.1 Single Query for multiple users

Please enter vehicle number?	*User C*
Please enter Day 1 from Monday to Sunday?	*Monday*
Please enter Time from 00:00:00?	*16:30:00*
Please enter Time to 23:59:59?	*17:30:00*

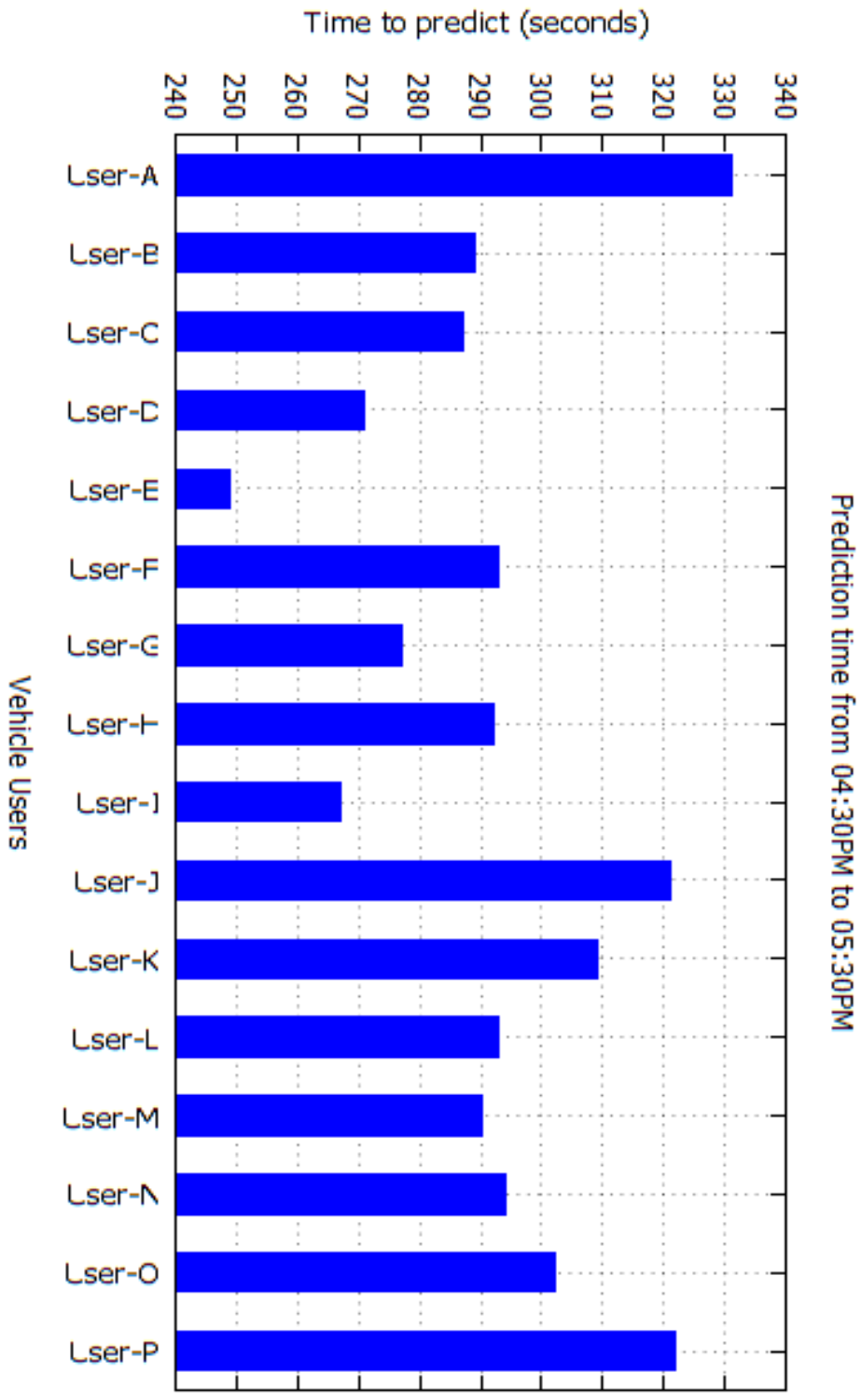

Figure 3.5 Prediction time from 04:30 PM to 05:30 PM

3.1.3 Multiple Queries for a Multiple User

In this section, we are analyzing the time taken by multiple users for multiples temporal queries for a single day. We are considering ‘Monday‘, and we are querying the twenty-four hour window for each user. The number of users for this query are sixteen. Table 3.2 displays the arguments for query. We kept changing the temporal part of the query for each user with one hour increment and saved the results as shown in figures Figure 3.6 to Figure 3.20.

These graphs can be used to develop an understanding of why certain process takes more time than the others which takes less time to generate results. One reason can be because the number of unique records for a query are too much so the processes takes more time than the others. Another reason could be because the numbers of records in the one that took lesser time are too less the than the others. Which means those users had very less activity for the time they were queried.

Table 3.2 Multiple queries for a multiple user

Please enter vehicle number?	*User A*
Please enter Day 1 from Monday to Sunday?	*Monday*
Please enter Time from 00:00:00?	*00:00:00*
Please enter Time to 23:59:59?	*01:00:00*

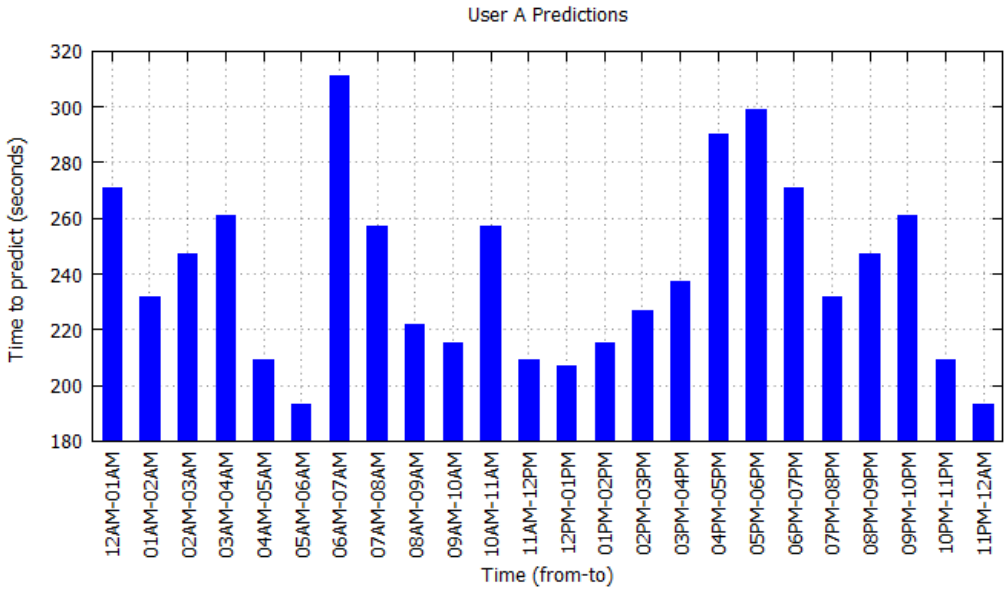

Figure 3.6 User A prediction for 24 hours window

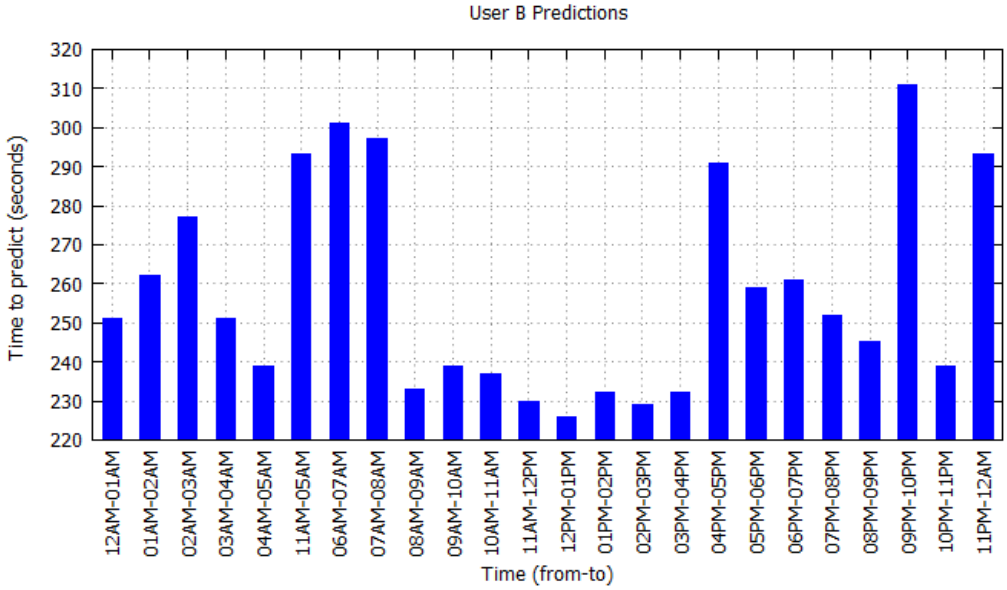

Figure 3.7 User B prediction for 24 hours window

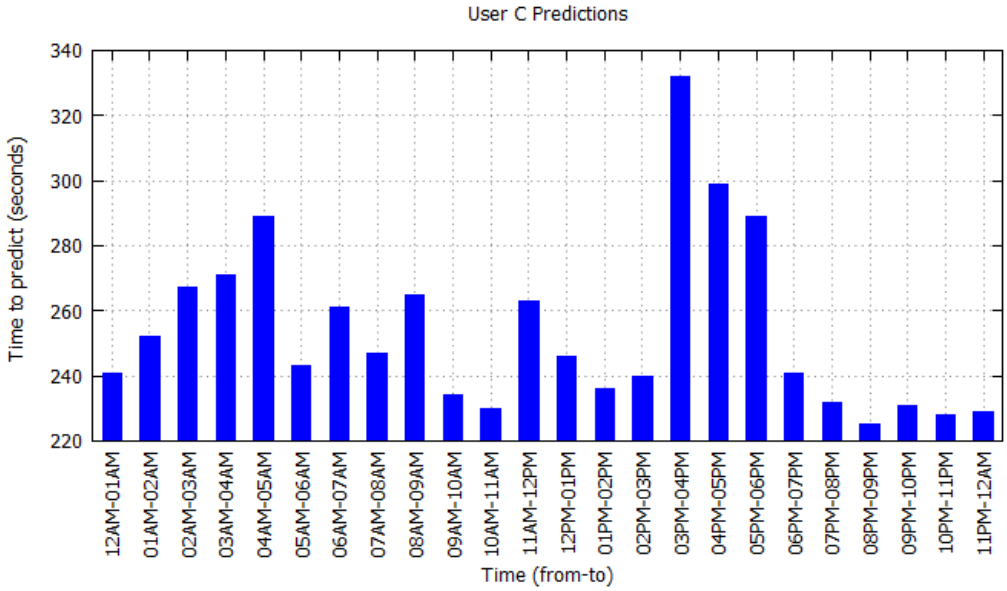

Figure 3.8 User C prediction for 24 hours window

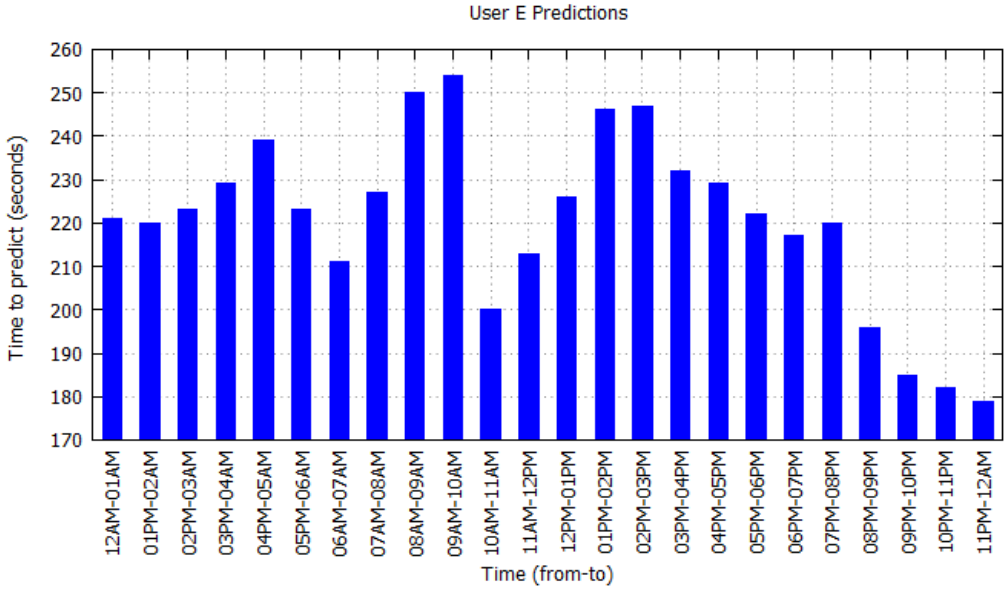

Figure 3.9 User E prediction for 24 hours window

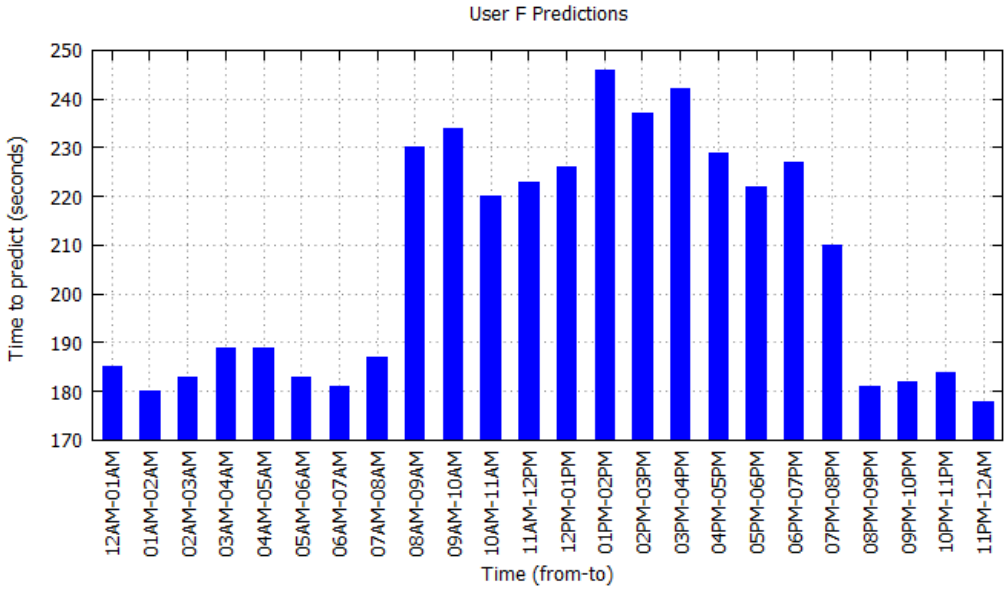

Figure 3.10 User F prediction for 24 hours window

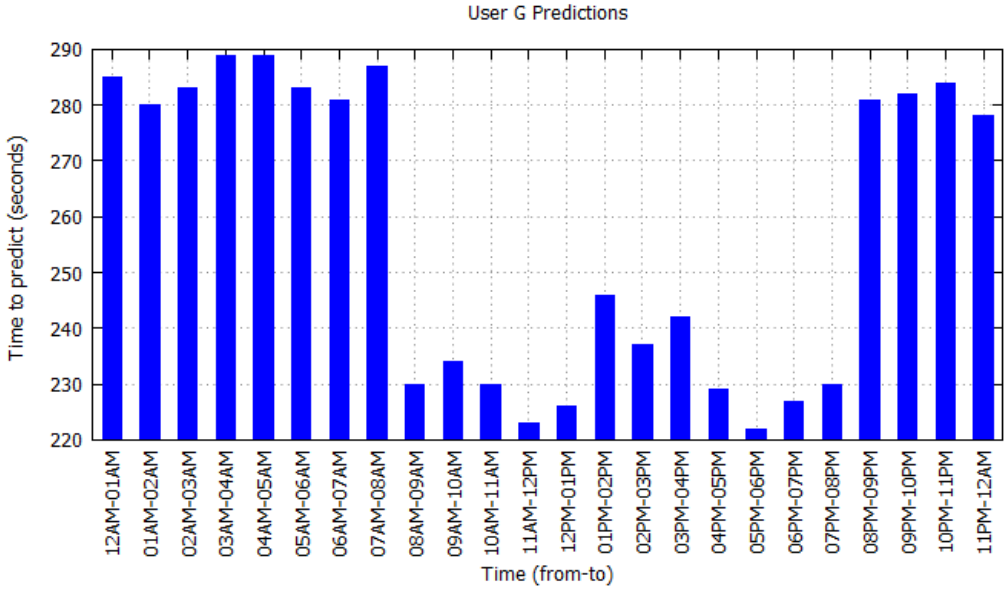

Figure 3.11 User G prediction for 24 hours window

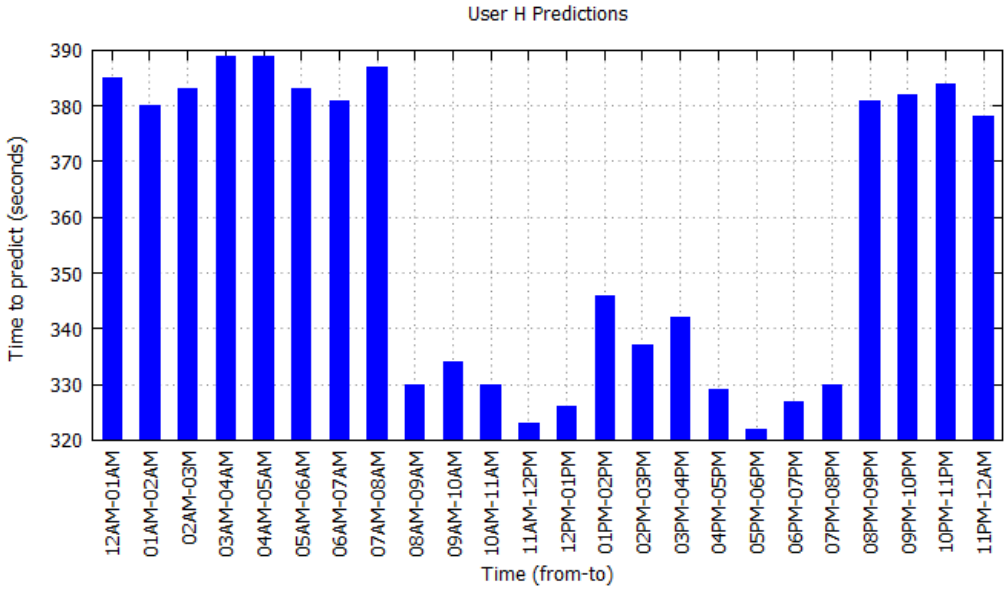

Figure 3.12 User H prediction for 24 hours window

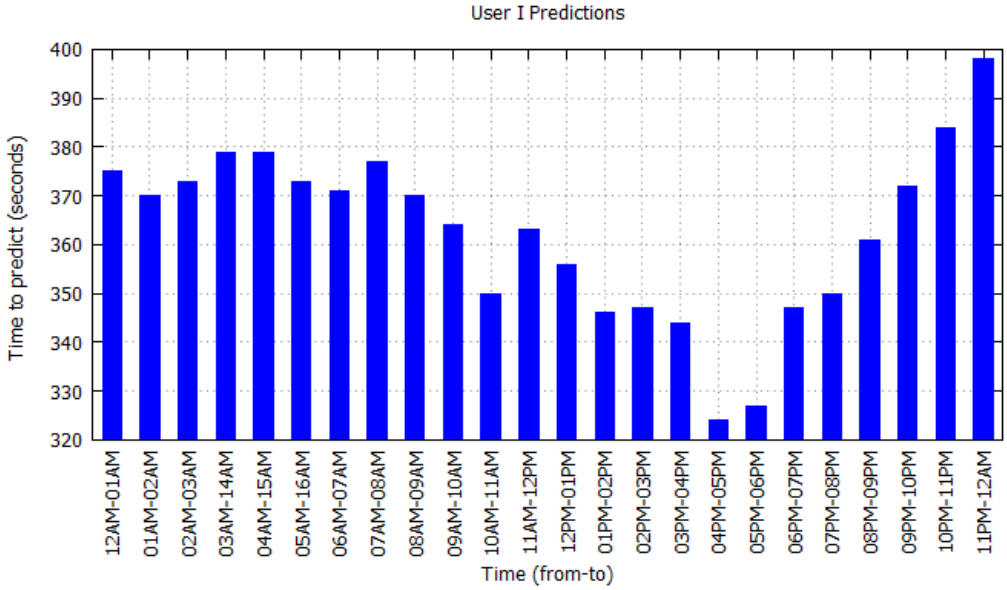

Figure 3.13 User I prediction for 24 hours window

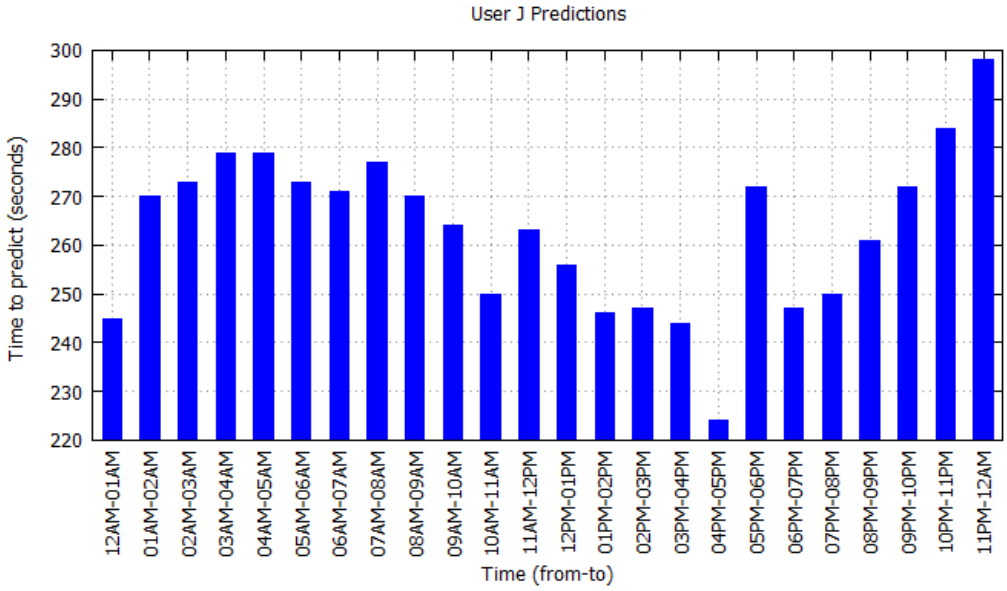

Figure 3.14 User J prediction for 24 hours window

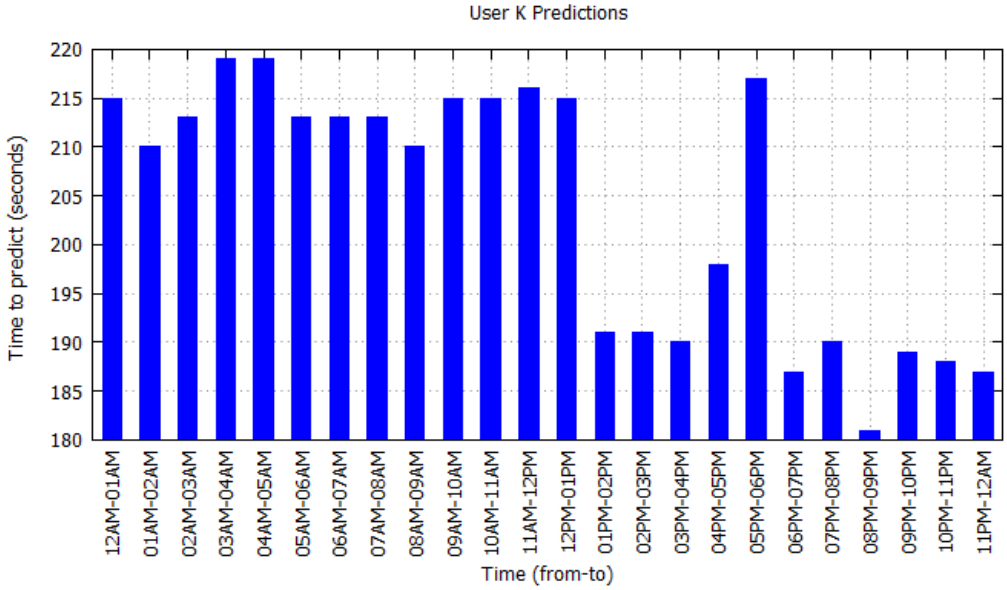

Figure 3.15 User K prediction for 24 hours window

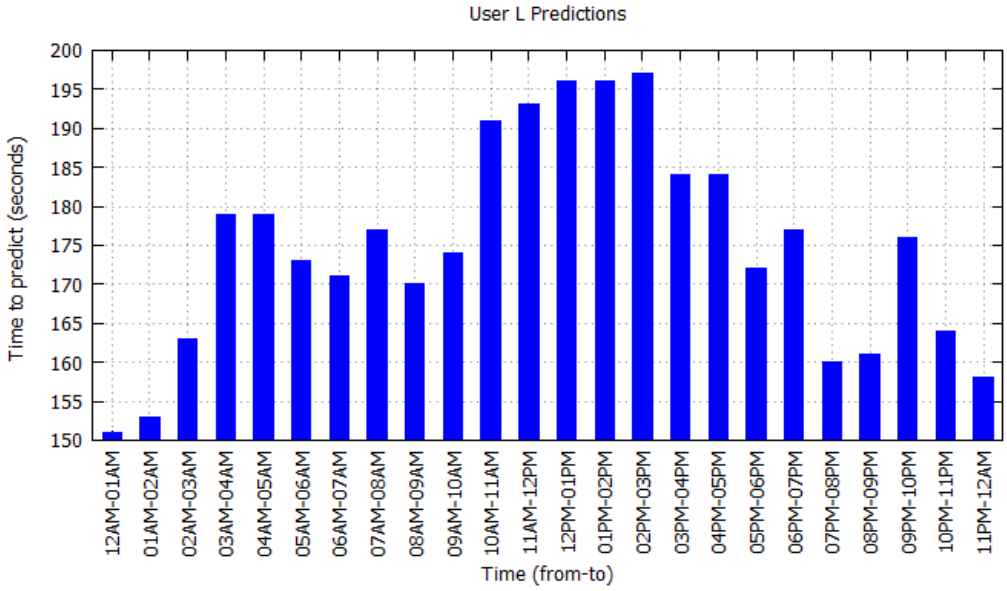

Figure 3.16 User L prediction for 24 hours window

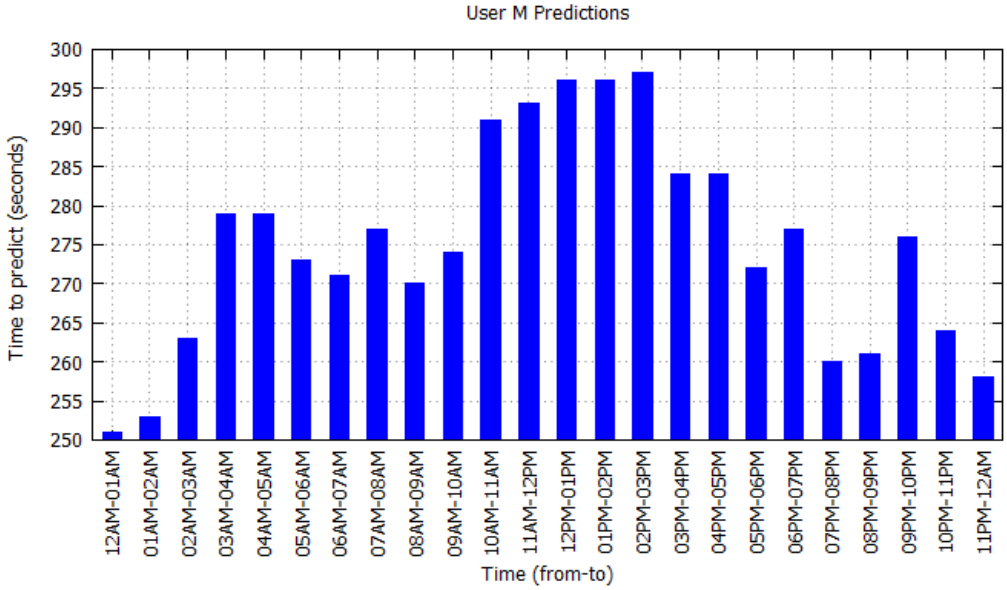

Figure 3.17 User M prediction for 24 hours window

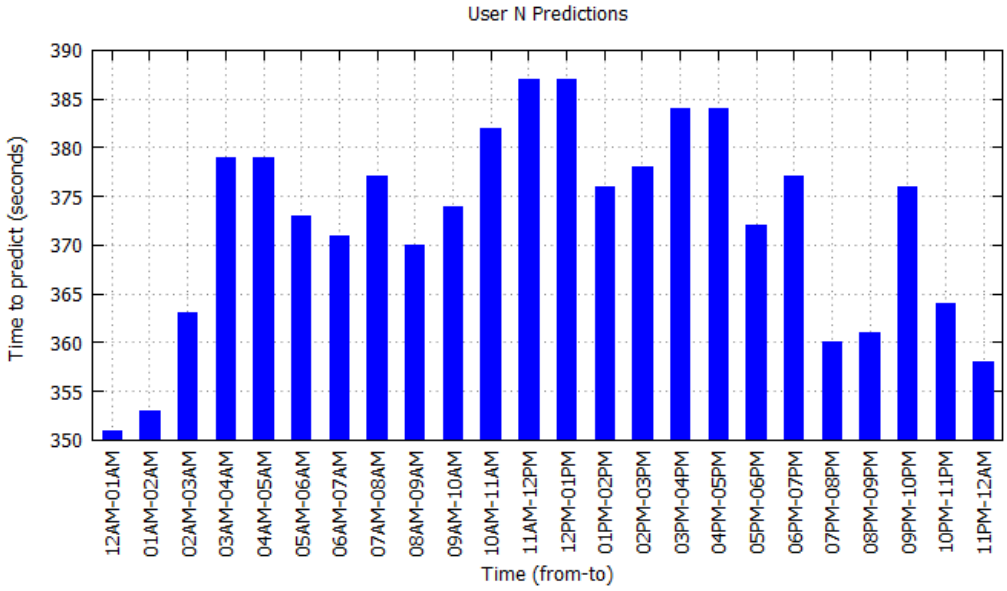

Figure 3.18 User N prediction for 24 hours window

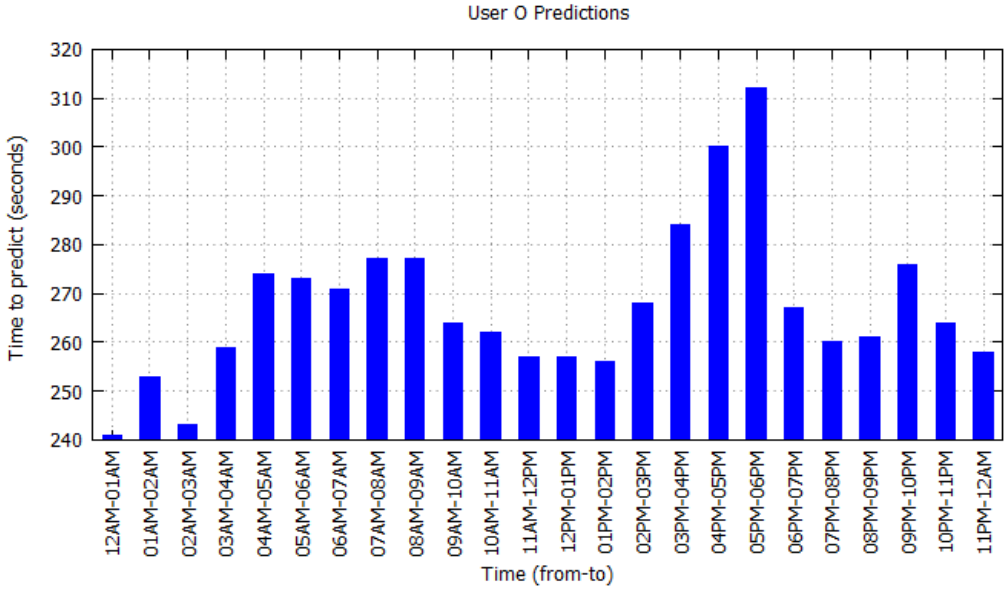

Figure 3.19 User O prediction for 24 hours window

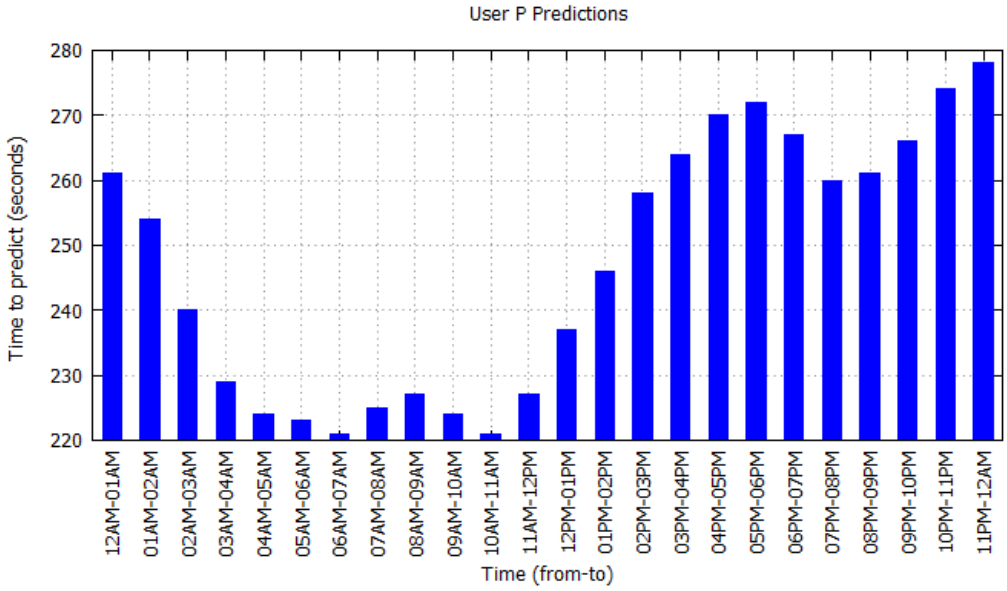

Figure 3.20 User P prediction for 24 hours window

3.1.4 Single User on Different Days of the Week

Here we are querying the location of different users temporally on different days for the same time window. We are considering the time between 09 am and 10 am. The attributes passed to the query as in Table 3.3. Results are illustrated in Figure 3.21 to Figure 3.36 and explained at the end.

The results illustrates the user's activities on the respective days. The lesser the time taken for the query execution on the day and time the lesser the activities the user carried out on that day of the week. The more the time taken for execution of the query. The more activities were performed by the users.

Table 3.3 Single user on different days of the week

Please enter vehicle number?	*User A*
Please enter Day 1 from Monday to Sunday?	*Monday*
Please enter Time from 00:00:00?	*09:00:00*
Please enter Time to 23:59:59?	*10:00:00*

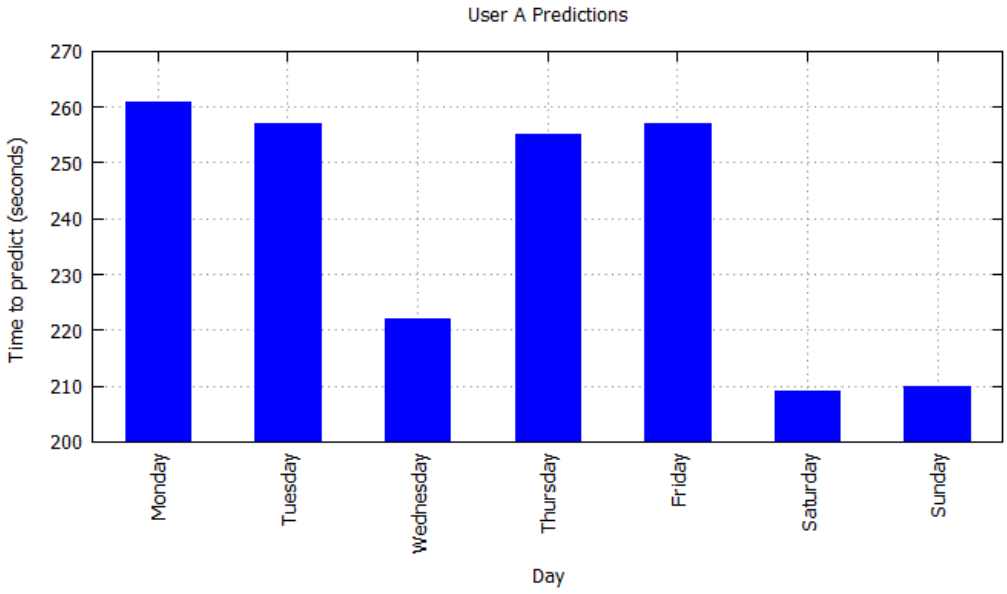

Figure 3.21 User A predictions for each day

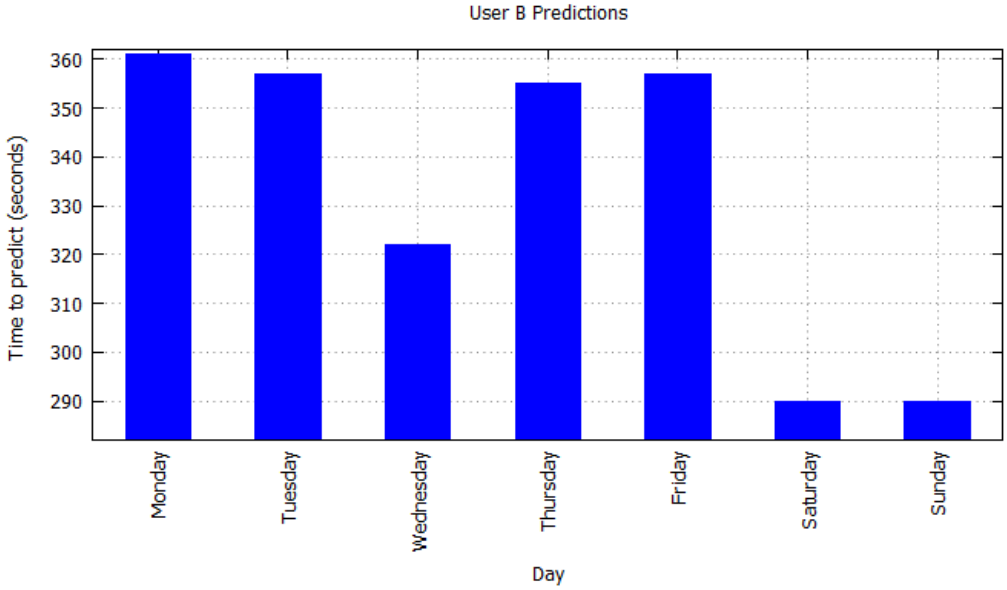

Figure 3.22 User B predictions for each day

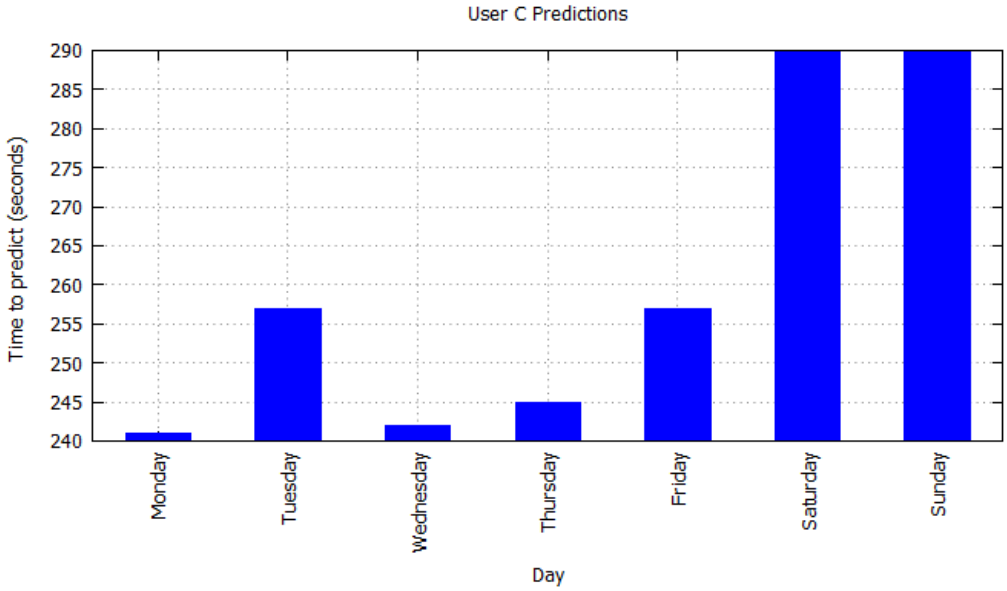

Figure 3.23 User C predictions for each day

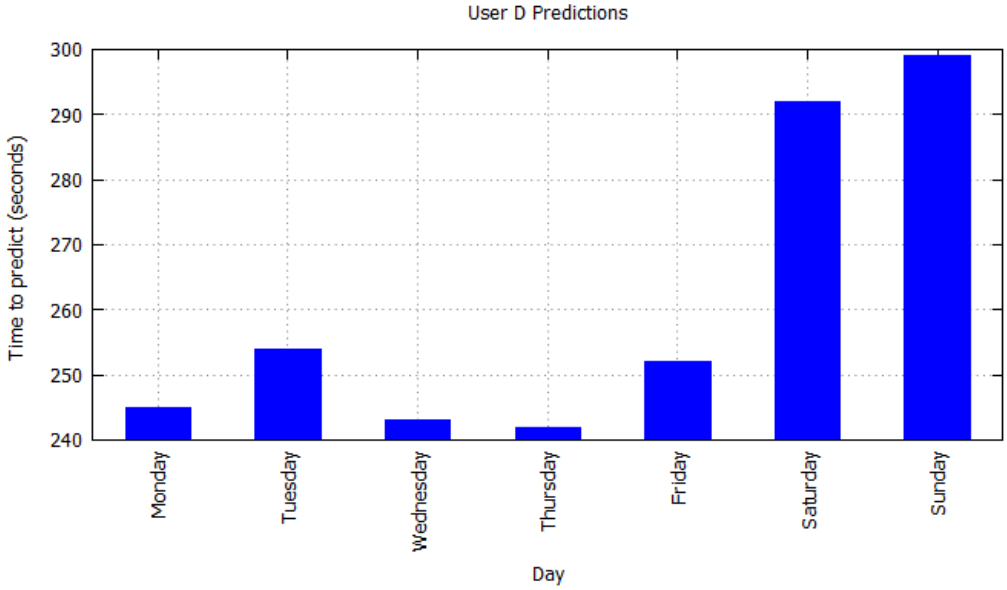

Figure 3.24 User D predictions for each day

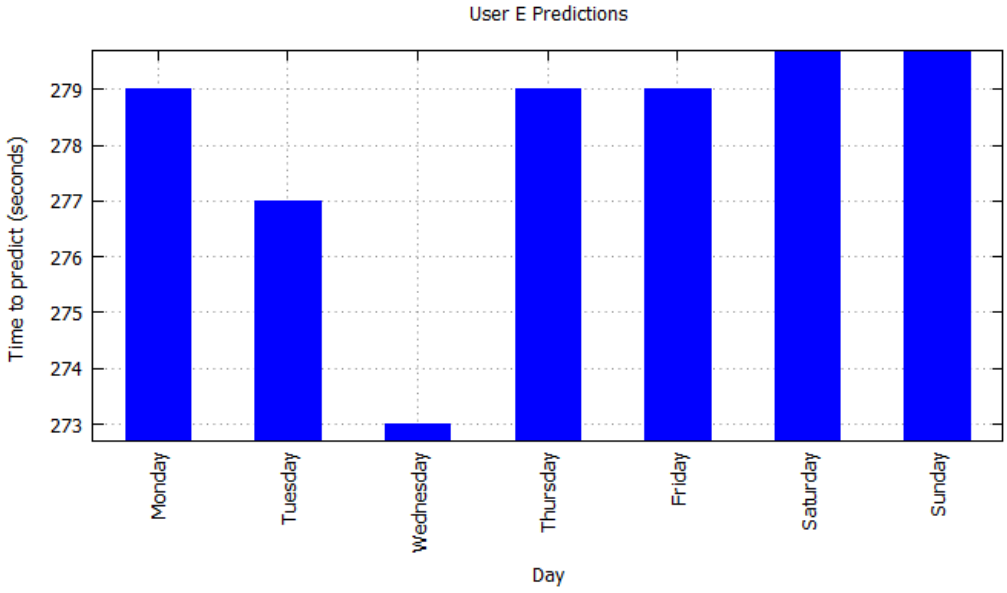

Figure 3.25 User E predictions for each day

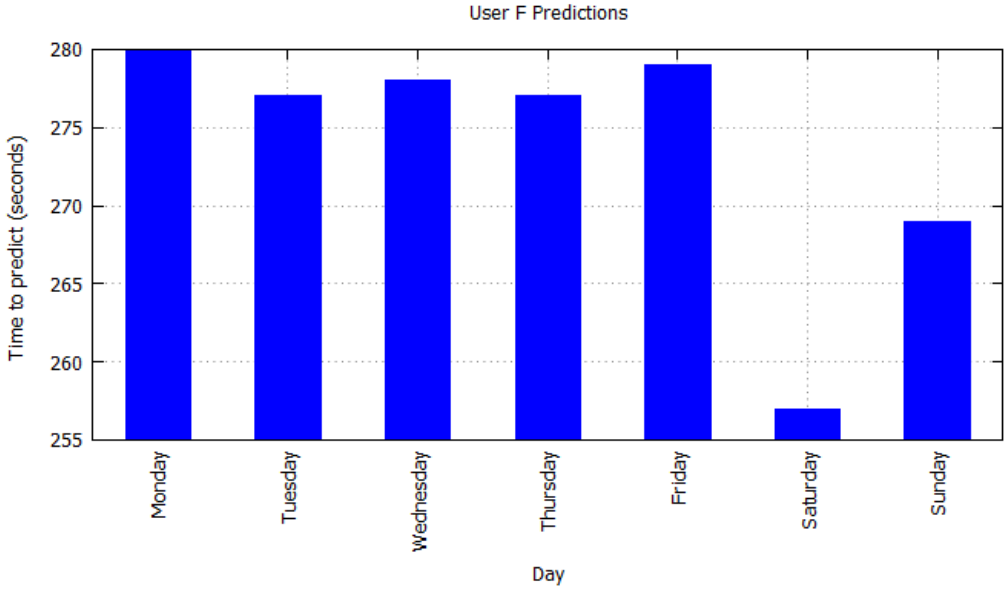

Figure 3.26 User F predictions for each day

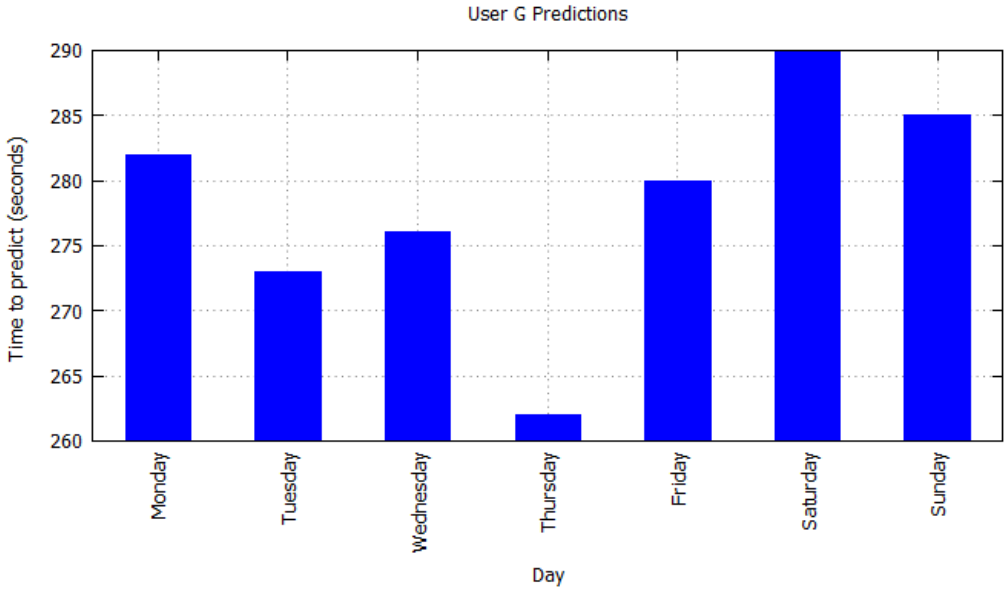

Figure 3.27 User G predictions for each day

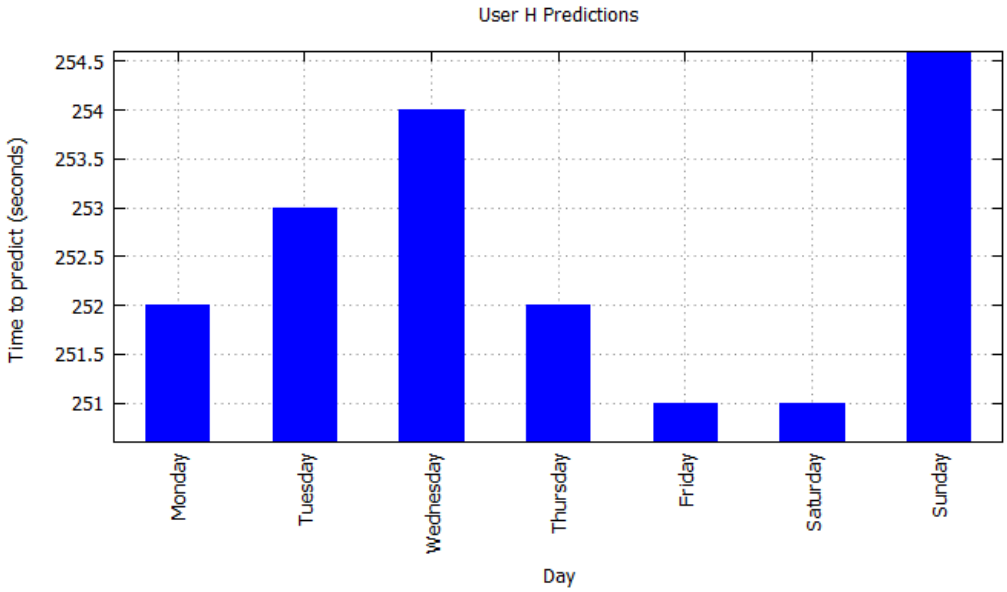

Figure 3.28 User H predictions for each day

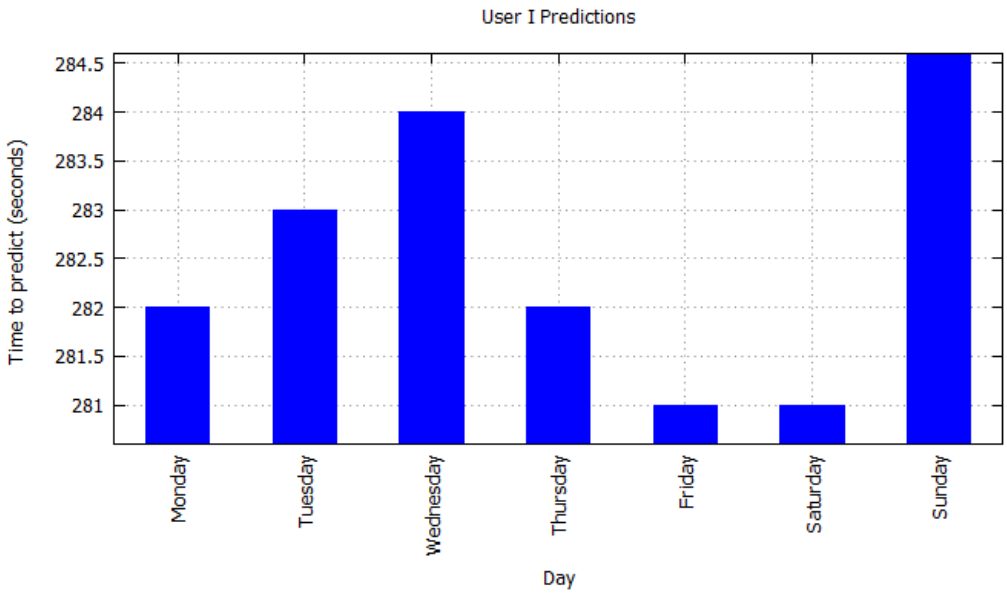

Figure 3.29 User I predictions for each day

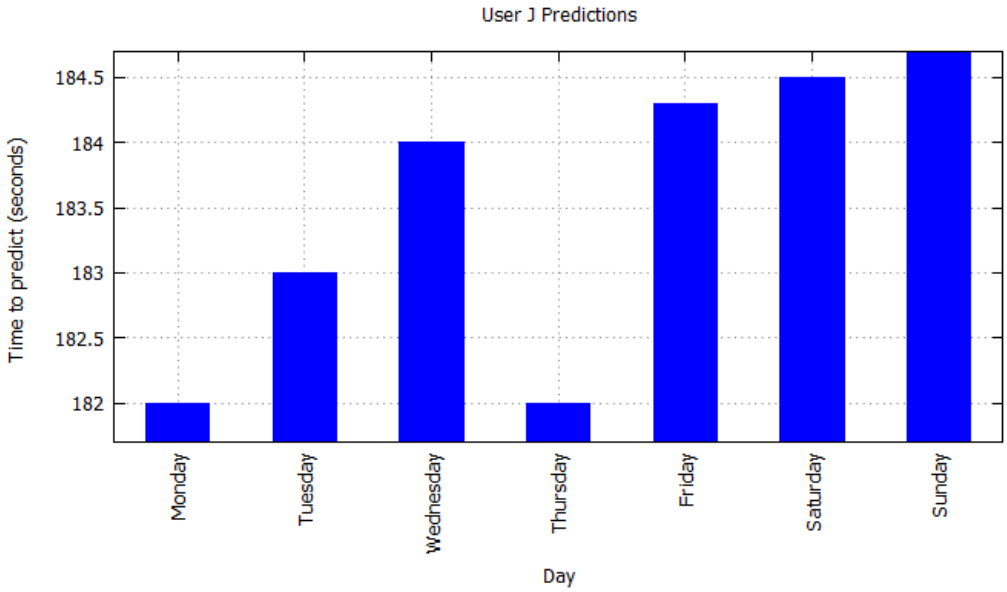

Figure 3.30 User J predictions for each day

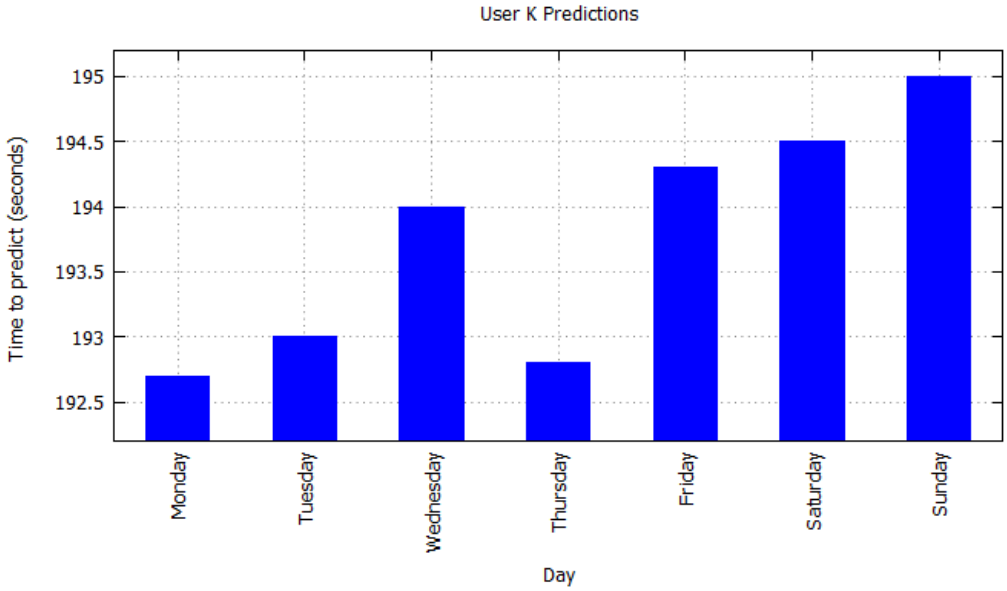

Figure 3.31 User K predictions for each day

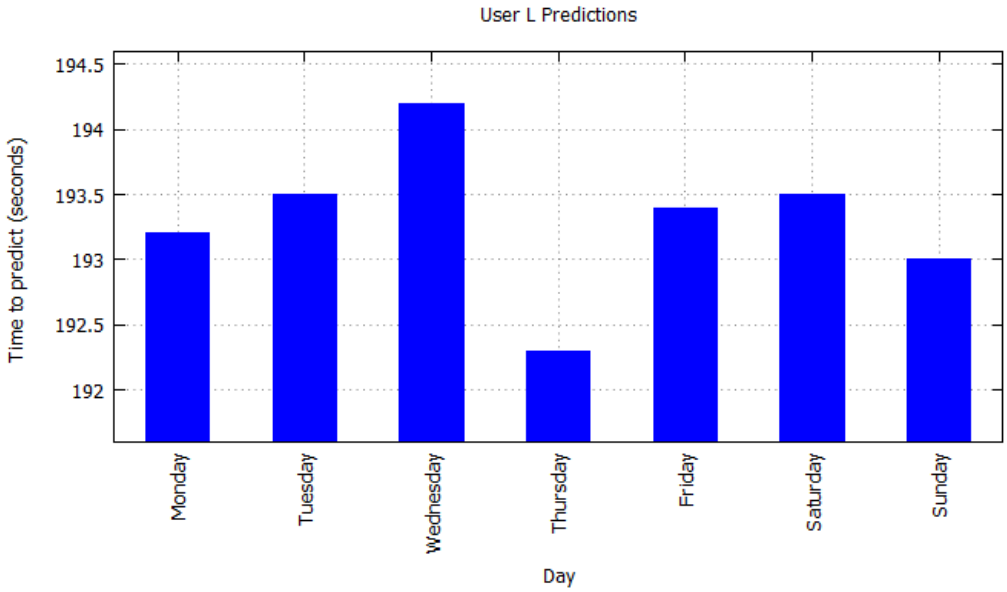

Figure 3.32 User L predictions for each day

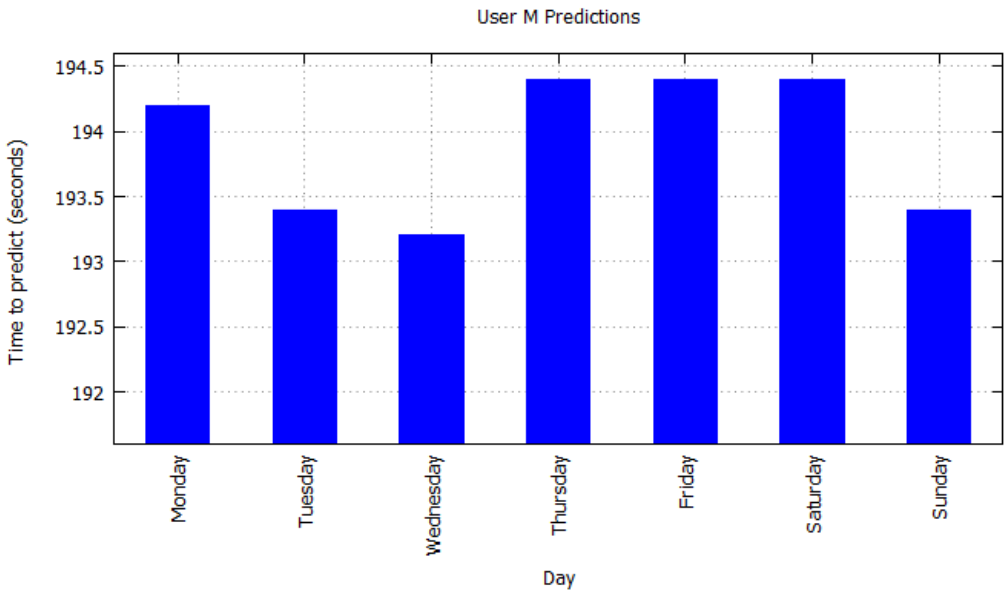

Figure 3.33 User M predictions for each day

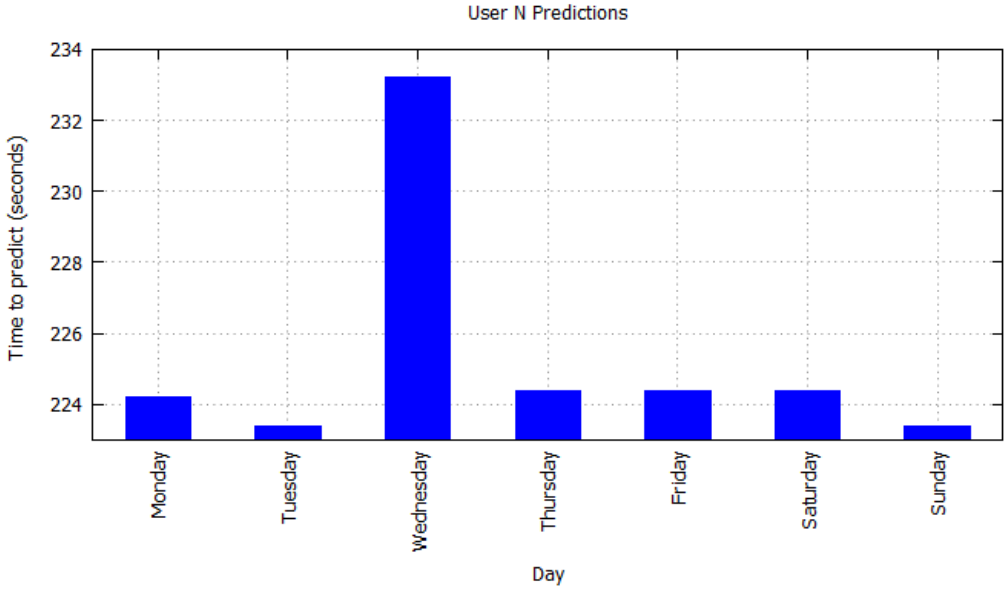

Figure 3.34 User N predictions for each day

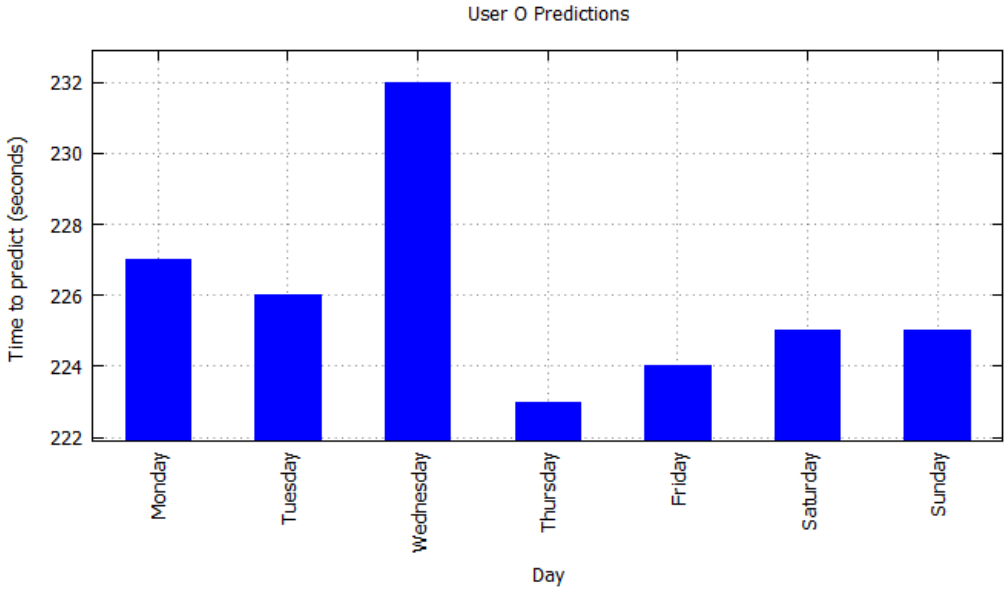

Figure 3.35 User O predictions for each day

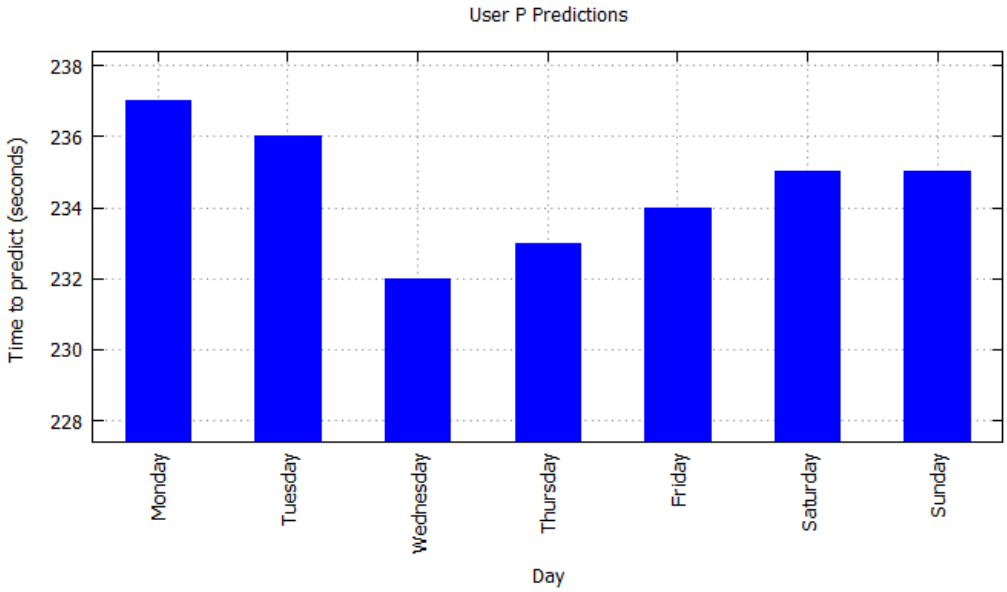

Figure 3.36 User P predictions for each day

Table 3.4 Queries for with and without using Apache Spark

Query No.	Vehicle No.	Day	Time from	Time to
Query 1	User A	Monday	10:00:00	12:00:00
Query 2	User B	Tuesday	11:00:00	13:00:00
Query 3	User C	Wednesday	12:00:00	14:00:00
Query 4	User D	Thursday	13:00:00	15:00:00
Query 5	User E	Friday	14:00:00	16:00:00

3.1.5 Queries Before and After Using Apache Spark

One of the objective of our research was to reduce the processing time of big GPS data. We initially developed our algorithm in python and used some libraries of pandas for predicting the future locations. We had some issues for running the queries on our dedicated system as the query used to occupy all the Ram available. We made all the RAM available for the process and then ran our queries. Figure 3.37 shows the results of the time taken in seconds by each query of Table 3.4.

After this when we developed the algorithm using Apache Spark for the queries shown in Table 3.4. We achieved a remarkable amount of decrease in the time taken by the queries. The job took more than two thousands seconds without using Apache Spark. After using Apache Spark the queries took less than 300 seconds. Figure 3.38 illustrates the time taken in detail.

3.1.6 Location Prediction of Single User for Multiples Queries

We developed the algorithm to predict the future location of a vehicle queried by a user. We are going to discuss some queries and their results in this section. Table 3.5 illustrates all the queries while Table 3.6 to Table 3.10 illustrates results of the queries.

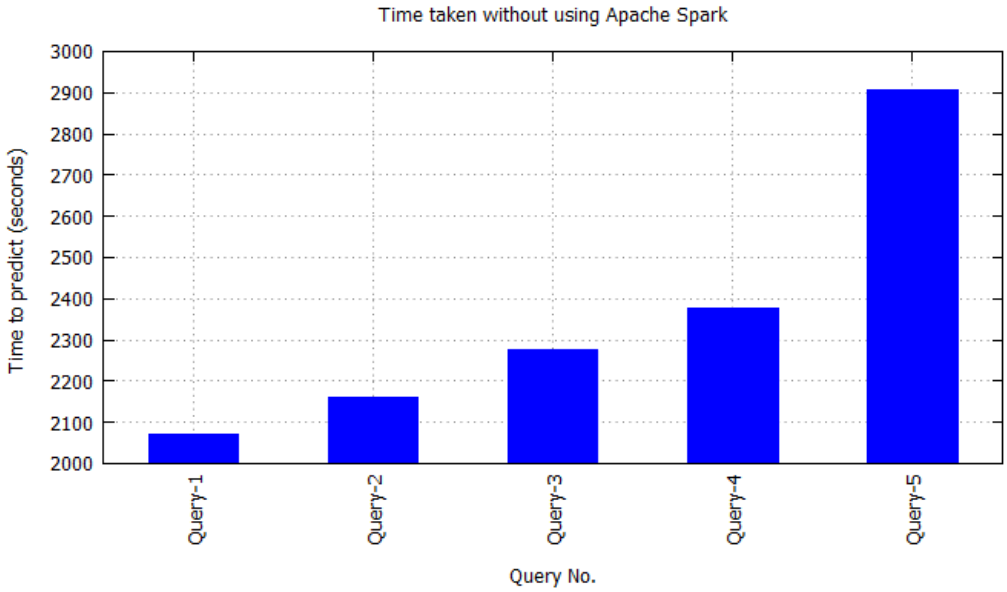

Figure 3.37 Time taken without using Apache Spark

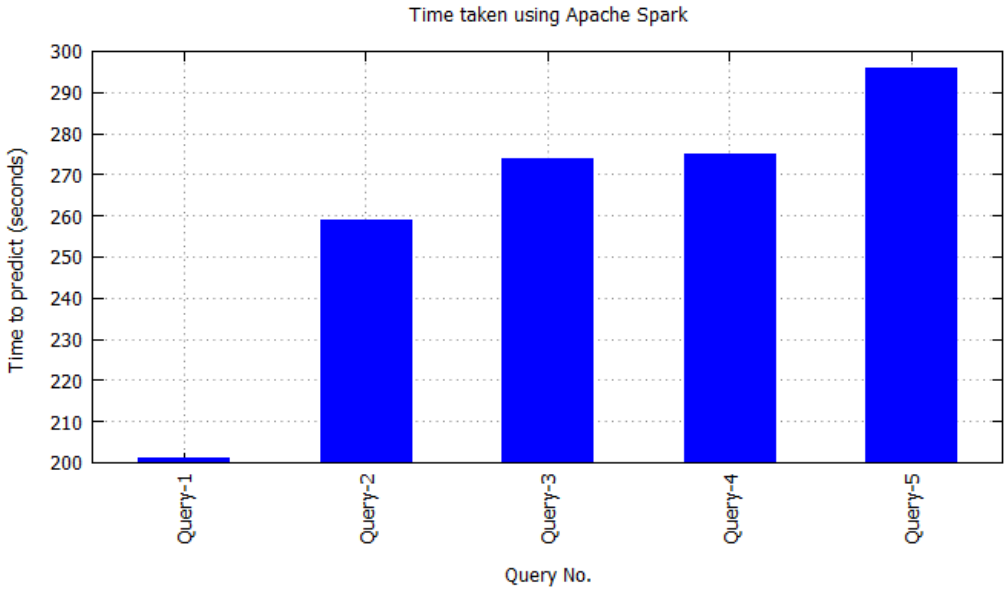

Figure 3.38 Time taken using Apache Spark

Table 3.5 User A Queries for Prediction

Query No.	Vehicle No.	Day	Time from	Time to
Query 1	User A	Tuesday	10:15:00	10:30:00
Query 2	User A	Tuesday	10:00:00	10:15:00
Query 3	User A	Tuesday	11:00:00	11:15:00
Query 4	User A	Wednesday	13:00:00	13:10:00
Query 5	User A	Friday	14:10:00	14:20:00

Table 3.6 Results of Query 1 Predictions

Top Probable	Prediction Percentage	Location Name	Latitude	Longitude
1st	61 %	Authorized Rest Area by FKTC Hamdani Hotel Danda Shah Bilawal	32.73745	71.87213333333334
2nd	23%	Wazir Ali Industries - Multan Bypass	30.16925	71.50620000000002
3rd	16%	Mastal F/S Dhok Mastal (Shell Site)	32.606966666666665	71.65476666666666

Table 3.7 Results of Query 2 Predictions

Top Probable	Prediction Percentage	Location Name	Latitude	Longitude
1st	52 %	LooniKot Check Post # 10	25.1946	67.85836666666665
2nd	32%	Shell Highway F/S (Rawalpindi)	33.321916666666667	72.78576666666666
3rd	26%	Shell Battle Axe F/S Shami Chowk Cantt.	31.522216666666665	74.36839999999998

Table 3.8 Results of Query 3 Predictions

Top Probable	Prediction Percentage	Location Name	Latitude	Longitude
1st	71 %	CALTEX DEPOT MACHIKE	31.7359	73.88689999999998
2nd	16%	Commercial market	31.32146	72.78543267
3rd	13%	Quaid e Azam Town Main Gate Lahore-Rawalpindi Bypass	32.182816666666667	74.1474

Table 3.9 Results of Query 4 Predictions

Top Probable	Prediction Percentage	Location Name	Latitude	Longitude
1st	78 %	FKTC Rest area Waan Banjra mianwali hotel Sarai Mohajir	31.5454333 3333333	71.2516333 3333333
2nd	12%	Nooriabad	25.18405	67.8199
3rd	10%	Usman Ghani Chowk Near Gopal Singh Rice Sheller Jandiala Road Sheikhupura	31.75	73.9620333 3333334

Table 3.10 Results of Query 5 Predictions

Top Probable	Prediction Percentage	Location Name	Latitude	Longitude
1st	92 %	FKTC Rest area Waan Banjra mianwali hotel Sarai Mohajir	31.5454333 3333333	71.2516333 3333333
2nd	6%	Nooriabad	25.18405	67.8199
3rd	2%	Usman Ghani Chowk Near Gopal Singh Rice Sheller Jandiala Road Sheikhupura	31.75	73.9620333 3333334

3.2 Accuracy of Predicted Locations

In this section we are going to discuss how we find the accuracy of a predicted location. We made separate files from the twelve month data. Six months were used to predict the future locations of users and then the data from the next six months were used to find the accuracy of the predicted locations. We compared the real-time locations from the next six months data with the predicted output for the queries. Table 3.11 mentions the queries whose accuracy percentage is illustrated in Figure 3.39.

The three bars for each query in Figure 3.39 shows the accuracy of the top three locations that were queried.

3.3. Geo-Visualization of Predicted Future Locations

After we had predicted the top three probable locations a vehicle would be found. We wanted to display the resulted locations on map for any further processing and analytics. We choose python libraries as we were already working in python to display the three locations on web browser. We used ‘folium’ library of python to display the location on the map. When a user queries, it generates the top three probable locations. Once the query has finished execution, a web page is created from the results of the query in the same directory. We plotted three markers with info-windows over them with their information. We also developed some maps to visualize the resulted locations. The queries are displayed in Table 3.12 and their respective resulting maps are visualized as follows in Figure 3.40 to Figure 3.42.

Table 3.11 Queries to find Accuracy

Query No.	Vehicle No.	Day	Time from	Time to
Query 1	User A	Monday	00:15:00	00:30:00
Query 2	User B	Tuesday	11:00:00	11:15:00
Query 3	User C	Wednesday	11:00:00	11:15:00
Query 4	User D	Thursday	11:00:00	11:10:00
Query 5	User F	Friday	09:10:00	09:30:00

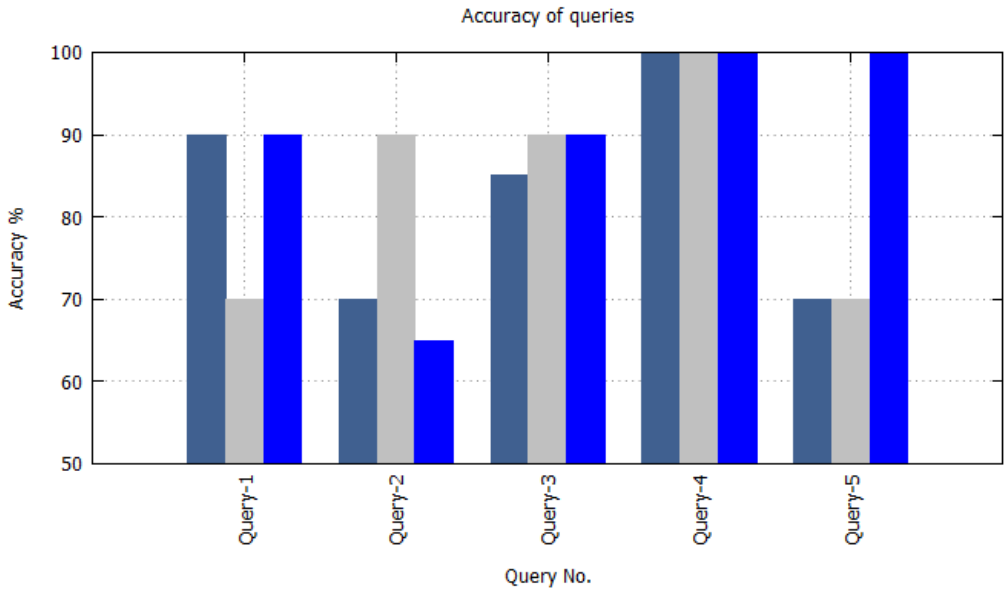

Figure 3.39 Accuracy of queries

Table 3.12 Queries for Geo-visualization

Query No.	Vehicle No.	Day	Time from	Time to
Query 1	User A	Wednesday	12:00:00	01:00:00
Query 2	User Z	Monday	09:00:00	10:00:00
Query 3	User Z	Tuesday	09:00:00	10:00:00

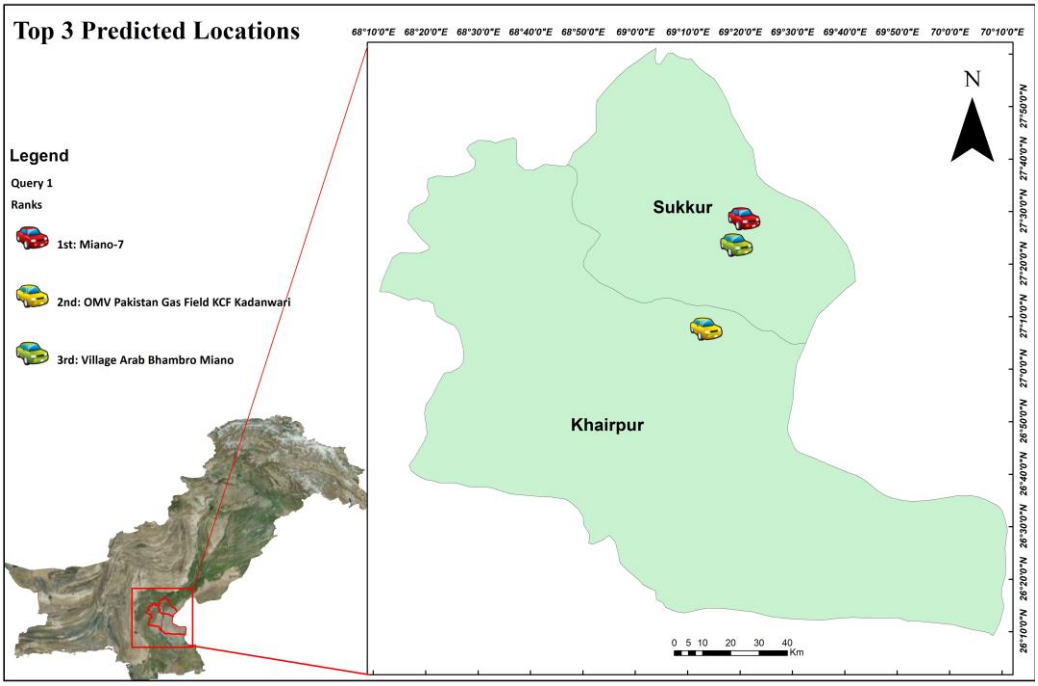

Figure 3.40 Geo-visualization of Query 1

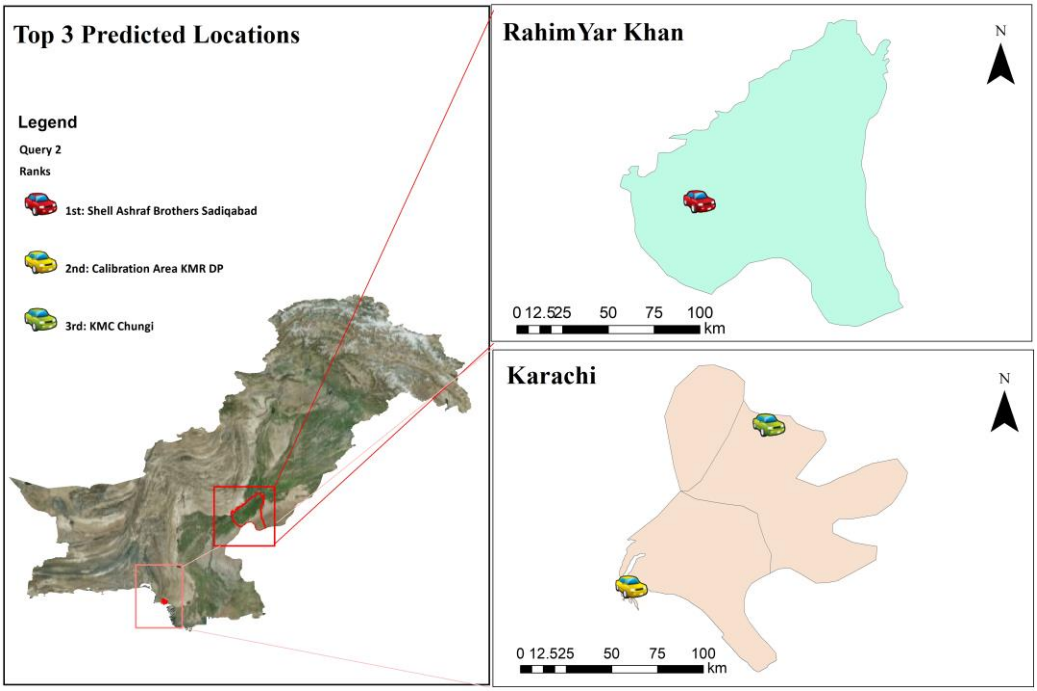

Figure 3.41 Geo-visualization of Query 2

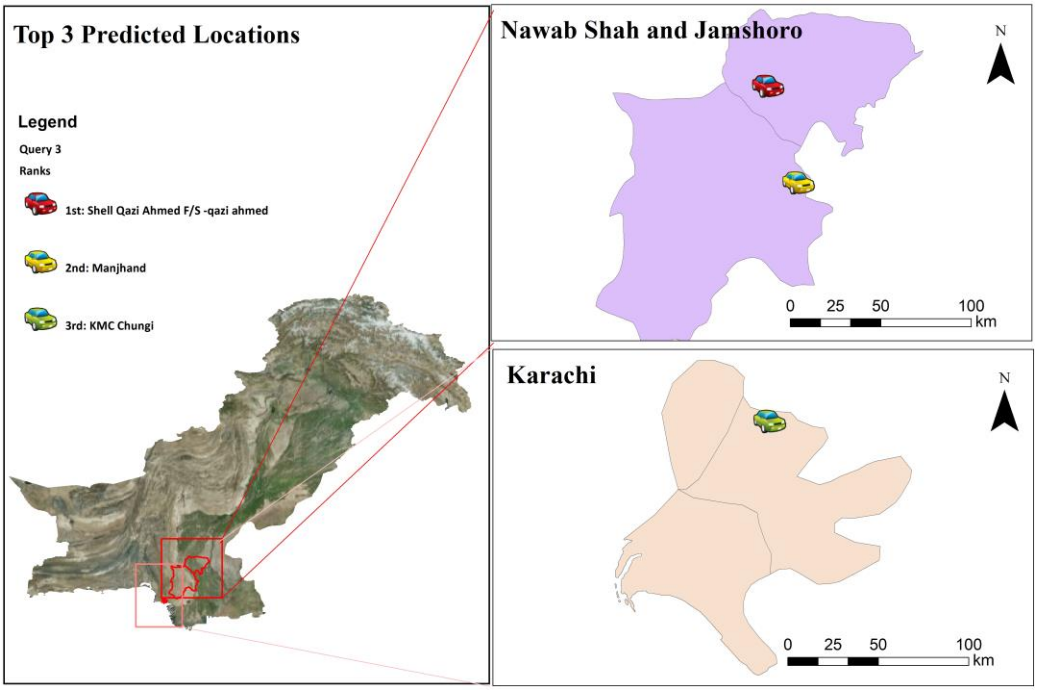

Figure 3.42 Geo-visualization of Query 3

Further visual illustration for result of Query 3 are display in the following figures. The 1st ranked probable location is visualized in Figure 3.43 and Figure 3.44. Similarly, 2nd ranked probable location is visualized in Figure 3.45 and Figure 3.46. Lastly, the 3rd ranked probable location is visualized in Figure 3.47 and Figure 3.48 with different base maps.

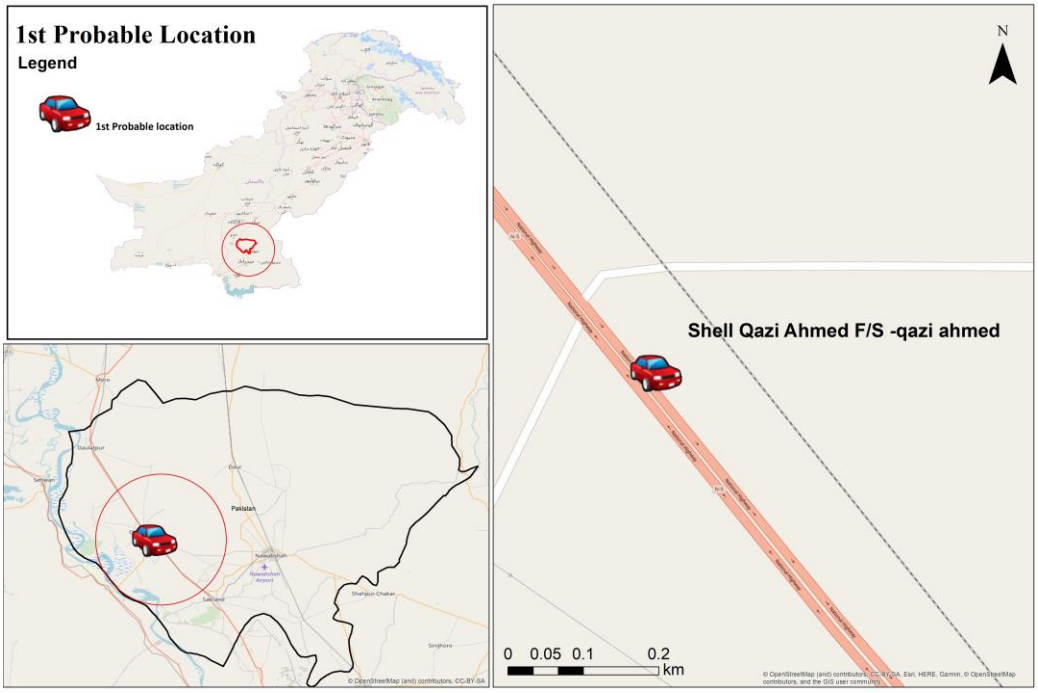

Figure 3.43 First Probable Location using OSM

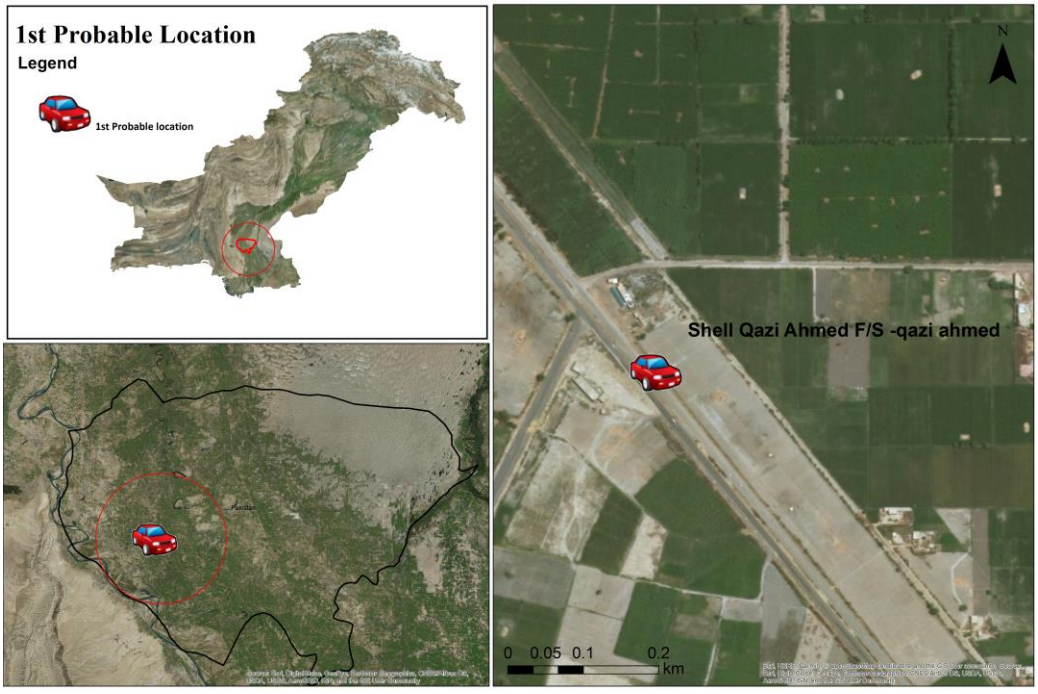

Figure 3.44 First Probable Location using World Imagery

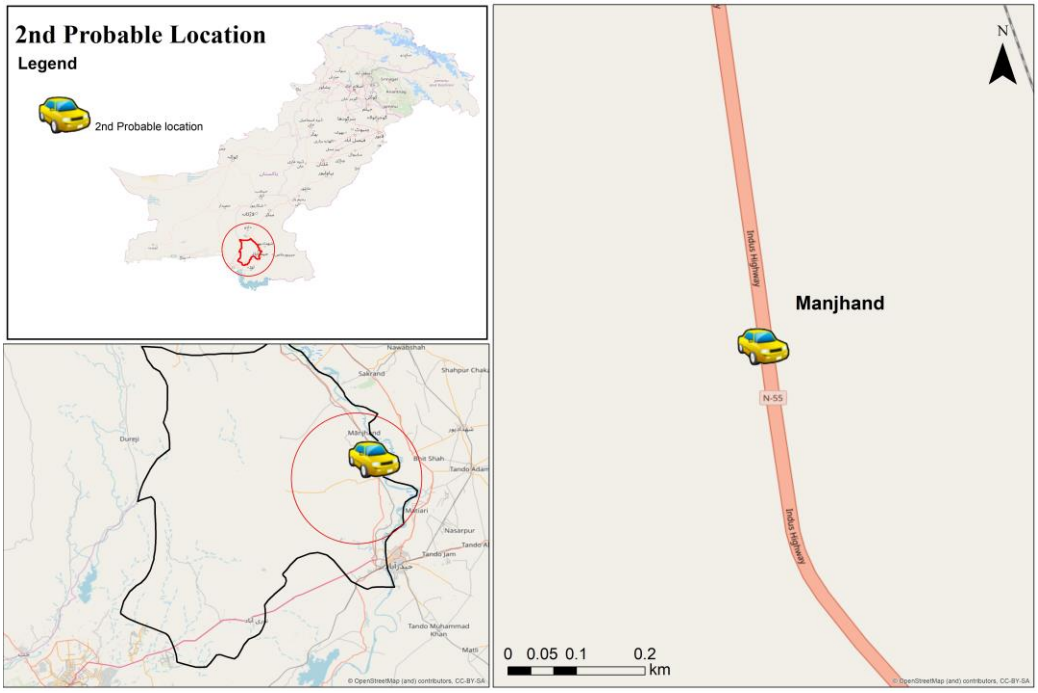

Figure 3.45 Second Probable Location using OSM

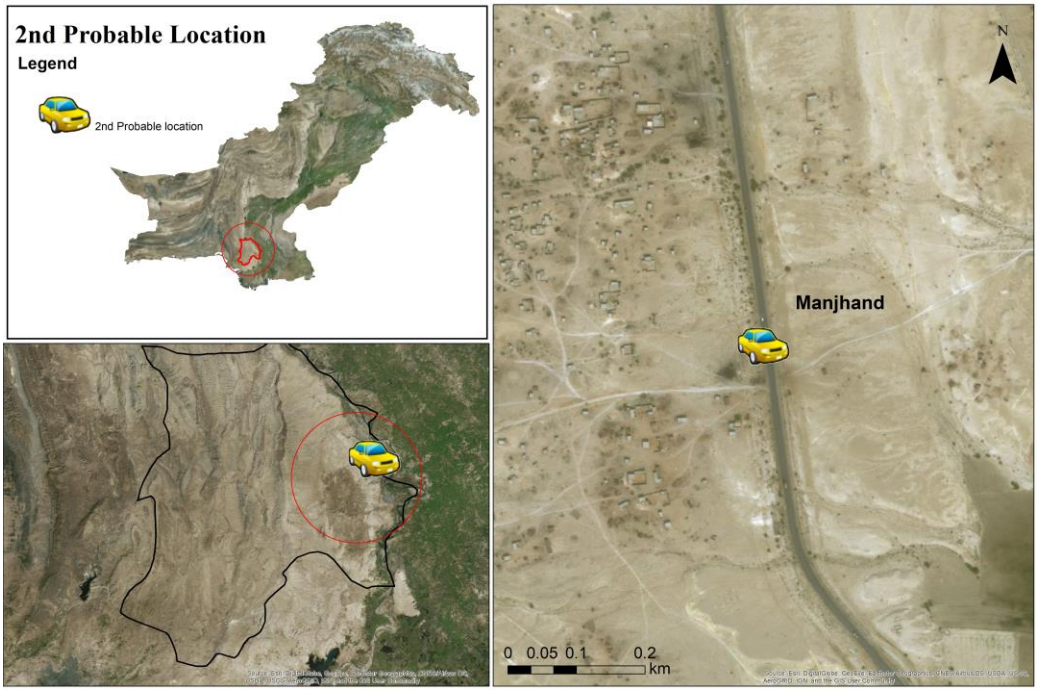

Figure 3.46 Second Probable Location using World Imagery

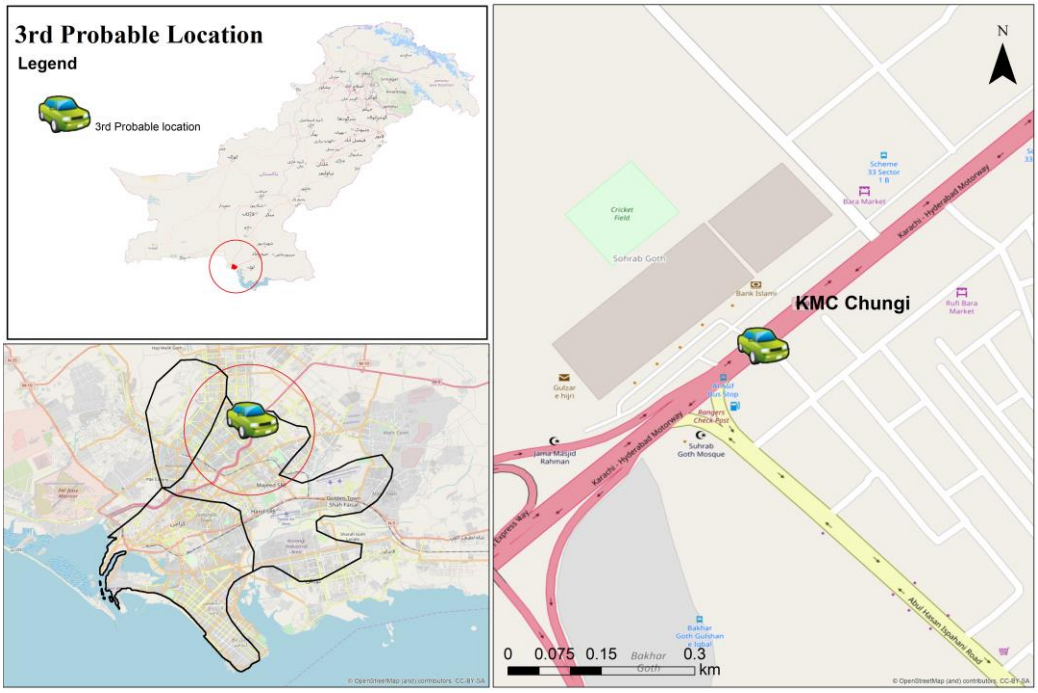

Figure 3.47 Third Probable Location using OSM

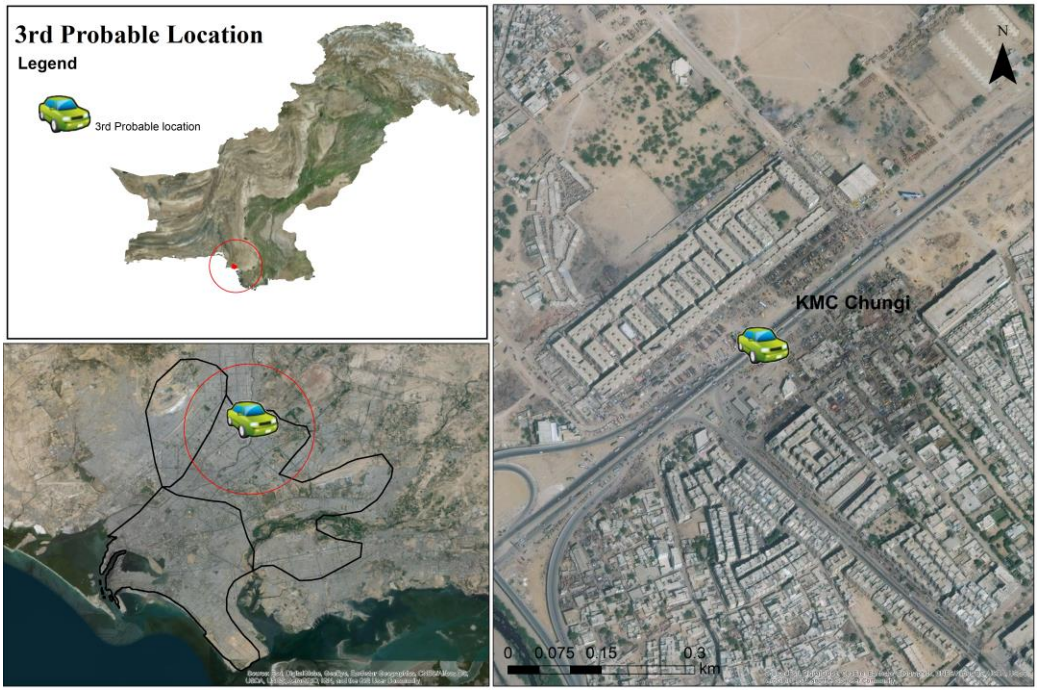

Figure 3.48 Third Probable Location using World Imagery

Chapter 4

Conclusion and Recommendations

4.1 Conclusions

Spatial analysis performance was the primer focus of this research on raw GPS big data collected through installed GPS devices in vehicles. Conversion of this data into meaningful information was carried out using Apache Spark, big data platform. Vehicle user's patterns were predicted using a novel algorithm which was developed on python. However, we found that simple software's has limitations in predicting location from big data. As the number of data increases the processing capabilities of simple algorithm over simple libraries decreases. Which results in higher latency rate for the job at hand. Using Apache Spark helped improve the efficiency and latency of the algorithm developed on pyspark. Apache Spark reduced the processing time to up to 300% then the time consumed without running the algorithm on Apache Spark.

Predicting the future location of a user can assist different spatial task the user performs and can improve the analytical knowledge gained from understanding their behavior. This research work can also help solve commercial and security related problems. The user behavior is observed to understand their space time patterns. Although Apache Spark is a powerful big data tool which supports built-in spatiotemporal data types, however the range of functions it provides needs a learning curve to be familiar with its functionalities. The tutorials available helped overcome this limitation by learning Apache Spark from the tutorials provided by Apache Spark officials. By using spatial temporal data this process can be

generalized for analysis in different domains. This area of research is yet to be unexplored in context of Pakistan. However, this study can help in giving a meaning direction towards other applications of GPS data, mobility data in context to Pakistan.

4.2 Recommendations for further research

GPS mobility data analysis is an active area for research. There are still a lot of studies going on in finding patterns and predicting future or next location. Based on the current work there is another application of we can use Apache Spark for real-time streaming big data as well along with batched data. Apache Spark supports distributed processing which can be very helpful for reducing the big data load in a single processer.

Another such area of focus could be introducing more nodes to the distributed processing to enhance the efficiency of the system running the queries. More data attributes can be introduced to analyze additional information which can reveal meaningful information and patterns for real time applications. Further analysis can be carried out in answering the question how and why a user visited a particular location. This can be helpful in finding the semantics of trajectories and carrying out their analysis.

Similarly, another future area in the algorithm can be predicting the next location of a vehicle using its previous history i.e. where a user will be next after a specific location. Commercial, security and traffic aspects can also be looked after using this study. By making a system to predict a route for vehicles that will be congested for specific time and ask the vehicle user if they wants to avoid that road. This study opens up further future directions for researchers.

References

Apache Spark History. (2018, September 27). *Apache Spark*. Retrieved from https://spark.apache.org/history.html

APIs, A. S. (2019). Apache Spark APIs. Retrieved from https://databricks.com/blog/2016/07/14/a-tale-of-three-apache-spark-apis-rdds-dataframes-and-datasets.html

Ashbrook, D., & Starner, T. (2003a). Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous computing, 7*(5), 275-286.

Ashbrook, D., & Starner, T. (2003b). Using GPS to learn significant locations and predict movement across multiple users. *P & UC, 7*(5), 275-286.

Cheng, C., Jain, R., & van den Berg, E. (2003). Location prediction algorithms for mobile wireless systems. Paper presented at the Wireless Internet Handbook.

Cho, S.-B. (2016). Exploiting machine learning techniques for location recognition and prediction with smartphone logs. *Neurocomputing, 176*, 98-106.

DataFrame. (2019). Difference between DataFrame, Dataset, and RDD in Spark. Retrieved from https://stackoverflow.com/questions/31508083/difference-between-dataframe-dataset-and-rdd-in-spark

Do, T. M. T., & Gatica-Perez, D. (2012). Contextual conditional models for smartphone-based human mobility prediction. Paper presented at the Proceedings of the 2012 ACM Conference on Ubiquitous Computing.

Froehlich, J., & Krumm, J. (2008). Route prediction from trip observations (0148-7191). Retrieved from https://www.sae.org/publications/technical-papers/content/2008-01-0201/

Gambs, S., Killijian, M.-O., & del Prado Cortez, M. N. (2012). Next place prediction using mobility markov chains. Paper presented at the Proceedings of the First Workshop on Measurement, Privacy, and Mobility.

Gao, H., Tang, J., & Liu, H. (2012). Mobile location prediction in spatio-temporal context. Paper presented at the Nokia Mobile Data Challenge Workshop.

Garnier, J., Osguthorpe, D. J., & Robson, B. (1978). Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of Molecular Biology, 120*(1), 97-120.

Giannotti, F., & Pedreschi, D. (2008). Mobility, data mining and privacy: Geographic Knowledge Discovery: Springer Science & Business Media.

Gonzalez, J. E., Xin, R. S., Dave, A., Crankshaw, D., Franklin, M. J., & Stoica, I. (2014). GraphX: Graph Processing in a Distributed Dataflow Framework. Paper presented at the OSDI.

Guessoum, D., Miraoui, M., & Tadj, C. (2016). Contextual location prediction using spatio-temporal clustering. *International Journal of Pervasive Computing and Communications, 12*(3), 290-309.

Güting, R. H., Behr, T., & Düntgen, C. (2010a). SECONDO: A Platform for Moving Objects Database Research and for Publishing and Integrating Research Implementations. *IEEE Data Eng. Bull., 33*(2), 56-63.
Güting, R. H., Behr, T., & Düntgen, C. (2010b). Secondo: A platform for moving objects database research and for publishing and integrating research implementations: Fernuniv., Fak. für Mathematik u. Informatik.
Hadoop, A. (2018). Apache Hadoop. Retrieved from https://hadoop.apache.org/
Hermes. (2009). Hermes. Retrieved from http://incubator.apache.org/projects/hermes.html
Ho, W., Ho, G. T., Ji, P., & Lau, H. C. (2008). A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence, 21*(4), 548-557.
Jensen, F. V. (1996). An introduction to Bayesian networks (Vol. 210): UCL press London.
Jeung, H., Liu, Q., Shen, H. T., & Zhou, X. (2008). A hybrid prediction model for moving objects. Paper presented at the Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on.
Karbassi, A., & Barth, M. (2003). Vehicle route prediction and time of arrival estimation techniques for improved transportation system management. Paper presented at the Intelligent Vehicles Symposium, 2003. Proceedings. IEEE.
Kim, B., Kang, S., Ha, J.-Y., & Song, J. (2015). Agatha: predicting daily activities from place visit history for activity-aware mobile services in smart cities. *International Journal of Distributed Sensor Networks*.
Krumm, J. (2008). A markov model for driver turn prediction (0148-7191). Retrieved from https://www.microsoft.com/en-us/research/publication/markov-model-driver-turn-prediction/
Krumm, J., & Horvitz, E. (2006). Predestination: Inferring destinations from partial trajectories. Paper presented at the International Conference on Ubiquitous Computing.
Lefèvre, S., Vasquez, D., & Laugier, C. (2014). A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH, 1*(1), 1.
Lin, M., & Hsu, W.-J. (2014). Mining GPS data for mobility patterns: A survey. *PMC, 12*, 1-16.
Liu, Q., Wu, S., Wang, L., & Tan, T. (2016). Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts. Paper presented at the AAAI.
Lu, J., & Güting, R. H. (2012). Parallel secondo: boosting database engines with hadoop. Paper presented at the 2012 IEEE 18th International Conference on Parallel and Distributed Systems.
Mapreduce, S. V. (2019). Spark Vs. Mapreduce. Retrieved from https://intersog.com/blog/apache-spark-vs-hadoop-mapreduce/
Marketos, G., Damiani, M., Pelekis, N., Theodoridis, Y., & Yan, Z. (2013). Trajectory collection and reconstruction.
Matekenya, D., Ito, M., Shibasaki, R., & Sezaki, K. (2016). Enhancing location prediction with big data: evidence from dhaka. Paper presented at the Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct.

Mobile. (2013). Mobile. Retrieved from http://itunews.itu.int/en/3741-Mobile-subscriptions-near-the-78209billion-markbrDoes-almost-everyone-have-a-phone.note.aspx

Monreale, A., Pinelli, F., Trasarti, R., & Giannotti, F. (2009). Wherenext: a location predictor on trajectory pattern mining. Paper presented at the Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Morris, B. T., & Trivedi, M. M. (2008). A survey of vision-based trajectory learning and analysis for surveillance. *IEEE transactions on circuits and systems for video technology*, *18*(8), 1114-1127.

Morzy, M. (2006). Prediction of moving object location based on frequent trajectories. Paper presented at the International Symposium on Computer and Information Sciences.

Morzy, M. (2007). Mining frequent trajectories of moving objects for location prediction. *Machine Learning and Data Mining in Pattern Recognition*, 667-680.

Petzold, J. (2005). Augsburg Indoor Location Tracking Benchmarks. Context Database, Institute of Pervasive Computing, University of Linz, Austria.

Petzold, J., Bagci, F., Trumler, W., & Ungerer, T. (2004). Confidence estimation of the state predictor method. Paper presented at the European Symposium on Ambient Intelligence.

Petzold, J., Bagci, F., Trumler, W., & Ungerer, T. (2005). Next location prediction within a smart office building. *Cognitive Science Research Paper-University of Sussex CSRP*, *577*, 69.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81-106.

Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine*, *3*(1), 4-16.

Renso, C., Spaccapietra, S., & Zimányi, E. (2013). Mobility Data: Cambridge University Press.

Scellato, S., Musolesi, M., Mascolo, C., Latora, V., & Campbell, A. T. (2011). NextPlace: a spatio-temporal prediction framework for pervasive systems. Paper presented at the International Conference on Pervasive Computing.

Simmons, R., Browning, B., Zhang, Y., & Sadekar, V. (2006). Learning to predict driver route and destination intent. Paper presented at the Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE.

Spaccapietra, S., Parent, C., Damiani, M. L., de Macedo, J. A., Porto, F., & Vangenot, C. (2008). A conceptual view on trajectories. *Data & Knowledge Engineering*, *65*(1), 126-146.

Spark, A. (2018). Apache Spark. Retrieved from https://spark.apache.org/

Spark, A. (2019). Introduction to Apache Spark. Retrieved from https://aws.amazon.com/big-data/what-is-spark/

Specht, D. F. (1990). Probabilistic neural networks. *Neural Networks*, *3*(1), 109-118.

Tao, Y., Faloutsos, C., Papadias, D., & Liu, B. (2004). Prediction and indexing of moving objects with unknown motion patterns. Paper presented at the Proceedings of the 2004 ACM SIGMOD international conference on Management of data.

Tran, L. H., Catasta, M., McDowell, L. K., & Aberer, K. (2012). Next place prediction using mobile data. Paper presented at the Proceedings of the Mobile Data Challenge Workshop (MDC 2012).

Vieira, M. R., Bakalov, P., & Tsotras, V. J. (2009). On-line discovery of flock patterns in spatio-temporal data. Paper presented at the Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.

Xu, Gao, S., Daneshmand, M., Wang, C., & Liu, Y. (2017). A survey for mobility big data analytics for geolocation prediction. *IEEE Wireless Communications, 24*(1), 111-119.

Xu, & Wolfson, O. (2003). Time-series prediction with applications to traffic and moving objects databases. Paper presented at the Proceedings of the 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access.

Yavaş, G., Katsaros, D., Ulusoy, Ö., & Manolopoulos, Y. (2005). A data mining approach for location prediction in mobile environments. *Data & Knowledge Engineering, 54*(2), 121-146.

Ying, J. J.-C., Lee, W.-C., Weng, T.-C., & Tseng, V. S. (2011). Semantic trajectory mining for location prediction. Paper presented at the Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.