University of Windsor
Scholarship at UWindsor

Electronic Theses and Dissertations | Theses, Dissertations, and Major Papers

12-12-2018

Big Data Mining to Construct Truck Tours

Vidhi Kantibhai Patel
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation

Patel, Vidhi Kantibhai, "Big Data Mining to Construct Truck Tours" (2018). *Electronic Theses and Dissertations*. 7620.
https://scholar.uwindsor.ca/etd/7620

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

Big Data Mining to Construct Truck Tours

By

Vidhi Patel

A Thesis

Submitted to the Faculty of Graduate Studies

through the School of Computer Science

in Partial Fulfillment of the Requirements for the

Degree of Master of Science

at the University of Windsor

Windsor, Ontario, Canada

2018

© 2018 Vidhi Patel

Big Data Mining to construct Truck Tours

By

Vidhi Patel

APPROVED BY:

H. Maoh

Department of Civil and Environmental Engineering

P. Zadeh

School of Computer Science

M. Kargar, Co-Advisor

School of Computer Science

J. Chen, Advisor

School of Computer Science

December 12, 2018

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and the intellectual content of this thesis is the product of my own work and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyones copyright nor violate any proprietary rights and any ideas or techniques and all the assistance received in preparing this thesis and sources have been fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Cross-Border shipping of goods among different distributors is an essential part of transportation across Canada and U.S. These two countries are heavily dependent on border crossing locations to facilitate international trade between each other. This research considers the identification of the international tours accomplishing the shipping of goods. A truck tour is a round trip where a truck starts its journey from its firm or an industry, performing stops for different purposes that include taking a rest, fuel refilling, and transferring goods to multiple locations, and returns back to its initial firm location. In this thesis, we present a three step method on mining GPS truck data to identify all possible truck tours belonging to different carriers. In the first step, a clustering technique is applied on the stop locations to discover the firm for each carrier. A modified DBSCAN algorithm is proposed to achieve this task by automatically determining the two input parameters based on the data points provided. Various statistical measures like count of unique trucks and count of truck visits are applied on the resulting clusters to identify the firms of the respective carriers. In the second step, we tackle the problem of classifying the stop locations into two types: primary stops, where goods are transferred, and secondary stops like rest stations, where vehicle and driver needs are met. This problem is solved using one of the trade indicator called Specialization Index. Moreover, several set of features are explored to build the classification model to classify the type of stop locations. In the third step, having identified the firm, primary and secondary locations, an automated path finder is developed to identify the truck tours starting from each firm. The results of the specialization index and the feature-based classification in identifying stop events are compared with the entropy index from previous work. Experimental results show that the proposed set of cluster features significantly add classification power to our model giving 98.79% accuracy which in turn helps in discovering accurate tours.

DEDICATION

Dedicated to my parents Kanti and Sunita Patel and my sister Pooja Raychura.

ACKNOWLEDGMENT

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

I would like to reflect on the people who have supported me and helped me throughout this period of my research. I would like to express my deep and sincere gratitude to my research advisors Dr. Jessica Chen and Dr. Mehdi Kargar for providing invaluable guidance, continuous support and motivation.

I would also like to thank my thesis committee members Dr. Hanna Maoh and Dr. Pooya Moradian Zadeh for their valuable guidance, comments and suggestions that added more value to my thesis work.

I would especially like to deeply thank Dr. Mina Maleki. As my teacher and mentor, she has taught me more than I could ever give her credit for here. I would like to thank my mentors Dr. Hanna Maoh and Dr. Mina Maleki from my research at Cross-Border Institute (CBI) for giving me the opportunity to be a part of this institute for my thesis work.

Also I would like to thank all the staff of Graduate Society of Computer Science for their kindness. I am extending my thanks to all my friends and colleagues who supported and helped me during this period.

On a personal note, I would like to express my deepest gratitude to my parents for their love, prayers, caring and sacrifices for educating and preparing me for my future.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Symbol	Definition
GDP	Gross Domestic Product
GPS	Global Positioning system
DBSCAN	Density Based Spatial Clustering Of Applications With Noise
STING	Statistical Information Grid
CLIQUE	Clustering in QUest
SVM	Support Vector Machine
UTC	Universal Transverse Mercator
TOA	Time Of Arrival
CDMA	Code Division Multiple Access
SQL	Structured Query Language
SNN	Shared Nearest Neighbor
OPTICS	Ordering points to identify the clustering structure
CLARANS	Clustering Large Applications Based On Randomized Search
SI	Specialization Index
EI	Entropy Index
FI	Formulated Index
CF	Cluster Features
TF	Temporal Features

Chapter 1

INTRODUCTION

1.1 Motivation

Major roads and highways constitute the primary transportation infrastructure for the transit movement of goods. The historical trend of highway investment has taken into account the role of trucking as the predominant mode of transport. Trucking accounts for 31% of revenue in Canada's commercial transportation sector by gross domestic product (GDP). The remaining modes including air-based, rail-based, and marine based travel represent 12%, 11%, and 2% of Canada's commercial transportation GDP, respectively [4]. By volume, 72% of domestic goods are transported by trucks, while rail and marine modes only haul 21% and 7%, respectively [5]. Major link between Canada and U.S. for freight transportation on highways are the three border crossing locations i.e. Ambassador Bridge, Peace Bridge, and Blue Water Bridge. Moreover, trade between Canada and the U.S. relies heavily on trucks as the dominant method of transport. Therefore, domestic and Canada-U.S. trade are both highly dependent on trucks as a major source of commercial transportation to ship goods.

One of the technology used in transportation to gain useful information about the navigation and truck tracking is Global Positioning System (GPS). GPS is one such satellite based technology that helps in tracking of vehicles in real-time. Sensors in trucks and other freight vehicles also give real-time information about how the vehicle is performing, how fast it is going, how long it is on the go, how long it is standing still etc. This real-time data obtained from GPS sensors could be processed and mined to get meaningful information about the vehicle and its transportation activities. This processing and mining of the

transportation data from GPS sources usually involves the handling of Big Data. Big Data in transportation research plays an important role in the information retrieval within freight transportation. Mining freight information from GPS data using concepts and techniques in computer science and spatial science highly motivated the present research work.

The continuous growth of the usage of GPS (Global Positioning System) devices in transportation management systems has evolved into generation of huge amounts of GPS pings. The enormous spatio-temporal data produced by such devices has greatly increased the interest in the application of big data mining algorithms. Extracting meaningful information from raw GPS data is a crucial task in most location-aware applications. Hence, this task is becoming more and more interesting. A lot of recent research has focused on mobile phones data, while the commercial vehicles sector is almost unexplored. The problems typically involve detection of interesting places and classification of the detected locations.

This thesis focuses on one of the major tasks of mining the GPS data coming from the commercial fleet transportation systems to identify firms for each carrier that are associated with truck stop locations and then discover the tours starting from each firm. Also, this thesis includes major contribution in proposing a different approach of categorizing stop locations using an economic geography index known as the specialization index [6] and the feature-based classification technique. The proposed approach presented in this research deals with a combined approach of using clustering and classification techniques. Clustering technique is basically used for solving two tasks in this research. The first task involves finding firm location by clustering the stop points of each carrier or fleet individually. A firm is an establishment of an industry hub that produces the goods that needs to be transported. Hence, a firm location is unique to each carrier and so we process the stop points of each carrier individually for finding firm location. Information on the locations of individual firms can be obtained from commercial organizations like Google or InfoCanada [7]. The actual location of a firm provides very useful data for activity based

freight transportation research. The second task involves the use of clustering technique to cluster all the stop points and then applying statistical metric to classify the purpose of each stop location. The data used in this thesis are collected by Shaw Tracking and provided to us by the Cross-Border Institute at the University of Windsor. The obtained dataset contains individual GPS pings for a large group of Canadian-owned freight carriers with truck movements across Canada and the U.S. This thesis uses the observed GPS records for the month of March, 2016. Here, a total of 569 carriers with 40,650 individual trucks with approximately 75 million GPS pings (i.e. data points) are analyzed. Each GPS ping results in a data record containing the carrier ID (CID), truck ID (PID), latitude, longitude, and time. The elapsed time between GPS pings and a dwell time that accumulates if the truck is stopped can be derived from the input GPS data. Pings associated with meaningful dwell times represent stop events that can be classified as (1) primary, which occurs when goods are transferred between the truck and location (or another truck), or (2) secondary, which occurs when a truck is stationary for other purposes like driver breaks or fuel refills [3]. Identification of the purpose of truck stops is important in mining meaningful information out of the truck GPS data. Primary stops are particularly important for models since they denote trip ends for a given truck. The firm is important to be discovered accurately as it denotes tour ends for a given truck. Likewise, secondary stops are useful since they complement primary stops by providing a complete picture on the nature of truck movements over space. Hence, it is really important to make accurate distinction between the types of stops since they correspond to different activities.

The truck tours discovered from GPS trajectory in this research help in building activity based freight models that represent the microscopic movement of commercial trucks [8]. These micro-simulation models helps traffic engineers and planners simulate freight activities as part of long-range urban and regional planning exercises. However, the development of such micro models requires observed information about the generated tours. The infor-

mation needed includes the length of the tours (i.e. duration), the frequency of the stops made by each truck, as well as the type of the stops (i.e. primary stop for picking or delivering goods, or secondary stops for resting and/or refueling).

1.2 Research Objective & Solution Outline

This research outlines the identification of the tours accomplishing the shipping of goods across Canada and US. This broader problem is solved by dividing it into three task: finding a firm location for each carrier, classifying the stop locations into primary and secondary stops, and forming the tours that typically starts from a firm location making multiple stops and return to the same firm location.

A truck tour is ideally defined as a round trip where a truck starts from the depot or establishment to perform one or more stops including primary stops for transferring goods and secondary stops for taking rest or fuel refilling and return to the same initial depot. It is also called trip chaining in terms of transportation as shown in Fig 1.1.

The objective of this thesis is to mine a large set of truck GPS data using the techniques of clustering and classification mainly to solve two major tasks. The first task involves applying clustering technique to the stop points of each carrier individually and identify the firm location specific to each carrier. The second major task involves categorizing the stop locations into primary and secondary stops using statistical metric called specialization index and feature-based classification technique. The classification results obtained from the proposed method in this thesis are compared with the one using the entropy index [3] from previous work. The Entropy Index is a measure of the level of order or disorder a system has. Typically, a system with a high level of order is said to have a low entropy. On the other hand, higher levels of disorder will be associated with higher entropy. The work in [3] uses the Shannon entropy to represent the level of homogeneity (i.e. order) versus

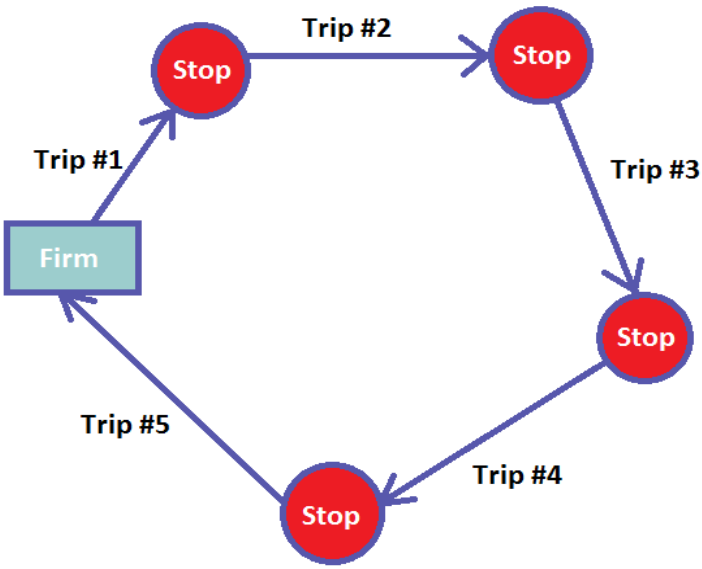

Figure 1.1: Tour

heterogeneity (i.e. disorder) at a given location. The general idea in [3] is to be able to determine if a location has a homogeneous cluster in which trucks pertaining to only one carrier are found at the location. On the contrary, if a location has a heterogeneous cluster then it will be associated with trucks from a variety of carriers, which makes it more likely to be a secondary stop. Experimental results show that the feature-based classification significantly outperforms the latter. The accuracy results showing the comparison with the entropy index are reported in chapter 5.

1.3 Structure of thesis

The remainder of this thesis is organized as follows.

Chapter 2 presents a brief introduction to GPS Architecture and its basics, together with an overview of clustering techniques and several clustering models. One of the famous clustering technique named DBSCAN and its algorithm is also introduced in this chapter. This chapter also covers basic algorithms used for classification problem in data mining.

Chapter 3 briefly describes previous studies in the field of mining GPS transportation data to find out meaningful information such as deriving stop locations from the raw GPS pings, and categorizing of the stop locations. Also, DBSCAN and its variations for finding clusters with varied shape, size and density, and other similar approaches are summarized. Several works including the variety of classification features for stop categorization are also included in this chapter.

Chapter 4 presents the primary contribution of this thesis and describes the implementation details of our approach. The major contribution of this research are the following:

- Implementing DBSCAN with automatic calculation of Eps (cluster radius) parameter.
- Implementing Specialization Index for classifying the purpose of stop locations.
- Implementing novel stop cluster features and building classification model.

Finally, Chapter 5 reports and discusses the experimental results.

Chapter 6 provides the summary to conclude the thesis along with the directions of possible future work.

Chapter 2

BACKGROUND STUDY

2.1 Global Positioning System

The Global Positioning System (GPS) is a satellite-based navigation system that consists of 24-orbiting satellites, each of which makes two circuits around the earth at every 24 hours [9]. GPS project was developed in 1973 to overcome the limitations of previous navigation system. GPS was developed and introduced by U.S DEPARTMENT OF DEFENSE and made freely accessible to everyone [9]. It is also known as NAVSTAR GPS (Navigation Satellite Timing and Ranging Global Positioning System).

2.1.1 GPS Overview & Architecture

The GPS system provides accurate, continuous, worldwide, three-dimensional position and velocity information to users with the appropriate receiving equipment. It also disseminates a form of Coordinated Universal Time (UTC) [9]. The satellite constellation nominally consists of 24 satellites arranged in 6 orbital planes with 4 satellites per plane. A worldwide ground control/monitoring network monitors the health and status of the satellites. This network also uploads navigation and other data to the satellites. The system utilizes the concept of one-way time of arrival (TOA) ranging. Satellite transmissions are referenced to highly accurate atomic frequency standards on-board the satellites, which are in synchronism with a GPS time base. The satellites broadcast ranging codes and navigation data on two frequencies using a technique called code division multiple access (CDMA); that is, there are only two frequencies in use by the system, called L1 (1,575.42

MHz) and L2 (1,227.6 MHz) [9]. Each satellite transmits on these frequencies, but with different ranging codes than those employed by other satellites. These codes were selected because they have low cross-correlation properties with respect to one another. Each satellite generates a short code referred to as the course/acquisition or C/A code and a long code denoted as the precision or P(Y) code. The navigation data provides the means for the receiver to determine the location of the satellite at the time of signal transmission, whereas the ranging code enables the user's receiver to determine the transit (i.e., propagation) time of the signal and thereby determine the satellite-to-user range. This technique requires that the user receiver also contain a clock. Utilizing this technique to measure the receiver's three-dimensional location requires that TOA ranging measurements be made by satellites. If the receiver clock were synchronized with the satellite clocks, only three range measurements would be required. However, a crystal clock is usually employed in navigation receivers to minimize the cost, complexity, and size of the receiver. Thus, four measurements are required to determine user latitude, longitude, height, and receiver clock offset from internal system time.

GPS architecture is comprised of three segments:

- Space segment - The space segment consists of a nominal constellation of 24 operating satellites that transmits one-way signals that gives the current GPS satellite position and time.

- Control segment - The control segment is composed of a master control station, a network of monitor stations which upload the clock and orbit errors, as well as the navigation data message to the GPS satellites.

- User segment - User segment consists of the GPS receiver equipment, which receives the signals from GPS satellites and uses the three dimensional position and time.

2.1.2 Working of GPS

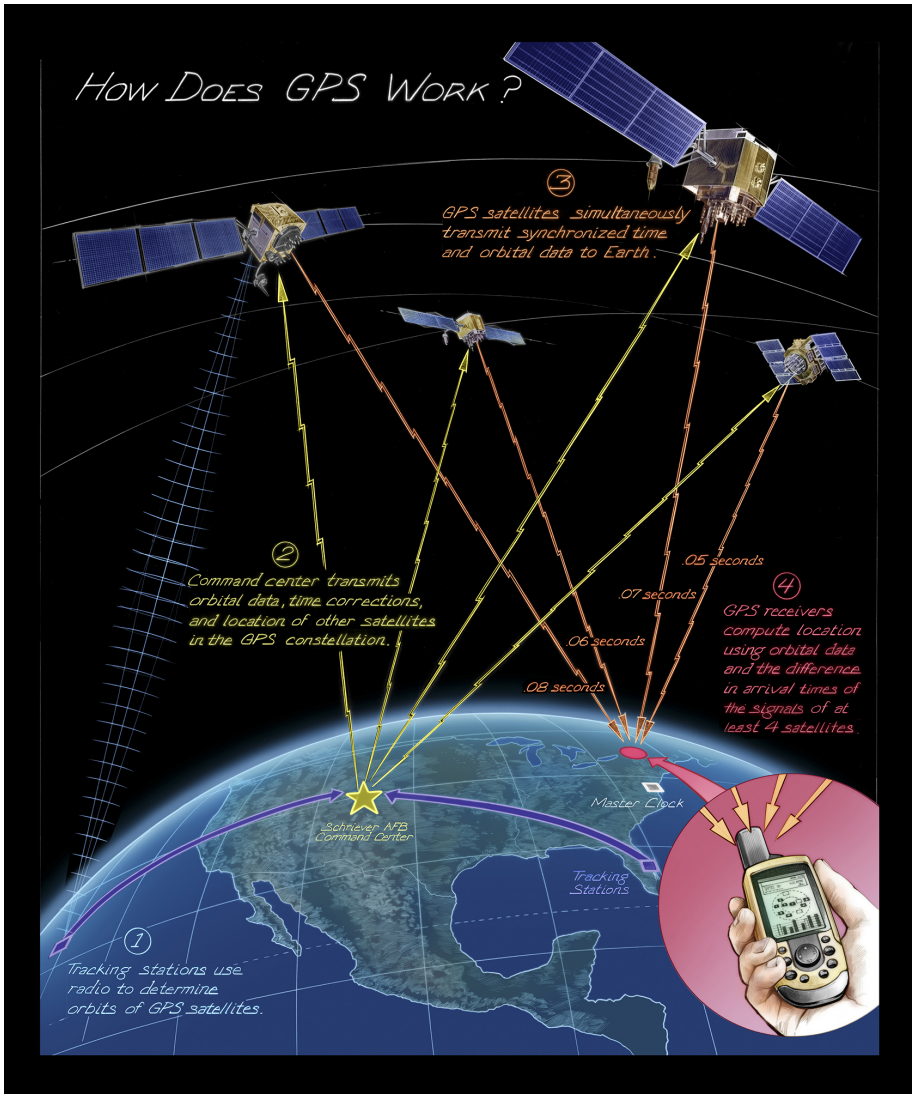

Figure 2.1: GPS [1]

GPS satellites are orbiting above the earth at an altitude of 11,000 miles. The orbits and position of satellites are known in advance. These satellites transmit 3-bits of information to the GPS receiver which includes Satellite number, Satellite position in space and Time at which the information is sent. Mostly, four nearest GPS satellites send information to the GPS receiver. GPS satellites uses the method of Trilateration [10] to find out the receivers position. Data from a single satellite narrows position down to a large area of the earth's surface. Adding data from second satellite narrows position down to the region where two

spheres overlap. Adding data from third satellite provides relatively accurate position. Data from fourth satellite enhances precision and also the ability to determine accurate elevation. A GPS receiver is composed of an Antenna, a Receiver processor, a highly stable clock and a display for showing location and speed information. GPS Receiver performs the following tasks.

- Selecting one or more satellites
- Acquiring GPS signals
- Measuring and tracking
- Recovering navigation data

2.1.3 GPS Services & Applications

GPS provides two types of service:

- Civilian Service
- Military Service

The civilian service is freely available to all users on a continuous worldwide basis. The military service is available to U.S and allied armed forces as well as approved government agencies.

GPS comes with a vast range of applications:

- Safety cameras
- Scientific experiments
- Traffic jams resolution

- Entertainment
- Outdoor activities

The most important applications of GPS are positioning and navigation. GPS tracking takes the normal functions of a GPS device a step further, by capturing and storing position data in the internal memory for later retrieval or by transmitting the location data in real time via the cellular data networks used by mobile phones. A vehicle can be tracked if it has a GPS device in it. The location information is sent by the GPS device via cell tower to mobile phone. GPS data is displayed in different message formats over a serial interface. All GPS receivers generally output NMEA data. The NMEA standard is formatted in lines of data called sentences [11]. Each sentence contains various bits of data organized in comma delimited format. The GPS data contains information showing the time stamp, latitude, longitude, number of satellite seen and the altitude for each ping in comma delimited format. The outstanding performance of GPS over many years has earned the confidence of millions of users worldwide. It has proven its dependability in the past and promises to be beneficial to users throughout the world in future.

2.2 Clustering

Clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than those in other groups [12] . These groups of similar objects formed are known as clusters. Clustering is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics. Popular notions of clusters include groups with small distances between cluster members, dense areas of the data space, inter-

vals or particular statistical distributions. The appropriate clustering algorithm and parameter settings (including parameters such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and error. It is often necessary to perform data preprocessing and modify model parameters until the result achieves the desired properties.

Different cluster models have been developed and for each of these cluster models, different algorithms can be given. The notion of a cluster, as found by different algorithms, varies significantly in its properties. Understanding these cluster models is key to understand the differences between the various algorithms. Here we explain some major cluster models.

2.2.1 Grid-based Algorithms

The grid-based clustering approach differs from the conventional clustering algorithms in a way that it is concerned not with the data points but with the value space that surrounds the data points. This is the approach in which we quantize space into a finite number of cells that form a grid structure on which all of the operations for clustering is performed. So, having a set of records and we want to cluster the data with respect to some attributes, we divide the related space (plane) into a grid structure and find the clusters [13]. In general, a typical grid-based clustering algorithm consists of the following five basic steps [14] :

- Creating the grid structure, i.e., partitioning the data space into a finite number of cells
- Calculating the cell density for each cell
- Sorting the cells according to their densities

- Identifying cluster centers
- Traversal of neighbor cells

The following are some techniques that are used to perform Grid-Based Clustering:

- CLIQUE (CLustering In QUest)
- STING (STatistical Information Grid)
- WaveCluster

STING is used for performing clustering on spatial data. Spatial data may be thought of as features located on or referenced to the Earth's surface. Its main benefit is that it processes many common region oriented queries on a set of points efficiently [15]. It clusters the records that are in a spatial table in terms of location. Placement of a record in a grid cell is completely determined by its physical location. The spatial area is divided into rectangular cells (Using latitude and longitude). Each cell forms a hierarchical structure. This means that each cell at a higher level is further partitioned into smaller cells in the lower level. The computational complexity is O(k) where k is the number of grid cells at the lowest level. Usually $k << N$, where N is the number of records. STING is a query independent approach, since statistical information exists independently of queries.

CLIQUE is a density and grid based subspace clustering algorithm that discovers the clusters by taking density threshold and number of grids as input parameters. CLIQUE operates on multidimensional data not by operating all the dimensions at once but by processing a single dimension at first step and then grows upward to the higher one [16]. It could generate overlapping clusters within a subspace with one data point belonging to more than one cluster.

WaveCluster is a novel clustering approach based on wavelet transforms which uses multiresolution property of wavelet transforms to effectively identify arbitrary shape clusters at different degrees of accuracy [17].

2.2.2 Centroid-based Algorithms

In centroid-based clustering [12] , clusters are represented by a central vector, which may not necessarily be a member of the data set. Basically, the similarity of two clusters is defined as the similarity of their centroids. One of the famous centroid based clustering algorithm is K-means algorithm.

K-means clustering aims at partitioning n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. The inputs to the algorithm are the number of clusters and the data set. The data set is a collection of features for each data point. The number of clusters K can either be randomly generated or randomly selected from the data set. The algorithm works as a two-step approach and it iterates between these two steps.

- Data assignment step: Firstly, based on the user input of the number of clusters K, the algorithm will place K centroids c_1, ..,c_k at random locations. Each data point is then assigned to its nearest centroid, based on the squared Euclidean distance.

- Centroid update step: In this step, the centroids are recomputed. This is done by taking the mean of all data points assigned to that centroid's cluster and this mean point in the cluster will become a new centroid. Based on the new centroid recomputed, again check all points to see which cluster they are near to based on Euclidean distance and assign the points to the nearest cluster.

The algorithm iterates between steps one and two until a stopping criteria is met i.e. no data points change clusters, the sum of the distances is minimized, or some maximum number of iterations is reached. The k-means algorithm is known to have a time complexity of $O(n^2)$, where n is the input data size. Most k-means-type algorithms require the number

of clusters k to be specified in advance, which is considered to be one of the biggest drawbacks of these algorithms. Also it does not work well with clusters of different size and different density, and it fails to detect outliers since it considers all the data points to form a cluster.

2.2.3 Density-based Algorithms

In density-based clustering, clusters are defined as areas of higher density than the remainder of the data set. Objects in the sparse areas that require separate clusters are usually considered to be noise and border points.

The most popular density based clustering method is DBSCAN. Density-based spatial clustering of applications with noise (DBSCAN) [18] is a data clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jrg Sander and Xiaowei Xu in 1996. Given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers those points in low-density regions.

The DBSCAN algorithm basically requires 2 parameters:

- Eps: the minimum distance between two points. It means that if the distance between two points is lower than or equal to this value (eps), these points are considered neighbors. In other words, it defines the radius n of neighborhood around a point p.

- MinPoints: the minimum number of neighbors with eps radius to form a dense region. For example, if we set the minPoints parameter to be 5, we need at least 5 points to form a cluster.

Consider a set of points in some space to be clustered. For the purpose of DBSCAN clustering, the points are classified as core points, density reachable points and outliers, as follows [19] :

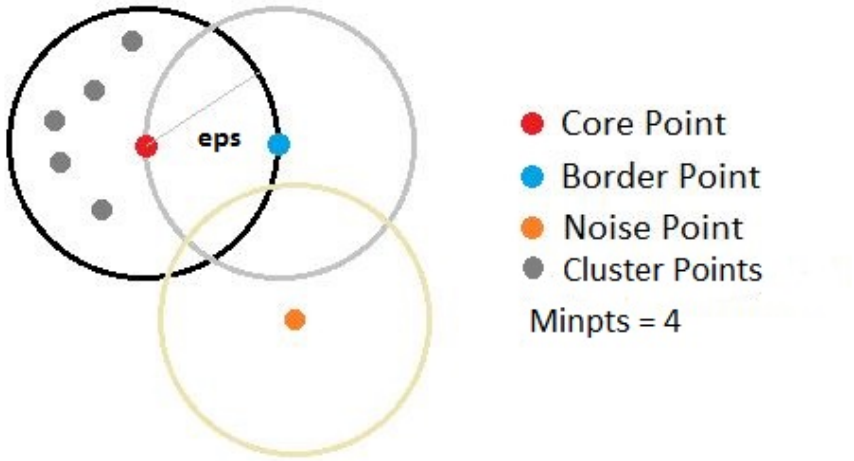

Figure 2.2: DBSCAN [2]

- A point p is a core point if at least minPts points are within eps distance (eps is the maximum radius of the neighborhood from p) of it (including p). Those points are said to be directly reachable from p.
- A point q is directly reachable from p if point q is within distance from point p and p must be a core point.
- A point q is reachable from p if there is a path p_1, .., p_n with p1 = p and p_n = q, where each $p_i + 1$ is directly reachable from p_i (all the points on the path must be core points, with the possible exception of q).
- All points not reachable from any other point are outliers.

Pseudo-code of DBSCAN Algorithm

Abstract DBSCAN Algorithm is described below:

- Find the (eps) neighbors of every point, and identify the core points with more than minPts neighbors

```
DBSCAN (D, eps, MinPts)
   C = 0
   for each unvisited point P in dataset D
         mark P as visited
         N = getNeighbors (P, eps)
         if sizeof(N) < MinPts
               mark P as NOISE
         else
               C = next cluster
               expandCluster(P, N, C, eps, MinPts)

expandCluster(P, N, C, eps, MinPts)
add P to cluster C
for each point P' in N
         if P' is not visited
         mark P' as visited
         N' = getNeighbors (P', eps)
         if sizeof (N') >= MinPts
               N = N joined with N'
         if P' is not yet member of any cluster
               add P' to cluster C
```

Figure 2.3: Traditional DBSCAN algorithm

- Find the connected components of core points on the neighbor graph, ignoring all non-core points
- Assign each non-core point to a nearby cluster if the cluster is an (eps) neighbor, otherwise assign it to noise

DBSCAN visits each point of the database, possibly multiple times (e.g., as candidates to different clusters). For practical considerations, however, the time complexity is mostly governed by the number of invocations to the distance calculation function. DBSCAN executes exactly one such query for each point, and if an indexing structure is used that executes a neighborhood query in O(log n), an overall average runtime complexity of O(n

log n) is obtained.

Advantages of DBSCAN

- Does not require a-priori specification of the number of clusters
- Able to identify noise data while clustering
- Able to find clusters of arbitrarily sizes and arbitrarily shapes

Disadvantages of DBSCAN

- DBSCAN algorithm fails in case of varying density clusters
- If the data and the scale are not well understood, choosing a meaningful distance threshold eps and minpts can be difficult

Applications of Clustering

- Medicine, Computational biology and bio-informatics: Medical imaging, Human genetic clustering, Sequence analysis
- Business and marketing: Market research that includes market segmentation, Product positioning and customer surveys
- World wide web: Social network analysis and Search result grouping
- Computer science: Image segmentation, Recommendation systems, Anomaly detection, Natural language processing
- Social science: Crime analysis and Educational data mining

2.3 Classification

Classification is a process of categorizing where objects are recognized and differentiated into the set of categories. In machine learning and statistics, supervised classification is a learning problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations whose category membership is known. Some of the well-known examples of classification algorithms include:

- Logistic Regression: It measures the relationship between a dependent variable and one or more independent variables by estimating probabilities using a logistic function. The model itself simply models probability of output in terms of input and it can be used to make a classifier, for instance by choosing a cut-off value and classifying inputs with probability greater than the cut-off as one class, below the cutoff as the other [20]. Logistic regression is used in various fields, including machine learning, most medical fields, and social sciences. Logistic regression may be used to predict the risk of developing a given disease (e.g. diabetes; coronary heart disease), based on observed characteristics of the patient. The technique can also be used in engineering, especially for predicting the probability of failure of a given process, system or product. It is also used in marketing applications such as prediction of a customer's propensity to purchase a product.

- Naive Bayes classifier: Naive Bayes is a classification technique based on Bayes Theorem with an assumption of independence among predictors [21]. In probability theory and statistics, Bayes theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Even if these features depend on each other or upon the existence of the other

features, all of these properties independently contribute to the probability [21]. It works well in many real-world situations such as document classification and spam filtering. It is easy to build and particularly useful for very large data sets.

- Support vector machines: Support vector machine is a representation of the training data as points in space separated into categories by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. It is effective in high dimensional spaces. It uses a subset of training points in the decision function so it is also memory efficient. In the case of support vector machines, a data point is viewed as a p-dimensional vector (a list of numbers), and we want to know whether we can separate such points with a (p-1) dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier.

- Random forest: Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement. This classifier works better in reducing over-fitting and it is more accurate than decision trees in most cases.

- K-nearest neighbor: The k-nearest-neighbors algorithm is a supervised classi-

fication algorithm. It takes a bunch of labeled points and uses them to learn how to label other points. To label a new point, it looks at the labeled closest points or the nearest k neighbors to that new point and has those neighbors vote, so whichever label the most of the neighbors have is the label for the new point [21]. This algorithm is simple to implement, robust to noisy training data, and effective if the training data is large.

Chapter 3

LITERATURE REVIEW

This chapter gives a brief overview of the research work carried out for processing GPS data and mining useful information from it. GPS technology is widely used in trucking companies for the purpose of fleet tracking. GPS data obtained from these trucking companies are in form of raw pings which basically includes details like Carrier ID, Truck ID, Latitude and Longitude Points and Time of each ping. To mine useful information out of these raw GPS pings, involves the usage of Big Data in freight transportation. A lot of research has been done in this area to process this GPS data and other useful information involved in freight transportation.

In this section, we briefly introduce different clustering techniques used to extract activity stops. Various types of clustering methods have been developed and different techniques are used as per different application needs. Moreover, which clustering techniques to select highly depends on the nature of the dataset. DBSCAN is the most popular among all density based clustering algorithms for geographical spatial data. Different versions and modifications to DBSCAN have been implemented to find activity stop locations. Moreover, graph based clustering is also widely used to discover trajectory clusters. In the following, we discuss related work used to find activity locations with different approaches.

3.1 Related works on the identification of stop locations

Extraction of stop locations in GPS trajectory is one of the most popular problem in mining geographical data. Previous studies covers variety of methods to deal with the stop identification task. To accomplish this task, two general approaches were used: static approach and dynamic approach. In the static approach [22, 23], all the important and popular lo-

cations that could possibly be the stop locations are predefined. So when extracting stops from trajectories, if a vehicle enters into a predefined location and the stay duration exceeds the duration threshold, this previously defined region is regarded as a stop location in the trajectory. The main drawback of static algorithms is that users need to specify their respective places of interest. As a result, this approach will fail to discover some of the additional and unknown interesting locations if they are not provided by users beforehand. Also this approach is quite difficult to apply on big datasets including millions of GPS records as it is not practical to define and cover all the important locations for the complete dataset.

In the dynamic approach, the user does not need to have a prior knowledge regarding the stop locations. Several classical clustering algorithms are introduced to extract stops from a trajectory under this approach. Considering only the spatial characteristics of the trajectory data, previous work has included the implementation of clustering techniques such as variation of traditional K-Means method [24] in order to detect stop locations. The selection of the value of parameter K and the initial clustering center is the main drawback because it heavily affects the results. Modified DBSCAN algorithm named DJ-Cluster (density and join-based clustering algorithm) [25], is proposed to detect personal meaningful places. These density-based clustering algorithms only take spatial dimensions into consideration and the temporal sequential features are ignored. Several studies have also considered temporal information along with the spatial one. Different DBSCAN enhancements with temporal sequential characteristic have been considered, and adopted by many researchers in order to extract stop positions [25–29]. An improved DBSCAN algorithm with gap treatment was proposed in [26] to detect stop episodes in a trajectory. The CB-SMoT (clustering-based stops and moves of trajectories) [27] algorithm was proposed to extract known and unknown stops where clusters are generated by evaluating trajectory sample points at a slower speed than the velocity threshold. In addition to the velocity threshold, distance threshold (eps) is obtained using a quantile function. The method pro-

posed in [28] improves the CB-SMoT algorithm by proposing an alternative for calculating the Eps parameter, but it is still difficult to calculate as it depends on users to distinguish the low speed part and high speed part. Additionally, by assigning different thresholds to different characteristics, some clustering approaches have been proposed [1821]. A two-step clustering technique to discover most frequently visited locations from the GPS trajectory data is introduced in the TDBC (a spatio-temporal clustering method used to extract stop points from individual trajectory) algorithm [21]. Additionally, a time-based clustering algorithm [30] was proposed considering both the clustering distance threshold and the time threshold.

Lei Gong et al. [31] used a two-step method for identifying activity stop locations. In the first step, modified version of DBSCAN algorithm is used to identify stop points and move points. In the second step, one of the machine learning algorithm called support vector machines (SVMs) method is used to distinguish between activity stops and non-activity stops from the identified stop points. Improved DBSCAN algorithm, named C-DBSCAN, is introduced in this paper. This improved algorithm works with two constraints that basically takes into account the time sequence constraint and a direction change constraint. These constraints are used to avoid errors due to moving points or points representing movement along a straight road at low speed. The first constraint says that all points in a cluster should be temporally sequential. So if the points are separated in a sudden, the cluster will be divided into two clusters at the point of sudden increase of distance and each one will be tested to see if it satisfies the condition of minimum number of points in one cluster. The second constraint says that the percentage of abnormal points in a cluster should not exceed a given threshold value. Pseudocode for C-DBSCAN algorithm is explained in detail. It requires five input parameters i.e. Trajectory data (T), neighborhood of core points (eps), minimum number of points in a cluster (Minpts), threshold percentage of abnormal points and direction change coefficient. Output of this algorithm are the stop points and the

moving points. In the second step, once the stop points and moving points are discovered, SVM is used to classify between activity and non-activity stops considering features like stop duration, mean distance to the centroid of a cluster of points at a stop location. This proposed algorithm was applied on GPS data collected using mobile phones in the Nagoya area of Japan in 2008. Experimental results were carried out to compare the results of the proposed algorithm with the traditional one with respect to four different indexes as described in the paper. Improved DBSCAN algorithm (C-DBSCAN) achieves an accuracy of 90% in identifying stop locations and the SVMs method is almost 96% accurate in distinguishing activity stops from non-activity stops. One of the drawback of this approach is that the dataset used in this paper does not include traffic congestion, hence it may be considered as an activity stop and current methods and attributes used in SVMs may not handle it well.

Zhongliang Fu et al. [30] proposed a two-step clustering technique to discover most frequently visited locations from the GPS trajectory data. In the first step, they extract stop points using spatio-temporal clustering algorithm based on time and distance. The second step involves applying improved clustering algorithm based on a fast search and identification of density peaks to discover the trajectory locations. Pseudocode of the novel spatio-temporal clustering technique based on time and distance, which is abbreviated to TDBC, is presented in this paper. The algorithm takes single trajectory T, time threshold dt and distance threshold dd as input conditions. It finds clusters which represent stop points. Furthermore, it also uses a function that checks the relationship between the current cluster and the previous cluster and process the cluster either by merging them or considering a new stop point based on this function check. In the second step, improved Clustering by Fast Search is applied to the discovered stop points to extract trajectory locations based on density peak. Steps has been explained in the paper to find the density peaks for finding locations. The proposed algorithm was tested on the GPS pings collected from the

data collection application installed in the smart phones by 30 volunteers. Three different datasets were tested for the number of stop points found for different distance and time thresholds and presented as graph representation in this paper. The proposed TDBC was compared with the other four algorithms, K-Medoids, DJ-Cluster [25], CB-SMoT [27] and Time-Based Clustering [32]. Summary table of the results obtained from these algorithms is represented with respect to time complexity, precision and recall. Results from the table shows that k-medoids, DJ-Cluster and CB-SMoT are more time-consuming than the other two algorithms due to their complex computation. The efficiency of the TDBC is similar to the Time-Based and both have high clustering efficiency. In terms of precision, the TDBC and DJ-Cluster are over 0.8, which is significantly larger than the other three algorithms. In terms of recall, compared with the others, the TDBC is slightly improved. However, the proposed approach requires some prior parameters, and manual intervention is necessary when the cluster centers are selected.

W. Chen et al. [33] proposed a modified density-based clustering algorithm, named T-DBSCAN, by considering the time sequential characteristics of the GPS points along a trajectory. Three formal terms called Trajectory, Stop, and Move used in T-DBSCAN are defined below:

A trajectory is the user-defined record of the evolution of the position of an object that is moving in space during a given time interval in order to reach a given destination.

trajectory: [tbegin, tend] $->$ space

A stop is a part of a trajectory, such that (1) The user has explicitly defined this part to represent a stop, (2) The temporal extent [tbeginstopx, tendstopx] of this part is a non-empty time interval, and (3) The traveling object does not move, i.e. the spatial range of the trajectory for the interval is a single point. (4) All stops are temporally disjoint, i.e. their temporal extents are always disjoint.

A move is a part of a trajectory, such that (1) The part is delimited by two extremities that

represent either two consecutive stops, or tbegin and the first stop, or the last stop and tend, or tbegin and tend. (2) The temporal extent [tbeginmovex, tendmovex] is a non-empty time interval, and (3) The spatial range of the trajectory for interval [tbeginmovex, tendmovex] is a spatiotemporal polyline defined by the trajectory function.

Four input parameters are required for T-DBSCAN: D is the set of points comprising the trajectory; Eps is an inner radius for identifying density-based neighborhood; C_{Eps} is an outer radius for limiting the density searching range; Eps is the search radius; and MinPts is the minimum number of neighboring points to identify a core point. Pseudocode of T-DBSCAN algorithm has been defined in this paper. Two tests were performed to provide a comparative analysis between DBSCAN and T-DBSCAN in terms of segmentation accuracy and computation efficiency. First test indicated that T-DBSCAN was significantly faster than DBSCAN in segmenting the trajectories at all data levels. Specifically, TDBSCAN took only 6.25% of DBSCAN's time to process up to 8000 points, and this ratio decreased to 5.72% and further 4.34% when 8000˜20000 points and 20000˜30000 points were processed, respectively. The second test involved comparison of accuracy between the two methods. Compared to the real stops from field verification, all T-DBSCAN deduced clusters had correct match except those clusters that resulted from a traffic jam. In comparison, serious overlapping occurred between many pairs of clusters identified with DBSCAN.

Benoit Thierry et al. [34] proposed a novel kernel-based activity location detection algorithm in comparison with the classical detection method based on distance and time thresholds [35], [36], [37]. This novel algorithm detects (i) known activity locations and (ii) time spent at a given location, depending on algorithm bandwidth value, GPS noise level and actual stop duration. Kernel density estimation is a non-parametric method where a symmetrical kernel function is first superimposed over each event. The set of overlapping functions is then summed to create a continuous density surface. Kernel densities are

frequently used for point pattern analysis and hotspot exploration. The proposed algorithm runs globally by calculating a kernel density surface [38] instead of grouping spatially nearby points. This creates a smoothed surface corresponding to the probability density function of 2D points. This smoothed surface is controlled by the bandwidth. The peaks of this surface determines the candidates for actual stops. Explanation of this kernel-density algorithm (Akd) and of the classical fixed threshold algorithm (Aft) are presented in this paper. Experiments were conducted on randomly generated GPS tracks and results show that the proposed algorithm outperforms the fixed threshold algorithm for almost all indicators, correctly identifying the three artificially generated stops with varying duration and noise levels. Similarly, although Aft had the best spatial accuracy with smaller bandwidths and for the lowest noise levels, Akd succeeded in maintaining a better overall accuracy across all bandwidths and noise categories.

All the techniques seen so far in the related work, need to consider reasonable threshold values for both the distance and time parameter. Also, while calculating the density of GPS points, most clustering-based algorithms take the number of GPS points within a given distance into account, without considering their sequential characteristics. The concept of move ability [39] is introduced where the density of GPS points will be calculated using the adjacent points over the trajectory, not the overall spatial points. In [39], Kun Fu et al. introduced a novel approach of move-ability and hybrid feature based density clustering which considers temporal and spatial properties to find stop points in GPS trajectory data. Various definitions are described in this paper for defining the concept of move-ability and a formula for calculating it. In the first step, move-ability is calculated for each points in the trajectory. Basically, stop points in a trajectory should have lower move-ability and higher density of GPS points. Hence in the second step, an improved DBSCAN is applied which calculates density to define a core point. Experiments were conducted on Geolife dataset containing total of 17,621 trajectories. Each trajectory in this dataset consists of

a sequence of temporal, ordered, time-stamped points; each point contains geographical coordinate information, such as longitude, latitude and altitude. To validate the proposed technique, results were compared to four other stop-detection algorithms: the CB-SMoT algorithm [27], DBSCAN algorithm, DJ-Cluster algorithm [25], and time-based clustering [32]. Results show that this novel approach is more robust to fake stops which occurs due to congestion owing the concept of move-ability.

Kevin Gingerich et al. [3] applied an alternative approach where both the distance and time measures were used, also considering the sequential characteristics of GPS pings for identifying stop locations. So, in order to find the truck stop locations, the GPS pings for a given truck is sorted sequentially according to the registered time stamp and the location of a first ping is compared to the location of the next ping. If the distance between two consecutive pings is less than a certain distance threshold, the dwell time is set equal to the elapsed time between the two pings. If the distance of the third ping from the first ping is also less than the threshold value, the dwell time continues to accumulate. The elapsed time of all these pings will keep on adding up to dwell time until there is a ping located outside the buffer threshold distance at which point the dwell time is reset. A reasonable distance threshold with radius of 250 m was selected in a way to avoid cutting a stop short if a vehicle moved a limited range within a given property and also to avoid the spatial errors that might arise due to bad GPS readings. To investigate the potential of false positives in the data, an area containing Highway 401 road links between Highway 407 and Highway 403/410 in Ontario, Canada (latitude between 43.588 and 43.638; longitude between -79.819 and -79.661) was considered. This area was selected since it has the largest concentration of GPS pings in the dataset and occurs along a heavily congested highway corridor in the Toronto metropolitan area in Ontario. Only 48 out of 32,174 stop events in the examined area are false positive stop events suggesting that the potential for obtaining erroneous results are kept to a minimum with the proposed techniques.

All the previous study considered most of the characteristics of geographical data such as distance threshold, time threshold, direction change, adjacent sequential data, and group of whole data for clustering to extract stop points from the GPS data points. Also, the novel approach of move ability applied in [39] shows good results in finding out stop locations. Other works considering distance and time measures in [3] along with the entropy concept to process real-time individual truck pings in sequential manner with the distance and time threshold parameter also generated quite good results and were validated manually to check its correctness. This approach is proved feasible for large volume of data by evaluating the pattern that emerges from analyzing stop events over space. Hence, we apply the same technique to process the GPS data points carrying the same nature of the dataset.

3.2 Related works on the variations and enhancements of the clustering techniques

Clustering techniques considering varied shape, size and density of the clusters

Adriano Moreira et al. [40] described implementation of two density based clustering algorithms: DBSCAN [41] and SNN [42] to identify clusters from geographical data based on their spatial density to characterize geographic regions. This paper mainly introduces working of DBSCAN and SNN algorithm which also includes discussion on how the values for the input parameters to be selected for these algorithms. These algorithms were implemented in Visual Basic 6.0. It also compares the cluster results obtained from both algorithms. Although DBSCAN can find clusters with different shape and size, it fails to find clusters with different density. On the other hand, SNN algorithm can find clusters of different densities. Two different approaches of the SNN algorithms were implemented. The first approach is a Core Approach. It creates the clusters around the core points. The second approach is a Graph Approach in which clusters are identified by the points that are connected in a graph. The graph is constructed by linking all the points whose similar-

ity is higher than the Eps value. The results obtained through the use of these algorithms show that SNN performs better than DBSCAN since it can detect clusters with different densities while DBSCAN cannot. The only drawback here is that both the density based algorithms require two parameters Eps and MinPts to be inputted manually and the cluster results changes greatly with the change in the parameter values inputted to the algorithm.

Mohammed T. H. Elbatta et al. [43] proposed an enhancement of DBSCAN algorithm called Dynamic Method DBSCAN (DMDBSCAN) that has the ability to detect the clusters of different shapes, sizes that differ in local density. DMDBSCAN uses dynamic method to find suitable value of Eps for different density level in the data set instead of using global Eps value. Distance from point is calculated to its kth nearest neighbor, which is defined as k-dist. These k-dists are computed for all data points for some k value inputted by user, and a graph is plotted for this value of k-dists sorted in ascending order. The sharp curve in the graph corresponds to suitable value of Eps for each density level of data set. Lastly, DBSCAN is applied for each Eps value to find the clusters with different density. Experiments were conducted on three artificial two-dimensional data sets as well as three real dataset and results obtained using proposed approach were compared with traditional DBSCAN and DVBSCAN with different parameter values. Also, a table showing the results across average error index and the number of generated clusters for each algorithm is presented in this paper. Results show that the proposed algorithm DMDBSCAN gives more stable estimates of the number of clusters than existing DBSCAN or DVBSCAN over many different types of data of different shapes and sizes.

Peng Liu et al. [44] implemented new variation of DBSCAN algorithm called VDBSCAN to get clusters from varied-density data. The basic idea of this algorithm is to determine different values of eps parameter for different density of the data points instead of single global eps value as used in traditional DBSCAN for the purpose of generating varied density-based clusters. The approach used to determine the parameters Eps and MinPts is

to look at the behavior of the distance from a point to its kth nearest neighbor, which is called k-dist. The k-dists are computed for all the data points for some k, sorted in ascending order, and then plotted using the sorted values. As a result, a sharp change is expected to see. The sharp change at the value of k-dist corresponds to a suitable value of Eps. This value of Eps that is determined in this way depends on k, but does not change dramatically as k changes. VDBSCAN is implemented with 2 steps. In the first step, VDBSCAN calculates and stores k-dist for each point and partition k-dist plots. The number of densities is determined by k-dist plot so it selects Eps_i parameter value for each density. In the second step, DBSCAN algorithm runs for each Eps_i (i=1,2,3,.....,n. n is the number of density levels). Eps_i have been ordered as k-dist line curves, that is Epsi $<$ Eps_i+1 (i$<$n). Before performing DBSCAN for Eps_i+1, it marks the points in clusters corresponding with Eps_i as C_i - t (t is a natural number), which indicates that the points belong to the cluster t in density level i. Marked points will not be processed by DBSCAN again. Non-marked points after all the Eps_i process are recognized as outliers. The experimental results show that VDBSCAN generates different clusters with different density, while it takes the same time complexity as DBSCAN.

EDBSCAN (An Enhanced Density Based Spatial Clustering of Application with Noise) algorithm [45] is another extension of DBSCAN which clusters the data points with varying densities effectively. It keeps tracks of density variation which exists within the cluster. It calculates the density variance of a core object with respect to its -neighborhood. If the density variance of a core object is less than or equal to a threshold value and also satisfying the homogeneity index with respect to its neighborhood, then it will allow the core object for expansion. It calculates the density variance and homogeneity index locally in the Eps neighborhood of a core object. Steps for the proposed algorithm are described in detail. The idea is to use varied values for Eps according to the local density of the starting point in each cluster. The clustering process starts from the highest local density point towards

the lowest local density one. For each value of Eps, DBSCAN is adopted to make sure that all density reachable points with respect to current Eps are clustered. At the next process, the clustered points are ignored, to avoid merging among denser clusters with sparser ones. The proposed algorithm starts by first finding the k-nearest neighbors for each point in given dataset as DBSCAN does, but here the enhanced algorithm does not build the sorted k-dist graph. Based on the k-nearest neighbors, a local density function is used to find the local density at each point. The overall density of the data space can be calculated as the sum of the influence functions of all data points. The influence function can be seen as a function which describes the impact of a data point within its neighborhood, and it is applied to each data point. The main steps of the proposed algorithm are described in detail in the paper. The time required for a neighborhood query is O (log n) by using a spatial access method such as R*-tree. The proposed method arranges the points according to their local densities using quick sort, which requires O (n log n). Hence the overall run time complexity is O (n log n) which is the same as that of DBSCAN. The proposed algorithm was implemented in C++. Eight different data sets containing 2D points with different geometric shapes were used to test the proposed technique. Results show that the proposed approach is efficient enough as it discovers correct clusters.

M. Venkata Sowjanya and T. Maruthi Padmaja [46] proposed an algorithm which is a variation of DENGRAPH (Density Based Graph Clustering) called varied Density DENGRAPH. It overcomes the drawback of original DENGRAPH that fails to find varied density cluster for a given network graph by finding different Eps value for different density using the method of plotting graph between k-dist (calculated by finding distance to kth nearest neighbour) and data points sorted by distance to kth nearest neighbiour. For any new points dynamically getting added to the network, the proposed approach uses incremental updates as described in the paper. Experiments were performed on the dataset collected from Facebook network that contains 46,952 nodes and 876,993 edges among them. Clus-

ter Results were compared to both DENGRAPH and the proposed DENGRAPH method with different Eps values. Varied Density DENGRAPH algorithm identifies better density connected compact communities when compared with DENGRAPH.

Algorithm named MKEIDBSCAN [47] which is based on EIDBSCAN [48] is proposed with multi density to discover the significant places. The idea to estimate the Eps radius for different density is to fix the radius value and change the MinPts value. For finding different minpts, k-mediods is used to find k center points in input data and plotting sorted Minpt-values in ascending order. Finding sharp change corresponds to find suitable value of Minpts for each density level. Experiments were performed on Geolife dataset collected from April 2007 to August 2012. A table showing clustering results using different algorithms is listed in the paper. Results shows that significant places are detected more accurately and the running time is reduced.

Majority of the potential methods and variations of DBSCAN algorithm listed above apply the concept of finding different radius threshold (eps) for different density level using the technique of k-dist plot of the distance sorted based on the k-nearest neighbors. This technique helps finding clusters with varied size and density. One of the variation of DBSCAN also developed a density function and homogeneity index that keeps track of density variance within the cluster and generating clusters within the bigger cluster. Other way of generating varied size and density clusters in mentioned in the previous work is by fixing the radius threshold globally and considering multiple values of minpts using the k-medoids to find suitable minpts for each density level. Although this work helps in generating the fine clusters with different size and density, the requirement of this research is to apply clustering to the stop points of each carrier individually and identify one such cluster as a firm out of the generated ones. This firm cluster is further identified using other features of the dataset, hence at this point, we do not require to generate varied density clusters. Hence the traditional DBSCAN with global eps and minpts parameter values is

sufficient for this part of the task.

DBSCAN Enhancements for the automatic generation of input parameters (Eps, Minpts)

Xiaowei Xu et al. [49] introduced new clustering algorithm called DBCLASD (Distribution Based Clustering of Large Spatial Databases). Unlike CLARANS, DBCLASD discovers clusters of arbitrary shape and unlike DBSCAN, it does not require any input parameters. This is in contrast to the clustering algorithm DBSCAN requiring two input parameters which may be difficult to provide for large databases. DBCLASD is an incremental algorithm, i.e. the assignment of a point to a cluster is based only on the points processed so far without considering the entire cluster or even the entire database. DBCLASD incrementally augments an initial cluster by its neighboring points as long as the nearest neighbor distance set of the resulting cluster fits the expected distance distribution. The set of candidates of a cluster is constructed using circle query with suitable radius m. When all candidates of the current cluster have been processed, the unsuccessful candidates of that cluster are considered again. DBCLASD will further merge neighboring clusters whenever a candidate generated is already assigned to some other cluster. The algorithm terminates when (i) all of the data points are selected at most once as a starting point for the generation of a new cluster and (ii) no more candidates exist: the unsuccessful candidates are not considered further if none of them fits the current cluster. The algorithm is explained with a pseudocode in this paper. DBCLASD is implemented in C++ using R*-tree and is applied on real database from an earthquake dataset of 40,000 km^2 region of the central coast ranges in California. Cluster results and the running time results were compared for this algorithm with DBSCAN and CLARANS. The run time of DBCLASD is roughly twice the run time of DBSCAN, while DBCLASD outperforms CLARANS by a factor of at least 60. DBCLASD algorithm detects 4 clusters corresponding to the 2 main seismic faults without requiring any input parameters. This paper concludes that DBCLASD also

works effectively on real databases with non-uniform data.

Xiaopeng Yu et al. [50] proposed a novel clustering implementation where an enhanced DBSCAN is implemented based on k-nearest neighbors (KNN) that merges KNN and DBSCAN together. Firstly, the window-width of each data point is determined and the entire data set is partitioned into some fuzzy cluster (FC) by the K" based on KDE. Furthermore, the local parameters, Eps and Minpts of each FC are determined according to the entropy formula. Finally, each local Eps is mapped to the global Eps, and each FC is separately clustered. Pseudocode of the proposed KNNDBSCAN algorithm is presented in this paper. The time complexity of KNNDBSCAN depends mainly on the calculation of KNN, local Eps and DBSCAN. Time complexity for all the three factors is nearly equal to O(n log n). Hence, the time complexity of KNNDBSCAN is O(n log n), which is the same as that of traditional DBSCAN. The quality of the resulting clusters is not sensitive to the parameter k. The proposed algorithm is robust, and has better quality and efficiency.

Erica Rosalina et al. [51] proposed a method to automatically estimate the input parameters required for two well known density based clustering algorithm called DBSCAN and HDBSCAN, avoiding the manual intervention for inputting the parameter values. In DBSCAN algorithm, for finding Eps value, it uses a proposed technique from [41] that determines sharp curve value obtained from the plot of sorted k-dist graph as Eps. Psuedocode for finding sharp curve value for Eps is presented in this paper along with the plot. For minpts, it uses the heuristic applied by [52] which suggests that the Minpts be defined by mpts = ln(N) which is the natural log of N, where N is the total number of points in dataset. However, instead of the total size of dataset, it uses total size of the visible points in the dataset based on the resolution. Similar approach used in DBSCAN for finding minpts is adopted in HDBSCAN with mpts = ln(N). The second parameter, the minimum cluster size (mclSize), is calculated using two approaches. The first one is the normal approach called HDBSCAN Normal, where mclSize = mpts. The second approach is mode

approach called HDBSCAN Mode wherein minimum cluster size is calculated based on the most frequent number of neighbours that are within knee of core distances. Pseudocode for mode approach is described in the paper. Experiments were performed on the road crash dataset taken from Victorias road network data, Australia from the period of 1 January 2006 through to 30 June 2013, containing 72176 accident nodes. They compare the best results of DBSCAN and the best results of HDBSCANs two approaches (HDBSCAN Normal & HDBSCAN Mode), and identify the best out of the three approaches by using the cluster indices and visualization for both low and high resolutions. Visualization shows that HDBSCAN Mode is better than HDBSCAN Normal in high resolution. HDBSCAN outperforms DBSCAN both in high and low resolutions. Although the performance of DBSCAN is average in the high resolution scenario, it may not be suitable for a low resolution application.

AGED [53] is one of the enhanced version of DBSCAN algorithm used for automatic generation of multiple Eps for different density levels that may exist in the dataset. The proposed method uses k-nearest neighbors to determine a suitable value of Eps for clustering. It finds the k-nearest neighbors for each point in given dataset and uses a local density function to find the local density at each point. The local density function is defined as the sum of the distances among the point x and its k-nearest neighbors. The average local density function gives the different local density values for each point. From these values, to automatically determine the values of Eps, min-max normalization is applied to transform the average local density function. Binning technique is applied to distribute the different values of average local density into the respective buckets. If any bucket contains more than k elements, then we take the average value of the bucket elements and apply the reverse of min-max normalization on the same. The value presented in each bucket indicates all possible Eps. Experiments have been carried out on two and multidimensional datasets of non-varied and varied density points. The value of Eps derived using AGED gives more

accuracy than the approximate value of Eps taken in DBSCAN. The clusters generated using the Eps value derived from AGED algorithm gives promising results in terms of other performance measures too. The graphical representation of the clusters generated using AGED and DBSCAN, for two dimensional datasets with various color coding has also been presented in the paper.

AutoEpsDBSCAN [54] is one of the variations of DBSCAN algorithm to get rid of the global eps parameter and determine different range of Eps value automatically to identify clusters with varied density. Firstly, it computes the average of the distances of every point to all k of its nearest neighbors [55], [56]. This is unlike VDBSCAN, where only the kth nearest neighbor is considered during the distance computation. Secondly, these averaged k-distances in an ascending order is plotted and knee or sharp curve value is observed which represents a change in density distribution amongst points. All such knee values corresponds to a set of Eps's for identifying all the clusters having different density distributions. DBSCAN is applied to each Eps_i value and the mark points in clusters corresponding to Eps_i as C_i are not processed while applying DBSCAN for Eps_i+1. Moreover, Minpts are calculated as the average of the number of points in Eps neighborhood of point i. The entire psudeocode for this proposed algorithm is presented in this paper. The algorithm was implemented in java and the experiment was conducted on the compound dataset obtained from research department of The University of EDINBURGH Schools of informatics. Experimental results show that the proposed algorithm gives better output compared to the existing DBSCAN algorithm.

A lot of study has come up with different solutions to one of the major drawback of DBSCAN clustering technique of determining the input parameters automatically avoiding the manual intervention. Since DBSCAN is very sensitive to its input parameters in generating clusters, it is quite important to reasonably decide the threshold for its input parameters. Popular approach used in the previous study involves finding different set of eps value at

different levels of density variation. This sharp density value is obtained from the plot of data points which are sorted based on k-dist of the neighbors. This K-dist is calculated using K-nearest neighbor distance method. For finding Minpts value automatically, we refer to the simple heuristic applied in [52] which suggests minpts value is safe to determine by taking the natural log of the size of the dataset. We consider the same approach of finding minpts automatically for our dataset. Since, we consider global eps for entire dataset, we have used the concept of finding suitable eps by calculating the distance of each point with all its neighbors and find the average distance to sort them and further filtering this sorted distance based on the spatial measure threshold. We set this spatial measure threshold for the eps parameter i.e. radius ranging between 100 and 500 metres. This range has been selected considering the actual physical region of the firm or industry which we believe should not be less than the diametric area covering 200 meters or larger than the diametric area covering 1000 metres. Hence, with an intention of identifying firm location, we filter the sorted distance within the area between 100 and 500 metres and we take average of this filtered distance list. This average distance value sets the eps radius that is automatically determined based on the data points.

Clustering techniques handling small clusters

S. Gayathri et al. [57] proposed a new Proclus clustering algorithm to overcome the drawback of traditional Proclus algorithm which ignores the cluster with small data points. The proposed technique here combines Proclus with Shared nearest neighbor (SNN) density based algorithm in order to cluster small data points in high dimensional data. Proclus algorithm stands for projected clustering which is a partitioned clustering algorithm based on the idea of k-medoids clustering. SNN algorithm considers the similarity among the points by looking at the number of the nearest neighbors that any two points share. The SNN density is calculated as the sum of those similarities of the nearest neighbors of a point. The proposed algorithm is explained in detail in this paper, starting with using Pro-

clus method followed by SNN algorithm to cluster the small data points which are not clustered using Proclus method. The experiment was tested on synthetic data sets. The Proclus method was tested using Elki cluster tool kit and SNN was tested in WEKA tool. The results are represented in terms of graphs which show that even small data points with sparse density are clustered using combined approach as proposed in this paper.

Previous study focused on generating small size clusters from the high dimensional data using the Proclus clustering algorithm combined with the SNN clustering algorithm. For firm identification task, we do not require to find smaller clusters with sparse density as we look for the dense clusters that are visited by large number of distinct trucks and more number of truck visits. However, for the second part where we cluster the stop points and classify them, we need to include the small clusters as well. In this research, we modify the DBSCAN algorithm by considering just the eps radius and ignoring the minpts while finding the neighbors for a point. This makes the sparse points to be a part of a cluster with fewer data points. Then in the second step of expanding the cluster with the neighboring points, we take into account the minpts parameter so that we can define accurate clusters by including the boundary points which are denser and excluding the sparse boundary points.

The summary of various studies showing DBSCAN enhancements and variations is listed in Table 3.1.

Graph based clustering techniques on the GPS dataset

Diansheng Guo et al. [58] proposed a graph-based approach that converts trajectory data to a graph-based representation and apply graph partitioning method to discover natural regions defined by trajectories and use the discovered regions to search and visualize trajectory clusters. Truck trajectory data with 276 trajectories and 112,203 GPS points has been used to evaluate this approach. First, GPS points are processed by removing redundant GPS locations to extract representative points of repeated GPS measures. Second, with the extracted representative points of GPS locations, each trajectory is interpolated

Author Name & Year	Method Name	Key Objective	Input Parameters *Eps & Minpts*
Adriano Moreira et al. [33], 2005	DBSCAN and SNN	Traditional implementation of DBSCAN [34] and SNN [35] on geographical data.	DBSCAN: User dependent and SNN: Computed automatically
Mohammed T. H. Elbatta et al. [36]	Dynamic Method DBSCAN (DMDBSCAN)	Dynamic method to find set of Eps for different density level	Computed automatically
Peng Liu et al. [37]	VDBSCAN	Clustering uneven dataset varied in density efficiently	Computed automatically
Fahim et al. [38]	EDBSCAN	Perform efficient clustering without requiring any user supplied input parameters using density variance and homogeneity index	Not Required
Xiaowei Xu et al. [42]	DBCLASD	Incremental algorithm to discover clusters of arbitrary shape	Not Required
Xiaopeng Yu et al. [43]	KNNDBSCAN	Offer DBSCAN to determine density threshold in an unsupervised way	Calculated automatically
S. Neha et al. [46]	AGED	automatic generation of multiple Eps for different density levels	Calculated automatically
M. N. Gaonkar et al. [47]	AutoEpsDBSCAN	determine different range of Eps value automatically to identify clusters with varied density	Calculated automatically
D. Birant et al. [45]	ST-DBSCAN	1.Discovering Cluster on spatial-temporal data, 2.Identification of adjacent Clusters. 3.Identification of Noise objects from cluster with different densities	Calculated automatically

Table 3.1: Summary on DBSCAN Enhancements

with a modified shortest distance measure. After these two steps, a graph is constructed, where representative points are nodes and a connection is added between a pair of nodes if they are on the same trajectory. This is a weighted graph and the weight for each pair of nodes (edge) is the total number of connections they have (i.e. the total number of trajectories that they share). A spatially constrained graph partitioning method [59] is then applied to find natural regions within the trajectories, where locations inside a region share more trajectories with each other than with locations in other regions.

Mohamed Khalil El Mahrsi et al. [60] presented a clustering approach to discover groups of road segments that are often travelled by the same trajectories. This paper models the interactions between segments with respect to their similarity as a weighted graph to ap-

ply a community detection algorithm to discover meaningful clusters. Firstly, a similarity measure is defined between road segments based on the comparison of the common trajectories that visited them. Based on this measure, a graph G is built (G = (V; E; W), where V = v_1; v_2;..; v_n are vertices, and edges in E are weighted with $w_{ij} >= 0$ and $w_{ij} = w_{ji}$) that depicts the relationships between different road segments. The graph is then partitioned using a modularity-based community detection algorithm in order to discover a hierarchy of nested segment clusters. The implementation of the hierarchical modularity-based clustering is described. The pseudo-code for the proposed algorithm is presented in this paper. Results of this modularity based clustering proposed here was compared with two other graph clustering techniques called spectral clustering and label propagation clustering. Performance of these three algorithms were tested on five synthetic datasets using the Old-enburg road network. The full data is composed of 6105 vertices and about 14070 road segments. Each dataset contains 100 trajectories visiting a various amount of road segments. Cluster results were compared and it was proved that modularity-based clustering is superior to other defined techniques.

Bita Shams and Saman Haratizadeh [61] presented a novel framework called GraphLoc that uses network community detection problem to discover significant locations. The first step to this approach is to calculate the co-location probability between two points considering the local structure around each one. This is calculated using symmetric relative distance between two points. The resulting probabilities are then used to represent data as a KNN graph reflecting useful information for location discovery. A KNN graph is defined as G_k = (V, E, W) in which, V is the set of the nodes representing the stay points. There is an edge in E, connecting two nodes if at least one of the two nodes is among the k-nearest neighbors of the other one. W defines the co-location probability of its corresponding nodes. The next step is to find significant locations from the graph i.e sub-graphs whose inside links are much stronger than their outside links. To find reasonably important

locations in a city, or their equivalent partitions in the KNN graph, an existing hierarchical agglomerative community detection algorithm proposed by Clauset et al. [62] was used. The pseudo-code of the algorithm, called Hierarchical Graph Location (HGL) is presented in this paper. All experiments are conducted on Geolife trajectory data-set collected by Microsoft Research Asia from April 2007 to October 2011. The list of interesting locations discovered by GraphLoc is listed in this paper. The results of GraphLoc were compared with those from the implementation of DBSCAN and grid clustering. GraphLoc achieves the best performance regarding prediction accuracy and Point-to-Point distance. Its accuracy is about 56%, with DBSCAN the accuracy varies from 45% to 50% based on the best values for DBSCANs parameters. The running times of the three algorithms are quite close, while the running time of GraphLoc is sometimes a little less than that of DBSCAN.

A graph based clustering is also one of the novel technique in clustering road network data. Previous work includes the concept of modularity based community detection and similarity measure approach for building similarity graph such that the nodes represents the stop points and the edges represents the strong similarity between the stop points. This graph concept can be applied to determine the firm location but is not relevant in clustering the data points for classification purpose. A firm can be identified as a node that has highest edge similarity indicating high amount of visits by the trucks. But we consider the simple DBSCAN approach to cluster the stop points and identify a firm cluster using various statistics that also includes the maximum distinct truck count along with the highest number of visits.

3.3 Related works on the classification of stop locations

Lei Gong et al. [31] considered three different features in GPS trajectories that are selected as input parameters for implementing SVMs in order to distinguish activity stops and non-activity stops. The first feature is the stop duration and the paper also presents the

distribution of stop duration for both of the activity and non-activity stops in the training data trajectories which shows that majority of the activity stops have a much longer duration than non-activity stops. A threshold of 105 s is taken, giving 92.5% of stops being accurately classified. Significant observation shows that almost 7.5% of activity stops have a stop duration from 30 to 105 s while 7.5% of non-activity stops have a stop duration from 105 to 170 s. The second feature is the mean distance of GPS points to the cluster centroid with the observation determined in this work that almost all the non-activity stops have an average distance from the centroid of less than 30 meters. While the activity stops have larger average distance from the centroid greater than 30 metres. The third feature is the shorter of the distances from the current location to home and to the workplace by activity stop and non-activity stop in three dimensional space.

Kevin Gingerich et al. [3] applied the concept of Entropy to mine the GPS data to classify the stopped truck points. These stop points were categorized into two types: (1) Primary stop where actually goods are transferred and (2) Secondary stop where a driver makes a stop either for rest, eating or for fuel refilling. The proposed entropy technique measures the diversity of truck carriers with trucks at all stop locations. The entropy to quantify the variety of carrier fleets for a particular stop location q is defined as the number of truck stop events occurring for a given carrier c at location q to the total number of truck stop events for all carrier at q. This means that a stop location that is visited by a larger number of truck belonging to few carriers, will have smaller entropy value and will have more credibility to be defined as a primary stop. A stop location that is visited by a smaller number of trucks belonging to a large number of varied carriers, will have larger entropy value and will likely to be defined as a secondary stop. The SQL clustering approach was also applied to identify the primary shipping depot for each carrier. In this paper, the authors have used the GPS records for the month of March, 2013 that includes a total of 40,650 individual trucks with approximately 101.6 million GPS pings. The Entropy method was implemented in SQL

using SQL clustering approach. Histogram of entropy results for all clusters of stop points are plotted in this paper. To examine the validity of the results, kernel density estimation was applied in ArcGIS 10.1 as an alternative method to check the carrier's shipping depot. Also, 10% of the results were manually validated using Google Maps and Street View. The analysis found that all of the locations resulting from the SQL clustering approach were correctly located at a carriers shipping depot.

The combined problem of solving the stop identification task and classification from the GPS data having heterogeneous dataset of commercial fleets from diverse industries is also introduced in [63]. The dataset used in this paper typically consist of the raw GPS pings, providing information on the position of the vehicles as well as the work order status messages, providing information on the schedule and progress of the jobs executed by the drivers. Based on this data, a spatio-temporal clustering procedure is developed that assigns a type to each GPS message and then gathers them into groups of GPS pings that is define as stops. At the first place, the pings are categorized into three classes: engine off i.e. pings with an engine off event, idling i.e. pings where the engine is on, but the vehicle is still or moving slowly in a small area, and finally the journey pings i.e. pings that are neither engine off nor idling. These pings are further sorted chronologically for each vehicle, and all the idling and engine off pings are assembled together, forming a group for all the consecutive pings which are not separated by journey pings. The groups of pings created in this manner represents the identified vehicle stops that are classified at a later step. Stop locations in this paper is mainly classified into two different categories: work related and non-work related based on the purpose. This paper uses random forest classifier by developing different category of features for classifying the stop locations. These features include stop-wise features, points of interest (POI) features, stop cluster features, and sequential features. A large number of attributes are considered within each category of feature. The authors took labeled dataset containing 702446 records that is randomly split into training and test sets

using 10 fold cross validation. Area under the ROC is selected as a metric to evaluate the performance for each pair of parameters. Finally the performances of the proposed model on engine off and idling stops are evaluated separately, obtaining respectively 0.903 and 0.957 as ROC AUC value.

The problem of classification of stops of commercial vehicle fleets is extremely useful for the fleet intelligence companies which helps them in getting correct automatic classification of activity and non-activity stops. A few research articles addresses this problem with different techniques. Previous work in [31] considered features like stop duration and the mean distance to the center of the cluster centroid using the SVM classifier to identify stops using GPS data. Usually, the primary or activity or work stop have larger stop time as compared to the secondary or non-activity or the non-work stop. So, generally the loading/unloading and transferring of goods takes some more time than the non-work activities where driver stops for eating or gas refilling. We do not adapt the stop duration feature as there is no fixed pattern of the stop duration for both category of stops. It has been observed in our dataset, that the trucks going for long tours typically take rest stops overnight near the highways for more than 4-5 hours approximately. Also the stay duration is uncertain at the locations where goods are transferred. These observations determines the stop duration feature a highly non-discriminative one, making it difficult to differentiate the purpose of the stop events. Hence, due to such nature of data in our dataset, it is not meaningful to include stop duration as a feature for classification. Huge number of features used in [63] is possible due to the good amount of information provided in the dataset that includes work order status messages along with the GPS raw pings. In our settings, we do not have any information about the schedule and progress of the jobs executed by the drivers. We only have the raw GPS information about the position of the truck along with the timestamp at every location. The concept of entropy presented in [3] measures the diversity of the truck fleets with trucks that dwell for 15 minutes or longer at a given location. Similar approach

that we use in this research is calculating the homogeneity and heterogeneity of a stop location using the Specialization Index. Since this index alone is not accurate enough to classify the stop location, we also develop certain cluster features like calculating minimum, maximum, and standard deviation, total number of truck stop pings, unique count of carriers and unique count of truck for each cluster that helps increasing the overall performance of the classification of truck stops in fleet management.

Chapter 4

METHODOLOGY

This chapter presents a three-step method to achieve the objective of this research. The proposed method starts by applying clustering technique on GPS data points. DBSCAN algorithm presented here calculates its input parameters (Eps and Minpts) automatically from the data points itself and further form clusters using these input parameters. Obtained clusters are processed further to find firm location for each carrier. The second step targets the classification problem of categorizing the truck stop events into two types: primary stops, where goods are transferred, and secondary stops, where vehicle and driver needs are met, such as eating, taking rest or fuel refilling. This problem is solved using one of the trade indicator called Specialization Index [6]. Moreover, a classification model is designed, developing various cluster features that acts as an input features to our classification model. We compute the cluster features such as min, max and standard deviation of the percentage of truck count per carrier within the cluster; total number of the stops within the cluster; total number of unique truck within the cluster and total number of unique carriers within the cluster. These features typically help in classifying the stop points using diversity of the trucks from the variety of carriers at a stop cluster. In the third step, SQL script is developed to find tours starting from each firm.

4.1 Data Processing

This thesis utilizes an existing GPS dataset that was acquired by the Cross-Border Institute from Shaw Tracking. Although the full dataset covers a span of 6 consecutive months; this research uses the GPS records for the month of March 2016. It consists of 2 million

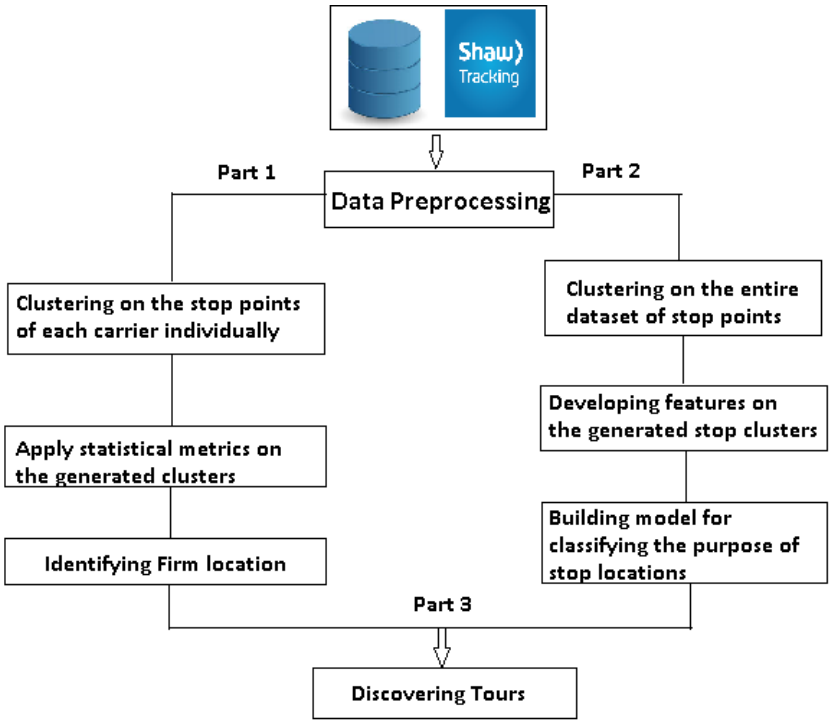

Figure 4.1: Overview of Proposed Method

records (stop points) that pertain to a total of 43,142 individual Canadian registered trucks, owned by 569 Canadian carriers. Each GPS ping corresponds to some truck movement information by location (i.e. latitude and longitude) and time stamp. Sequential numerical identifiers per ping are provided to differentiate freight trucks and their corresponding carriers. These identifiers are anonymous to protect the identity of the firms they belong to. We process the raw GPS pings, which consist of the following features.

- Carrier ID
- Truck ID
- Latitude
- Longitude
- Time

Carrier ID (CID)	Truck ID (PID)	Latitude	Longitude	Timestamp
253	116	58.3966	-104.1936	2016-03-16 15:33:38.000
253	125	47.5692	- 87.4203	2016-03-12 20:49:26.000
312	173	42.6800	-86.7580	2016-03-23 08:43:49.000
548	106	42.1431	-68.1918	2016-03-31 18:16:07.000

Table 4.1: An example of information from raw GPS data

The information shown in table 4.1 represents the format of original raw data containing records that correspond to individual GPS pings. This dataset consists of the truck pings covering both Canada and the U.S as the trucks travel across both of these countries for shipping of goods. However, this particular GPS data source corresponds specifically to Canadian owned trucks. The truck ID (PID) field is only unique for a given carrier and may be repeated by other carriers. Therefore, unique ID values are created by concatenating the carrier and truck ID fields together which is addressed as CPID in the dataset. The time stamp for each GPS ping is provided to the nearest second. While the time stamp is very likely to be correct, a minor error will not affect the analysis as long as the time shift is consistent across all pings. The potential accuracy issues arise from the location attributes. The latitude and longitude representing a geographical location are expressed in the format of decimal degrees with a precision of 5 decimal places (e.g latitude of 43.96254). The coordinate system used by the GPS devices is the 1984 World Geodetic System (WGS), where the last decimal place represents approximately 1.1 meters, although the length of longitude varies based on location.

These raw GPS pings are preprocessed using the techniques presented in Gingerich et al. [3] to find some of the meaningful data. First of all, sequential numerical identifiers per ping are provided in order to uniquely identify each GPS record with respect to the freight trucks and their corresponding carriers. This sequential identifier is addressed as SeqID in

the dataset. Also, we derive the sequential numerical identifier that uniquely identify each truck ping per each carrier. This identifier is named as SeqPID, which is unique for all the truck pings belonging to each carrier and it is reset every time for different carrier pings. These identifiers are anonymous to protect the identity of the firms they belong to. Elapsed time is calculated by taking the time difference of successive GPS pings. The time interval between each ping for a given truck can vary considerably from a few seconds to several hours or longer in some cases. Also, some trucks tend to have a shorter elapsed time between successive GPS pings compared to others. Over 2 million stops are discovered in the GPS dataset. Each stop was identified when a truck had multiple GPS pings in close proximity to each other with a total dwell time greater than 15 min. To determine the truck stopped events, previous studies considered travel speed as calculated from the distance and time of the consecutive pings. However, such a metric may result in erroneous stop identification for vehicles moving at low speeds due to congested conditions. As an alternative, a distance measurement along with time measurement is used to determine truck stops points in our dataset as shown in Fig. 4.2. With the pings for a given truck sorted sequentially according to the registered time stamp, the location of a first ping (P1) is compared to the location of the next ping (P2). If the distance is less than a certain threshold 'd', the dwell time is set equal to the elapsed time between the two pings. If the next ping (P3) is also less than 'd' from the first ping, the dwell time continues to accumulate. This continues until there is a ping (say P_n) located outside the buffer threshold at which point the dwell time is reset. The buffer threshold used to determine a stopped vehicle is set to a large radius of d = 250 m. This is imposed to avoid cutting a stop short if a vehicle moved a limited range within a given property. In this case, it is more reasonable to assume that the ending of a stop event occurs when the vehicle leaves the property location. The chosen threshold also accommodates any spatial errors that might arise due to bad GPS readings. After processing the original data, our final dataset consists of 48,866,985 (48

million) records. These processed records are mined to apply clustering technique to find out the firm for each carrier, finding out primary stops and later discovering tours for each truck. This approach considers distance and time to process real-time individual truck pings in sequential manner with the distance and time threshold parameter that generated quite good results and were validated manually to check its correctness. This approach is proved feasible for large volume of data by evaluating the pattern that emerges from analyzing stop events over space. Hence, we apply the same technique to process the GPS data points to extract the stop locations as we have the same dataset as the one used in [3]. The data shown in table 4.2 shows the sample processed data.

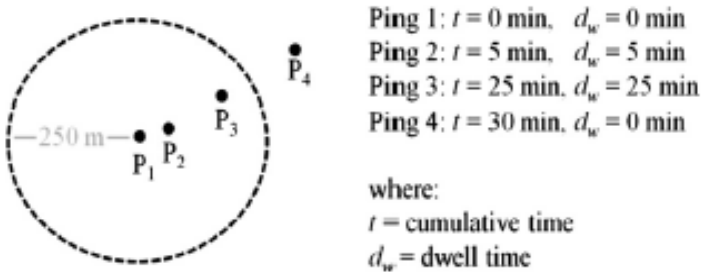

Figure 4.2: Distance based dwell time calculation [3]

SeqID	SeqPID	CID	PID	CPID	Latitude	Longitude	TimeEntry	ElapsedTime	DwellHours	Stop
12	1	186	42	186-42	49.893	-93.366	2016-03-08 04:43:30	00:04:00	00:00:00	NULL
13	2	186	42	186-42	49.901	-99.3121	2016-03-08 05:53:04	01:09:34	00:00:00	NULL
14	3	186	42	186-42	50.396	-105.1936	2016-03-08 12:54:56	00:12:45	07:01:52	1
15	1	186	43	186-42	50.486	-105.3271	2016-03-08 13:03:59	00:09:03	00:00:00	NULL
16	2	186	43	186-43	43.279	-80.5208	2016-03-08 13:07:58	00:03:59	00:00:00	NULL

Table 4.2: An example of processed GPS data

4.2 Firm identification using clustering technique

We apply one of the most popular density based clustering algorithm called DBSCAN on the stop locations of each carrier individually and discover industry location for each carrier.

4.2.1 Proposed DBSCAN with Self Regulating Eps and Minpts

DBSCAN is one of the most popular density based clustering algorithms. It is very sensitive to its input parameter, which greatly impacts the resulting clusters. Hence it is very important to select relevant and reasonable values of Eps and Minpts based on the data points. In this thesis, Eps is calculated automatically based on the density of the neighbourhood of data. The algorithm starts by calculating distance of each point to all other points in the dataset. This distance is sorted and then used to calculate the average of the k-nearest neighbors. This value of k is determined using the approach presented in [52] for calculating minpts. Once we calculated the average of the k-dist of the neighbors for each point, we again sort this average distance of all the data points and further filter it with appropriate range to quantify the value of eps parameter. Since, our dataset consists of GPS pings, we are dealing with spatio-temporal data which means we have data points moving in space with respect to time. However, we can quantify these spatial-temporal measures in finding cluster of dense points. As the goal of clustering in this research is to identify firm location, we contend to quantify the spatial measure i.e. defining the range of radius for a firm location or a primary location (industry hub) where transfer of goods takes place. A reasonable range of radius for a firm or a primary hub can be defined from 100 metres to 500 metres of radius. Logically, there cannot be any hub location smaller than 200 metres region and greater than 1000 metres (1 km) region. So, we can safely filter the distance value which does not fall under the range of 100-500 metre radius. After filtering the distance, we take average of the filtered distance and that represents the Eps

value. Hence, the general idea is to find eps by calculating the sorted average distance for each point, filtering too small and too large distance value (outliers) and then find average of the filtered distance to obtain reasonable distance that represents Eps value. For finding Minpts value automatically, we refer to the simple heuristic applied by [52] which suggests Minpts = ln(N), where N is the size of the data points.

We use this automatically calculated Eps and Minpts value to apply DBSCAN algorithm on the stop points of individual carrier to generate clusters and identify firm location out of those clusters.

Pseudocode of proposed algorithm

```
Eps (D)
For each point P in dataset D
        NeighborDistance [ ] = getDistance (P)
        For each point P' in NeighborDistance
                AvgDist [ ] = (Σ NeighborDistance [ ])/size (NeighborDistance [ ])
        SortAvgDist [ ] = sort (AvgDist [ ])
        FilteredList [ ] = SortAvgDist [ ] .sublist (100, 500);
        Eps = (Σ FilteredList [ ])/size (FilteredList [ ])


DBSCAN (D, eps, MinPts)
    C = 0
    for each unvisited point P in dataset D
            mark P as visited
            N = getNeighbors (P, eps)
            if sizeof(N) < MinPts
                    mark P as NOISE
            else
                    C = next cluster
                    expandCluster(P, N, C, eps, MinPts)

expandCluster(P, N, C, eps, MinPts)
add P to cluster C
for each point P' in N
            if P' is not visited
            mark P' as visited
            N' = getNeighbors (P', eps)
            if sizeof (N') >= MinPts
                    N = N joined with N'
            if P' is not yet member of any cluster
                    add P' to cluster C
```

Figure 4.3: Pseudocode of proposed DBSCAN algorithm to identify firm cluster

Having discovered the clusters, we calculate the following statistic measures for each

cluster to discover firm location.

1. Count of unique truck: the cluster having the maximum count of distinct truck is marked as a firm.

2. Count of truck visits (cycles): the cluster having the maximum count of truck visits is marked as a firm.

As firm is the hub from where all of its trucks start from to transfer goods and come back to the same location, it should have maximum truck count and maximum number of visits by the trucks running for it. Note that in practice, a stop location with maximum truck count also has the maximum number of visits, and vice versa. So, we consider both metrics together to determine if the cluster is a firm location. Therefore, a cluster that represents maximum number of unique truck count and maximum number of visits or cycles is identified as a firm.

4.3 Stop purpose classification

We propose two potential approach of classifying the stop locations in this thesis. The first approach is a point-based classification and the second approach is a cluster-based classification. For both of these approach, we derive two different sets of novel features to determine the purpose of stopped truck events and to categorize them into primary and secondary stops. The first feature is the trade measure used in transportation and economic geography called Specialization Index which measures how specialized or diversified a freight terminal is in relation to a group of locations. The second set of features presents the list of several stop cluster features that we develop and build the supervised classification model to classify the stop locations.

4.3.1 Classification Features

Classification using the Specialization Index

The Specialization Index is one of the well-known trade indicator used to classify a location to know if it is specialized or diversified in terms of its activities. In transportation, to find out if a terminal is specialized in the transshipment or the handling of a particular merchandise or if, inversely, it transfers a wide variety of merchandises, a specialization index [6] can be calculated. This index looks at the level of concentration of an activity at a location such as a terminal in relation to a group of locations. The specialization index (SI) is calculated using the following formula:

$SI = \sum_i t^2_i / (\sum_i t_i)^2$

It is the total of squares of tonnage (or monetary value) of each type of merchandise i (t_i) handled at a terminal over the square of the total volume tonnage (or monetary value) of merchandise handled at the terminal. If the index value is closer to 1, such a result indicates that the terminal is specialized. If, inversely, the index value is close to 0, it means that the terminal's activity is highly diversified. Thus, the specialization index evaluates the degree of specialization or diversification of any type of terminal or location.

In the present research, the Specialization Index is applied to classify locations to know if they are specialized to be labeled as primary stop for handling shipping from certain types of carriers or it is diversified to be labeled as secondary stop for handling shipping from multiple types of carrier. Stop event classification using the specialization index is calculated by following formula:

$SI = \sum_c t^2_c / (\sum_c t_c)^2$

where c is the carrier and t_c is the truck count belonging to carrier c. Hence specification index here is the total sum of squares of truck count for each carrier over the square of the

total of truck count for each carrier. The value of this index ranges between 0 and 1. Value closer to 1, suggest that the stop location is specific to some carrier indicating primary stop for loading/unloading of goods. However, if the value is close to 0, then this indicates that the stop location is general, suggesting secondary stop for general purpose. To apply the concept of specialization index, the locations of stops where trucks dwelled for over 15 min are considered. SQL script is developed to calculate SI for each stop location. The script applies SQL clustering to obtain clusters of stop locations for each stop by capturing nearby stop events with a predefined radius (250 meters) and then applying specialization index on the clusters. The value for specialization index fall in a range of 0 to 1. A value close to 0 indicates secondary stops while value close to 1 indicates primary stops. A threshold value of 0.1 was selected to differentiate between primary and secondary stops. So the stop points with specialization index value greater than or equal to 0.1 are marked as primary stops and the ones with specialization index value smaller than 0.1 are marked as secondary stops. This threshold was selected based on the experimental results as reported in the next chapter.

Classification using the measure of stop cluster features

In this approach of classification, we develop a set of several cluster features to measure the homogeneity and heterogeneity of a stop location. To measure this diversity of truck in terms of variety of different carriers at particular stop location, we calculate the percentage of truck count for each carrier stopping at a cluster of stop points. Based on this percentage values, we define the statistical cluster features that helps in training the model for classifying stop locations into primary and secondary stops. The following are the list of features that we have identified to improve the accuracy of classifying stop locations.

Classification Features We divide the list of classification features into two set: Stop Cluster Features and Temporal Features

1. Stop Cluster Features: As shown below, we generate a number of statistical features for each cluster and provide these attributes as an input to our classification model.

    - Minimum: The minimum percentage of truck count per unique carrier.
    - Maximum: The maximum percentage of truck count per unique carrier.
    - Standard Deviation: The variation or dispersion of truck count per unique carrier.
    - Total Trucks: The total number of trucks stopping at the stop cluster.
    - Unique Carrier Count: The number of distinct carriers at the stop cluster.
    - Unique Truck Count: The number of distinct trucks at the stop cluster.

    Ideally, a primary location should have large number of trucks from few carriers and a secondary location should have trucks from large variety of carriers.

2. Temporal Features: Based on previous studies that considered temporal measures as one of the prominent features for stop purpose classification, we also include such measures to check the performance of our model. We consider the below mentioned temporal features:

    - Dwell Time: A measure of the amount of time that a truck has stopped at a particular stop location.

- Time of the day: Refers to the time or the hour of the day when truck makes the stop at a location.

It has been observed in our dataset, that the trucks going for long tours typically take rest stops overnight near the highways for more than 4-5 hours approximately. Also, the stay duration is uncertain at the locations where goods are transferred. These observations determines that the stop duration feature is highly non-discriminative in categorizing the purpose of the stop events. Hence, with our dataset, it is not meaningful to include Dwell Time as a feature for classification. We also observe that the secondary stops taken for the rest on the highways are mostly during the night hours while a primary location where goods are transferred are mostly performed during day time. Consequently, the dwell time combined with the time of the day could be a good indicator to differentiate primary and secondary stops.

Based on the above listed features, we first apply classification technique on the specialization index in combination with the cluster features and then combine them with the temporal features. Performance results for all the cases are reported in the next chapter.

4.3.2 Point-Based Classification

In this approach, we calculate the feature values for each stop point individually. We start by considering a single stop and for each stop point, we cluster its neighbours together based on the selected reasonable radius. We calculate the specialization index and the cluster features considering the diversity of the neighbours for this point. So this approach performs the feature calculation on each stop point individually by applying simple clustering with a defined radius. Several experiments have been conducted to determine a reasonable radius in order to get more accurate classification results.

4.3.3 Cluster-Based Classification

In this approach, we focus on clustering the stop points using the DBSCAN algorithm based on the density of points. Once we group the dense stop points together into a stop cluster, we calculate the specialization index and the cluster features for each cluster surrounding the stop event in consideration. Hence, this approach performs the feature calculation on each cluster individually. To perform cluster-based classification, we modify the traditional DBSCAN algorithm to apply clustering on the stop points. The pseudocode of the modified DBSCAN algorithm is shown in Fig. 4.4.

```
DBSCAN (D, eps)
        C = 0
        for each unvisited point P in dataset D
                mark P as visited
                N = getNeighbors (P, eps)
                C = next cluster
                expandCluster(P, N, C, eps, MinPts)

expandCluster(P, N, C, eps, MinPts)
MinPts = ln(D)
add P to cluster C
for each point P' in N
        if P' is not visited
                mark P' as visited
                N' = getNeighbors (P', eps)
                if sizeof (N') >= MinPts
                        N = N joined with N'
                        if P' is not yet member of any cluster
                        add P' to cluster C
```

Figure 4.4: Pseudocode of Modified DBSCAN for finding stop clusters

This algorithm works in a similar way as the traditional algorithm. The only modification applied to its working is that, we consider only the eps parameter while forming the initial cluster with radius defined by the eps value. That is, starting from a stop point, we form a cluster including all its neighbors that falls within the eps distance without considering the

density of the stop points around that point. Hence, even if there is one neighbour around or few neighbors around, we consider them to be part of a cluster. Basically, we do not declare the sparse points as outliers and include them initially in a cluster. In the second step of expanding the cluster, we consider the minpts parameter with the value of minpts determined by the natural log function over the size of the dataset. This constraint of minpts allows the cluster to expand only when the cluster boundary is dense enough, otherwise the cluster will be left unexpanded. This modification is developed for the purpose of preserving all the stop locations since we need to classify and label all of them to primary or secondary. Several experiments has been conducted to determine a reasonable radius and minpts in order to achieve more accurate classification results.

4.4 Finding Truck Tours

Once the firm is discovered for each carrier, SQL script is developed to find the tours for each truck starting from a firm and returning to the firm. The approach to find a tour is to first sort all the data by truck ID and time-stamp. Sorting the GPS pings by time stamp for a given truck helps in observing the travel patterns of the vehicles. Iterating through this sorted data, we check two consecutive records to see if they have the same Truck ID and it belongs to the cluster marked as a firm. If that is the case, we update the tour, otherwise we fetch the next record and again check for the same condition. This logically says that a tour is formed if the same truck that begins at a firm location comes back to that same firm location after a certain period.

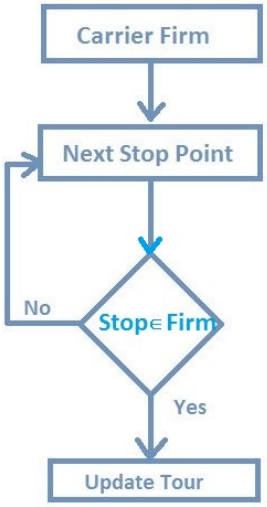

Figure 4.5: Workflow of SQL script for finding Tour

Chapter 5

RESULTS AND DISCUSSIONS

The GPS data which needs to be processed for finding truck tours are stored in database tables in MS SQL server 2008. To discover the firm for each carrier, in the first step, our clustering algorithm was implemented in Java on Eclipse platform. The clusters generated by this algorithm are then stored in database tables and processed further to apply statistics to identify firm cluster. SQL script has been developed to apply these statistics on generated clusters to find the firm. In the second step, specialization index is calculated using SQL script to classify the stop events. The value of the cluster features that we propose in our classification model are also calculated using the SQL scripts. The classification model we use is a Random Forest classifier [64], which consists of a group of decision trees, and is widely believed to be among the best choices for standard classification tasks. The experimental results for both the point-based and cluster-based approach are presented in this section. Fig. 5.1 shows the sample stop points considered for the point-based classification. The stop points were randomly selected from the complete dataset and labeled manually using the Google Maps and Google Street View to determine the type of the stop. Labeled information of this sample data is provided in Table 5.1.

For the cluster-based approach, our training dataset consists of three different samples corresponding to three different geographical locations all over Canada and the US as shown in Fig 5.2. The stop points of these sample regions are clustered using the clustering algorithm proposed in Fig. 4.4. These generated clusters are labeled manually using the Google Maps. We further pick the primary and secondary clusters randomly from the labeled data and create final sample data for training the model. Labeled sample data information for each sample region is described as in Table 5.2. The model has been trained by

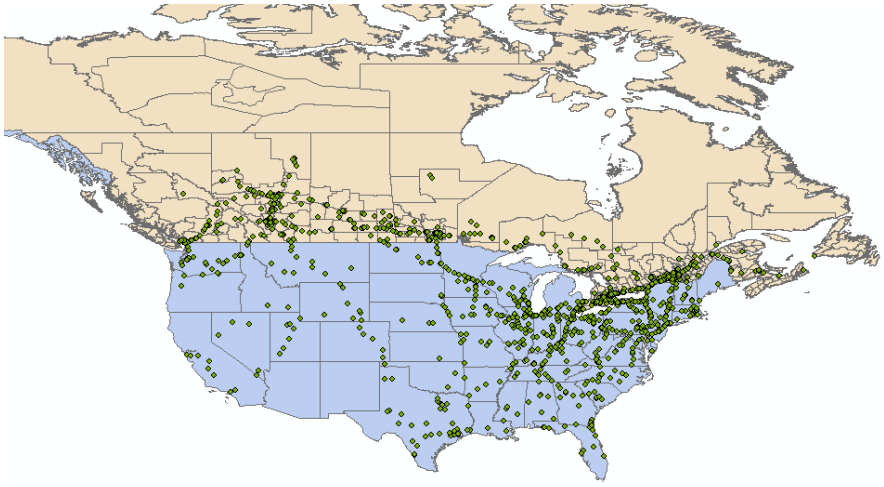

Figure 5.1: Sample stop points

Training Data	Primary Stops	Secondary Stops	Total Stops
Sample Data	2721	2693	5414

Table 5.1: Sample training data for point-based classification

means of a 10 fold cross-validation procedure to choose the best number of trees and their maximum depth. This proposed model is developed using the Weka tool. Finally, to derive truck tours starting from a firm, a script in SQL has been implemented which processes the GPS pings sorted based on date and time. The truck tours that we discover are presented using a tool called ArcGIS where we can visualize all the truck pings for the entire tour over a geographical map.

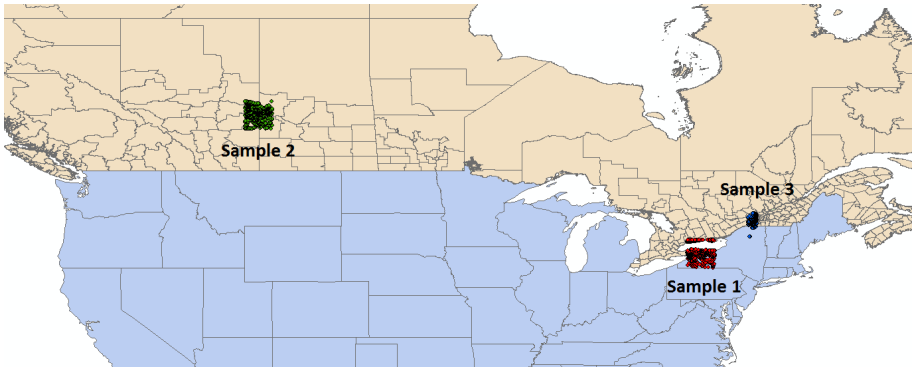

Figure 5.2: Sample Regions

Training Data	#TotalStops	#Clusters	#ActualPrimary	#ActualSecondary
Sample 1	21089	307	10411	10678
Sample 2	6792	160	3764	3028
Sample 3	9293	93	3999	5294

Table 5.2: Sample training data for cluster-based classification

5.1 Firm Location Validation

A sample of stop points of 14 carriers were processed to identify their firms. The stop points of each carrier is first processed to obtain the clusters. The generated clusters are further processed to calculate two statistical metric i.e count of distinct trucks and count of truck visits to identify the firm cluster. The cluster results along with the calculation of the statistical metrics for two sample carriers are reported in Table 5.3 and 5.4.

The identified firm locations were validated manually using street view provided by Google Maps. The firm identification results along with their statistical metric values for 14 sample carriers are presented in Table 5.5.

Cluster ID	#Distinct Trucks	#Cycles	Firm
C1	4	7	No
C2	2	9	No
C3	5	9	No
C4	1	8	No
C5	5	10	No
C6	4	16	No
C7	5	5	No
C8	2	10	No
C9	1	10	No
C10	1	8	No
C11	1	7	No
C12	4	9	No
C13	1	8	No
C14	1	28	No
C15	3	5	No
C16	1	8	No
C17	19	92	Yes
C18	1	6	No
C19	1	16	No

Table 5.3: Firm information for sample carrier 1

Cluster ID	#Distinct Trucks	#Cycles	Firm
C1	6	2	No
C2	8	0	No
C3	7	2	No
C4	23	106	Yes
C5	3	6	No
C6	5	3	No
C7	11	12	No
C8	10	11	No
C9	12	3	No
C10	17	32	No
C11	6	2	No
C12	11	8	No
C13	22	73	No
C14	10	15	No
C15	8	8	No
C16	8	5	No
C17	6	3	No

Table 5.4: Firm information for sample carrier 2

Carrier ID	#Stops	#Clusters	#Distinct Truck	#Cycles
C1	1145	19	19	92
C2	998	18	10	170
C3	1102	34	14	109
C4	1298	17	23	106
C5	1937	47	29	156
C6	1206	32	5	290
C7	6519	123	56	278
C8	1916	30	27	255
C9	17612	239	217	1688
C10	3029	48	38	333
C11	7382	118	102	1758
C12	1528	18	15	50
C13	40	3	3	15
C14	2382	24	36	372

Table 5.5: Firm information for 14 sample carriers

5.2 Analysis on the performance of point-based classification approach

5.2.1 Analysis on deciding Specialization Index threshold value for classifying stop locations

Since specialization value close to 1 indicates specialized location and value close to 0 indicates diversified location, we need to safely determine the threshold to differentiate between primary and secondary stops. All the stop points in the training sample data with specialization value greater than 0.2 are clearly observed as primary stops. Similarly, ma-

jority of stop points with value smaller than 0.09 are observed to be secondary stops. This is clearly observed in Fig. 5.4 and all the SI values ranging from 0.09 to 0.2 holds both the primary and secondary locations. Hence it is important to accurately determine the threshold value between 0.09 and 0.2 to classify the purpose of stop events. The histogram shown in Fig. 5.3 depicts the specialization threshold value.

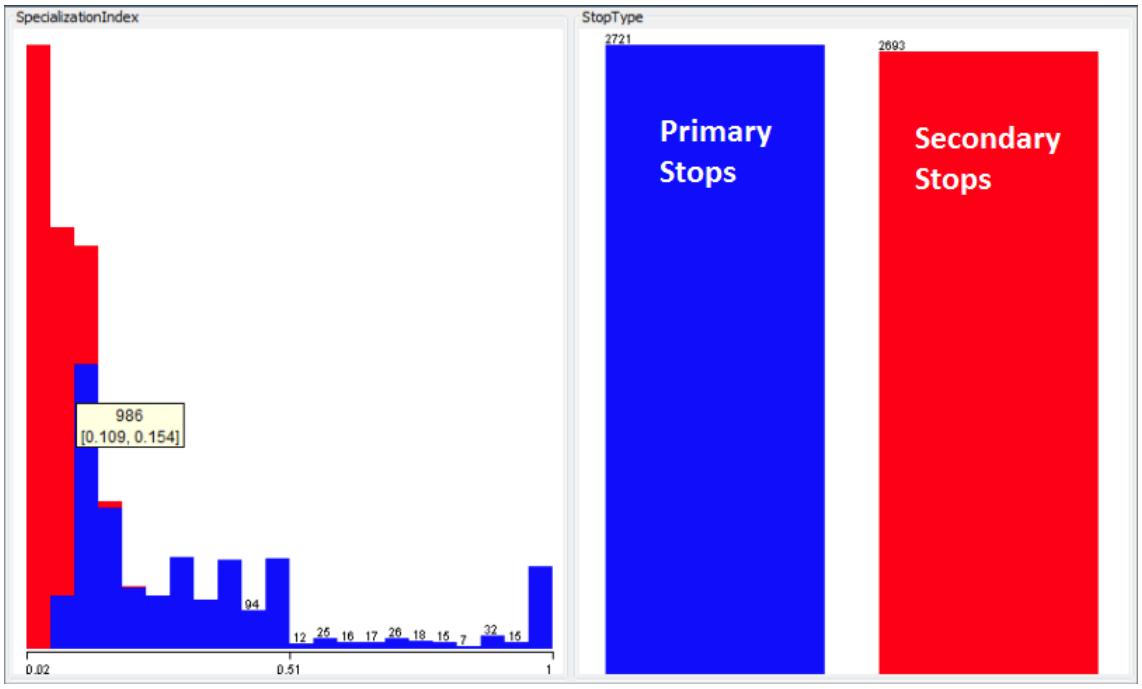

Figure 5.3: Specialization Index threshold for classifying Stop locations

According to the plot, we can clearly see that all the smaller values corresponds to the secondary stops and all the higher values corresponds to the primary stops, and with the value between 0.109 and 0.154, the count of secondary stops starts decreasing and the count of primary stops starts increasing.

In order to find accurate threshold value such that the errors i.e. false positives and false negatives is minimized, we have developed SQL script that finds the count of primary and secondary stops for each SI value in the sample training data. Values having only the count of primary stops and no count for secondary stops corresponds to the true positives, and values having no count of primary stops and only the count of secondary stops corresponds

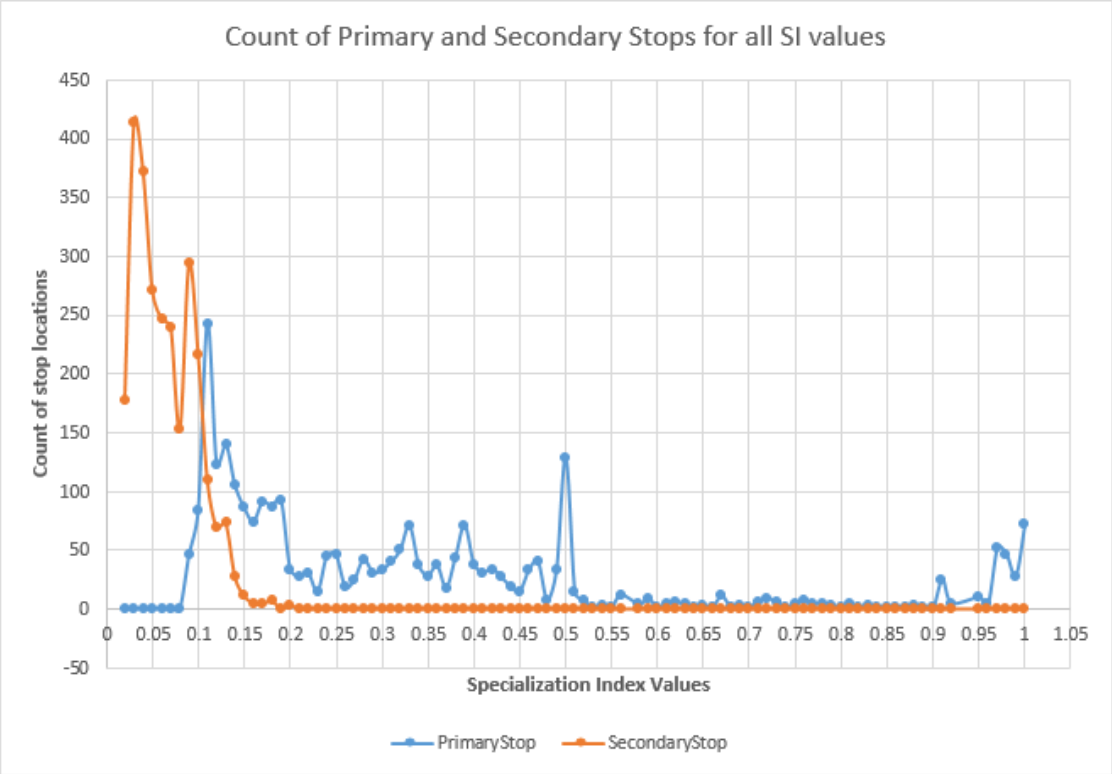

Figure 5.4: Count of Actual Primary and Secondary stops for all SI values

to the true negatives. So for the specialization values having the count of both primary and secondary stops corresponds to the type I and type II errors (False positives and false negatives). Our goal is to find such value that minimize both the errors, so that we get good accuracy on classifying stop locations. The following function defines the mathematical expression to find the accurate threshold value.

V = Min [fp (V) + fs (V)]

Where, the function fp defines the count of primary stops for specialization value V and the function fs defines the count of secondary stops for specialization value V. The threshold value V is selected when the above function gives the least value.

Table 5.6 shows the table with Specialization Index value holding the count of false negatives and false positives along with the count of true positives (primary stops) and true negatives (secondary stops). Based on this table, a chart has been formed across the Specialization values and the total error rate (FP + FN). The chart and the table in Fig. 5.5

	SI	PrimaryStop	SecondaryStop	FP	FN	Total
1	0.09	46	294	524	46	570
2	0.10	84	216	308	130	438
3	0.11	243	109	199	373	572
4	0.12	123	69	130	496	626
5	0.13	140	73	57	636	693
6	0.14	105	27	30	741	771
7	0.15	86	11	19	827	846
8	0.16	74	5	14	901	915
9	0.17	91	4	10	992	1002
10	0.18	87	7	3	1079	1082
11	0.20	34	3	0	1206	1206

Table 5.6: Table showing accuracy and error values for SI between 0.09 and 0.20

clearly depicts that the specialization index value at 0.1 holds the least error rate with less value of both the false positives and the false negatives count.

Hence, a threshold of 0.1 was selected based on the noticeable separation in the proportion of primary and secondary stops. Based on the threshold selected, we can clearly see the percentage of the false positives and false negatives as per the results reported in Fig. 5.6. Out of 5414 stop locations, based on the threshold value of 0.1, total of 2899 points were classified as primary locations of which 2591 is the actual count of primary locations with 308 secondary locations as validated manually. This leaves 11% of false positives that represents secondary locations mislabeled as primary locations. Similarly, out of 5414 stop locations, 2515 locations were classified as secondary stops of which 2385 is the actual count of secondary locations with 130 primary locations as validated manually. This corresponds to 5% of miss rate indicating false negatives that represents the primary locations mislabeled as secondary locations.

5.2.2 Experiments to show the selection of reasonable radius for stop clusters

A reasonable radius of 250 meters was selected to form a cluster of stop points while calculating specialization index. Several experiments were performed to calculate SI value

SI	FP	FN	Total Error (FP+FN)
0.09	524	46	570
0.1	**308**	**130**	**438**
0.11	199	373	572
0.12	130	496	626
0.13	57	636	693
0.14	30	741	771
0.15	19	827	846
0.16	14	901	915
0.17	10	992	1002
0.18	3	1079	1082
0.2	0	1206	1206

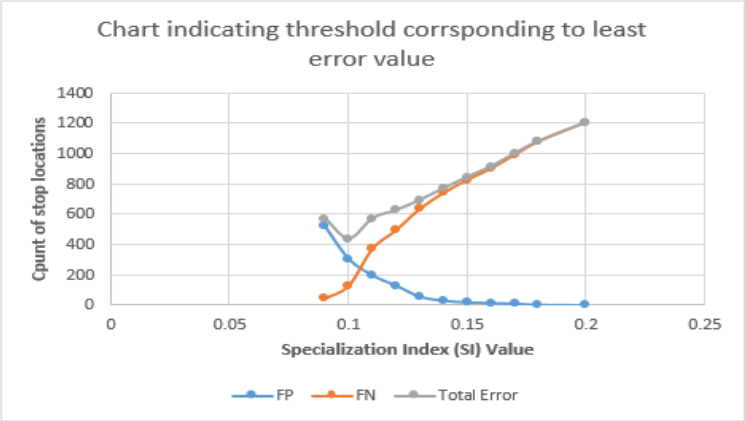

Figure 5.5: Table and Chart showing Specialization threshold with least error value

to verify the results in terms of classifying the number of primary and secondary locations accurately. Table 5.7 shows both the type I and type II errors for different radii. Table 5.8 reports the accuracy and the error rates. Moreover, for each cluster radius, we checked the corresponding SI threshold value using decision tree classifier (J48) in WEKA. The SI threshold value consistently comes to be 0.1 for each cluster radius. Hence, we can safely set the SI threshold value as 0.1 to classify the stop points accurately.

If we increase the radius of the cluster, there are chances of missing primary locations as the locations tend to combine into a big cluster. Also the number of secondary location increases indicating increase in false negative rate that represents the primary locations mislabeled as secondary locations. On the other hand, if we keep decreasing the cluster radius, there will be no effect on the number of primary stops; it will not further increase but there are chances of getting more secondary locations which are misclassified as primary stops leading to get high false positive rate. Therefore, it is very important to safely

Specialization Index			
>0.1		<=0.1	
Primary	2591	Primary	130
Secondary	308	Secondary	2385
Total	2899	Total	2515

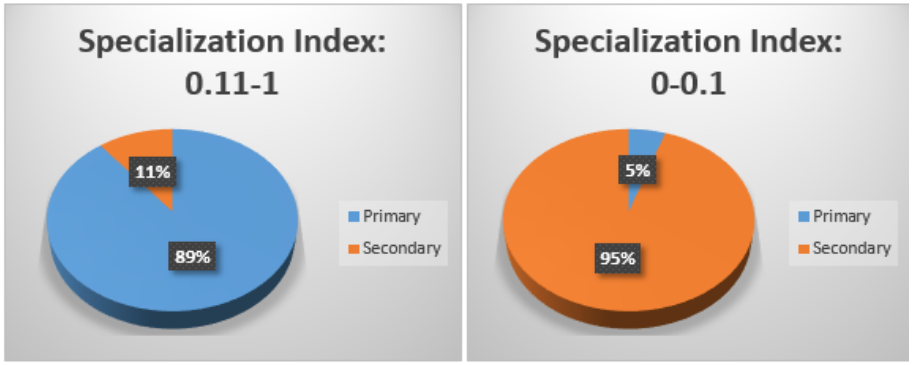

Figure 5.6: Charts showing percentage of error for defined SI threshold

Specialization Index Threshold (0.1)							
Cluster Radius (metres)	Actual Primary Stops	Actual Secondary Stops	Primary Stops (TP)	Secondary Stops (TN)	False Positives (FP)	False Negatives (FN)	Total (FP+FN)
350	2721	2693	2316	2372	321	405	726
300	2721	2693	2449	2395	298	272	570
250	**2721**	**2693**	**2591**	**2385**	**308**	**130**	**438**
200	2721	2693	2591	2333	360	130	490
150	2721	2693	2681	2187	506	40	546

Table 5.7: Effect on the stop classification with varied cluster radius

determine the radius of the stop cluster such that we minimize both the false positive and false negative rates.

Fig. 5.7 shows the graph of the accuracy and the error rate for each cluster radius. As we go on increasing the cluster radius too large or decreasing too small, the accuracy starts dropping eventually with increase in the error rates. Radius of 250 meters shows the balanced count of the actual primary and secondary stops. Hence, we can safely decide to consider 250 meter radius for the stop cluster.

5.2.3 Analysis on the performance of the proposed point-based model for stop purpose classification

To evaluate the accuracy of the classification using different cluster radius, we have

Specialization Index Threshold (0.1)									
Cluster Radius (metres)	**Actual Primary Stops**	**Actual Secondary Stops**	**Identified Primary Stops**	**Identified Secondary Stops**	**True Positive Rate (TPR)**	**True Negative Rate (TNR)**	**False Positive Rate (FPR)**	**False Negative Rate (FNR)**	**Total Accuracy**
350	2721	2693	2316	2372	85.12%	88.08%	11.92%	14.88%	86.59%
300	2721	2693	2449	2395	90.00%	88.93%	11.07%	10.00%	89.47%
250	**2721**	**2693**	**2591**	**2385**	**95.22%**	**88.56%**	**11.44%**	**4.78%**	**91.91%**
200	2721	2693	2591	2333	95.22%	86.63%	13.37%	4.78%	90.95%
150	2721	2693	2681	2187	98.52%	81.21%	18.79%	1.48%	89.92%

Table 5.8: Accuracy results for varied cluster radius

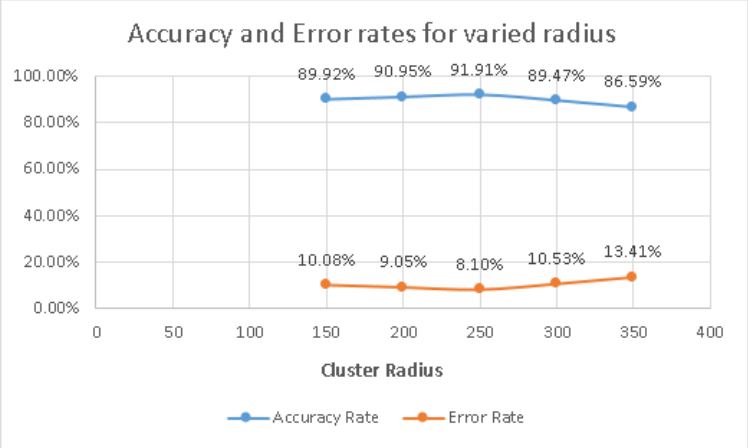

Figure 5.7: Error rate for varied cluster radius

selected four major derivations from the confusion matrix; the true positive rate (TPR), the true negative rate (TNR), the false positive rate (FPR) and the false negative rate (FNR) where:

TPR = TP / P

TNR = TN / N

FPR = FP / N

FNR = FN / P

and TP and TN are the true positives and true negatives respectively. P is the real positive cases in the data and N is the real negative cases in the data. In our setting, primary stops are represented by the real positive cases and secondary stops are represented by the real negative cases. TP corresponds to the correctly predicted primary stops, TN corresponds to the correctly predicted secondary stops, FP corresponds to the actual secondary stops mis-

classified as the primary stops and FN corresponds to the actual primary stops misclassified as the secondary stops.

We consider evaluating the overall accuracy for classification in a way that we look for reducing the overall error rate i.e reducing both the false negative and the false positive rate. This means that we do not look for just getting majority of the correctly identified primary locations but we also consider getting less false positive rate i.e less misclassification of actual secondary stops to the primary ones. Accuracy is defined by the following:

Accuracy = (TP + TN) / (P + N)

We build our classification model by first using the specialization index that we have proposed as our classification index. The model is further trained by progressively adding the cluster features to it. We also add the temporal features to our model. This is done with the aim of evaluating the performance gain that we obtain with the features developed in this research work for our model. We compare our results with the entropy index proposed in [3], that is applied to the stop locations belonging to Canada and the USA. We also add our cluster features to the entropy index to observe the performance of our proposed model. CF in the result table refers to the cluster features that includes all the six features i.e. Total Trucks, Unique Trucks, Unique Carriers, Minimum, Maximum and Standard deviation. We also test the accuracy of our classification model by generating a random function index. Results show that the proposed features when combined with this function index greatly improves the performance as compared to the function index used independently as a classification index. This suggests that adding the proposed features would certainly add more power to the classification model.

Analysis on the result of the specialization index and the cluster features for stop purpose classification

As per the results shown in Table 5.9, the accuracy of stop classification using the spe-

	Random Forest Classifier		
Model	**True Positive Rate**	**True Negative Rate**	**Accuracy**
Specialization Index (SI)	**95.22 %**	**88.56 %**	**91.9099 %**
Specialization Index + (Cluster Features) Cluster Features: Total Trucks, Unique Carriers, Unique Trucks, Minimum, Maximum and Standard Deviation	**98.89 %**	**98.70 %**	**98.7994 %**
Specialization Index + (Cluster Features) + (Temporal Features) Temporal Features: Dwell Time and Time of the day	**89.08 %**	**86.74 %**	**87.9202 %**

Table 5.9: Performance of the model over the specialization index and its combination with the cluster features

cialization index is 91.90% and when it is combined with the cluster features, the accuracy is increased to 98.79%. As we discussed before about the nature of temporal features in our dataset, we can clearly observe its impact on our model. The model when combined with the temporal features, reduces the classification performance and the accuracy drops to 87.92%.

Analysis on the result of the entropy index and the cluster features for stop purpose classification

Table 5.10 shows the performance of Entropy Index and its combination with the cluster features. It is clearly observed that the cluster features naturally improves the performance of entropy index. The model is 95.14% accurate with entropy index while its combination with the cluster features increases the accuracy to 98.74%. Similarly, we also check the results combining the temporal features with the entropy index. We could observe the same situation where the accuracy is dropped to 87.97% which is almost the same as in the case of specialization index.

	Random Forest Classifier		
Model	**True Positive Rate**	**True Negative Rate**	**Accuracy**
Entropy Index (EI)	**96.39 %**	**93.87 %**	**95.1422 %**
Entropy Index + (Cluster Features) Cluster Features: Total Trucks, Unique Carriers, Unique Trucks, Minimum, Maximum and Standard Deviation	**98.93 %**	**98.55 %**	**98.744 %**
Entropy Index + (Cluster Features) + (Temporal Features) Temporal Features: Dwell Time and Time of the day	**90.52 %**	**85.40 %**	**87.9756 %**

Table 5.10: Performance of the model over the entropy index and its combination with the cluster features

Analysis on the result of the Specialization index and the Entropy index with the cluster features for stop purpose classification

	Random Forest Classifier		
Features	**True Positive Rate**	**True Negative Rate**	**Accuracy**
SI	**95.22 %**	**88.56 %**	**91.9099 %**
EI	**96.39 %**	**93.87 %**	**95.1422 %**
SI + CF	**98.89 %**	**98.70 %**	**98.7994 %**
EI + CF	**98.93 %**	**98.55 %**	**98.7440 %**
SI + EI + CF	**99.23 %**	**98.88 %**	**99.0580 %**

Table 5.11: Performance of the model over the combination of SI, EI, and CF

Table 5.11 shows the impact of the specialization and the entropy index combined with the cluster features. Both the index combined together with cluster features gives 99% accuracy.

Analysis on the result of the Formulated index and the cluster features for stop purpose classification

	Random Forest Classifier		
Model	**True Positive Rate**	**True Negative Rate**	**Accuracy**
Formulated Index (FI)	**85.81 %**	**83.36 %**	**84.5955 %**
Formulated Index + (Cluster Features) Cluster Features: Total Trucks, Unique Carriers, Unique Trucks, Minimum, Maximum and Standard Deviation	**98.71 %**	**98.73 %**	**98.7255 %**
Formulated Index + (Cluster Features) + (Temporal Features) Temporal Features: Dwell Time and Time of the day	**87.61 %**	**83.51 %**	**85.5744 %**

Table 5.12: Performance of the model over the formulated index and its combination with the cluster features

We define an arbitrary function index as shown below:

$FI = \sum_c t^3{}_c \; / \; (\sum_c t_c)^3$

where, c is the carrier and t_c is the truck count belonging to carrier c. This index is formed with an intention to show that there can be many such index which can be formed using the percentage of truck count per carrier as listed in this way, but not necessarily each index can independently give accurate performance in classifying the purpose of stop locations. But we can definitely improve the performance of the overall classification model by including the additional features to the model. As shown in Table 5.12, the formulated index is only 84% accurate, but when combined with the cluster features, the accuracy is increased to 98%. In this research, the cluster features that we define, really proved its relevance in classifying the stop locations.

	Random Forest Classifier		
Model	**True Positive Rate**	**True Negative Rate**	**Accuracy**
Cluster Features: Total Trucks, Unique Carriers, Unique Trucks, Minimum, Maximum, Standard Deviation	98.82%	98.84%	98.8364 %
Cluster Features: Total Trucks, Unique Carriers, Unique Trucks	97.57%	97.36%	97.4695 %
Cluster Features: Minimum, Maximum, Standard Deviation	95.92%	97.66%	96.7861 %

Table 5.13: Performance of the model over the formulated index and its combination with the cluster features

Analysis on the result of the cluster features for stop purpose classification

The results presented in Table 5.13 depicts the direct impact of the cluster features with respect to the performance of our model. Cluster features are accurate enough to independently classify the stop locations giving 98.83% accuracy. Out of the six cluster features, the first three features mainly the total trucks, unique carriers and the unique trucks are the most prominent features as compared to the minimum, maximum and the standard deviation as per the reported results. Since minimum, maximum and the standard deviation are the statistical features calculated based on the total trucks and unique carriers, we can say that total trucks and unique carriers are the base features giving highest accuracy.This is the reason why it greatly improves the performance of the overall model when combined with different index used in classifying the stop locations.

5.2.4 Analysis on the correlation of the features in classifying stop locations

Table 5.14 shows the correlation value of each feature to the target class of classifying the purpose of stop location. This attribute ranking is performed using one of the attribute selection method called InfoGainAttributeEval method provided in the weka tool. This method evaluates the worth of an attribute by measuring the information gain with respect to the prediction class. The correlation values generated using this method specifies how

Attributes	Correlation Value
Total Trucks	0.9158
Unique Carriers	0.89
Unique Trucks	0.8842
Specialization Index	0.8615
Entropy Index	0.8291
Maximum	0.7797
Formulated Index	0.737
Standard Deviation	0.5668
Time of the Day	0.088
Dwell Time	0.0681
Minimum	0.0554

Table 5.14: Feature ranking based on the correlation values of the Features

relevant the features are in classifying the target prediction class. The top five relevant features to our classification model are Total Trucks, Unique Carriers, Unique Trucks, Specialization Index and the Entropy Index. These features depicts high relevance to the target class as compared to the other features. The features with high correlation value are more relevant in classifying while the one with low value are less relevant. The following section presents the graphs showing how the highly correlated features are related to the target class.

The graph plot showing how each feature is related to other features and the final target class is shown in Fig 5.8. The blue points refers to the primary stops and the red points refers to the secondary stops in the correlation plot.

Fig. 5.9 shows the correlation between Specialization Index and the number of unique carriers. The graph shows that locations having lower Specialization index value and higher value of unique carriers corresponds to the secondary location while the ones with higher specialization value and less variety of carriers corresponds to the primary locations.

Fig. 5.10 clearly shows that the primary locations hold more number of trucks with less variety of carries while the secondary locations hold the trucks with large variety of carriers.

Fig. 5.11 depicts that primary locations tend to have high maximum percentage value

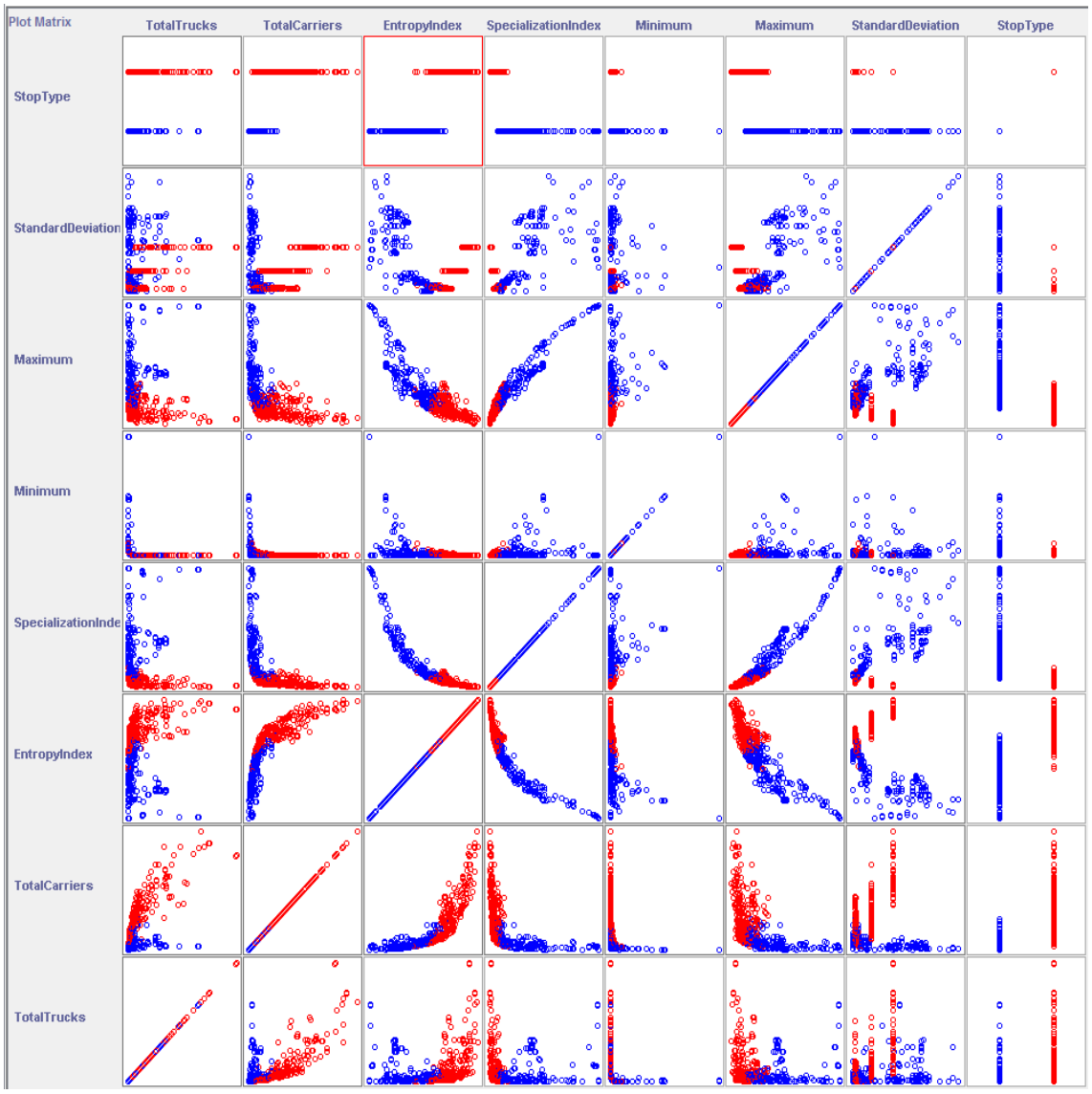

Figure 5.8: Correlation graph of the features (Blue:Primary & Red:Secondary)

with high specialization Index and low variety of carriers. While Secondary locations have low maximum percentage value with low specialization Index and high variety of carriers.

The remaining list of features such as Dwell Hours, Time of the day, and Minimum are the features which do not show strong relation with the target class as seen in the correlation graph in Fig. 5.12.

The overall experimental results demonstrate that the cluster features developed in this research remarkably improves the performance of our classification model.

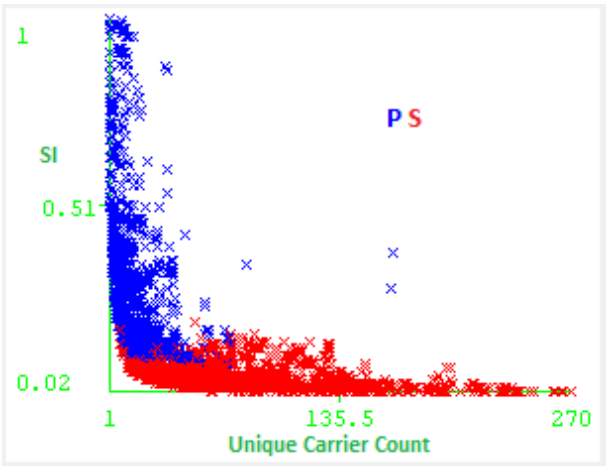

Figure 5.9: Relationship between Specialization Index and Unique carrier count

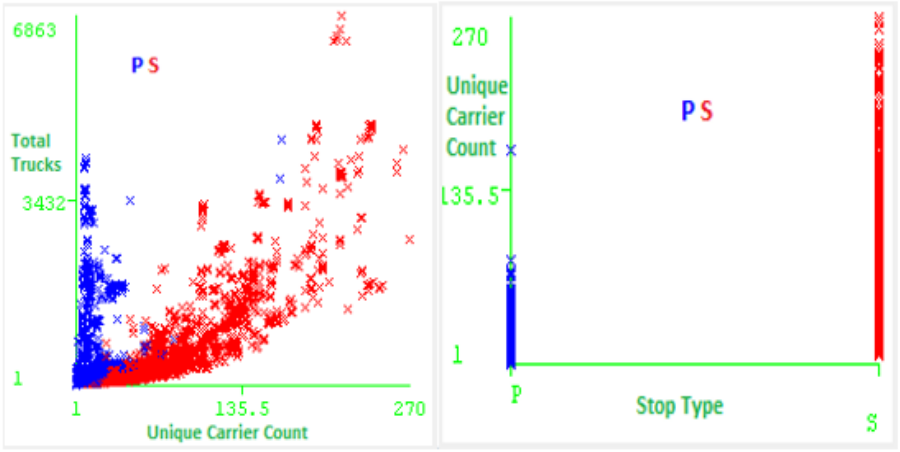

Figure 5.10: Relationship between Total Trucks, Unique carrier count and Stop Type

5.3 Analysis on the performance of cluster-based classification

5.3.1 Analysis on Input Parameters for Clustering

Experiments to select a reasonable radius for stop clusters

Several experiments were performed using different cluster radii to calculate the accuracy of classifying the stop clusters using the specialization index. This experiment was

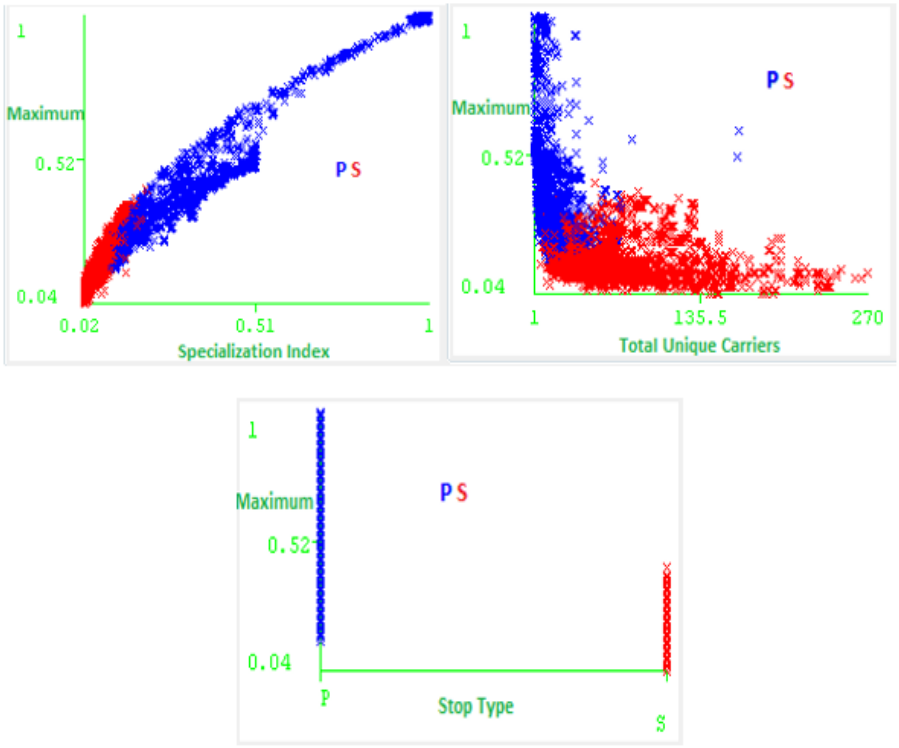

Figure 5.11: Relationship between Specialization Index, cluster features and Stop Type

conducted on two different sample data.

Table 5.15 and 5.16 shows the accuracy and error results for two different samples with six different radius values. The corresponding graphs showing the accuracy rate for each radius are also reported in Fig. 5.13 and 5.14. The results clearly show that if we increase the radius of the cluster gradually, there are chances of missing primary locations as the locations tend to be combined into bigger clusters. Also the number of secondary location increases indicating the increase in false negative rate that represents the primary locations mislabeled as secondary locations. On the other hand, if we decrease the cluster radius gradually, there are chances of getting more primary stops as a large amount of locations tend to split into the smaller ones. This increase in the primary locations also increases the false positive rate that represents the secondary locations mislabeled as primary locations. For example, if there is a bigger cluster consisting of combination of primary and secondary locations, then the reasonable smaller radius can help split the bigger cluster, giving more

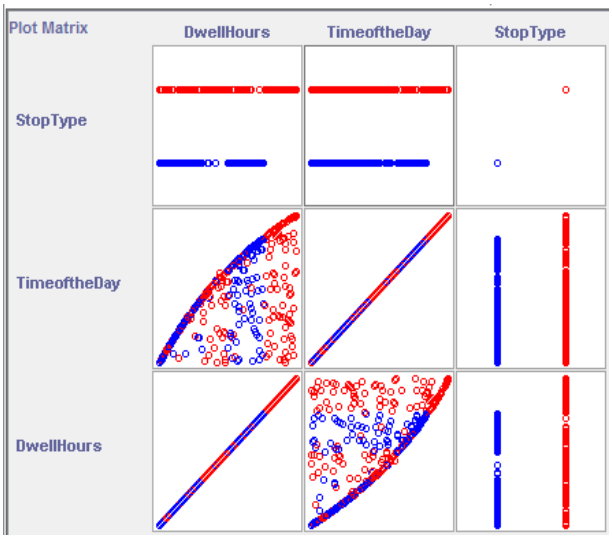

Figure 5.12: Correlation graph of Temporal Features

Cluster Radius (metres)	No of Clusters	True Positive Rate (TPR)	True Negative Rate (TNR)	False Positive Rate (FP)	False Negative Rate (FN)	Total Error	Total Accuracy
100	532	96.42 %	95.14 %	4.86 %	3.58 %	4.2297 %	95.7703 %
150	307	99.77 %	94.34 %	5.66 %	0.23 %	2.9779 %	97.0221 %
200	205	99.69 %	94.42 %	5.58 %	0.31 %	2.9779 %	97.0221 %
250	129	99.39 %	94.65 %	5.35 %	0.61 %	3.0063 %	96.9937 %
300	123	99.47 %	94.65 %	5.35 %	0.53 %	2.9684 %	97.0316 %
350	119	99.47 %	94.65 %	5.35 %	0.53 %	2.9684 %	97.0316 %

Table 5.15: Sample 1 - Effect of varied cluster radius on the stop classification

primary locations instead of declaring the entire bigger cluster as a secondary location. Hence, there is a chance of reducing the false negative error rate with the smaller radius. However, if we keep decreasing the cluster radius beyond certain reasonable value, there will be no more effect on obtaining the correct primary stops, but there will be more chances of getting large number of secondary locations which are misclassified as primary stops leading to high false positive rate. Therefore, it is very important to determine the proper radius of the stop cluster such that we minimize both the false positive and false negative rates.

Results for the two samples show that too small or too big cluster radius affects the overall accuracy. Based on the accuracy results reported for all the samples in Table 5.17,

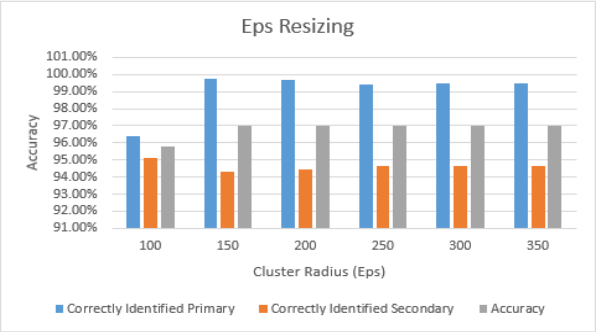

Figure 5.13: Sample 1 - Accuracy graph with varied cluster radius

Cluster Radius (metres)	No of Clusters	True Positive Rate (TPR)	True Negative Rate (TNR)	False Positive Rate (FP)	False Negative Rate (FN)	Total Error	Total Accuracy
100	275	97.02 %	90.92 %	9.08 %	2.98 %	5.6979 %	94.3021 %
150	160	99.23 %	92.80 %	7.20 %	0.77 %	3.6366 %	96.3634 %
200	108	95.29 %	88.73 %	11.27 %	4.71%	7.6266 %	92.3734 %
250	80	87.85 %	95.37 %	4.62 %	12.14 %	8.7898 %	91.2102 %
300	77	87.99 %	95.04 %	4.96 %	12.01 %	8.8634 %	91.1366 %
350	75	88.71 %	94.51 %	5.49 %	11.29 %	8.7014 %	91.2986 %

Table 5.16: Sample 2 - Effect of varied cluster radius on the stop classification

it is clearly observed a safe threshold to choose is 150 metres. This is reasonable, as we start with a smaller radius first, and further expand the cluster based on the density of its border points. With this approach, we reduce the chance of missing the primary locations, and also reduce the false negative rate. But if we consider a larger radius, we have a high possibility of missing majority of primary locations at the first place, increasing the false negative rate.

Hence we choose 150 as the reasonable threshold such that not only the false negative rate is considered, but also the false positive rate is taken into account, which will eventually minimize both the error rates. Radius of 150 metres gives the highest accuracy with minimum error rate.

Experiments to show the selection of reasonable Minpts for stop clusters

Once we have selected a reasonable eps value, we fix its value and perform several

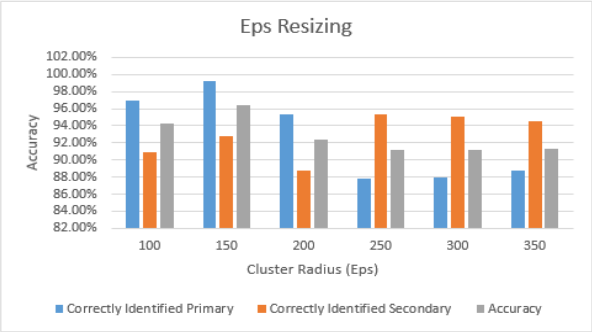

Figure 5.14: Sample 2 - Accuracy graph with varied cluster radius

Cluster Radius (metres)	No of Clusters	True Positive Rate (TPR)	True Negative Rate (TNR)	False Positive Rate (FP)	False Negative Rate (FN)	Total Error	Total Accuracy
100	807	92.24 %	92.70 %	7.30 %	7.76 %	7.532 %	92.468 %
150	467	97.65 %	91.30 %	8.70 %	2.35 %	5.4733 %	94.5267 %
200	313	99.03 %	88.55 %	11.45 %	0.97 %	6.1224 %	93.8776 %
250	209	98.94 %	87.33 %	12.67 %	1.06 %	6.768 %	93.232 %
300	200	99.24 %	86.95 %	13.05 %	0.76 %	6.8039 %	93.1961 %
350	194	99.43 %	81.18 %	18.82 %	0.57 %	7.0837 %	92.9163 %

Table 5.17: Combined Samples - Effect of varied cluster radius on the stop classification

experiments using different minpts and observe its impact on the accuracy of classifying the stop locations. Since we consider the log function over the size of the data points to determine the value of minpts, we take several values close to it to check the accuracy results for classification.

The accuracy results of different minpts value (3,6,12,15) on the two sample datasets are presented in this section as shown in Table 5.18, 5.19 and 5.20. For all the reported sample data, it is observed that there is no significant variation on the accuracy with the change in the minpts value. Lower minpts will lead to the expansion of the initial cluster even if the border points are within sparse region. If the region around the cluster is dense, the cluster will expand greatly but if the surrounding region is sparse, the cluster will not expand noticeably. Larger minpts values will only invoke the expansion of the initial cluster if it is within the dense region and for the sparse regions, it will have no effect on the initial cluster size. Since we have fixed the cluster radius to get minimum classification error rate,

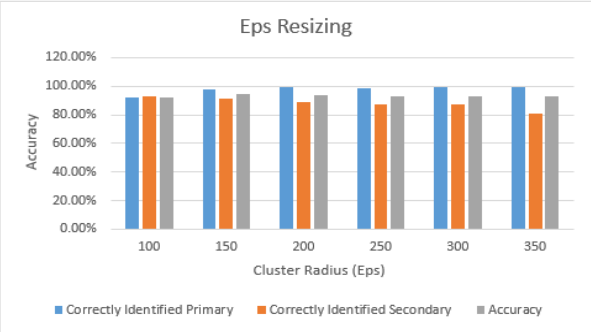

Figure 5.15: Combined Samples - Accuracy graph with varied cluster radius

Minpts	No of Clusters	True Positive Rate (TPR)	True Negative Rate (TNR)	False Positive Rate (FP)	False Negative Rate (FN)	Total Error	Total Accuracy
3	246	100 %	94.21 %	5.79 %	0 %	2.9304 %	97.0696 %
6	276	100 %	94.14 %	5.86 %	0 %	2.9684 %	97.0316 %
10	307	99.77 %	94.34 %	5.66 %	0.23 %	2.9779 %	97.0221 %
12	322	99.73 %	94.48 %	5.52 %	0.27 %	2.9257 %	97.0743 %
15	339	99.73 %	93.90 %	6.10 %	0.27 %	3.2197 %	96.7803 %

Table 5.18: Sample 1 - Effect of varying Minpts on the stop classification

we have reduced the sensitivity of minpts parameter on the overall accuracy which is also observed in the reported results.

5.3.2 Analysis on the performance of the proposed cluster-based classification model

We perform classification on the training data from the three samples considered for cluster-based approach. The results presented in this chapter performs classification by treating all the stops within a given cluster in the same way. However, this treatment introduces bias in the training of the model since all the points within the cluster are assumed to have inherit the same characteristics of the the cluster they belong to. In reality, this is not the case since each point has its own temporal features. The classification results with this approach are presented in this chapter.

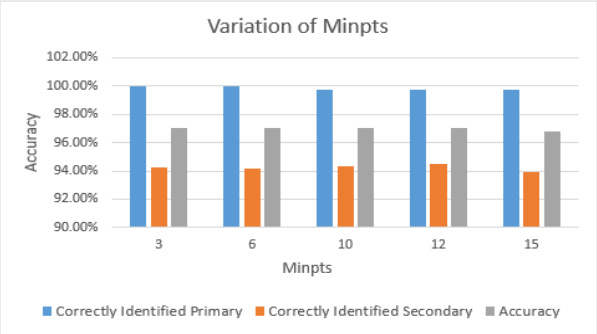

Figure 5.16: Sample 1 - Accuracy graph with varied Minpts

Minpts	No of Clusters	True Positive Rate (TPR)	True Negative Rate (TNR)	False Positive Rate (FP)	False Negative Rate (FN)	Total Error	Total Accuracy
3	136	98.64 %	93.46 %	6.54 %	1.36 %	3.6661 %	96.3339 %
6	149	98.91 %	93.30 %	6.70 %	1.09 %	3.5925 %	96.4075 %
9	160	99.23 %	92.80 %	7.20 %	0.77 %	3.6366 %	96.3634 %
12	175	99.68 %	92.90 %	7.10 %	0.32 %	3.3422 %	96.6578 %
15	185	99.76 %	91.87 %	8.13 %	0.24 %	3.7544 %	96.2456 %

Table 5.19: Sample 2 - Effect of varying Minpts on the stop classification

Analysis on the result of the specialization index and the cluster features for stop purpose classification

As per the results shown in Table 5.21, the accuracy of stop classification using the specialization index is 95.50% and when the specialization index is combined with the cluster features, the accuracy is increased to 99.54%. Combining the temporal features with the specialization index reduces the accuracy of the model which drops to 89.01%.

Analysis on the result of the Entropy index and the cluster features for stop purpose classification

Table 5.22 shows the performance of Entropy Index and its combination with the cluster features and the temporal features. It is clearly observed that the cluster features naturally improves the performance of entropy index. The model is 95.21% accurate with entropy index while its combination with the cluster features increases the accuracy to 99.63%.

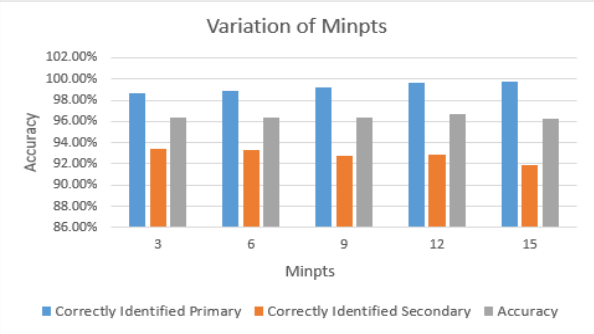

Figure 5.17: Sample 2 - Accuracy graph with varied Minpts

Minpts	No of Clusters	True Positive Rate (TPR)	True Negative Rate (TNR)	False Positive Rate (FP)	False Negative Rate (FN)	Total Error	Total Accuracy
3	382	99.51 %	89.99 %	10.01 %	0.49 %	5.172 %	94.828%
6	425	97.16 %	92.12 %	7.88 %	2.84 %	5.319 %	94.681%
9	467	97.65 %	91.30 %	8.70 %	2.35 %	5.4733 %	94.5267%
12	497	97.45 %	91.97 %	8.03 %	2.55 %	5.2401 %	94.7599%
15	524	97.11 %	91.67 %	8.33 %	2.89 %	5.5629 %	94.4371%

Table 5.20: Combined Sample - Effect of varying Minpts on the stop classification

Temporal features reduces the accuracy in case of entropy index as well.

Analysis on the result of the Formulated index and the cluster features for stop purpose classification

As shown in Table 5.23, the formulated index is 92.25% accurate, but when combined with the cluster features, the accuracy is increased to 99.54%. Temporal features impacting this index also drops the accuracy to 86.58%.

Analysis on the result of the cluster features for stop purpose classification

The results presented in Table 5.24 depicts the direct impact of the cluster features with respect to the performance of our model. Cluster features are accurate enough to independently classify the stop locations giving 99.54% accuracy. Out of the six cluster features, the first three features, i.e., the total trucks, unique carriers and the unique trucks are the

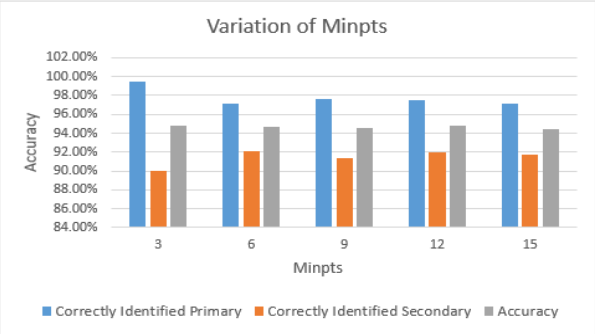

Figure 5.18: Combined Sample - Accuracy graph with varied Minpts

	Random Forest											
Model	Sample 1			Sample2			Sample3			Combined Sample		
	TPR	TNR	Accuracy	TPR	TNR	Accuracy	TPR	TNR	Accuracy	TPR	TNR	Accuracy
SI	99.68%	94.56 %	97.08%	99.10%	93.10 %	96.42%	100%	99.50 %	99.72%	97.92 %	93.1 9%	95.50%
SI + CF	99.95%	99.48 %	99.71%	99.57%	98.84 %	99.24%	100%	99.77 %	99.88%	99.82 %	99.2 %	99.54%
SI + TF	96.07%	94.40 %	95.22%	78.80%	65.69 %	72.95%	95.7%	94.05 %	94.79%	89.75 %	88.3 %	89.01%
SI + CF + TF	99.29%	97.96 %	98.61%	98.20%	94.88 %	96.71%	99.9%	99.52 %	99.73%	99.88 %	99.7 %	99.79%

Table 5.21: Performance of the model over the specialization index and its combination with the cluster and temporal features

most prominent ones compared to the minimum, maximum and the standard deviation as per the reported results. Since minimum, maximum and the standard deviation are the statistical features calculated based on the total trucks and unique carriers, we can say that total trucks and unique carriers are the base features giving highest accuracy.This is the reason why it greatly improves the performance of the overall model when combined with different index used in classifying the stop locations.

Analysis on the performance of the proposed cluster-based classification model with single point per cluster

We perform classification on the training data for the combined sample considered for

	Random Forest											
Model	Sample 1			Sample2			Sample3			Combined Sample		
	TPR	TNR	Accuracy	TPR	TNR	Accuracy	TPR	TNR	Accuracy	TPR	TNR	Accuracy
EI	98.54%	96.46 %	97.48%	96.01%	95.60 %	95.83%	98.17 %	98.75 %	98.51%	96.81 %	93.6 7%	95.21%
EI + CF	99.79%	99.73 %	99.76%	99.54%	99.24 %	99.41%	99.89 %	99.84 %	99.88%	99.64 %	99.6 2%	99.63%
EI + TF	93.11%	93.45 %	93.28%	94.63%	87.68 %	91.53%	95.24 %	93.89 %	94.49%	89.28 %	89.0 5%	89.16%
EI + CF + TF	99.57%	98.27 %	98.91%	99.12%	97.45 %	98.38%	100%	99.52 %	99.74%	99%	95.6 2%	97.27%

Table 5.22: Performance of the model over the entropy index and its combination with the cluster and temporal features

	Random Forest											
Model	Sample 1			Sample2			Sample3			Combined Sample		
	TPR	TNR	Accuracy	TPR	TNR	Accuracy	TPR	TNR	Accuracy	TPR	TNR	Accuracy
FI	83.10%	98.77 %	91.03 %	94.02%	87.54 %	91.13%	100%	99.49 %	99.72%	89.41 %	94.9 6%	92.25%
FI + CF	99.95%	99.48 %	99.71%	99.57%	98.84 %	99.24%	100%	99.77 %	99.88%	99.82 %	99.2 6%	99.54%
FI + TF	86.09%	91.36 %	88.76%	73.27%	50.59 %	63.16%	95.69 %	94.05 %	94.77%	85.44 %	87.6 7%	86.58%
FI + CF + TF	99.82%	98.48 %	99.14%	98.00%	95.47 %	96.87%	99.97 %	99.52 %	99.73%	98.97 %	95.6 2%	97.26%

Table 5.23: Performance of the model over the formulated index and its combination with the cluster and temporal features

cluster-based approach. The classification results presented in Table 5.25 were performed considering only a single stop point per cluster. Since the approach of considering all the stop points within a cluster bias the training model, we take only one stop point per cluster to train the classification model to generate unbiased results. Table 5.25 shows the performance of the specialization index and its combination with the cluster features for the combined sample considering one point per each cluster. We can observe that the overall accuracy is noticeably reduced because the accuracy in classifying the secondary locations is quite low. The true negative rate is affected by majority of the bigger cluster containing

	Random Forest Classifier		
Model	True Positive Rate	True Negative Rate	Accuracy
Cluster Features: Total Trucks, Unique Carriers, Unique Trucks, Minimum, Maximum and Standard Deviation	99.82%	99.2%	99.54%
Cluster Features: Total Trucks, Unique Carriers & Unique Trucks	99.82%	99.2%	99.54%
Cluster Features: Minimum, Maximum and Standard Deviation	99.79%	98.21%	98.98%

Table 5.24: Performance of the model over the cluster features

	Random Forest Classifier		
Model	**True Positive Rate**	**True Negative Rate**	**Accuracy**
Specialization Index (SI)	95.5%	25.2%	75.5793 %
Specialization Index+ (Cluster Features) Cluster Features: Total Trucks, Unique Carriers, Unique Trucks	**92.1**%	**38.4**%	**76.8271** %
Specialization Index+ (Cluster Features) + (Temporal Features) Temporal Features: Dwell Time and Time of the day	96.3%	23.3%	75.5793 %

Table 5.25: Performance of the cluster-based with single point classification model

both the primary and secondary stops.

Analysis on the performance of the proposed classification model on the unseen test data

Experiments to predict the class labels for unseen data are also carried out on both the point-based and cluster-based classification model to test their accuracy. A total of 53 stop locations are provided as the unseen test data to our model. The predicted class labels were manually checked using Google Maps to verify the accuracy of the classification model. The manual validation shows that out of 53 stop locations, 7 secondary stops were

misclassified as primary stops and 1 primary stop was misclassified as secondary stop using the cluster-based approach considering all the stop points per cluster. This corresponds to 85% accuracy. On the other hand, the cluster-based clasification with one point per cluster gives 9 secondary stops misclassified as primary stops and 4 primary stops misclassified as secondary stops resulting in 75% accuracy. The point-based approach misclassify 4 secondary stops as primary stops and 1 primary stop as a secondary stop. The point-based approach corresponds to the maximum accuracy giving 91% accurate predictions on the unseen test data.

5.4 Discussion on the effectiveness of Point-Based and Cluster-Based approach

Both the proposed approach has their own advantages and disadvantages in terms of efficiency and accuracy. Point-Based approach has a major drawback in terms of efficiency. Since point-based method processes each point individually, it is less efficient as compared to the cluster-based approach. While cluster-based approach, on the other hand, processes each stop cluster individually, it is more efficient in terms of computational complexity in performing classification. In terms of accuracy, the point-based method proves to be more accurate since we apply simple clustering without any cluster expansion. This simple clustering without cluster expansion helps in distinguishing between a primary and secondary location when they are close to each other. The cluster-based approach fails to distinguish accurately when both the primary and secondary locations are nearby due to the cluster expansion process resulting both the locations combined into one cluster giving low accurate classification results. Summarizing both the approach, we can conclude that the point-based method produces more accurate results but the cluster based is more time efficient. Since we need to discover the tours at the end, it is really important to accurately classify

the stop points.Therefore, the point based approach is selected to identify the purpose of truck stop events to construct the truck tours in the thesis. As there is a trade-off between accuracy and efficiency, we choose accuracy in order to obtain accurate tours.

5.5 Analysis on the Tours

The truck tours that we discover starting from each firm, making multiple stops including both the primary and the secondary and again returning to the initial industry location are presented with all the moving and stop pings with complete labeling of firm, primary locations and secondary locations. The stop locations of the tours presented in this section are labeled using our proposed classification model. We provide the stop locations as the unseen test data to our classification model and obtain the predicted class labels for all the stops. A total of 53 stop locations belonging to three sample tours are provided as the unseen test data to our model. The predicted class labels were manually checked using Google Maps to verify the accuracy of the classification model. The manual verification shows that out of 53 stop locations, 4 secondary stops were misclassified as primary stops and 1 primary stop was misclassified as secondary stop. Hence, the proposed model gives good accuracy for unseen data as well. The accurate predicted class labels obtained using our model helps in discovering the accurate tours.

A sample of one carrier and GPS pings of three different trucks belonging to the sample carrier are considered for reporting the results of truck tours starting from a firm.

The GPS pings of three different sample trucks belonging to the same sample carrier shown in Fig. 5.19 depicts the firm location of the carrier as all the trucks starts from its initial firm location for making a tour.

Fig. 5.20 shows the visualization of three tours made by one of the truck with the firm and both the primary and secondary stops labeled.

Fig 5.21 shows the sample tour which constitutes to a long tour making large number

of stops while forming a round-trip. This tour makes a total of 12 primary stops and 18 secondary stops in one tour.

The second sample tour constitutes of 2 primary stops and 8 secondary stops as shown in Fig. 5.22.

The last sample tour represents a small tour that takes the same route while returning to the firm making 2 primary stops and 5 secondary stops as presented in Fig. 5.23.

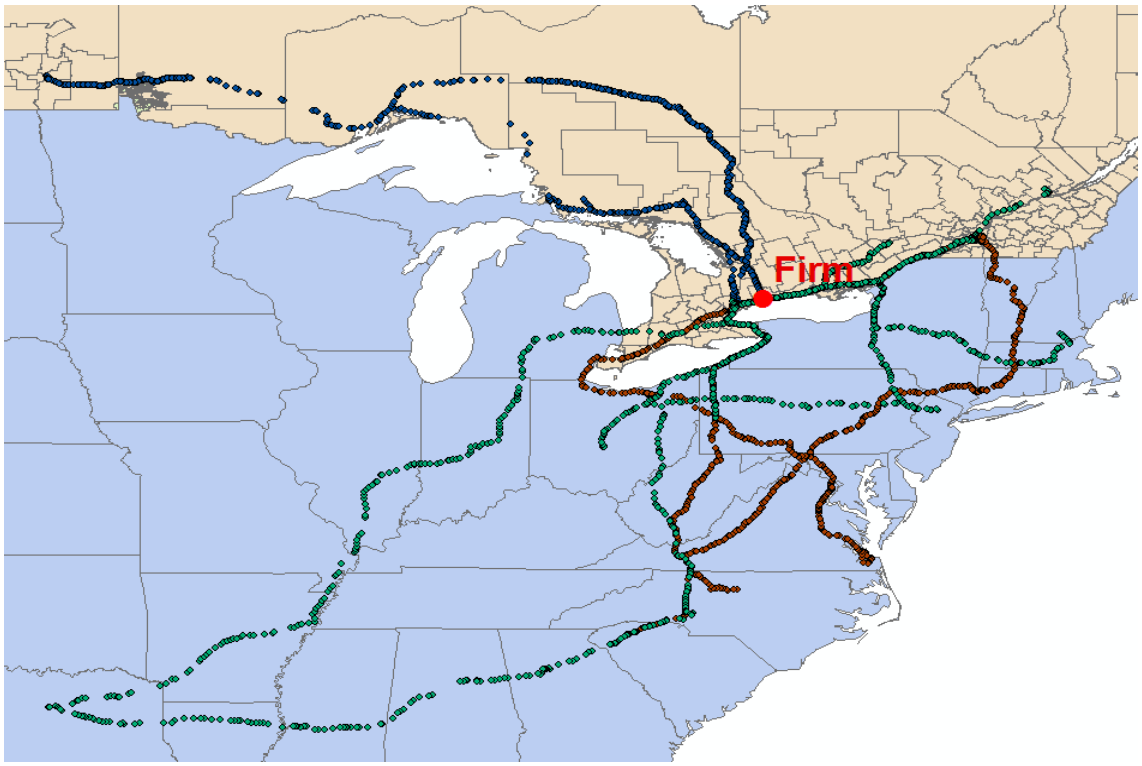

Figure 5.19: GPS pings of three trucks for a sample carrier showing Firm

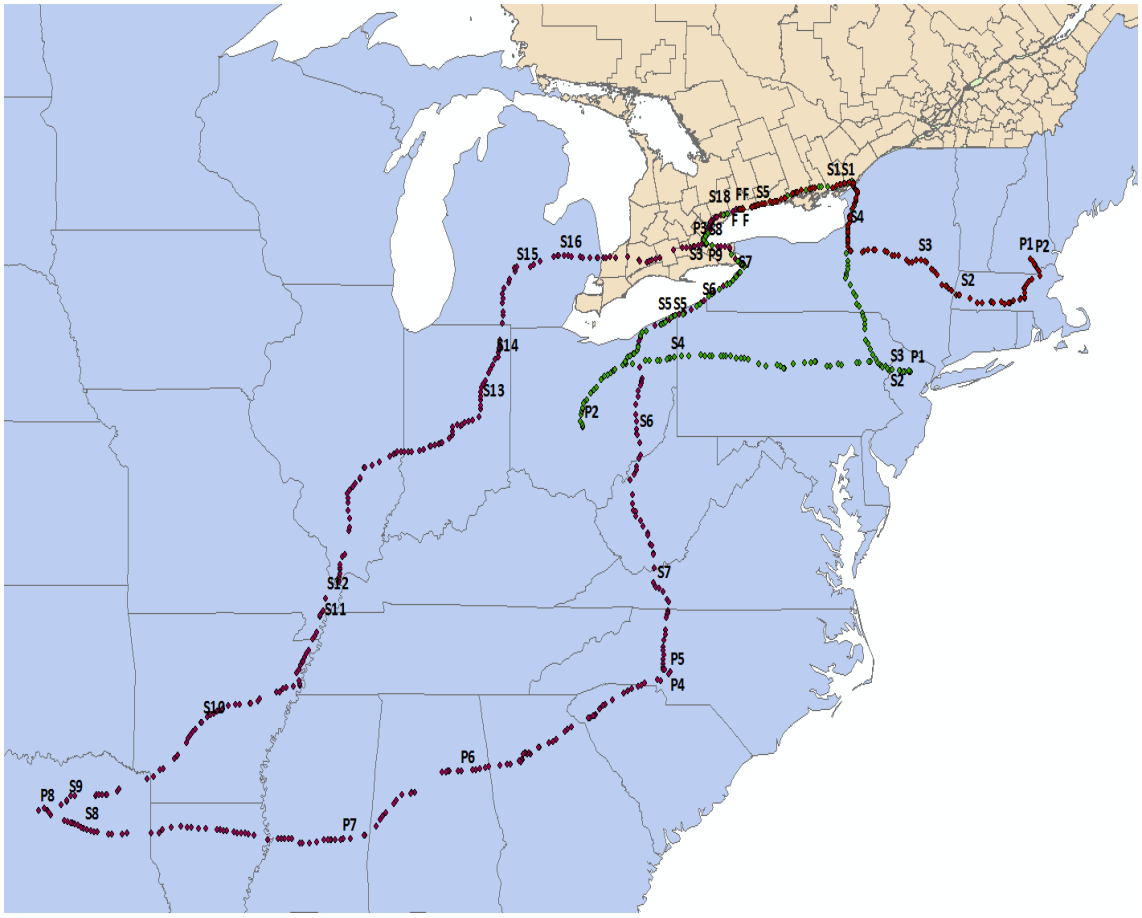

Figure 5.20: Three Sample Tours of a truck

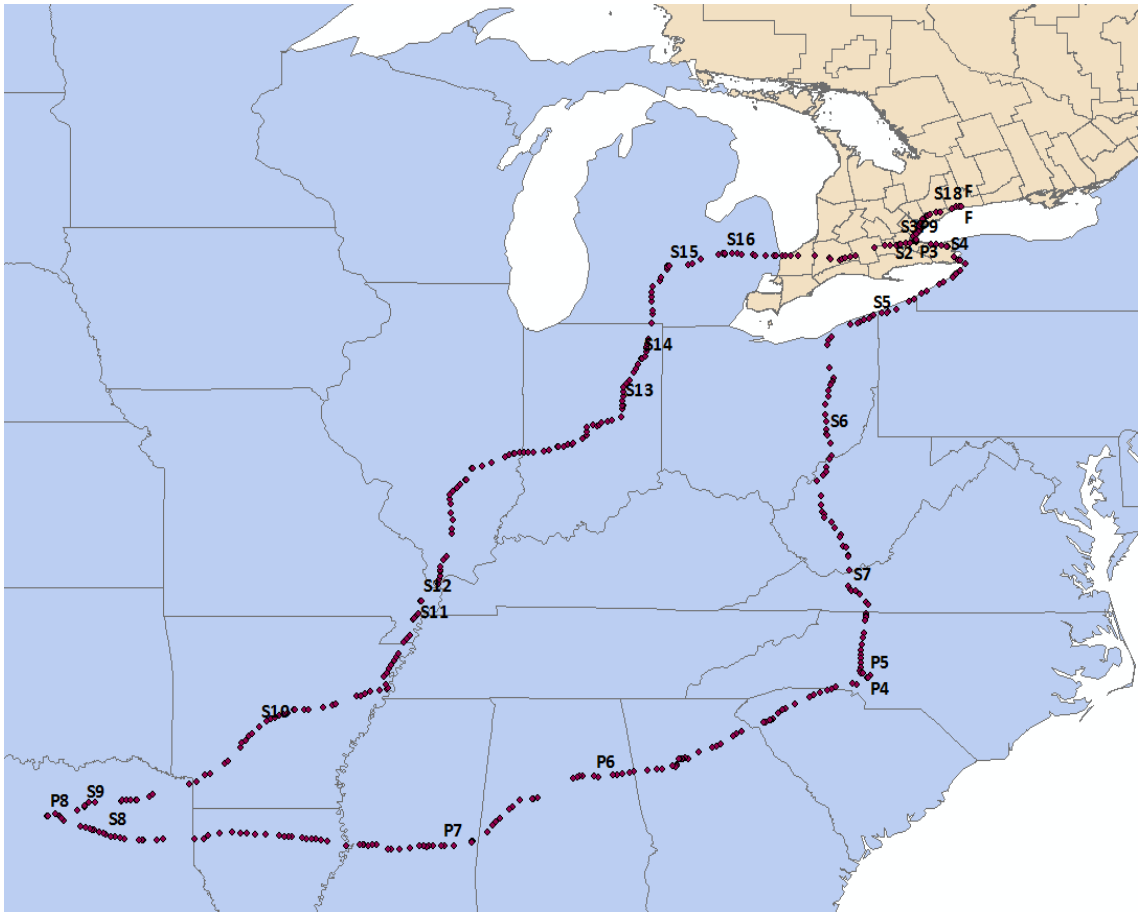

Figure 5.21: Sample Tour 1

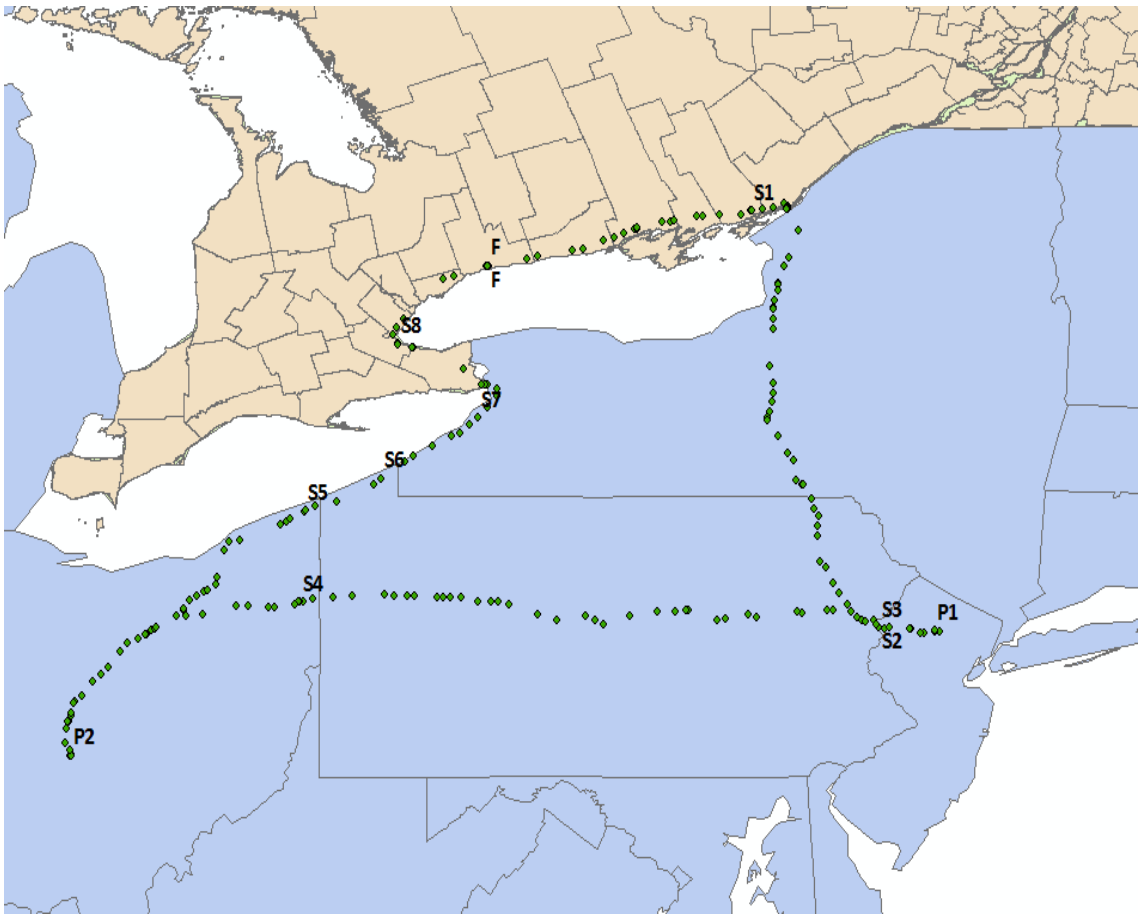

Figure 5.22: Sample Tour 2

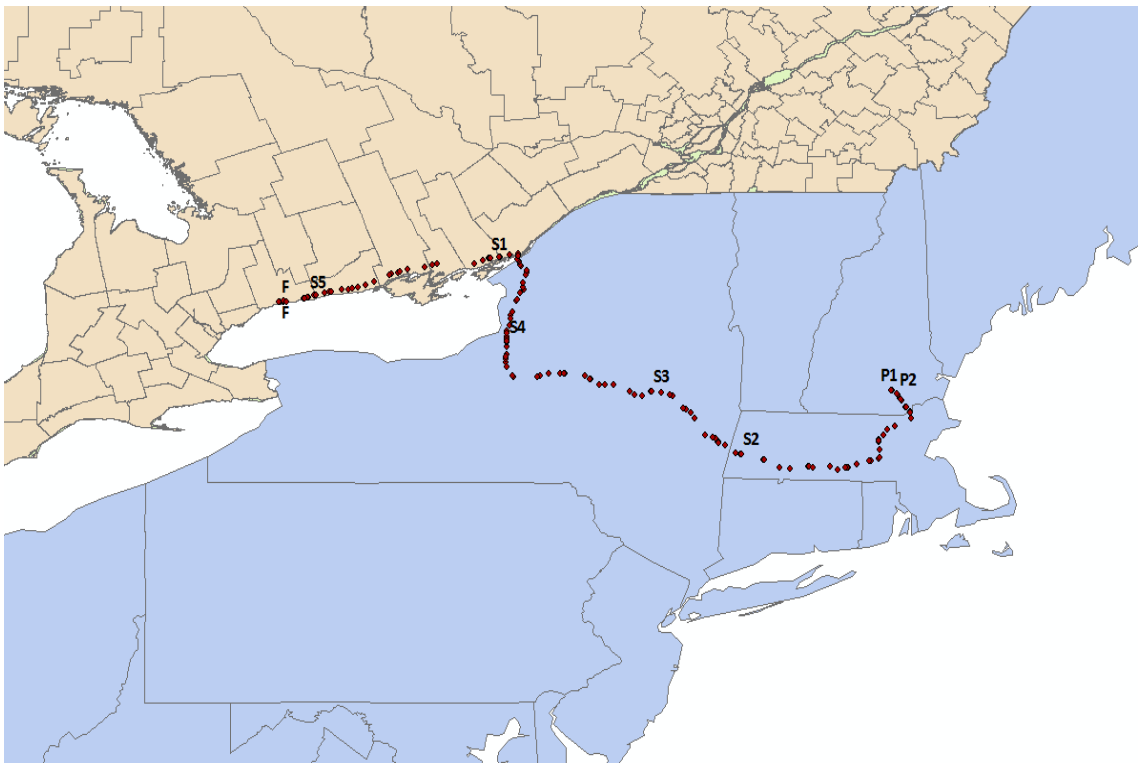

Figure 5.23: Sample Tour 3

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this research work, we have focused on clustering techniques to mine truck GPS data to interpret and discover meaningful information related to freight transportation. Major contribution of this research is the method for classifying the stop locations into primary and secondary locations. To solve this classification problem, several statistical measures has been applied to obtain better accuracy. The first statistical index named specialization index gives 91.90% accuracy while the entropy index is 95.14% accurate in classifying the stop locations. To further improve the accuracy, various stop cluster features such as minimum, maximum, standard deviation of the percentage of truck count per unique carrier for each stop cluster, total trucks, unique carrier count and unique truck count were developed that defines the homogeneity and heterogeneity of a stop cluster. Out of these six cluster features listed, three features, i.e., the Total Trucks, Unique Carrier and Unique Trucks proved to be the best features for the model since they are the base features for the stop purpose classification. Combining these cluster features with the specialization index greatly improves the classification results giving the best accuracy i.e.98.79% as compared to the entropy index which is 95.14% accurate. Other temporal features such as dwell time and time of the day are also considered and combined with the specialization index to check its impact on our model. The results show that temporal features reduces the accuracy of the model to 87.92% as they do not prove relevant in case of our dataset. The results demonstrated that the cluster features are important impact features in solving the classification problem of identifying the purpose of stop truck events.

6.2 Future work

Potential future work could be conceptualized in a direction of classifying the stop locations more accurately. The current cluster-based approach encounters the problem of generating bigger clusters consisting of the combination of both primary and secondary locations. The goal is to focus on improving the cluster-based classification approach by extending the current clustering technique in a way to obtain a fine quality clusters. The idea is to perform iterative clustering on the bigger clusters and generate smaller clusters in a way to avoid the combination of different location type into one cluster. This method of generating fine clusters could be a possible future work in classifying the stop locations more accurately.

BIBLIOGRAPHY

[1] B. Morser, "How does gps work?" 2012. [Online]. Available: https://timeandnavigation.si.edu/multimedia-asset/how-does-gps-work

[2] A. Annaldas. A GENTLE INTRODUCTION TO DBSCAN. [Online]. Available: https://mineracaodedados.wordpress.com/2018/02/09/a-gentle-introduction-to-dbscan/

[3] K. Gingerich, H. Maoh, and W. Anderson, "Classifying the purpose of stopped truck events: An application of entropy to GPS data," *Transportation Research Part C: Emerging Technologies*, vol. 64, pp. 17–27, mar 2016.

[4] "Transportation and the economy." [Online]. Available: https://www.tc.gc.ca/eng/policy/anre-menu-3016.htm#footnotes

[5] "Transportation in canada." [Online]. Available: https://www.tc.gc.ca/media/documents/policy/2015_TC_Annual_Report_Overview-EN-Accessible.pdf

[6] J.-P. Rodrigue, C. Comtois, and B. Slack, *Methods in Transport Geography*, 01 2016.

[7] M. Ferguson, H. Maoh, J. Ryan, P. Kanaroglou, and T. H. Rashidi, "Transferability and enhancement of a microsimulation model for estimating urban commercial vehicle movements," *Journal of Transport Geography*, vol. 24, pp. 358 – 369, 2012.

[8] K. Gingerich, H. Maoh, and W. Anderson, "Modelling the determinants of truck tours within canadian markets," in *In Proceedings of Annual Canadian Transportation Research Forum (CTRF) conference*, 2015, pp. 307–321.

[9] E. D. Kaplan and C. J. Hegarty, "Understanding gps: Principles and applications." [Online]. Available: http://d1.amobbs.com/bbs_upload782111/files_33/ourdev_584835O21W59.pdf

[10] "Global positioning system(GPS)." [Online]. Available: https://www.slideshare.net/shifasadia/global-positioning-systemgps-44678563/3

[11] GPS Basics. [Online]. Available: https://learn.sparkfun.com/tutorials/gps-basics/

[12] "Cluster analysis — Wikipedia, the free encyclopedia." [Online]. Available: https://en.wikipedia.org/wiki/Cluster_analysis#Applications

[13] *Grid-Based Clustering Algorithms*, ch. 12, pp. 209–217. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9780898718348.ch12

[14] P. Grabusts and A. Borisov, "Using grid-clustering methods in data classification," in *Proceedings. International Conference on Parallel Computing in Electrical Engineering*, Sept 2002, pp. 425–426.

[15] "Clustering grid-based methods." [Online]. Available: http://dbmanagement.info/Books/MIX/07_DataMining_Clustering_GridBased_Data_Mining.pptx

[16] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," *SIGMOD Rec.*, vol. 27, no. 2, pp. 94–105, Jun. 1998.

[17] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "Wavecluster: A wavelet-based clustering approach for spatial data in very large databases," *The VLDB Journal*, vol. 8, no. 3-4, pp. 289–304, Feb. 2000.

[18] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996, pp. 226–231.

[19] "DBSCAN — Wikipedia, the free encyclopedia." [Online]. Available: https://en.wikipedia.org/wiki/DBSCAN

[20] T. H. R. T. Gareth James, Daniela Witten, *Classification*.

[21] M. Sidana. Types of classification algorithms in machine learning. [Online]. Available: https://medium.com/@sifium/machine-learning-types-of-classification-9497bd4f2e14

[22] L. O. Alvares, V. Bogorny, B. Kuijpers, J. A. F. de Macedo, B. Moelans, and A. Vaisman, "A model for enriching trajectories with semantic geographical information," in *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*, ser. GIS '07. New York, NY, USA: ACM, 2007, pp. 22:1–22:8. [Online]. Available: http://doi.acm.org/10.1145/1341012.1341041

[23] K. Xie, K. Deng, and X. Zhou, "From trajectories to activities: A spatio-temporal join approach," in *Proceedings of the 2009 International Workshop on Location Based Social Networks*. New York, NY, USA: ACM, 2009, pp. 25–32. [Online]. Available: http://doi.acm.org/10.1145/1629890.1629897

[24] D. Ashbrook and T. Starner, "Using gps to learn significant locations and predict movement across multiple users," *Personal and Ubiquitous Computing*, vol. 7, no. 5, pp. 275–286, Oct 2003. [Online]. Available: https://doi.org/10.1007/s00779-003-0240-0

[25] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen, "Discovering personally meaningful places: An interactive clustering approach," *ACM Trans. Inf. Syst.*, vol. 25, no. 3, jul 2007.

[26] S. Hwang, C. Evans, and T. Hanke, *Detecting Stop Episodes from GPS Trajectories with Gaps*. Springer International Publishing, 2017, pp. 427–439. [Online]. Available: https://doi.org/10.1007/978-3-319-40902-3_23

[27] A. T. Palma, V. Bogorny, B. Kuijpers, and L. O. Alvares, "A clustering-based approach for discovering interesting places in trajectories," in *Proceedings of the 2008 ACM Symposium on Applied Computing*. ACM, 2008, pp. 863–868.

[28] Z. Xiu-Li and X. Wei-Xiang, "A clustering-based approach for discovering interesting places in a single trajectory," in *2009 Second International Conference on Intelligent Computation Technology and Automation*, vol. 3, Oct 2009, pp. 429–432.

[29] J. A. M. R. Rocha, V. C. Times, G. T. D. Oliveira, L. O. Alvares, and V. Bogorny, "Db-smot: A direction-based spatio-temporal clustering method," *2010 5th IEEE International Conference Intelligent Systems*, pp. 114–119, 2010.

[30] Z. Fu, Z. Tian, Y. Xu, and C. Qiao, "A two-step clustering approach to extract locations from individual gps trajectory data," *ISPRS Int. J. Geo-Information*, vol. 5, p. 166, 2016.

[31] L. Gong, H. Sato, T. Yamamoto, T. Miwa, and T. Morikawa, "Identification of activity stop locations in gps trajectories by density-based clustering method combined with support vector machines," *Journal of Modern Transportation*, vol. 23, no. 3, pp. 202–213, 09 2015.

[32] M. Lv, L. Chen, Z. Xu, Y. Li, and G. Chen, "The discovery of personally semantic places based on trajectory data mining," *Neurocomputing*, vol. 173, pp. 1142 – 1153, 2016.

[33] W. Chen, M. Ji, and J. Wang, "T-dbscan: A spatiotemporal density clustering for gps trajectory segmentation," *International Journal of Online Engineering (iJOE)*, vol. 10, p. 19, 10 2014.

[34] B. Thierry, B. Chaix, and Y. Kestens, "Detecting activity locations from raw gps data: a novel kernel-based algorithm," *International Journal of Health Geographics*,

vol. 12, no. 1, pp. 1–10, Mar 2013.

[35] R. Hariharan and K. Toyama, "Project lachesis: Parsing and modeling location histories," in *Geographic Information Science*. Springer Berlin Heidelberg, 2004, pp. 106–124.

[36] D. Ashbrook and T. Starner, "Learning significant locations and predicting user movement with gps," in *Proceedings. Sixth International Symposium on Wearable Computers,*, Oct 2002, pp. 101–108.

[37] J. H. Kang, h. Welbourne, B. Stewart, and G. Borriello, "Extracting places from traces of locations," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 3, pp. 58–68, Jul. 2005.

[38] B. Silverman, *Density Estimation For Statistics And Data Analysis*, 01 1986, vol. Vol. 26.

[39] T. Luo, X. Zheng, G. Xu, K. Fu, and W. Ren, "An improved dbscan algorithm to detect stops in individual trajectories," vol. 6, p. 63, 02 2017.

[40] A. Moreira and M. Y. Santos, "Density-based clustering algorithms - DBSCAN and SNN," 2005.

[41] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." AAAI Press, 1996, pp. 226–231.

[42] L. Ertz, M. Steinbach, and V. Kumar, "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data," in *in Proceedings of Second SIAM International Conference on Data Mining*, 2003.

[43] M. Elbatta and W. Ashour, "A dynamic method for discovering density varied clusters," vol. 6, pp. 123–134, 02 2013.

[44] P. Liu, D. Zhou, and N. Wu, "Vdbscan: Varied density based spatial clustering of applications with noise," *2007 International Conference on Service Systems and Service Management*, pp. 1–4, 2007.

[45] A. M. Fahim, G. Saake, A.-B. M. Salem, F. A. Torkey, and M. A. Ramadan, "An enhanced density based spatial clustering of applications with noise," *2009 IEEE International Advance Computing Conference*, pp. 1475–1478, 2009.

[46] M. V. Sowjanya and T. M. Padmaja, "Varied density based graph clustering algorithm for social networks," *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 520–524, 2017.

[47] T. H. Chuyen Luong, Son Do and Deokjai, "A method for detecting significant places from gps trajectory data," in *Journal of Advances in Information Technology*, vol. 6, no. 1, February 2015, pp. 44–48.

[48] C. Tsai and C. Sung, "An extended improving dbscan algorithm with sampling techniques," *Int. J. Bus. Intell. Data Min.*, vol. 5, no. 1, pp. 94–111, December 2010.

[49] X. Xu, M. Ester, H. . Kriegel, and J. Sander, "A distribution-based clustering algorithm for mining in large spatial databases," in *Proceedings 14th International Conference on Data Engineering*. IEEE, Feb 1998, pp. 324–331.

[50] X. Yu, D. Zhou, and Y. Zhou, "A new clustering algorithm based on distance and density," in *Proceedings of ICSSSM '05. 2005 International Conference on Services Systems and Services Management, 2005.*, vol. 2, June 2005, pp. 1016–1021.

[51] E. Rosalina, F. D. Salim, and T. Sellis, "Automated density-based clustering of spatial urban data for interactive data exploration," in *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, May 2017, pp. 295–300.

[52] D. Birant and A. Kut, "St-dbscan: An algorithm for clustering spatial-temporal data," *Data Knowl. Eng.*, vol. 60, no. 1, pp. 208–221, Jan. 2007.

[53] S. Neha and G. Amit, "AGED (automatic generation of eps for dbscan)," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 14, no. 5, pp. 536–559, May 2016.

[54] M. N. Gaonkar and K. Sawant, "Autoepsdbscan : Dbscan with eps automatic for large dataset," *International Journal on Advanced Computer Theory and Engineering (IJACTE)*, vol. 2, pp. 11–16, 2013.

[55] P. Liu, D. Zhou, and N. Wu, "Vdbscan: Varied density based spatial clustering of applications with noise," in *2007 International Conference on Service Systems and Service Management*, June 2007, pp. 1–4.

[56] S. Mitra and J. N, "Kddclus: A simple method for multi-density clustering," 2010.

[57] S. Gayathri, M. M. Metilda, and S. S. Babu, "A shared nearest neighbour density based clustering approach on a proclus method to cluster high dimensional data," *Indian Journal of Science and Technology*, vol. 8, no. 22, 2015. [Online]. Available: http://www.indjst.org/index.php/indjst/article/view/79131

[58] D. Guo, S. Liu, and H. Jin, "A graph-based approach to vehicle trajectory analysis," *Journal of Location Based Services*, vol. 4, no. 3-4, pp. 183–199, 2010.

[59] D. Guo, "Flow mapping and multivariate visualization of large spatial interaction data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1041–1048, Nov 2009.

[60] M. K. El Mahrsi and F. Rossi, "Graph-based approaches to clustering network-constrained trajectory data," in *Proceedings of the First International Conference on New Frontiers in Mining Complex Patterns*. Springer-Verlag, 2013, pp. 124–137.

[61] B. Shams and S. Haratizadeh, "Graphloc: a graph based approach for automatic detection of significant locations from gps trajectory data," vol. 63, no. 1, pp. 115–134, 2018.

[62] A. Clauset, M. E J Newman, and C. Moore, "Finding community structure in very large networks," vol. 70, 01 2005.

[63] L. Sarti, L. Bravi, F. Sambo, L. Taccari, M. Simoncini, S. Salti, and A. Lori, "Stop purpose classification from gps data of commercial vehicle fleets," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, Nov 2017, pp. 280–287.

[64] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

VITA AUCTORIS

NAME:	Vidhi Patel
PLACE OF BIRTH:	Gujarat, India
YEAR OF BIRTH:	1993
EDUCATION:	Bhartiya Vidya Bhavan's Narmada Vidyalaya
	Gujarat, India, 2011
	Dharmsinh Desai University, B.Tech in Computer Engineering
	Gujarat, India, 2015
	University of Windsor, M.Sc
	Windsor, ON, 2019