TRAFFIC ANALYSIS
USING GPS TRAJECTORY RECORDS
OF COURIER MOTORCYCLES

by

Yiğit Çetinel

B.S., Civil Engineering, Boğaziçi University, 2014

M.S., Civil Engineering, Boğaziçi University, 2017

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Civil Engineering

Boğaziçi University

2024

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and deepest appreciation to my thesis supervisor Prof. Ilgın Yaşar for her invaluable guidance, innovative suggestions, continued encouragement, remarkable patience, and enthusiasm throughout this study which consists of this thesis, one journal, and one conference paper.

I wish to thank Prof. Özer Çinicioğlu for his unconditional and valuable guidance for this thesis. I would like to thank the member of the thesis committee, Assoc. Prof. Gürkan Günay for his valuable comments and precious suggestions.

I am indebted to my family for the endless support, encouragement, and love they have given. I cannot forget the support of my dear wife, mother, and father.

ABSTRACT

TRAFFIC ANALYSIS USING GPS TRAJECTORY RECORDS OF COURIER MOTORCYCLES

In recent years, the use of motorcycles has witnessed a remarkable surge in urban areas, paralleled by a growing demand for research and development in the motorcycle industry. Furthermore, the widespread adoption of GPS-enabled devices over the last few decades has opened up exciting possibilities, particularly in the realm of data analysis, where motorcycle GPS data has emerged as a valuable resource for various applications. This study presents a novel methodology for estimating the travel duration of powered two-wheelers (PTWs) in heterogeneous traffic using GPS data generated by motorcycles on urban road networks. The proposed methodology has the potential to offer valuable insights into the behavior of PTWs in heterogeneous traffic environments. By analyzing big data generated by GPS-based trajectory data, researchers can identify areas with high motorcycle density and pinpoint potential bottlenecks that impact travel times. Temporal data storing with bearing information in hexagonal shards called "bubbles" enables researchers to utilize big data more efficiently. Spatial transformation, Kalman filtering, and map-matching of the trajectory data significantly enhance the quality of the data. In this study, the 10-minute interval is performed as optimal for estimating travel time with 4.3% MAPE. Furthermore, combining historical bubble data with a 0.35 scale factor improves MAPE by 9.6%. Despite the limitations, not only is the transferability of this methodology noteworthy, but it is also opening the door to broader applications in diverse urban settings.

ÖZET

KURYE MOTORSİKLETLERİNİN GPS ROTA VERİLERİYLE TRAFİK ANALİZİ

Son yıllarda, motosiklet kullanımı kentsel alanlarda dikkate değer bir artış göstermiş, bu durum motosiklet endüstrisinde araştırma ve geliştirme talebinin artmasıyla paralellik göstermiştir. Ayrıca, son birkaç on yılda GPS etkin cihazların yaygınlaşması, özellikle veri analizi alanında, çeşitli uygulamalar için değerli bir kaynak olarak ortaya çıkan motosiklet GPS verileriyle ilgili heyecan verici olasılıkları açmıştır. Bu çalışma, kentsel yol ağlarındaki motosikletler tarafından üretilen GPS verilerini kullanarak heterojen trafikte motorlu iki tekerlekli araçların (PTW) seyahat süresini tahmin etmek için yenilikçi bir metodoloji sunmaktadır. Önerilen metodoloji, heterojen trafik ortamlarında PTW'lerin davranışlarına ilişkin değerli içgörüler sunma potansiyeline sahiptir. GPS tabanlı yol verileri ile üretilen büyük verileri analiz ederek, araştırmacılar yüksek motosiklet yoğunluğuna sahip alanları belirleyebilir ve seyahat sürelerini etkileyen potansiyel darboğazları tespit edebilirler. Zaman verilerinin, 'kabarcık' adı verilen altıgen parçalarda yön bilgileriyle birlikte depolanması, araştırmacıların büyük verileri daha verimli kullanmalarını sağlar. Yol verilerinin mekansal dönüşümü, Kalman filtrelemesi ve harita eşleştirilmesi, veri kalitesini önemli ölçüde artırır. Bu çalışmada, seyahat süresini tahmin etmek için en uygun 10 dakikalık aralık %4.3 MAPE ile gerçekleştirilmiştir. Ayrıca, tarihi kabarcık verilerinin 0.35 ölçek faktörü ile birleştirilmesi MAPE'yi %9.6 oranında iyileştirmektedir. Sınırlamalarına rağmen, bu metodolojinin taşınabilirliği sadece dikkate değer değil, aynı zamanda çeşitli kentsel ortamlarda daha geniş uygulamalara kapı açmaktadır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
TABLE OF CONTENTS vi
LIST OF FIGURES viii
LIST OF TABLES xi
LIST OF ACRONYMS/ABBREVIATIONS xii
1. INTRODUCTION & BACKGROUND 1
1.1. Problem Statement 4
1.2. The Goal and Objectives 5
1.3. Literature Review 6
1.3.1. Heterogeneous Traffic with Powered Two-Wheelers 7
1.3.2. Trajectory Datasets for Powered Two-Wheelers 8
1.3.3. GPS as an Economic Intelligent Transportation System . 9
1.3.4. Navigation Systems 10
1.3.5. Simulation Models 10
1.3.6. Accident Analytics 11
1.3.7. GPS Processing 11
1.3.8. Discussions & Conclusion 13
1.4. Thesis Outline 14
2. THEORY 15
2.1. Spatial Transformation 16
2.2. Kalman Filtering 19
2.3. Bubble Generation 22
2.4. K-Means Clustering 25
2.5. Exponentially Weighted Moving Average 27

3. METHODOLOGY 31
3.1. Proposed Method 32
3.2. Study Area 39
3.3. Data Structure and Spatial Transformations 41
3.4. Distance and Speed Calculations 43
3.5. Data Filtering 45
3.6. Bubbles and Locational Grouping 48
3.7. Bearing Clustering 51
3.8. Map Matching 54
3.9. Speed Analysis 56
3.10. Travel Time Estimation 57
3.11. Case Study 59
4. ANALYSIS RESULTS 62
5. CONCLUSION 69
REFERENCES 71

LIST OF FIGURES

Figure 1.1. Latitude, longitude, time, and accuracy differences. 2

Figure 1.2. One hour and one day of the dataset in a specific area (Rumelihisarı, İstanbul). 3

Figure 2.1. Algorithm representation of EPSG 5637 (TUREF) projection. 18

Figure 2.2. Kalman filter state estimation. 20

Figure 2.3. Square vs hexagonal packing of circles, and hexagons. 22

Figure 2.4. Aggregated data visualization in different bin sizes. 25

Figure 2.5. Line plot between K and inertia. 27

Figure 3.1. Main sections of the proposed methodology. 32

Figure 3.2. Detailed outline of the proposed methodology. 33

Figure 3.3. Get the location and time information using the Android API. 34

Figure 3.4. Extract data with different options using Spark. 35

Figure 3.5. Python PyProj transformer example. 36

Figure 3.6. Lag error in Kalman filtering. 37

Figure 3.7. Creating a hexagonal spatial index in Python. 38

Figure 3.8. Location of the study area (Nispetiye Street). 40

Figure 3.9. One-hour trajectory points on Nispetiye Street encapsulated by a 100-meter buffer zone. 40

Figure 3.10. Anomalies in speed profiles calculated using raw data. 44

Figure 3.11. True position (red), measurement (green), estimation (gray). 45

Figure 3.12. (Python) Filtering speed profiles using Kalman Filter. 46

Figure 3.13. Some of the speed profiles (km/h) before and after the smoothing. 47

Figure 3.14. The speed profiles of the study area on the map. 48

Figure 3.15. Bubbles: GPS logs of motorcycles. 49

Figure 3.16. Data points of a "bubble". 50

Figure 3.17. Flow of points with bearing (angle) information of a "bubble". 51

Figure 3.18. Bearing (angle) information. 52

Figure 3.19. Histogram of bearing values in the sample bubble. 52

Figure 3.20. K-Means clustering to bearing values with four clusters. 53

Figure 3.21. Clusters of bearing values in polar coordinate system. 53

Figure 3.22. (Python) Extract road network from OpenStreetMap. 54

Figure 3.23. Beşiktaş road network (a) before and (b) after graph network filtering. 55

Figure 3.24. (Python) Getting nearest edges using OSMNX library. 56

Figure 3.25. (Python) Plotting speed distribution on map. 56

Figure 4.1. Lower triangle: KDE. Diagonal: histogram. Upper triangle: scatter plot. 63

Figure 4.2. Speed profiles over time (columns: weekday, rows: bearing clusters). 64

Figure 4.3. Hourly speed distribution of road segments. 65

Figure 4.4. Speed distribution in the Beşiktaş region. 66

Figure 4.5. Beşiktaş road network - motorcycle speed profiles at (a) 09:00, (b) 14:00, (c) 18:00. 67

LIST OF TABLES

Table 1.1. A typical example of spatial-temporal data. 2

Table 3.1. Sample data structure (EPSG:4326). 41

Table 3.2. Sample data structure after spatial transformation (EPSG:5637). .. 43

Table 3.3. Irregular time step in the dataset. 44

Table 3.4. The speed profiles after smoothing using the Kalman Filter. 47

Table 4.1. Calibration of the travel interval parameters. 68

Table 4.2. Comparison of the results of travel time methods. 68

LIST OF ACRONYMS/ABBREVIATIONS

API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
AWS S3	Amazon Web Services Simple Storage Service
BTT	Bubble-based Travel Time
COVID-19	Coronavirus Disease 2019
EBkSP	Entropy-Balanced k Shortest Paths
EPSG	European Petroleum Survey Group
ETT	Edge-based Travel Time
EWMA	Exponentially Weighted Moving Average
GARCH	Generalized Autoregressive Conditional Heteroscedasticity
GIS	Geographic Information Systems
GPS	Global Positioning System
HMM	Hidden Markov Model
ID	Identification
IQR	Interquartile Range
KDE	Kernel Density Estimation
LCC	Lambert Conformal Conic Projection
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
OSM	Open Street Map
PTW	Powered Two Wheelers
SRS	Spatial Reference System
SSE	Sum of Squared Errors
TUREF	Turkish National Reference Frame
UTM	Universal Transverse Mercator
VaR	Value at Risk
WGS	World Geodetic System

1. INTRODUCTION & BACKGROUND

Global Positioning System (GPS) enabled devices have become more and more common in the last decades. Especially for transportation purposes, GPS-based functionalities have been adapted to daily life. Besides the benefits to end-users, time and location data generated by these devices can be useful for decision-makers in governments, transportation communities, and companies.

On the other hand, traffic has been a major issue in developing and overly-populated cities. Although most parts of the cities were designed for four-wheel vehicles, such as passenger cars and buses, two-wheelers, such as bicycles and motorcycles, have been becoming popular in order to bypass traffic. In this study, GPS data generated by powered two-wheel vehicles are investigated, and useful and important insides are aimed to be provided.

GPS-enabled devices can generate location data where a typical example of location data consists of latitude, longitude, date, time, and accuracy (Table 1.1). Latitude and longitude values are in a specific map projection called the World Geodetic System 1984 (WGS 84) [1]. In addition to these fundamental data points, the timestamp of each recorded location allows for the chronological tracking of movement. By analyzing sequences of these data points, one can gain insights into patterns and behaviors over time.

Accuracy values represent the estimated horizontal accuracy radius in meters. Mining this data can provide more useful information, such as displacement, speed, acceleration, and deceleration. These values can lead us to create vehicle trajectories and location-based metrics, such as hotspots and problematic areas.

Table 1.1. A typical example of spatial-temporal data

Id	Time	Longitude	Latitude	Accuracy
101	2023-01-01 14:00:19	29.023170	41.118870	16.388
102	2023-01-01 14:00:42	29.023169	41.118890	13.401
103	2023-01-01 14:01:02	29.023167	41.118882	13.402
104	2023-01-01 14:01:13	29.023166	41.118889	13.352

The dataset of this study is from motorcycles in a one-year period. Averaged 0.006 latitude difference (around 670 meters in this study area) and 0.008 longitude difference (around 670 meters in this study area) are found in the dataset. It can be claimed that most of the trips in a circle have a roughly 670-meter radius from their starting point. The interquartile range (IQR) of the durations is 8 to 20 minutes, and the accuracies are 7 to 19 meters (Figure 1.1).

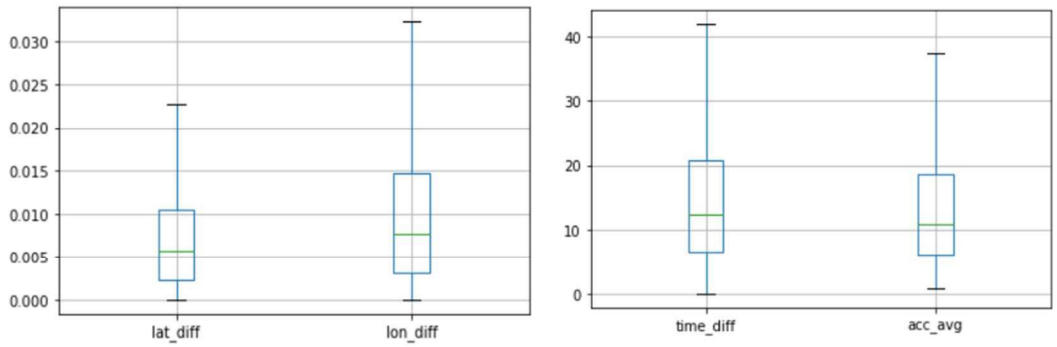

Figure 1.1. Latitude, longitude, time, and accuracy differences.

One-hour and one-day snapshots of the raw GPS data from motorcycles are visualized in Figure 1.2. Data cleansing should be applied in order to eliminate GPS-signal-loss-related deviations. After that, travel time calculations, traffic characteristics pattern recognitions, and hotspot analysis can be studied.



Figure 1.2. One hour and one day of the dataset in a specific area (Rumelihisarı, İstanbul).

The main goal of this study is to create a solid methodology to calculate location-based metrics and route characteristics for motorcycles using GPS data of motorcycles. Complete data analysis of GPS data is aimed at comprehending the spatial-temporal structure of a location. One of the main struggles of this analysis is to use one vehicle type and not a complete dataset. The results are to be achieved without a need for additional data, such as data from other vehicle types or sensors.

Taking everything into consideration, having knowledge of the effects of locations, time periods, route types, road structure, and other various factors is crucial to understanding deeply powered two-wheelers' behaviors. Using only the GPS data of motorcycles can be enough to investigate the transportation characteristics of powered two-wheelers (PTW).

1.1. Problem Statement

In recent years, urban areas have experienced interconnected shifts. More people are living in cities, intensifying the demand for transportation solutions. This growth in population has led to a significant increase in the number of vehicles on the roads. Simultaneously, the composition of vehicles has experienced a noticeable transformation, driven by both the COVID-19 pandemic and the growing concern over climate change. People's transportation choices have evolved. At the same time, increased awareness of environmental issues has started a strong interest in more sustainable modes of transportation, driving a shift towards eco-friendly alternatives.

Cities are changing how they get around due to traffic and environmental issues. More people are using bikes, electric scooters, public transport, and rideshare services for flexible, eco-friendly travel. Delivery services have also brought more motorcycles onto the roads. This trend is also seen in Europe, a region known for its advanced transportation networks, the proportion of motorcycles in traffic compositions has seen a slight rise [2]. These changes highlight the complex relationship between city growth, transport choices, and eco-awareness, calling for thoughtful planning from leaders to create strong, green city transport systems.

Urban areas encompass a diverse array of traffic participants, encompassing not only automobiles but also heavy and mid-sized vehicles, buses, motorcycles, and bicycles. Research indicates a growing prevalence of two-wheelers in daily traffic, particularly in developing countries. The traditional paradigms of traffic analysis are becoming inadequate to comprehend the complexities of mixed traffic scenarios. Moreover, studies related to traffic dynamics have primarily revolved around passenger cars, leaving a significant gap in understanding the behaviors of other vehicle types.

Existing traffic studies have predominantly centered on passenger cars, highlighting a potential avenue for exploring the behaviors of drivers operating different vehicle categories. Vehicles with narrower profiles, such as bicycles, exhibit distinct dynamics due to riders who may not strictly adhere to established road markings. Routing algorithms also necessitate differentiation, as these vehicles can navigate through narrower roads and execute U-turns with fewer space requirements. This study aims to address these gaps by examining trajectory data specifically related to motorcycles in the context of mixed traffic scenarios. In this context, there arises a clear need for research focused on these vehicles (motorcycles), involving the utilization of their data to delve into traffic analyses encompassing parameters such as traffic flow, speed estimates, and more.

1.2. The Goal and Objectives

It is mainly aimed to propose a methodology of an approach to traffic estimation for powered two-wheelers (PTW) in heterogeneous traffic using GPS data of motorcycles. To achieve this goal, the following research objectives are aimed:

(i) To apply big data extraction techniques on courier motorcycles' GPS records to create pre-processed trajectory data,

(ii) To reveal deceleration and acceleration zones using hotspot analysis to explore location-based patterns on travel times and to find out problematic areas for motorcycles,

(iii) To perform temporal frequency analysis and apply pattern recognition techniques to explore time-based patterns on travel times and to find out time-sensitive day parts for motorcycles,

(iv) To discover motorcycle-specific characteristics to propose production-ready parameters,

(v) To create a motorcycle-based travel time estimation method to be used in heterogeneous traffic estimations.

1.3. Literature Review

Motorcycle registrations in the largest European markets during the first quarter of 2023 have increased by 15.5% compared to the last year, and by 46.5% compared to five years ago [3]. Traffic characteristics of motorcycles have started to be studied more and more comprehensively within an extensive range of topics. It usually appears in articles related to mixed traffic studies, where multiple vehicle types and their interactions are explored [4]. For mixed traffic, vehicle following behavior models are proposed, using multinomial logistic regression [5], specifically at signalized intersections combined with agent-based simulation [6]. Some studies modeled interactions between buses and motorcycles [7]. Research shows mixed traffic behavior where the proportion of motorcycles is high [8]. One study models the maneuvers of different vehicles in disordered heterogeneous traffic [9].

This review focuses on the impact of GPS technology in studying powered two-wheelers (PTWs), such as motorcycles, within diverse traffic settings. PTWs are an essential part of urban transportation due to their affordability and ability to navigate through varied road conditions. However, they present unique challenges in traffic management and safety, especially when mixed with other types of vehicles.

GPS technology has emerged as a vital tool in addressing these challenges. It provides detailed insights into the movement and behavior of PTWs, helping to develop more accurate traffic models and safety strategies specifically designed for these vehicles. This is particularly important in areas where PTWs interact with a range of vehicle types, leading to complex traffic patterns.

Our review will explore how GPS data is used to improve our understanding of PTWs in traffic. This includes looking at traffic models that incorporate PTWs, how GPS data helps in analyzing PTW movements, and the role of GPS in creating intelligent transportation systems. We will also examine the use of advanced techniques like machine learning in processing GPS data, which allows for more in-depth analysis and effective traffic management solutions.

In sum, this review aims to shed light on the current uses of GPS in PTW studies and its potential to enhance traffic systems and policies, making urban transportation safer and more efficient for everyone.

1.3.1. **Heterogeneous Traffic with Powered Two-Wheelers**

Traffic characteristics of powered two-wheelers (PTW) have started to be studied more and more comprehensively within an extensive range of topics. It usually appears in articles related to mixed traffic studies [4]. Although PTWs are not only focused, multiple vehicle types and their interactions are explored. Traffic characteristics differ significantly based on the capabilities of vehicle types, their sizes, and behaviors. Drivers of power two-wheelers usually disobey lane discipline. That creates a need for special attention to investigate mixed traffic behaviors different than homogenous traffic.

This study provides an overview of existing mixed-traffic models including longitudinal and lateral movement models. There are different behavioral studies in mixed-traffic models, such as staggered and between two vehicles. However, the authors claim that the limited availability of trajectory data is one of the major obstacles to the development of mixed traffic driving models as the estimation of their parameters is required.

Another study examined how different types of vehicles follow each other on the road, focusing on the relationship between the size of the following (follower) and leading (leader) vehicles. Key findings include that smaller vehicles tend to directly follow larger vehicles; and that as the size of the following vehicle increases, it's more likely to adopt a staggered, rather than direct, following position. The proposed strategy is to use Multinomial Logistic Regression, their speeds and the distance between them also are dependent variables of the chosen following behavior [5].

Some studies use the agent-based simulation methodology to analyze the interactions between PTW and passenger cars in heterogeneous traffic [6]. The first step for calibration and validation is to process the real-life empirical microscopic trajectory data either from GPS or videos. Then, the agent-based model (ABM) helps researchers create an environment for observing the impacts of changing various variables.

1.3.2. **Trajectory Datasets for Powered Two-Wheelers**

Some projects are dedicated to creating trajectory datasets. Grab-Posisi is one of the real-world GPS trajectory datasets, from both developed and developing countries, with a 1-second sampling rate. The data is acquired from Android or iOS phone users driving cars or motorcycle-like vehicles. The authors suggest some application areas for this dataset, including one-way road detection using bearing information, and on-mode detection using standard deviation of bearing information.

Above-referred studies, like most research, are supported and based on datasets. Some projects aim to create datasets useful for mixed traffic and PTW-oriented studies. Some of the major extraction sources of these datasets are video recordings of major roads [10], offline vision-based systems [9] and drones [8, 11],

and GPS logs [12]. Video recordings need to be processed to generate trajectory datasets, as well as GPS-based data sources.

1.3.3. **GPS as an Economic Intelligent Transportation System**

Some studies utilize GPS technologies as a part of a combined system, such as a system of countermeasures to reduce fatalities and injuries [13]. The study combined several sources of data such as PTW sensors, context-based video retrieval systems, and GPS. It is suggested that training, infrastructure, and equipment adaptation can be prepared scientifically using the combination of the proposed data systems to reduce PTW fatalities.

For research on PTWs, although video datasets can provide detailed information on specific sections of a road, they are costly to implement over large areas. On the other hand, the research indicates that a 2–3% penetration of motorcycle GPS data in the driver population is enough to provide accurate measurements of the velocity of the traffic flow [14]. The authors experimented with 100 vehicles and compared the results with the probe data and loop detectors. The results match in significance even with a low rate of penetration.

On the other hand, using GPS data from multiple transportation modes, including motorcycles, is proposed as an economically adaptive ITS system for developing countries where non-lane-based traffic is the core [15]. Using low-cost GPS devices for motorcycles, it is claimed that the cost of developing, implementing, and maintaining this system is much lower than the traditional ITS systems.

Data mining techniques from mobile crowds are suggested particularly for developing countries not having ITS infrastructures [16]. In the proposed methodology, citizen observatories can collect and share traffic-related data with the

system for processing and analysis to resolve missing data to overcome the difficulty of estimating urban traffic in real-time at any road segment.

Another research on motorcycle taxis using GPS data proposed a methodology to analyze the impact of PTWs on a city with heterogeneous traffic [17]. The correlation of questionnaires with motorcycle density, which is extracted from motorcycle GPS records is investigated. It is shown that the mixed approach with GPS tracking despite the difficulty of obtaining large volumes of robust data is effective.

1.3.4. **Navigation Systems**

Some studies show that there is a need for a navigation system specialized for PTWs. In some urban areas, people started to prefer to use motorcycles over cars for various reasons such as it can pass narrow streets, and it can bypass the traffic in some specific conditions. This capability results in the inability to use standard navigation systems. A study proposed a system that uses the routing mechanisms entropy-balanced k shortest paths (EBkSP) to enable motorcycles to have a motorcycle-tailored navigation system [18].

1.3.5. **Simulation Models**

A study of a microscopic simulation model for two-wheelers claims that there is a gap in the literature that explains the movement of two-wheelers in mixed traffic conditions [19]. The behavior in mixed traffic includes various vehicle types moving together without strict lane adherence, collision avoidance, and reacting to the surrounding infrastructure and vehicles. The study suggests using perception lines which are imaginary lines drawn from the front seat of the subject motorcycle

outwards to identify the surroundings that influence the movement of the motorcycle.

1.3.6. **Accident Analytics**

Researchers used a machine learning approach including decision trees, random forests, logistic regression, naive Bayes, DBSCAN, and hierarchical clustering to analyze accidents [20]. The study shows that accidents involving motorcycles are the main factors that have a higher chance of resulting in victims. The locations of accidents are clustered to input the machine learning (ML) models to predict the accident hotspots.

The study suggests a four-step process used for identifying accident hotspots using machine learning (ML) techniques. First, data processing involves preparing various inputs like weather, road conditions, time, and geographic coordinates to be applicable for ML models. The second step involves clustering, where the data is analyzed to generate hotspots of frequent accidents. In the third step, different ML models such as decision trees, random forests, and others are trained and compared to determine their effectiveness. Finally, in the prediction phase, the chosen model is used to predict and identify where accident hotspots are likely to occur. This systematic approach leverages ML to enhance the understanding and prevention of accident-prone areas.

1.3.7. **GPS Processing**

As indicated before, GPS data is economical, widely distributed, and highly beneficial. However, to extract meaningful insights, it is essential to preprocess and clean the typically noisy records found in most GPS datasets. Studies indicate that

the Kalman filter and its variant, the extended Kalman filter, are among the most commonly used algorithms for processing motorcycle GPS data.

Researchers propose an algorithm for GPS of motorcycles having a bad measurement of trajectory stems from the declination of the body and obstacles near them [21]. The extended Kalman filter is constructed with x, y, and z coordinates, speed, standard deviation of velocity error, clock offset calculated by the nonlinear least-square method, and barometer values. The accuracy of trajectory calculated by the proposed method is less than 3m, and the estimation is equal to the exact location when a sufficient number of satellites is available.

Some other methods to reconstruct motorcycle trajectories from GPS records are smoothing cubic spline and Butterworth filters [22]. The spline method, unlike the interpolation function, does not pass exactly through all points but does come only reasonably close. The Mahalanobis distance is used for identifying multivariable outliers.

Researchers applied the chi-square test on two-wheeled vehicle traffic crashes for non-spatial analysis kernel density, nearest neighbor distances, and K-function analysis to determine the existence of clustering and hotspots [23]. They came to understand that a significant proportion of all accidents involve two-wheeled vehicles. It is recommended to use these key variables in the spatial analysis: ‘‘type of accident occurrence”, ‘‘number of vehicles involved in the accidents”, ‘‘day condition”, ‘‘season”, ‘‘month”, ‘‘hours in a day”, ‘‘road properties”, ‘‘weather properties”, ‘‘environment properties”.

1.3.8. **Discussions & Conclusion**

This review has systematically explored the intricate role of GPS technology in the realm of powered two-wheelers (PTWs) within heterogeneous traffic environments. Through our comprehensive analysis, it becomes evident that GPS not only serves as a pivotal tool for data collection and traffic analysis but also acts as a cornerstone in the development of intelligent transportation systems (ITS), particularly beneficial in settings with non-lane-based traffic common in many developing countries.

The studies reviewed here indicate a clear trend toward the utilization of GPS for more accurate and economical traffic management solutions. This is especially relevant for PTWs, which often display unique traffic behavior compared to other vehicle types. The incorporation of GPS data has been shown to significantly enhance the understanding of PTW dynamics, aiding in the development of targeted traffic models and navigation systems that cater to the specific needs of motorcyclists.

Moreover, the application of advanced data processing techniques like the Kalman filter to GPS data has opened new avenues for more precise trajectory analysis of PTWs. This advancement is crucial in addressing the challenges posed by the inherent noise and inaccuracies in raw GPS data, especially in the context of PTWs.

The integration of GPS data with machine learning techniques for accident analysis further underscores the potential of this technology in enhancing road safety for PTWs. By identifying high-risk areas and patterns, these methods offer valuable insights for policymakers and urban planners to develop more effective safety measures.

In conclusion, the reviewed literature collectively demonstrates that GPS technology is not only a key enabler in understanding and managing PTW traffic but also a vital component in advancing research in this field. As technology continues to evolve, the potential applications of GPS in PTW studies are expected to expand, offering more sophisticated and comprehensive solutions for traffic management, safety enhancement, and urban planning in the context of mixed-traffic environments.

The literature reveals a growing focus on the study of PTWs using GPS data during a surge in motorcycle use. While traditional methods like video recording and agent-based simulation are prevalent, there is an increasing shift towards GPS data for its economic and comprehensive coverage. However, a significant research gap exists in the methodology for processing and analyzing PTW GPS data. This study aims to bridge this gap by introducing refined analytical techniques, potentially setting a new standard in PTW GPS-related research, and offering valuable insights into motorcycle traffic behaviors.

1.4. Thesis Outline

The remainder of this thesis is organized as follows: In the next section, the background and theory of the methods which are used in this study are discussed. Then, in Chapter 3, the methodology of every step of this study is explained with examples as well as usage of the methods starting from how to obtain and store the data to how to extract information from the data. It is followed by the data analysis, the results of the analysis, and a comparison of the model performances to estimate travel times in Chapter 4. Finally, conclusions and recommendations are provided in Chapter 5.

2. THEORY

In this thesis, some algorithms, data mining, and machine learning methods such as spatial transformation, Kalman filtering, k-means clustering, and map matching, are used for creating a travel time estimation method for courier motorcycles.

Spatial transformations are used for mapping latitude and longitude information to coordinates that are more suitable for processing so that we can simply use meter-level calculation instead of dealing with skewed degree units.

Kalman filtering is used for eliminating noises, which are caused by nearby buildings, tunnels, bridges, and weather conditions, from the raw sensor data to generate pre-processed route data.

Bubble generation is used for grouping route points like hex-bin (hexagonal binning) methods so that location-based analytic data can be generated instead of analyzing every single point individually.

K-means clustering is used for grouping bearing information on motorcycles by separating them based on the closeness of the points so that we can distinguish data points on different roads at intersections and analyze them differently.

The map-matching methods are used for finding the nearest road parts to each location point so that we can estimate road-related parameters using individual route points.

Exponentially Weighted Moving Average (EWMA) is used for combining travel times from the edge and bubble-based estimation methods, which is a statistical technique used in time series analysis, to smooth out short-term fluctuations and highlight longer-term trends or cycles.

2.1. Spatial Transformation

Spatial transformation in the context of geospatial data refers to the process of modifying the spatial representation of data from one coordinate system to another. This process is essential for accurate data integration, analysis, and visualization, grounded in specific theoretical principles.

Spatial transformation involves converting the coordinates of features – such as points, lines, and polygons – from one coordinate system to another. This is often necessary when integrating data from different sources, each using different coordinate systems. It can include simple translation (shifting all points by a fixed distance), rotation (rotating around a point), scaling (increasing or decreasing size), or more complex transformations involving changes in the coordinate reference system.

Integrating data from various sources often necessitates spatial transformation because each dataset may use a different coordinate system. Converting all data into a common coordinate system is crucial for accurate analysis and comparison. Different coordinate systems are suited for specific purposes. For example, local coordinate systems can provide greater precision for small areas, whereas global systems are better for larger-scale applications. Transformations are particularly necessary for accurately displaying spatial data on maps, as they need to account for the Earth's curvature.

The Earth is an oblate spheroid, a complex shape requiring different coordinate systems for simplified representation of locations. Two primary types of coordinate systems are used: Geographic Coordinate Systems (GCS) and Projected Coordinate Systems (PCS). GCS, using latitude and longitude, is suited for global representations, while PCS, such as the Universal Transverse Mercator (UTM) system, uses meters and is better for detailed regional analysis.

Spatial transformation ensures that geospatial data from diverse sources can be accurately overlaid, analyzed, and visualized. By transforming data into a common coordinate system, analysts can ensure precision and consistency, whether working on local or global scales.

In the case of which is specialized for Türkiye, transforming data into this system allows for the use of Euclidean mathematics. This means calculations like distance, area, and angle measurements can be more straightforward and accurate as they are based on a flat plane concept, which is a good approximation for smaller areas. In geodesy, a horizontal control network is a series of points with known geographic coordinates spread across a region, forming the basis for mapping or spatial data collection within that area.

Transformation algorithms in the context of geospatial data are mathematical procedures used to convert coordinates from one spatial reference system (SRS) or coordinate system to another. These algorithms are essential in cartography, geographic information systems (GIS), and other fields that require accurate spatial data representation. The complexity and type of transformation algorithm used depend on the nature of the coordinate systems involved and the level of accuracy required. Example algorithm chain of TUREF is shown in Figure 2.1.

```
PROJCS["TUREF / LCC Europe",
    GEOGCS["TUREF",
        DATUM["Turkish_National_Reference_Frame",
            SPHEROID["GRS 1980",6378137,298.257222101],
            TOWGS84[0,0,0,0,0,0,0]],
        PRIMEM["Greenwich",0,
            AUTHORITY["EPSG","8901"]],
        UNIT["degree",0.0174532925199433,
            AUTHORITY["EPSG","9122"]],
        AUTHORITY["EPSG","5252"]],
    PROJECTION["Lambert_Conformal_Conic_2SP"],
    PARAMETER["latitude_of_origin",52],
    PARAMETER["central_meridian",10],
    PARAMETER["standard_parallel_1",35],
    PARAMETER["standard_parallel_2",65],
    PARAMETER["false_easting",4000000],
    PARAMETER["false_northing",2800000],
    UNIT["metre",1,
        AUTHORITY["EPSG","9001"]],
    AUTHORITY["EPSG","5637"]]
```

Figure 2.1. Algorithm representation of EPSG 5637 (TUREF) projection.

The process involves mathematical formulas and algorithms that convert coordinates from one system to another, taking into account the shape of the Earth and the characteristics of each system. The following formulas show the transformation of spherical coordinates to the plane via the Lambert conformal conic projection as

$$x = \rho \sin\left[n\,(\lambda - \lambda_0)\right], \tag{2.1}$$

$$y = \rho_0 - \rho \cos\left[n\,(\lambda - \lambda_0)\right], \tag{2.2}$$

$$F = \frac{\cos\phi_1 \tan^n\left(\frac{1}{4}\pi + \frac{1}{2}\phi_1\right)}{n}, \tag{2.3}$$

$$n = \frac{\ln\left(\cos\phi_1 \sec\phi_2\right)}{\ln\left[\tan\left(\frac{1}{4}\pi + \frac{1}{2}\phi_2\right)\cot\left(\frac{1}{4}\pi + \frac{1}{2}\phi_1\right)\right]}, \tag{2.4}$$

$$\rho = F \cot^n\left(\frac{1}{4}\pi + \frac{1}{2}\phi\right), \tag{2.5}$$

$$\rho_0 = F \cot^n\left(\frac{1}{4}\pi + \frac{1}{2}\phi_0\right), \tag{2.6}$$

$$\phi = 2\tan^{-1}\left[\left(\frac{F}{\rho}\right)^{1/n}\right] - \tfrac{1}{2}\pi\,, \tag{2.7}$$

$$\lambda = \lambda_0 + \frac{\theta}{n}\,, \tag{2.8}$$

$$\rho = \mathrm{sgn}\,(n)\sqrt{x^2 + (\rho_0 - y)^2} \quad \text{and} \tag{2.9}$$

$$\theta = \tan^{-1}\left(\frac{x}{\rho_0 - y}\right) \tag{2.10}$$

where λ is the longitude, λ_0 is the reference longitude, ϕ is the latitude, ϕ_0 is the reference latitude, and ϕ_1 and ϕ_2 are the standard parallels.

In summary, spatial transformation is a critical process in geospatial analysis, enabling the integration, accurate representation, and effective analysis of spatial data. It relies on mathematical and algorithmic principles to convert data between different coordinate systems, ensuring precision and relevance to the specific spatial context.

2.2. Kalman Filtering

Kalman filtering is a mathematical technique widely used in control systems, navigation systems, and signal processing to estimate the state of a system from a series of noisy measurements. It is particularly known for its ability to provide optimal estimates in systems where the noise and other uncertainties are represented as Gaussian processes.

Developed in the late 1950s and early 1960s by Rudolf E. Kalman [24], the filter addresses the general problem of trying to estimate the state of a dynamic system over time. In practical terms, the "state" of the system could be anything from the position and velocity of an aircraft to the temperature in a room as measured over time as in Figure 2.2.

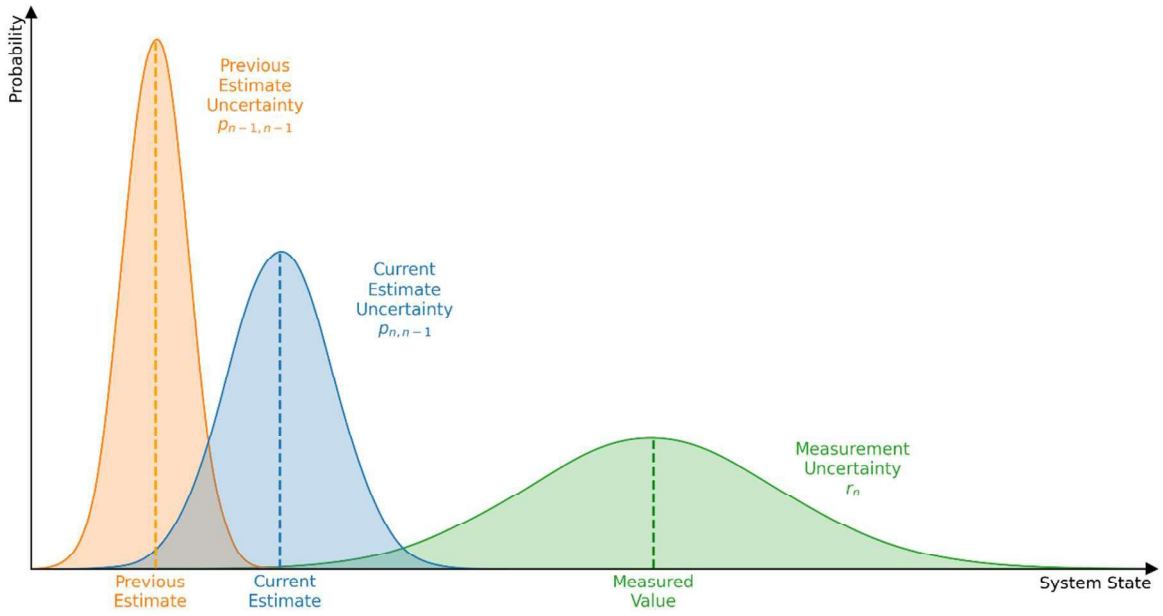

Figure 2.2. Kalman filter state estimation.

The current estimate $\hat{x}_{n,n}$ is calculated using the previous estimate $\hat{x}_{n,n-1}$, the Kalman gain K_n, and the measured value z_n as

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + K_n \left(z_n - \hat{x}_{n,n-1}\right) , \tag{2.11}$$

$$p_{n,n} = (1 - K_n)\, p_{n,n-1} \tag{2.12}$$

where the Kalman gain is a combination of the previous estimate uncertainty $p_{n,n-1}$ and the measurement uncertainty r_n.

The core idea of the Kalman filter is to combine two sources of information: the predictions about the state of the system and the new measurements. The system's state is estimated continuously over time by projecting the current state estimate ahead of time and then updating this projection with new measurements.

The prediction step uses the known dynamics of the system. This involves applying a model of the system's behavior to the last estimated state to predict what the next state should be. For example, if we know an aircraft's current position and speed, we can predict where it should be in a short time in the future. This prediction

is not perfect because of the uncertainties in the model and the possibility of unexpected changes in the system.

The update step incorporates the new measurement into the state estimate. The Kalman filter assumes that both the system dynamics and the measurements have some noise, modeled as random Gaussian processes. This noise represents the uncertainty in our knowledge about the system and the imprecision in our measurements. When a new measurement is received, the Kalman filter calculates a weighted average of the predicted state and the new measurement to produce a new estimate. The weighting is based on the relative certainty of the prediction and the measurement; more weight is given to the more certain source of information.

One of the key strengths of the Kalman filter is its ability to estimate the state of the system even when the measurements are noisy and the system itself is changing. It does this by constantly updating its estimate based on the latest data, thereby refining the prediction of the system's state over time.

The mathematics behind the Kalman filter involve several equations, primarily for the prediction and update steps. These equations use matrices to represent the system's dynamics, the measurement process, and the uncertainties in both. In the prediction step, the filter uses the state transition matrix and control input (if any) to predict the next state. In the update step, it uses the measurement matrix to incorporate the new measurement and update the state estimate.

The Kalman filter's effectiveness depends on accurately modeling the system's dynamics and the characteristics of the noise in the system and measurements. If these models are incorrect, the filter's performance can degrade. However, in many practical applications, the models are sufficiently accurate, making the Kalman filter a powerful tool for state estimation in a variety of fields.

In summary, the Kalman filter is a powerful and versatile algorithm for estimating the state of dynamic systems in the presence of noise. Its ability to combine predictions about a system's state with actual measurements and to adapt over time as new data is received makes it a valuable tool in many engineering and scientific applications.

2.3. Bubble Generation

Bubble generation is a technique to aggregate data points to investigate the effect of nearby in-bubble points on each other. The bubbles keep the information of all the points inside in an aggregated way. The aggregation method is usually the average of all values, but the median, count, summation, or standard deviation can be used.

Bubbles should be generated in a way that they cover as much as area they can. In the square packing, all bubbles in consecutive rows are positioned vertically aligned. However, the hexagonal packing shifts rows a radius so that they are packed more tightly, as shown in Figure 2.3. When the uncovered areas are assigned to the nearest bubble center, the shapes become hexagons [25].

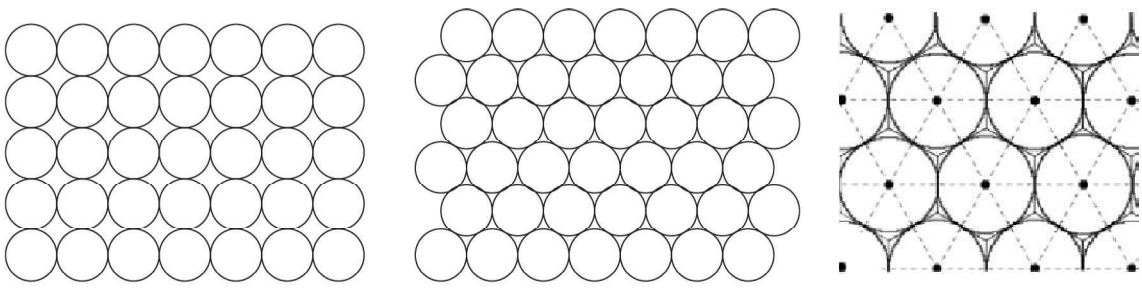

Figure 2.3. Square vs hexagonal packing of circles, and hexagons.

Hexagonal binning is a data aggregation and visualization technique particularly suited for handling large datasets. Unlike traditional scatter plots where each data point is plotted individually, hex-bin groups data points into hexagonally-shaped bins, allowing for a more efficient and clearer representation of data density and distribution. This method is especially useful in overcoming the limitations of overplotting that occur in scatter plots with large datasets.

The fundamental principle behind hexagonal binning is the partitioning of the dataset into hexagonal cells. Each cell, or bin, aggregates the data points that fall within its boundaries. The choice of hexagons, as opposed to other shapes like squares or rectangles, is based on their geometric efficiency. Hexagons have the highest number of sides among shapes that can tessellate, which allows for a closer approximation of a circle compared to squares or triangles. This property is advantageous because circles are the optimal shape for minimizing distance biases when grouping data points. In other words, hexagons ensure that data points are more evenly distributed within a bin, reducing the potential distortion in the representation of data density.

When implementing hexagonal binning, the process starts with laying out a grid of hexagons over the data space. The size of the hexagons is a critical decision as it determines the level of detail in the resulting visualization. Smaller hexagons capture more detail by creating more bins, each with fewer data points, while larger hexagons provide a more generalized view of the data by grouping more points into each bin. This bin size can significantly influence the interpretation of the data, as it affects the granularity of the visualization.

Once the grid is established, data points are assigned to the nearest hexagon based on their coordinates. This assignment is typically done by calculating the distance of each data point to the center of the nearest hexagon and assigning the

point to the hexagon with the smallest distance. After all points are binned, the data within each hexagon is aggregated. The most common form of aggregation is a simple count of the number of points in each bin, but other statistical measures like sums, averages, or medians can also be used depending on the nature of the data and the specific requirements of the analysis.

In the visualization stage, each hexagon is usually represented by a color or shade that reflects the aggregated value of the data points within that bin. A common approach is to use a color gradient where the intensity or hue of the color corresponds to the density of points in the bin. This method effectively conveys areas of high and low data density, allowing for the easy identification of patterns or clusters within the dataset.

As a data aggregation technique, hexagonal binning offers a balance between preserving the underlying structure of the data and providing a clear, interpretable representation of its distribution. Its ability to reduce the noise and clutter associated with overplotting in large datasets, while still retaining a high level of detail, makes it a valuable tool in the arsenal of data analysts and scientists [26]. The flexibility in bin size and the method of aggregation allows for a customizable approach to data visualization, making hexagonal binning a versatile and effective technique for exploring and communicating complex data patterns, as shown in Figure 2.4.

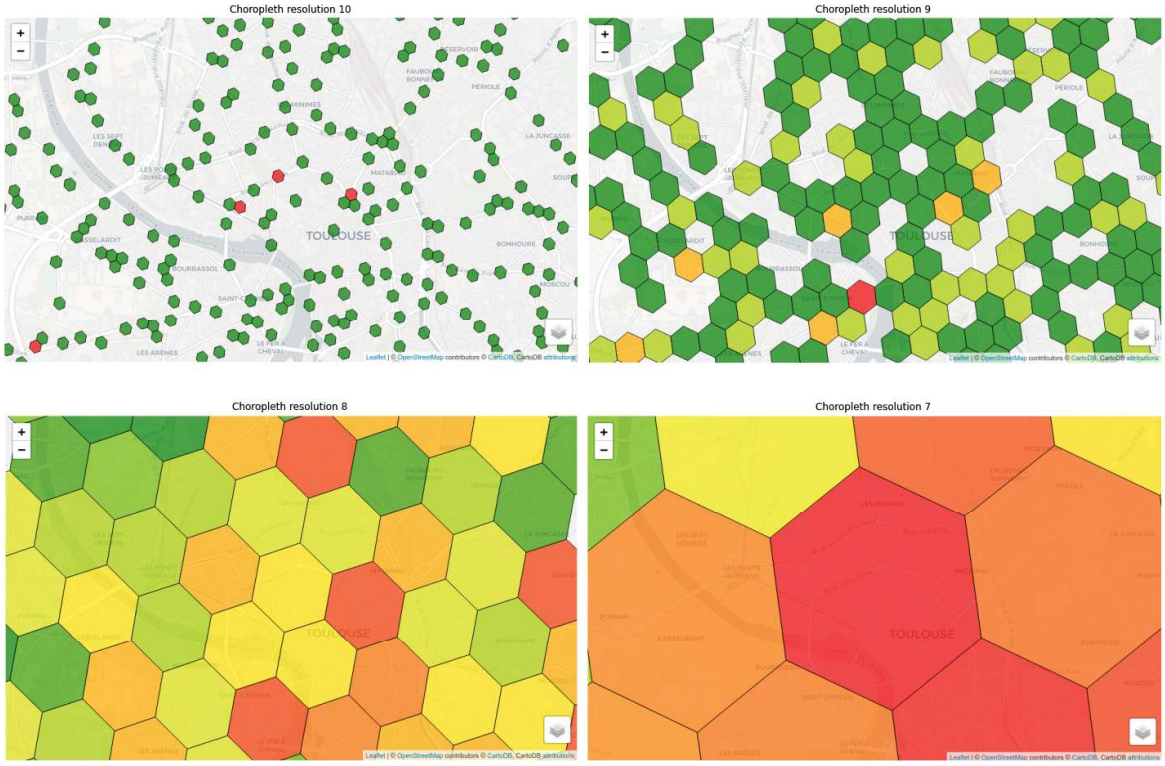

Figure 2.4. Aggregated data visualization in different bin sizes.

2.4. K-Means Clustering

K-means clustering is a popular and simple unsupervised machine learning algorithm used for partitioning a given dataset into K distinct, non-overlapping subgroups or clusters. The algorithm aims to classify the data points into clusters in such a way that similar data points are grouped together, thereby minimizing the variance within each cluster.

The process of K-means clustering begins with the selection of K initial centroids, which are points representing the initial center of a cluster. These centroids can be chosen randomly or based on a specific strategy. Each data point in the dataset is then assigned to the nearest centroid, based on a distance measure, usually the Euclidean distance. This assignment creates K initial clusters of data points.

Once the initial clusters are formed, the next step is to recalculate the centroids. The new centroid of each cluster is computed by taking the mean of all the points in that cluster. This mean point becomes the new centroid of the cluster. After recalculating the centroids, the data points are reassigned to the nearest new centroid. This step might cause data points to move from one cluster to another.

The algorithm iterates between the step of assigning data points to the nearest centroid and the step of recalculating centroids until a stopping criterion is met. The stopping criteria can be a set number of iterations, a threshold for the minimum amount of centroid change between iterations, or when no data points change clusters between successive iterations.

The key to K-means' efficiency and simplicity lies in its iterative nature, gradually improving the quality of the clusters with each iteration. However, this iterative process also brings some challenges. The result of K-means clustering is sensitive to the initial choice of centroids. Different initial centroids can lead to different final clusters. To mitigate this, the K-means algorithm is often run multiple times with different initial centroid configurations. The best clustering result, often measured by a metric such as the Sum of Squared Errors (SSE), is then chosen where K is the number of clusters, m is the number of observations, w_{ik} is belonging of x^i to the kth cluster, and μ_k is the mean of the kth cluster. The SSE is calculated as

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \left\| x^i - \mu_k \right\|^2. \tag{2.13}$$

Another challenge is choosing the right number of clusters, K. There is no straightforward way to determine the optimal number of clusters. A common method is the "elbow method," which involves running the K-means clustering for a range of K values and plotting the SSE. The point where the SSE starts to level off ("the elbow") suggests a suitable number of clusters, as shown in Figure 2.5. However, this method is subjective and may not always provide a clear cut-off point.

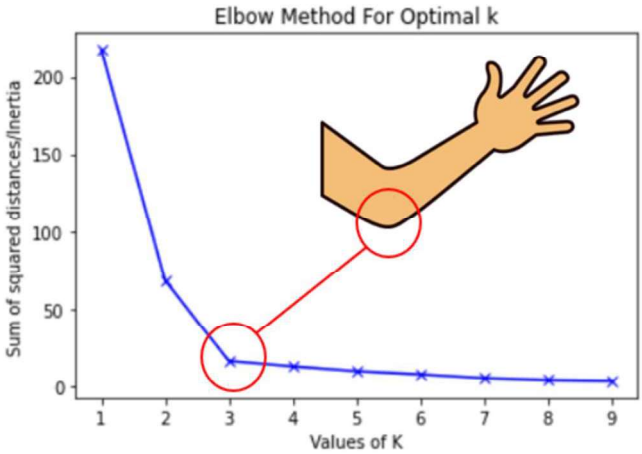

Figure 2.5. Line plot between K and inertia.

K-means clustering works best with spherical clusters of similar sizes and densities. It is less effective with clusters of different sizes and densities or non-spherical shapes. The algorithm also assumes that the mean is a good representation of the cluster center, which may not be true for some data distributions.

Despite these limitations, K-means clustering is widely used because of its simplicity, speed, and efficiency, particularly with large datasets. It is used in various applications like market segmentation, document clustering, image segmentation, and pattern recognition. Its ease of understanding and implementation makes it a popular choice for introductory machine learning and data mining courses.

2.5. Exponentially Weighted Moving Average

The Exponentially Weighted Moving Average (EWMA) is a statistical technique used in time series analysis, particularly in financial markets, to smooth out short-term fluctuations and highlight longer-term trends or cycles. The method is especially useful in forecasting and tracking applications where recent observations are considered more relevant than older ones. The key feature of EWMA is that it assigns exponentially decreasing weights to data points as they age, making it

different from simple moving averages where each data point in the period carries equal weight.

The algorithm for calculating the EWMA begins with selecting a smoothing factor, often denoted as λ, which determines the rate at which the weights decrease. The value of λ typically ranges between 0 and 1, and its selection is critical as it affects the sensitivity of the EWMA to recent changes in the data. A smaller λ gives more weight to recent observations, making the average more sensitive to recent changes. Conversely, a larger λ gives more weight to older data, resulting in a smoother average less responsive to recent fluctuations.

The calculation of EWMA for a time series Y is done iteratively. Let Y_t be the observation at time t, and S_t be the value of the EWMA at time t. The initial value of the EWMA, S_1, is typically set to Y_1, the first observation in the series. For each subsequent period ($t > 1$), the EWMA is calculated using the formula as

$$S_t = \lambda S_{\{t-1\}} + (1 - \lambda) Y_t. \tag{2.14}$$

This equation reflects the core principle of EWMA: the current smoothed value is a combination of the previous smoothed value and the current observation. The term $\lambda\, S_{\{t-1\}}$ represents the weighted contribution of the past data, while $(1 - \lambda)\, Y_t$ represents the weighted contribution of the current observation.

One of the main advantages of EWMA is its ability to react more quickly to recent changes in the data compared to a simple moving average. This feature makes it particularly useful in volatile markets where recent data is often more indicative of future trends. However, the choice of λ can significantly impact the performance of the model. A smaller λ can make the model too sensitive to short-term fluctuations, leading to overreaction, while a larger λ can make the model too sluggish to respond to real changes in the trend.

In practical applications, EWMA is often used in conjunction with other statistical techniques. For instance, it is frequently used in control charts for monitoring industrial processes, where it helps in detecting small shifts in process averages. In finance, EWMA models are used in risk management, especially in the calculation of Value at Risk (VaR) and for volatility forecasting in option pricing models.

The Exponentially Weighted Moving Average is a versatile and powerful tool for analyzing time series data, particularly in fields where recent data is more indicative of future trends. Its key advantage lies in its ability to give more weight to recent observations, allowing for a more responsive analysis that adapts quickly to changes in the underlying data. This responsiveness is controlled by the smoothing factor λ, which needs to be carefully chosen to balance the trade-off between sensitivity to recent changes and the stability of the moving average.

Despite its advantages, EWMA is not without limitations. One significant limitation is its reliance on a single parameter, λ, which can be difficult to choose without prior experience or domain knowledge. Additionally, EWMA is a univariate technique, focusing on a single time series without considering potential relationships with other variables. This makes it less suitable for complex analyses where multiple factors interact.

Moreover, EWMA is a linear model, assuming a constant linear relationship between past and present values. This assumption may not hold in cases where the data exhibits non-linear patterns or where external factors cause structural changes in the time series. In such scenarios, more sophisticated models, such as ARIMA (AutoRegressive Integrated Moving Average) or GARCH (Generalized Autoregressive Conditional Heteroskedasticity), might be more appropriate.

The practicality of EWMA also extends to its computational efficiency. Since it only requires the most recent observation and the previous EWMA value for calculation, it is particularly well-suited for real-time analysis in situations where data is continuously updated, such as in stock price monitoring or real-time process control in manufacturing. This efficiency makes it an attractive option for applications where computational resources are limited or where rapid decision-making is crucial.

In summary, the Exponentially Weighted Moving Average is a fundamental tool in time series analysis, particularly valuable for its simplicity and the intuitive way it assigns decreasing weights to older observations. Its adaptability makes it suitable for a wide range of applications, from financial forecasting to quality control in manufacturing. However, the effectiveness of EWMA largely depends on the correct selection of the smoothing factor and the context in which it is applied. Like any analytical tool, it should be used with an understanding of its assumptions and limitations, and often in conjunction with other analytical methods to provide a more comprehensive view of the data. The choice of λ remains a critical decision, requiring either empirical testing or domain expertise to determine the optimal value for a given application. The Exponentially Weighted Moving Average is a versatile and effective tool for time series analysis. Its simplicity, coupled with its ability to highlight recent trends, makes it a popular choice in various fields, particularly finance and industrial process control.

3. METHODOLOGY

In this section, a suggested approach of how to conduct analysis on motorcycle GPS data with detailed explanations. The methodology has yielded two key results (Figure 3.1): (1) the compilation of historical temporal speed profiles for bubbles and (2) the acquisition of recent speed values pertaining to road segments. The historical temporal speed profiles encapsulate data collected over a span of two weeks, providing a comprehensive understanding of the speed patterns within the bubbles. On the other hand, the recent speed values correspond to a narrower timeframe of ten minutes, offering insight into the most immediate past conditions of the road segments. In order to forecast the forthcoming hour, it is necessary to combine these two components, effectively formulating a predictive equation.

This section starts with descriptions of the location where the GPS data of motorcycles are gathered and the raw data structure (Figure 3.2). The preprocessing section, which has a final product of the speed profiles, includes spatial transformation, and Kalman filtering in order to solve the irregular time step problems and anomalies in the data. After explaining how to gather a map network, the speed profiles are used for map-matching to obtain related road segments. In parallel, the locations sharding proposal called "bubbles" is a way to store historical information along with bearing angle clusters, and weekday and hour breakdowns. Lastly, combining the speed distributions by the road segments and the seasonality and bearing information stored in bubbles is explained.

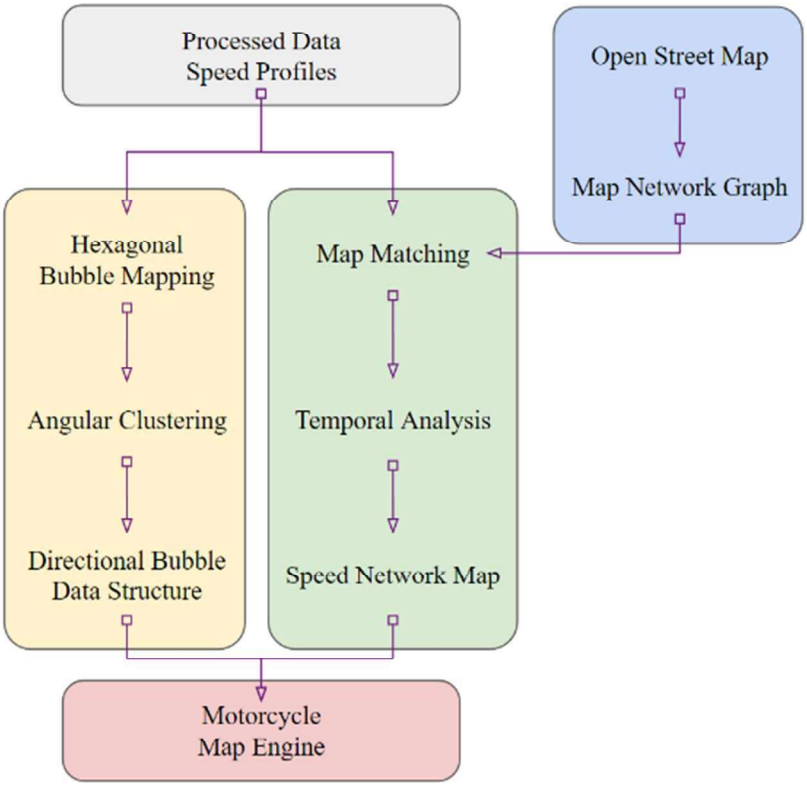

Figure 3.1. Main sections of the proposed methodology.

3.1. Proposed Method

Building on the research objectives outlined in the previous chapter, this section delves into the proposed methodology designed to address the core question of traffic estimation for powered two-wheelers (PTW) in heterogeneous traffic using GPS data of motorcycles. Recognizing the complexity and multifaceted nature of this issue, a mixed-method approach has been adopted. This method includes data extraction, processing, transformations, manipulations, and calculations using the GPS data, offering a comprehensive understanding of the characteristics of two-wheelers. The choice of this method is rooted in its proven efficacy in similar studies, particularly in GPS-related studies. The subsequent paragraphs will detail the sequential phases of this methodology, starting with the spatial transformation, followed by the speed calculation and the bubble generation, and concluding with the time estimation, as shown in Figure 3.2.

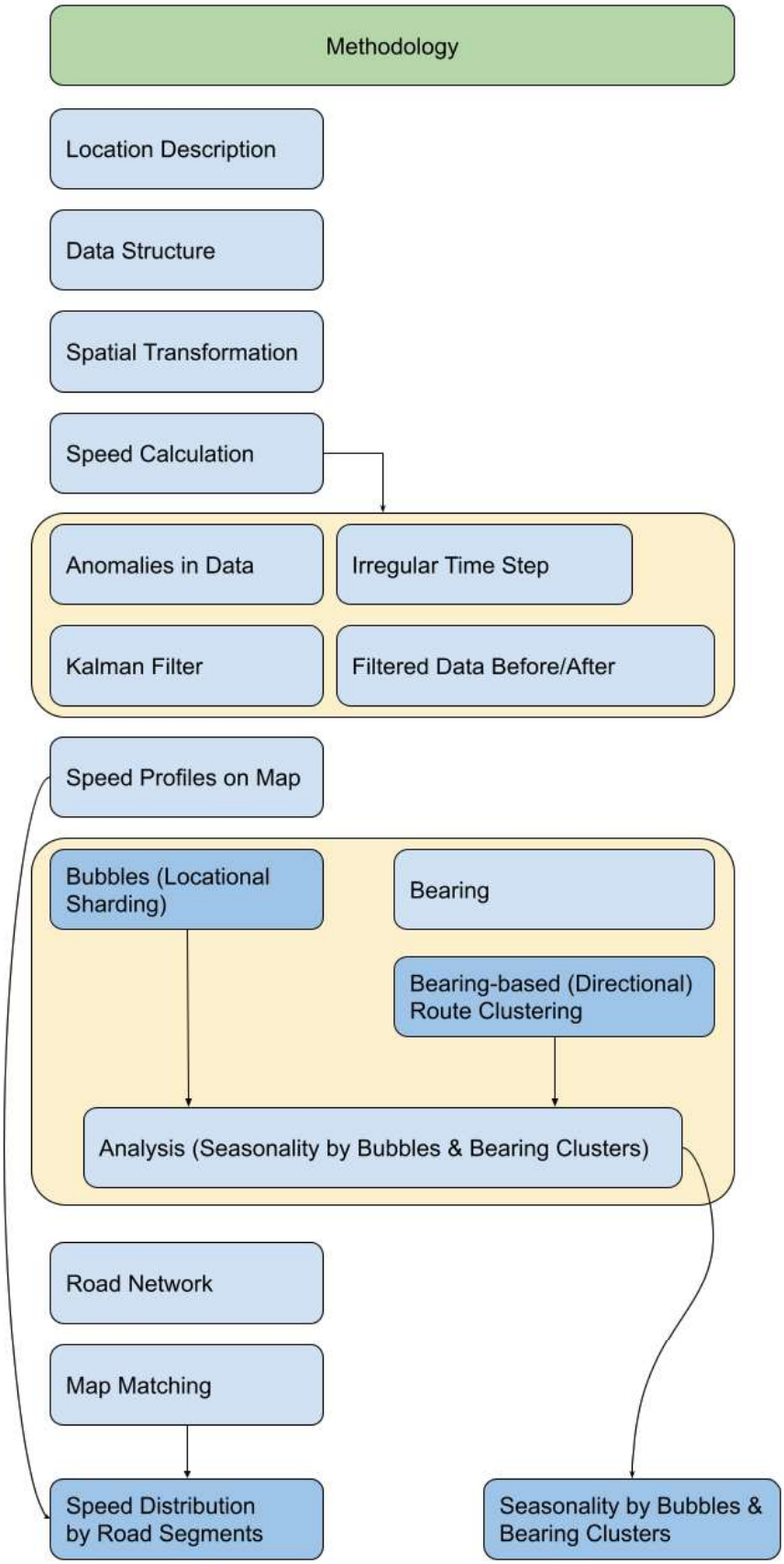

Figure 3.2. Detailed outline of the proposed methodology.

Firstly, the GPS records should be extracted into the proposed pipeline. Usually, the records are kept in a central database or a file system of one main server. Another possibility is that the records are distributed in multiple databases or servers. The last possibility is that the data is not collected from any device yet. If the data is not collected, one may choose the central or distributed way to keep the data. And, the records can be extracted via the device APIs. For example, the Android location manager can provide latitude and longitude values along with the device timestamp, it can be recorded using the SQLite database helper, as shown in Figure 3.3. After that, the records should be sent to the central or distributed servers.

```
private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 10;
private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 1;
locationManager = (LocationManager) mContext
    .getSystemService(LOCATION_SERVICE);
locationManager.requestLocationUpdates(
    LocationManager.GPS_PROVIDER,
    MIN_TIME_BW_UPDATES,
    MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
loc = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
latitude = loc.getLatitude();
longitude = loc.getLongitude();
```

```
// Get the current timestamp
long currentTimeMillis = System.currentTimeMillis();
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault());
String formattedTimestamp = sdf.format(new Date(currentTimeMillis));
```

```
private void insertLocation(double latitude, double longitude) {
    // Gets the data repository in write mode
    SQLiteDatabase db = dbHelper.getWritableDatabase();

    // Create a new map of values, where column names are the keys
    ContentValues values = new ContentValues();
    values.put("latitude", latitude);
    values.put("longitude", longitude);

    // Insert the new row, returning the primary key value of the new row
    long newRowId = db.insert("locations", null, values);
}
```

Figure 3.3. Get the location and time information using the Android API.

The records can be queried from the databases or fetched from file systems using Apache Spark. df refers to the "data frame" that stores information on datasets

in Figure 3.4. The different sources can be utilized to extract data such as Hadoop, AWS S3, Google Big Query, or other databases using Java Database Connectivity.

```
# Read single file
df = spark.read.csv('gps_records.csv')
# Multiple files
df = spark.read.csv(['gps1.csv','gps2.csv'])
# Folder
df = spark.read.csv('C:/path/to/directory')
# Hadoop File System
df = spark.read.csv('hdfs://path/to/directory')
# AWS S3 File System
df = spark.read.csv('s3://path/to/directory')
```

```
# From Google Big Query
df = spark.read.format("bigquery") \
    .option("table", f"{project_id}:{dataset_name}.{table_name}") \
    .load()

# Using JDBC
df = spark.read.format("jdbc").option("url", url) \
    .option("dbtable", table_name).load()
```

Figure 3.4. Extract data with different options using Spark.

The locational data usually is stored in degree format; latitudes are from 0° to 90° north-south, and longitudes are from 0° to 180° east-west. Converting to distance-based units will make the following calculations so much easier and more efficient. The spatial transformations are designed for this purpose and specific locations. EPSG (epsg.io) which is an open-source web service providing coordinates systems used in maps worldwide, can be used for finding the best coordinate system for the area of the dataset.

The first step of data manipulation is the spatial transformation of the chosen coordinate system. *pyproj* is a Python library that stores the conversion algorithms. The most locational data is in the World Geodetic System (WGS84) coordinate system, which is also called "EPSG:4326". The Python library can convert coordinates into the chosen coordinate system (for example EPSG:5637) with the

following code in Figure 3.5. The example coordinate system uses metric units. Therefore, Euclidian mathematics can be applied and the results will be in meters.

```
>>> from pyproj import Transformer
>>> transformer = Transformer.from_crs("EPSG:4326", "EPSG:5637")

>>> transformer
Description: Inverse of TUREF to WGS 84 (1) + Europe Conformal 2001
Area of Use:
- name: Türkiye (Turkey) - onshore and offshore.
- bounds: (25.62, 34.42, 44.83, 43.45)

>>> transformer.transform(41.015137, 28.979530)
(1815572.7774188793, 5545921.253366178)
```

Figure 3.5. Python PyProj transformer example.

After the spatial transformation, the x-y coordinate system, instead of latitude and longitudes, enables to use of basic mathematic calculations, such as calculating the distance between two coordinates. The sequential waypoints in a route can serve as the basis for distance calculations, ultimately enabling the computation of route speeds.

The coordinates from GPS devices do not precisely show the real location but have noise due to various reasons such as surrounding buildings and weather conditions. These deviations can lead to erroneous results, especially in high frequent data flows. For example, even if a device is staying still, the data that it is sending can show different locations and a sequence with a 5-meter diversion in a second can be seen as a vehicle driving at 18km/h speed.

Kalman filtering can help to better estimate true values by knowing that there is a linear relationship among the states and the measurements can have noises. To handle the relationship, common Kalman filtering models can be used such as constant velocity, and constant acceleration models. The choice of model depends on the motion characteristics of the vehicles being tracked and the required accuracy

and complexity of the tracking system. For most standard vehicle tracking scenarios, a constant velocity or constant acceleration model is often sufficient. The constant jerk model is typically reserved for specialized applications where vehicle motion is highly unpredictable and dynamic.

The measurement error and the process noise variance are parameters of the Kalman filtering algorithm. The measurement error can be taken from the device specifications or if not known, it can be measured by experimenting. The process noise variance can be set depending on how much the chosen model is trusted. If we assume that the model is accurate, we may set the process noise variance (q) to a very low value.

After the setup, with the GPS records as the measurements, the Kalman filtering model estimates the state values, which will be the corrected locations. The corrected and measured locations, displacements, velocities, and accelerations should be observed using plotting. If there is a lag error shown in Figure 3.6, the model or parameters should be revisited.

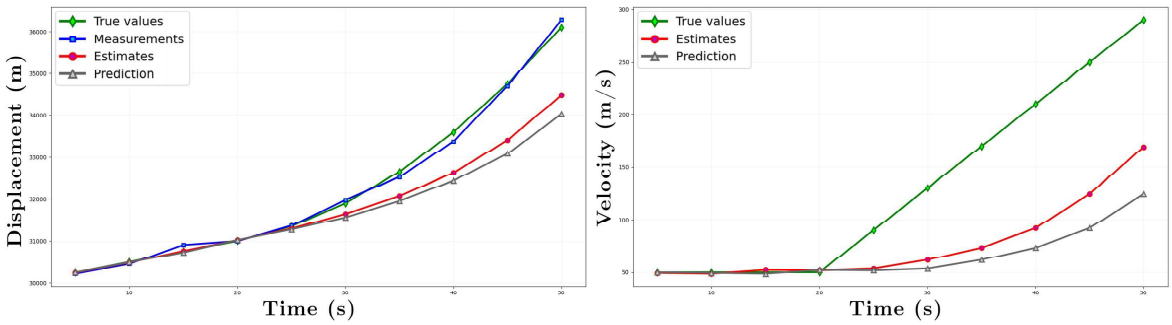

Figure 3.6. Lag error in Kalman filtering.

For the next step, "bubbles" (locational hexagonal groups) are to be calculated. To create hexagonal groups, there needs to be a reference system and determined bubble sizes. For a reference system, Uber's H3 Hexagonal Hierarchical Spatial Index can be used. The Python library *h3* has a function called *geo_to_h3*

that can convert latitude and longitude values to a hexagonal spatial index, as shown in Figure 3.7.

```
def calc_h3(row):
    return h3.geo_to_h3(row['lat'], row['lng'], 9)
df['h3'] = df[['lat', 'lng']].apply(calc_h3, axis=1)
```

Figure 3.7. Creating a hexagonal spatial index in Python.

The bearing clusters should be calculated in this step. Firstly, using the correction locations, the angles between two consecutive points are calculated. These values are between -180° and 180°. The angle 0° represents that a vehicle goes straight, whereas negative values show that a vehicle is turning left, and positive values show that a vehicle is turning right.

The K-Means clustering algorithm is applied to bearing information of all points in each bubble. To determine the optimum number of clusters, the elbow method can be used. Since there usually are four clusters representing four directions at intersections, the number four can be used as the initial value.

The spatial index and the bearing clusters should be used for aggregating the data along with time-related columns, such as weekday and hour information. The important statistics that should be calculated are mean, median, 15%, and 85% quantile speeds. The bearing, change in bearing, and acceleration values can be kept for analysis purposes.

For the next step, the routes should be matched on the roads. For this, a map network should be obtained. Open Street Map (OSM) is one of the sources that the map structure can be downloaded. The Python library *osmnx* can be helpful in extracting the data in a more programmatic way.

After downloading and creating a graph within a bounding box over the dataset, the map-matching functionalities of the library can be utilized. One of the functions is helpful for finding the nearest road to a point, which is a naïve approach. Another way for map-matching is to use the *leuvenmapmatching* library, which aligns GPS measurements to a map based on a Hidden Markov Model (HMM) with non-emitting states. Knowing which points belong to which roads enables one to have information on the street level.

At this stage, we have aggregated historical data on the bubble level and current traffic situations on the street level. The travel time estimation algorithm combines these two data flow pipelines. Using the live data to understand the traffic states, and the historical data to understand patterns. The correct combination will provide the estimates with low error rates.

3.2. Study Area

An arterial road called Nispetiye Street is selected as the study area, which is located between two freeways that are followed by two bridges on the Bosphorus Strait in Istanbul, Türkiye (Figure 3.8). The street comprises road segments containing different traffic features such as roundabouts, traffic lights, bus stops, and freeway connections. The study area has two parts in the Hisarüstü and Etiler neighborhoods. This region includes a university and schools, a hospital, and a restaurant district with high valet traffic and multiple traffic lights. There is traffic congestion even outside of peak hours. This also leads to a major uncertainty in terms of traffic, which makes testing this methodology here is therefore appropriate.



Figure 3.8. Location of the study area (Nispetiye Street).



Figure 3.9. One-hour trajectory points on Nispetiye Street encapsulated by a 100-meter buffer zone.

An area with a 100-meter buffer distance from the road is created. Then, the motorcycle trajectory points are filtered out using the area (Figure 3.9). Based on the sensor accuracy numbers, 99.9% of the data points have an accuracy that is lower than 74 meters. A buffer zone is employed to selectively filter data, preventing the processing of all information. The criterion can be modified for users by employing a methodology that applies the same approach to their data. A 100-meter threshold is chosen to accommodate users who may wish to bypass this stage. The points inside the zone are grouped by route identifier numbers, and 20 sample routes are selected for future analysis.

3.3. Data Structure and Spatial Transformations

The trajectory data consists of six columns: Data captured time, Route identifier string, Index of trajectory point within a route, Longitude, Latitude, and Accuracy of GPS (Table 3.1). The longitude and latitude information indicates a measurement of the location of a motorcycle at a given time. Latitude/longitude coordinate system based on the Earth's center of mass, used by the Global Positioning System (GPS), among others. The location columns are in the radian format of EPSG:4326 (also known as WGS 84), where EPSG is a geodetic parameter dataset that was originally created by the European Petroleum Survey Group.

Table 3.1. Sample data structure (EPSG:4326).

Time	**Route Id**	**Index**	**Longitude (degree)**	**Latitude (degree)**	**Accuracy (meter)**
2021-06-22 14:48:26.584	60d1f34...35f1f7	213	29.023263	41.077391	7.288
2021-06-22 14:48:27.418	60d1f34...35f1f7	214	29.023825	41.077437	7.494

Table 3.1. Sample data structure (EPSG:4326). (cont.)

Time	**Route Id**	**Index**	**Longitude (degree)**	**Latitude (degree)**	**Accuracy (meter)**
2021-06-22 14:48:28.349	60d1f34...35f1f7	215	29.024026	41.077456	7.661
2021-06-22 14:48:29.475	60d1f34...35f1f7	216	29.024224	41.077472	7.353
2021-06-22 14:48:31.532	60d1f34...35f1f7	217	29.024569	41.077495	7.589
...	...	...	...	...	...
2021-06-22 15:04:39.364	60d1f34...35f202	457	29.039334	41.083748	10.178
2021-06-22 15:04:42.436	60d1f34...35f202	458	29.039593	41.083948	10.587
2021-06-22 15:04:52.908	60d1f34...35f202	459	29.039931	41.084182	10.528
2021-06-22 15:04:53.538	60d1f34...35f202	460	29.040154	41.084314	11.234
2021-06-22 15:04:54.335	60d1f34...35f202	461	29.040263	41.084375	11.589

The geodetic parameter dataset includes grid-based spatial coordinate systems that use meter units. EPSG:5637 (TUREF / LCC Europe) is one the most used coordinate systems using meter units specialized for Türkiye. In this coordinate system, there is an equivalent metric coordinate for each radian coordinate. The TUREF coordinate system has the least transformation error values for Türkiye, there are other systems which are specialized for other countries. The dataset is transformed to EPSG:5637 for further analysis in order to get the benefits of Euclidean mathematics (Table 3.2).

Table 3.2. Sample data structure after spatial transformation (EPSG:5637).

Index	...	**Longitude (degree)**	**Latitude (degree)**	**Transformed Geometry (meter)**
213	...	29.023263	41.077391	POINT (5547681.208 1823033.918)
214	...	29.023825	41.077437	POINT (5547724.672 1823050.544)
215	...	29.024026	41.077456	POINT (5547740.072 1823056.735)
216	...	29.024224	41.077472	POINT (5547755.426 1823062.554)
217	...	29.024569	41.077495	POINT (5547782.215 1823072.238)
...	...	**...**	**...**	**...**
457	...	29.039334	41.083748	POINT (5548783.574 1824038.937)
458	...	29.039593	41.083948	POINT (5548798.579 1824065.341)
459	...	29.039931	41.084182	POINT (5548818.980 1824097.091)
460	...	29.040154	41.084314	POINT (5548833.100 1824115.652)
461	...	29.040263	41.084375	POINT (5548840.070 1824124.290)

3.4. Distance and Speed Calculations

The distance Δx is calculated using Euclidean distance between consecutive points. The time difference Δt is calculated using the difference in seconds between consecutive points. Speed (V) is calculated using distance and time difference as

$$\Delta t = t_i - t_{i-1}\,, \tag{3.1}$$

$$\Delta x = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}\,, \tag{3.2}$$

$$V = \frac{\Delta x}{\Delta t} \tag{3.3}$$

where x_i and y_i represent the transformed geometry coordinates at the ith index.

However, there appear to be some spikes in speed profiles calculated using the raw data (Figure 3.10). Those spikes can be identified as anomalies since there should be no data points with a speed of more than 120 km/h when considering the limitation of the motorcycles used in the study.

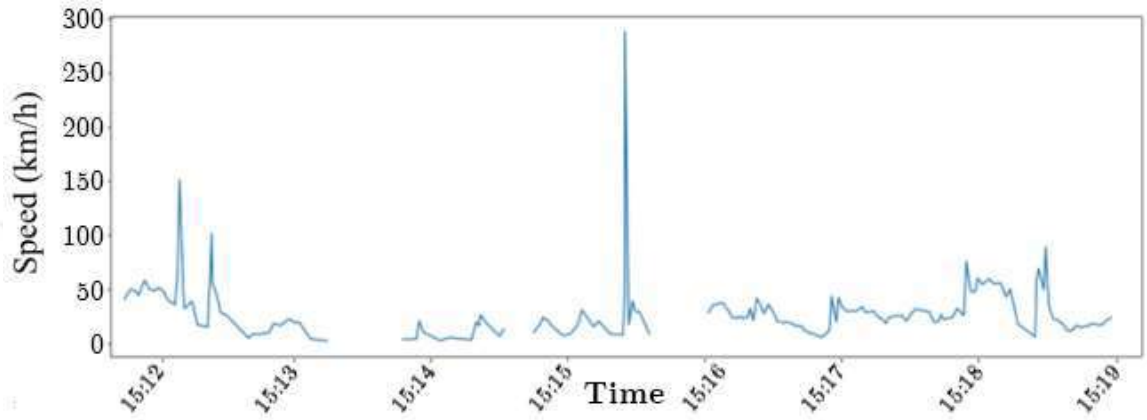

Figure 3.10. Anomalies in speed profiles calculated using raw data.

It can be claimed that anomalies in the speed profiles are caused by the irregular time-step problem. The data recordings are collected not in constant time intervals but in time intervals changing in 0.68 and 6.86 secs (Table 3.3).

Table 3.3. Irregular time step in the dataset

Distance (m)	**Speed (km/h)**	**Time**
24.945	34.394	2023-01-01 15:55:21.867
10.902	43.418	2023-01-01 15:55:22.771
11.123	45.401	2023-01-01 15:55:23.653
22.783	11.947	2023-01-01 15:55:30.518
59.049	312.154	2023-01-01 15:55:31.199
10.509	36.413	2023-01-01 15:55:32.238
10.908	39.113	2023-01-01 15:55:33.242

Global Positioning System (GPS) enables agents to have estimated location information. However, since the estimation quality depends on many variables, the locations will not be as precise as the real. Figure 3.11 shows an example with arbitrary coordinates x-y where the gray area is the distribution of possible real positions of the measurements. Data filtering and smoothing methods should eliminate outliers and correct the data.

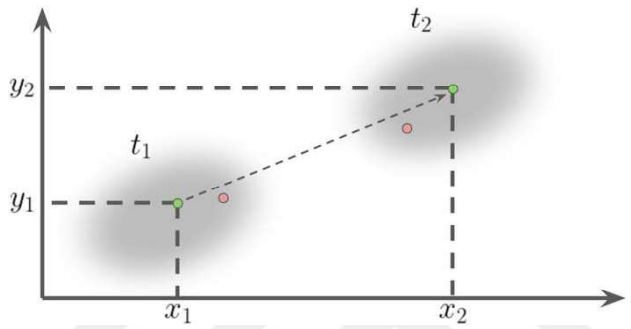

Figure 3.11. True position (red), measurement (green), estimation (gray).

3.5. Data Filtering

Smoothing using the Kalman Filter is one of the most used techniques to deal with the data with noise. The Kalman Filter is modeled with constant velocity constraints (Figure 3.12). The only information it has is the velocity in the driving direction. The x and y components of the velocity (x' and y') can be calculated from the absolute velocity and the heading of the vehicle. The filtered speed profiles are much more reasonable than the raw data (Table 3.4).

After the data filtering process, the dataset is mostly cleaned and almost ready for spatial analysis. Some of the examples are shown in Figure 3.13. Blue lines represent the calculated speed data using raw GPS records, and orange lines represent the speed profiles after the smoothing using the Kalman Filter. After the smoothing, most records of the filtered speed profiles are lower than 60 km/h, and

all of them are lower than 90 km/h, although the raw speed profiles contain many records of speeds higher than 100 km/h.

```
FILTER_SECONDS = 15
k = 0

rdf_list = []
for rid, rdf in tqdm(df.groupby('route_id')):
    rdf = rdf.reset_index(drop=True)
    rdf.sort_values('time', inplace=True)
    rdf['time_diff'] = rdf.time.diff()

    rdf_parts = []
    split_index = rdf.index[(rdf.time_diff.dt.total_seconds()>=FILTER_SECONDS)]
    p_i = 0
    for i in list(split_index) + [len(rdf)-1]:
        pdf = rdf.loc[p_i:i].copy()
        p_i = i
        if len(test_re) < 6:
            continue

        pdf.set_index('time', inplace=True)

        kalman = KalmanSmootherCV(pdf)
        filtered_traj = kalman.smooth()

        pdf['filtered'] = filtered_traj.geometry
        pdf.reset_index(inplace=True)

        pdf['filtered_dist'] = pdf.filtered.distance(pdf.filtered.shift(1))
        pdf.loc[(pdf.time_diff.dt.total_seconds()>FILTER_SECONDS), \
           'filtered_dist'] = None
        pdf['filtered_speed'] = pdf['filtered_dist'] / \
           test_re['time_diff'].dt.total_seconds() * 3.6
        pdf['filtered_acc'] = pdf['filtered_speed'].diff() / \
           test_re['time_diff'].dt.total_seconds()

        rdf_parts.append(pdf)

    if len(rdf_parts) == 0: continue

    rdf = pd.concat(rdf_parts)

    rdf_list.append(rdf)

    if len(rdf_list) % 1000 == 0:
        pd.concat(rdf_list).to_pickle(f'rdf_list_{k:03d}.pkl')
        k += 1
        rdf_list = []

pd.concat(rdf_list).to_pickle(f'rdf_list_{k:03d}.pkl')
```

Figure 3.12. (Python) Filtering speed profiles using Kalman Filter.

Table 3.4. The speed profiles after smoothing using the Kalman Filter

Time Difference (seconds)	Distance (m)	Speed (km/h)	Filtered Distance (m)	Filtered Speed (km/h)
0.904	10.902	43.418	8.706	34.670
0.882	11.123	45.401	7.273	29.687
6.865	22.783	11.947	57.624	30.218
0.681	59.049	312.154	12.248	64.751
1.039	10.509	36.413	19.077	66.100
1.004	10.908	39.113	16.856	60.442
2.075	24.991	43.358	29.457	51.106
1.929	25.183	46.999	23.431	43.729

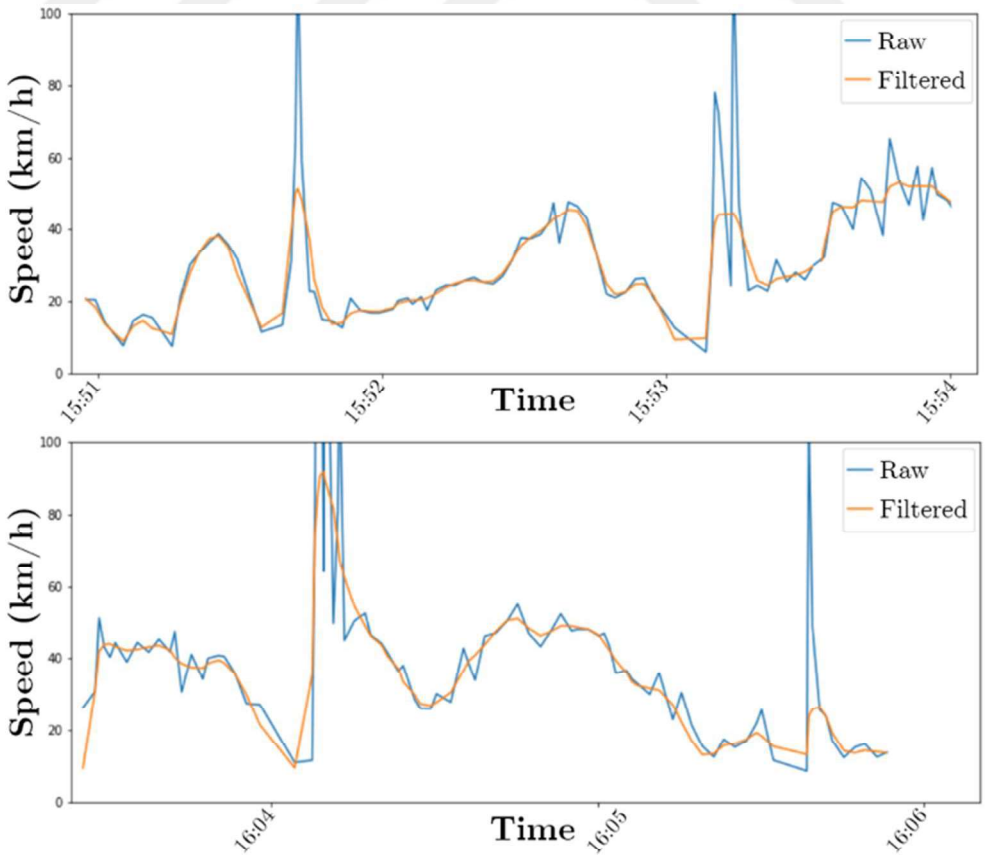

Figure 3.13. Some of the speed profiles (km/h) before and after the smoothing.

All of the sample routes selected are combined together to create a speed profile of the study area (Figure 3.14). Speed values are clustered into three groups using quantile methodology. The groups are indicated with colors, red as the slowest cluster (less than 15 km/h), green as the fastest cluster (more than 35 km/h), and yellow as the medium one.

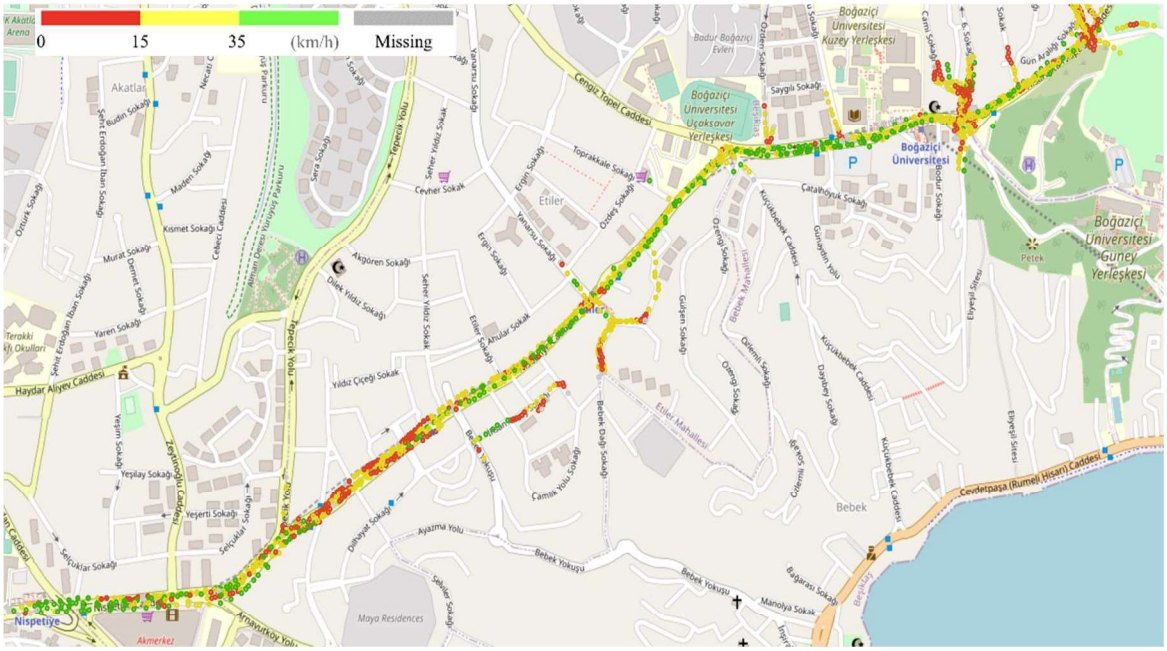

Figure 3.14. The speed profiles of the study area on the map.

3.6. Bubbles and Locational Grouping

The processed spatial-temporal of motorcycles can be converted into vehicular trajectory data using spatial analysis methodologies, such as map matching. Hence, traffic-related metrics can be calculated, such as speed, acceleration, and deceleration. From micro-scale (vehicle) to macro-scale (road, region, or hexagonal partitions), spatial statistical approaches can be applied. Hotspots and problematic areas can be identified using density estimation or clustering methods, such as Kernel Density Estimation.

On the other hand, historical big data can be processed in a newly proposed aggregated way, using so-called "bubbles". Since the above methods are computationally heavy, a light way that respects the time and space complexity of processing data is needed. One square kilometer of sample area is selected near Barbaros Avenue, as shown in Figure 3.15. The area is separated into "bubbles," which are hexagonal shapes with 100-meter side lengths. Orange shapes indicate "bubbles." Light blue lines indicate the GPS logs of motorcycles.



Figure 3.15. Bubbles: GPS logs of motorcycles.

One bubble has a side length of 100 meters, with an area of approximately 0.026 km^2 (Figure 3.16). The choice of a 100-meter bubble radius was determined by a comprehensive assessment, which involved analyzing the relationships between several bubble radii and important metrics. This analysis revealed no significant associations that would need a different size. Based on known evidence, a distance of 100 meters provides a suitable equilibrium between computational efficiency and physical interactions, taking into account the dimensions of the road and motorcycle. This choice is based in a practical physical setting.



Figure 3.16. Data points of a ‘‘bubble”.

One bubble is shown in Figure 3.17. Calculations are better to be executed at a hexagon level in order to get a summary of a bubble, such as speed profiles. Despite outlier points, which are due to GPS errors, the majority of points stay on the streets.



Figure 3.17. Flow of points with bearing (angle) information of a "bubble".

3.7. Bearing Clustering

In this step, the bearing (angle) information of motorcycles is utilized. This data indicates the angular orientation of motorcycles. The bearing values can differ from -180 to 180 degrees, as shown in Figure 3.18. Clockwise actions are set to positive, and the east direction is 0 degrees.

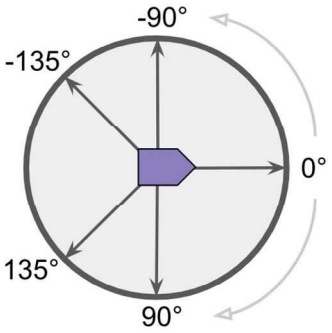

Figure 3.18. Bearing (angle) information.

The histogram of angles in the sample bubble shows that two big and two small spikes stand out (Figure 3.19). This information can be used for identifying the major road orientations.

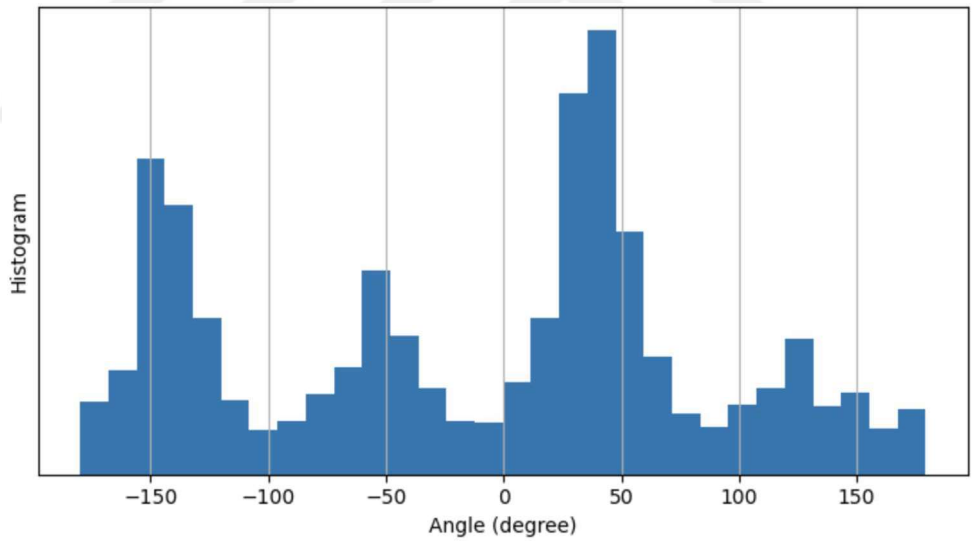

Figure 3.19. Histogram of bearing values in the sample bubble.

The results of the K-Means Clustering algorithm with 4 clusters are shown in Figure 3.20. The center points of clusters are -143.7, -52.8, 38.3, and 127.3 degrees. The first and third clusters have the most trips, which also stand on the main street in opposite directions. At the same time, the other two clusters represent the data on residential roads.

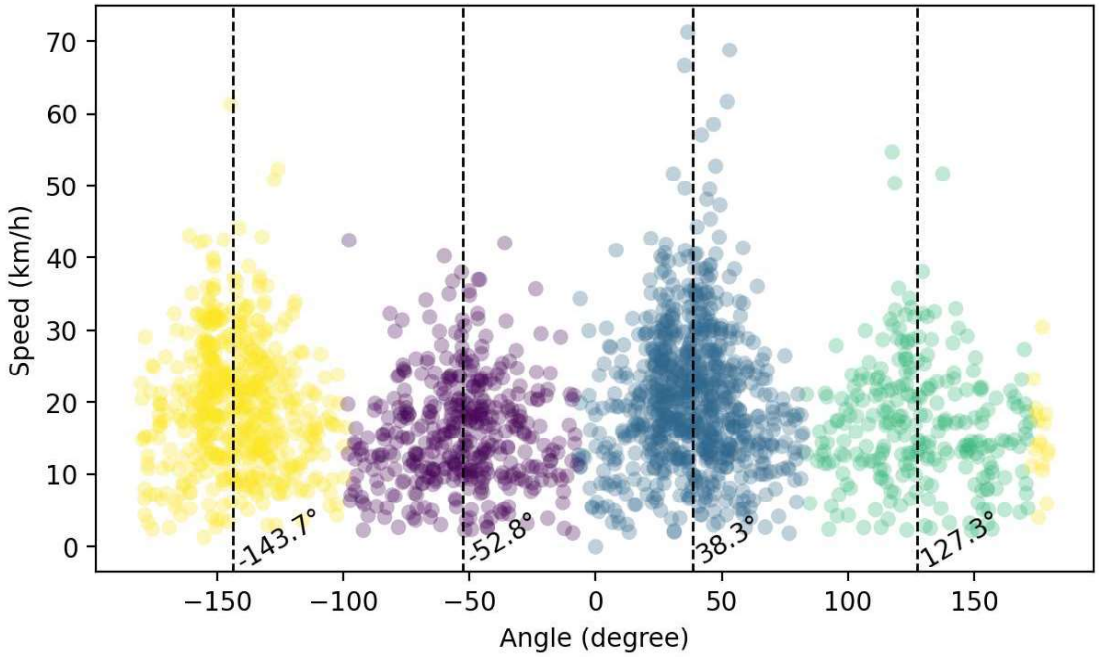

Figure 3.20. K-Means clustering to bearing values with four clusters.

The same results are converted into a polar coordinate system (Figure 3.21). It is to be noted that not all bubbles include such perpendicular cluster angles. The main street is positioned in the northwest-southeast direction, and its data is represented in blue-purple colors.

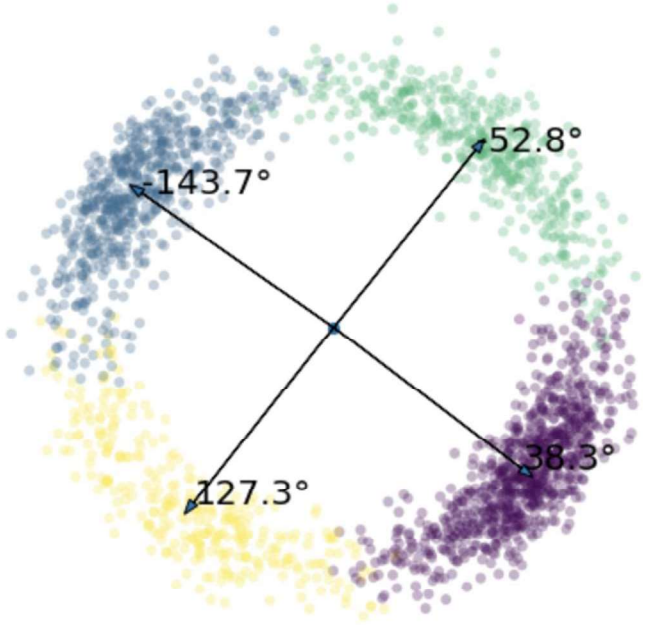

Figure 3.21. Clusters of bearing values in polar coordinate system.

Before using the bearing clusters, statistical correlations between features in a bubble could be investigated. Kernel Density Estimation (KDE), histograms, and scatter plots show the relationship between feature couples, which are distance, speed, acceleration, bearing (angle), and turning (angle difference).

3.8. Map Matching

The next step of the trajectory analysis methodology is map matching. Map matching is a process of determining individual data points to road segments. Firstly, road segments need to be obtained. City municipalities usually have their own road network datasets, which can be used for this process. For this research, the dataset from OpenStreetMap is used. An example of Python code is shared in Figure 3.22.

```
from shapely.geometry import shape
from osmnx import graph_from_polygon

polygon = {"type": "Polygon","coordinates":
[[[29.062787, 41.091255],[29.007409, 41.099922],
[28.983367, 41.028219],[29.040552, 41.049583],
[29.062787, 41.091255]]]}

G = graph_from_polygon(shape(polygon))
```

Figure 3.22. (Python) Extract road network from OpenStreetMap.

The road network in OpenStreetMap exists on a very granular level, including footway, pedestrian, steps, path, service, corridor, and cycleway road types. These road types should be excluded since motorcycles are not suitable to ride on these road types. Therefore, the map-matching algorithm will exclude these road segments for further analysis. Figure 3.23 shows before and after road network filtering. There are empty spaces, especially in the pedestrian-only parks.

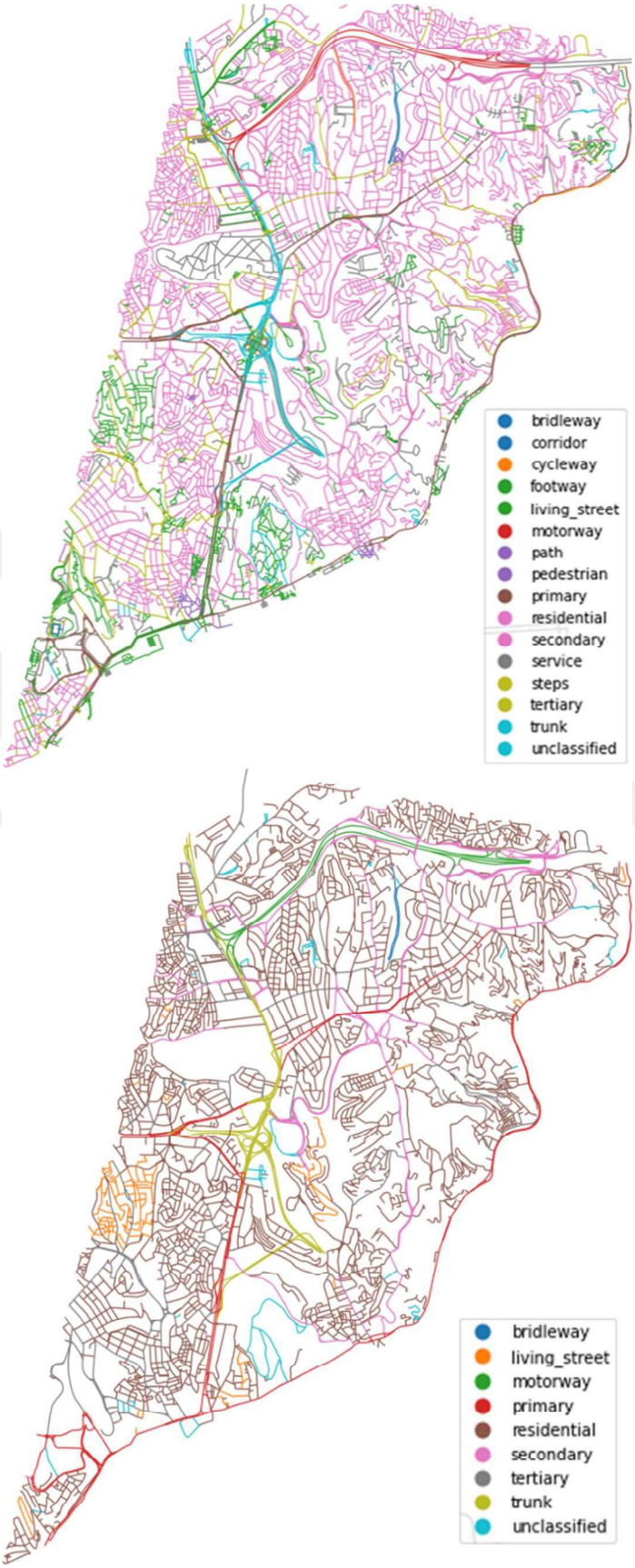

Figure 3.23. Beşiktaş road network (a) before and (b) after graph network filtering.

After the road network is extracted, the location data should be matched to the road segments. Whereas there are more complex map-matching algorithms, for the sake of brevity, the nearest edge can be considered as the matched result. OSMNX library has a built-in function for this feature, shown in Figure 3.24.

```
from osmnx import nearest_edges

edges = nearest_edges(G, df.lon, df.lat)
gdfs = ox.graph_to_gdfs(G, nodes=False)

df['nearest'] = gdfs.loc[edges].geometry
```

Figure 3.24. (Python) Getting nearest edges using OSMNX library.

3.9. Speed Analysis

A colored map of speed distribution can be generated using road network data and map-matched location data. The calculated speed information can be separated hourly and grouped by road segments that were matched. The average speed values of road segments can be visualized on a map (Figure 3.25).

```
road_avg = df.groupby('nearest').filtered_speed \
  .agg(['mean', 'size']) \
  .loc[lambda x: x['size']>1, 'mean'] \
  .rename('avg_speed')

road_avg['color'] = pd.cut(road_avg['avg_speed'],
  [0,15,35,100,1000],
  labels=['red', 'yellow', 'green', 'gray'])

clrs = gdfs.join(road_avg)['color'].fillna('gray')
gdfs.join(road_avg).plot(color=clrs, legend=True)
```

Figure 3.25. (Python) Plotting speed distribution on map.

3.10. Travel Time Estimation

The sample routes, whose real-life travel times are recorded, are tested. For each route, the following steps are applied. The recent data points within p-minute intervals, in which p will be a calibration parameter before the route starts are gathered to calculate the average speeds of the road segments. The lengths of the road segments are divided by the calculated average speeds of the road segments to estimate the travel times of the route segments.

Date timestamps are used for defining intervals to extract the smoothed data points, and to use them for calculating the recent average speed of road segments as

$$\bar{v}_{S_{(i)(t_0)}} = \frac{1}{N} \sum_{t_0 - p \le t} v_{S_{(i)}} \tag{3.4}$$

where $\bar{v}_{S_{(i)(t)}}$ is the average speed of the road segment i at time t, $v_{S_{(i)(t)}}$ is the smoothed speed of a data point near the road segment i at time t, t_0 is the starting timestamp of the route whose travel time will be estimated, N is number of data points in the time interval p.

The edge travel time calculation uses the average speed of road segments and lengths of road segments as

$$ETT_{(r)(t_0)} = \sum_{i \in R} \frac{\bar{v}_{S_{(i)(t_0)}}}{L_i} \tag{3.5}$$

where $ETT_{(r)(t_0)}$ is the edge travel time of the route r starting at t_0, R is the set of road segments in the route r, L_i is the length of the road segment i. At this point, we will have edge travel times of the routes, which can differ by their starting times.

On the other hand, bubble average speed is calculated by using the data points in the bubble at the specific time period as

$$\bar{v}_{B_{(j)(h)(d)(c)}} = \frac{1}{N}\sum_{h,d} v_{B_{(j)(c)}} \tag{3.6}$$

where $\bar{v}_{B_{(j)(h)(d)(c)}}$ is the average speed of the bubble j at hour h, weekday d, N is number of data points in the bubble j at the time period, $v_{B_{(j)(c)}}$ is the average speed in the bubble j at the time period.

The bubble travel time calculation uses the historical average speeds of bubbles and the average distance of the bubble as

$$BTT_{(r)(t_0)} = \sum_{j \in R} \frac{\bar{v}_{B_{(j)(h)(d)(c)}}}{L_j} \tag{3.7}$$

where $BTT_{(r)(t_0)}$ is the bubble travel time of the route r starting at t_0, R is the set of bubbles through the route r, L_j is the average distance of the bubble j. At this point, we will have bubble travel times of the routes, which can differ by their starting times.

In this proposed methodology, the combined travel time calculation, the exponentially weighted moving average (EWMA) is used as

$$\bar{v}_{S_{(i)(t)}} = \alpha \times \bar{v}_{S_{(i)(t-1)}} + \beta \times \bar{v}_{B_{(j)(h)(d)(c)}} \tag{3.8}$$

where is the $\bar{\nu}_{S_{(i)(t)}}$ average speed of the road segment i at time t, $\bar{\nu}_{B_{(j)(h)(d)(c)}}$ is the average speed of the bubble j at hour h, weekday d, bearing cluster c, α is the scale factor for the recent speed value, β is the scale factor for historical temporal speed value $(1-\alpha)$.

3.11. Case Study

The methodology including data extraction, data processing, data analysis, and travel time estimation is implemented in a case study area. The case study area, as it is introduced before in Section 3.2, is an arterial road called Nispetiye Street. The road in question consists of various segments, each characterized by unique traffic elements including roundabouts, traffic signals, bus stations, and connections to highways. In its vicinity, there are educational institutions like a university and schools, a medical facility, and a bustling area of restaurants known for heavy valet activity and numerous traffic lights. Congestion is a common issue here, not just during rush hours. Such variability and unpredictability in traffic conditions make this location an ideal choice for testing the proposed methodology.

The location data of couriers driving motorcycles is collected for more than two weeks. The data is collected from their Android devices using Android's Location API. The records are stored in the central cloud database. The road structure with a 100-meter buffer zone also is uploaded into the database. The buffer zone is used for querying the records, and only records in the buffer zone are extracted for this analysis.

In this case study, the motorcycle trajectory data is structured into six columns: Data captured time, Route identifier string, Index of trajectory point within a route, Longitude, Latitude, and Accuracy of GPS. This data formation facilitated the precise measurement of motorcycle locations at specific times using the latitude/longitude coordinate system, which is central to the functioning of the Global Positioning System (GPS). The coordinates were initially in the radian format of EPSG:4326, commonly known as WGS 84.

However, for enhanced analysis and application of Euclidean mathematics, the dataset needs to be transformed. The database of coordinate systems (EPSG.io website prepared by the MapTiler Team) is used for searching for a suitable coordinate system. The EPSG:5637, a grid-based spatial coordinate system tailored for Turkiye, is selected. The transformation process is done with the help of the PyProj library in Python. The degree-based location data is transformed into meter-based data, where the x and y coordinates have units of meters.

Displacement and time difference information is calculated using the consecutive points and their timestamps. Speed and acceleration information is calculated using these values. However, the raw data contains noise due to the nature of GPS systems, which causes unnatural spikes in the results, especially in speed and acceleration values.

Filtering and smoothing are some of the solutions to noisy sensor data. Kalman filtering is one of the most suitable ways to eliminate noise from the data. The primary objective of the Kalman Filter is to yield more accurate and realistic speed profiles by considering the motorcycle's velocity and direction. The constant speed modal of the Kalman filtering is implemented for this case study. After the filtering, the spikes are disappeared from the results, and the outlier values are decreased dramatically. This filtering notably reduced the inconsistencies and improbabilities in the speed data, enhancing the reliability of the subsequent analysis.

Bubbles: A novel approach in this study is the introduction of ‘‘bubbles,” which are hexagonal segments measuring 100 meters per side. These bubbles are pivotal in converting the complex spatial-temporal data of motorcycles into more manageable vehicular trajectory data. This method significantly lightens the computational load while retaining the integrity of spatial and temporal data analysis. The bubbles also facilitate the application of spatial analysis methodologies,

such as map matching and traffic-related metrics calculations, ranging from micro to macro scales.

The study employs the K-Means Clustering to analyze the bearing (angular orientation) of the motorcycles. This method is instrumental in discerning the primary road orientations within the dataset, differentiating between the trajectories on main streets and those on residential roads. The clustering results provide insightful perspectives on traffic patterns and orientations, essential for understanding vehicular movement within the study area.

An integral part of the trajectory analysis is map matching, which involves aligning individual data points to specific road segments. This process utilizes the comprehensive road network data from OpenStreetMap, excluding paths unsuitable for motorcycles to ensure accuracy. The map matching is streamlined using the OSMNX library, which helps in identifying the nearest edges to the data points, thereby accurately placing the motorcycle trajectories on the mapped road network.

In the final step, a color-coded map depicting the speed distribution across various road segments is created. By categorizing the speed information based on time and road segments, the study provides a detailed visual representation of traffic flow and speed dynamics. This analysis is important in identifying areas of congestion, high speed, and other traffic patterns essential for urban planning and traffic management.

In summary, this case study exemplifies a comprehensive approach to analyzing motorcycle trajectory data. Through a blend of sophisticated data filtering, spatial analysis, clustering algorithms, and map matching techniques, it provides valuable insights into traffic patterns and speed profiles, offering a model for similar studies in urban traffic analysis and planning.

4. ANALYSIS RESULTS

This chapter delves into the results derived from the comprehensive analysis of motorcycle trajectory data on Nispetiye Street, employing the methodology outlined in the previous chapter. The process involved data extraction, data filtering, spatial analysis, and the use of clustering algorithms and map-matching techniques. Given the unique traffic elements and the diverse nature of this arterial road, the analysis aimed to unveil patterns and insights that could contribute to urban traffic management and planning strategies. The introduction of "bubbles" as a methodological innovation played a pivotal role in simplifying the complex spatial-temporal data, enabling a more effective analysis of vehicular movements and traffic patterns.

The turning-acceleration and speed-acceleration graphs indicate road-specific driving characteristics of motorcycles for each bubble, which can be observed in Figure 4.1. On the diagonals of the figure, we see histograms of the five outputs; displacement (distance), speed, acceleration, bearing (angle), and change of bearing. On the upper triangles, there are scatter plots of the combination of the outputs. On the lower triangle, the contour density graphs generated with the Kernel Density Estimation method appear.

The displacement and the speed values seem to have a linear correlation since there are usually fixed time steps in the data so when the displacement increases, the speed increases. Showing that the acceleration values are mostly zero, the constant speed assumption for Kalman filtering seems to be correct.

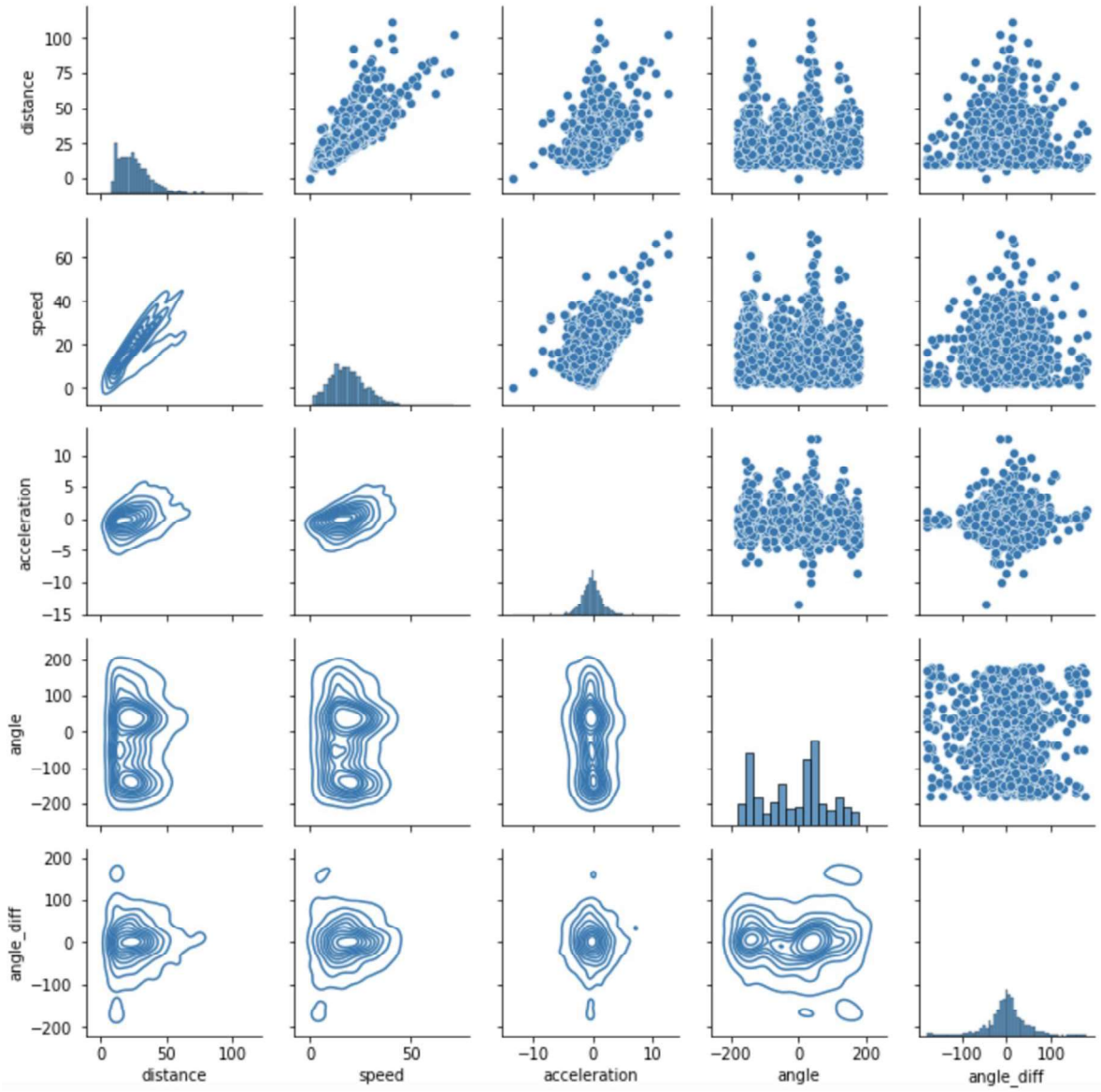

Figure 4.1. Lower triangle: KDE. Diagonal: histogram. Upper triangle: scatter plot.

With bearing clusters, weekday and hour-based investigation of speed profiles can be worth studying for important areas such as intersections. However, as seen in Figure 4.2, there is no significant pattern for bearing cluster, weekday, and hour breakdowns in the selected bubble. Each column represents a different day of the week, showing the frequency of speed values, while each row corresponds to a specific bearing cluster, illustrating variations in speed across different orientations or directions. This visualization aids in identifying patterns and anomalies in speed behavior relative to time and directional clusters.

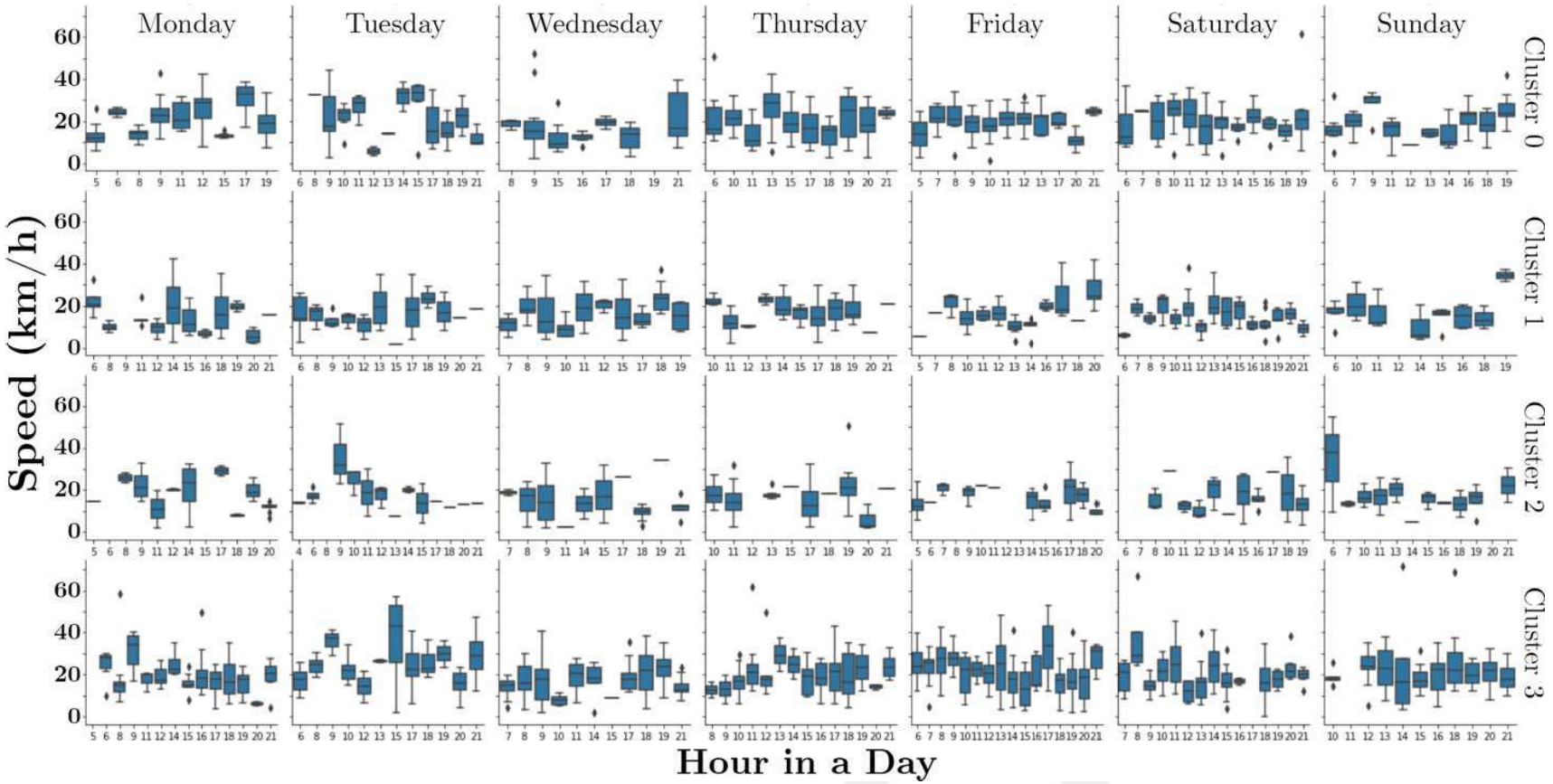

Figure 4.2. Speed profiles over time (columns: weekday, rows: bearing clusters).

The speed distributions over time are not significantly different. The effect of evening peak hours shifts the distribution left. The median speed decreases from 18 to 15 km/h (Figure 4.3).

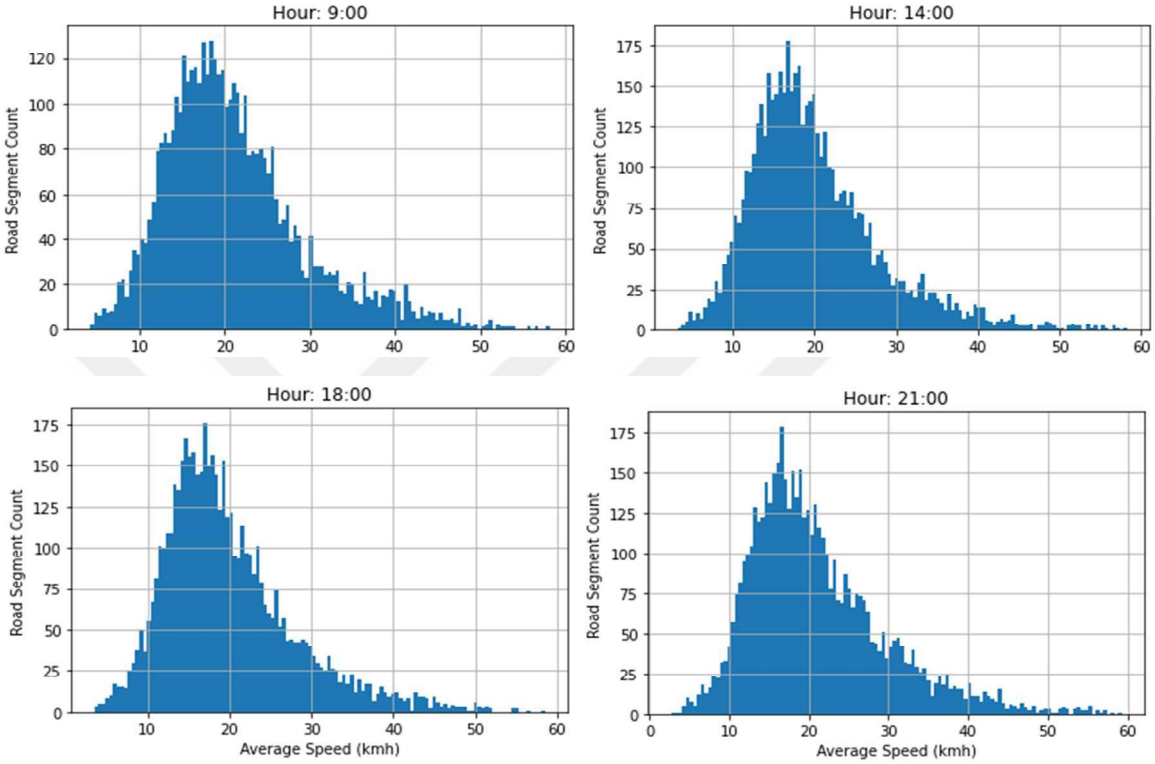

Figure 4.3. Hourly speed distribution of road segments
at (a) 09:00, (b) 14:00, (c) 18:00, (d) 21:00.

On the other hand, in the big picture, bubbles can be used for the representation of city-level speed distribution. On the main arterials, motorcycles tend to drive faster, approximately 35-40 km/h. On residential roads, the average speed is 15-20 km/h. Speed distribution in the Beşiktaş region is shown in Figure 4.4.

The residential areas can be seen as more congested at evening peak hours (Figure 4.5). Another observation is that roads with multiple lanes experience less

traffic congestion for motorcycles, even though those roads are heavily congested for cars at the peak hours.

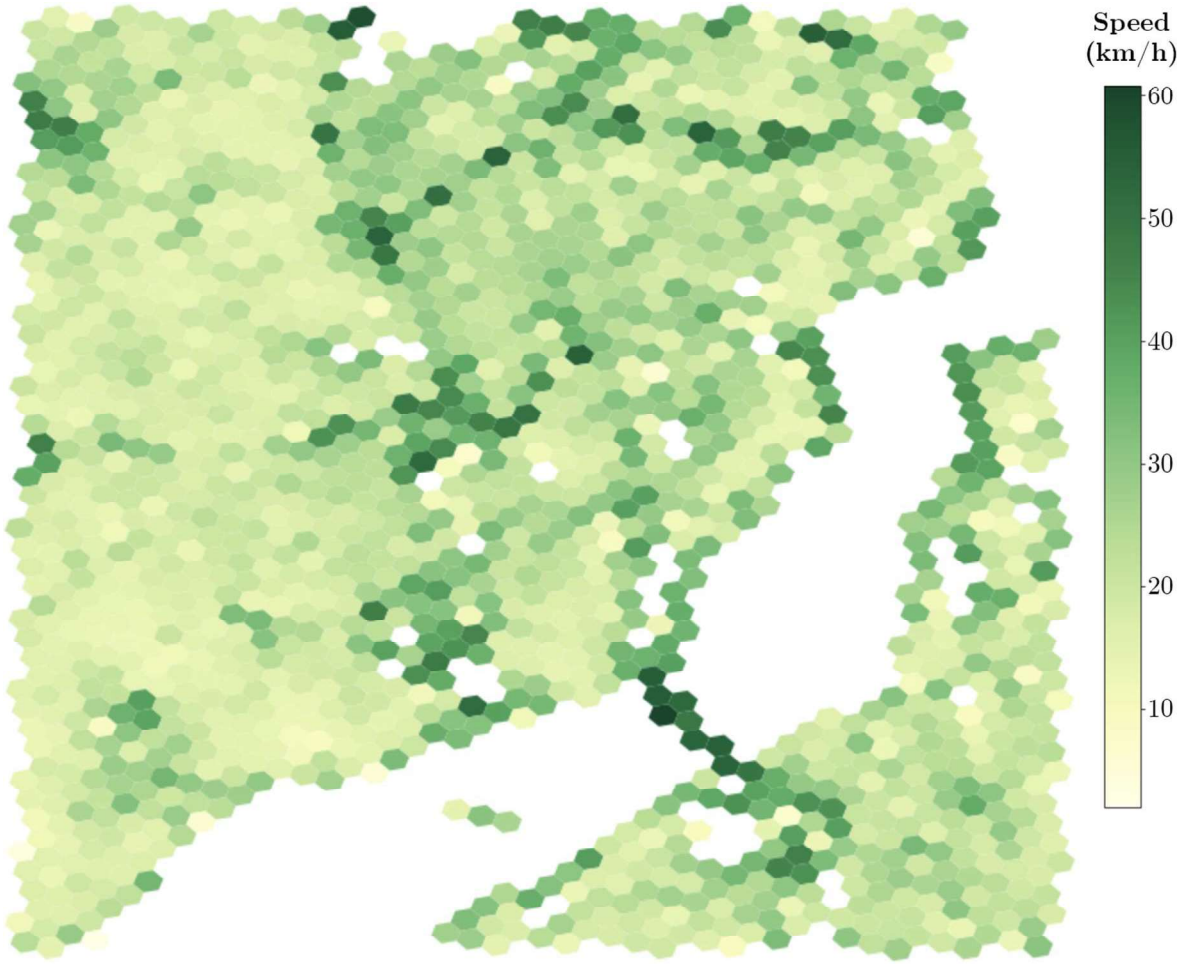

Figure 4.4. Speed distribution in the Beşiktaş region.

The results are compared with the realized travel times in Table 4.1. The mean absolute percentage error (MAPE) is 4.3%, and the mean absolute error (MAE) is 0.11 minutes, where the average travel time is 2.82 minutes, and the average speed is 24,98 km/h. The best-performing time interval is the ten-minute interval. There is 4.8% MAPE and 0.15 minutes MAE with the fifteen-minute interval, and 4.4% MAPE and 0.12 minutes MAE with the five-minute interval.

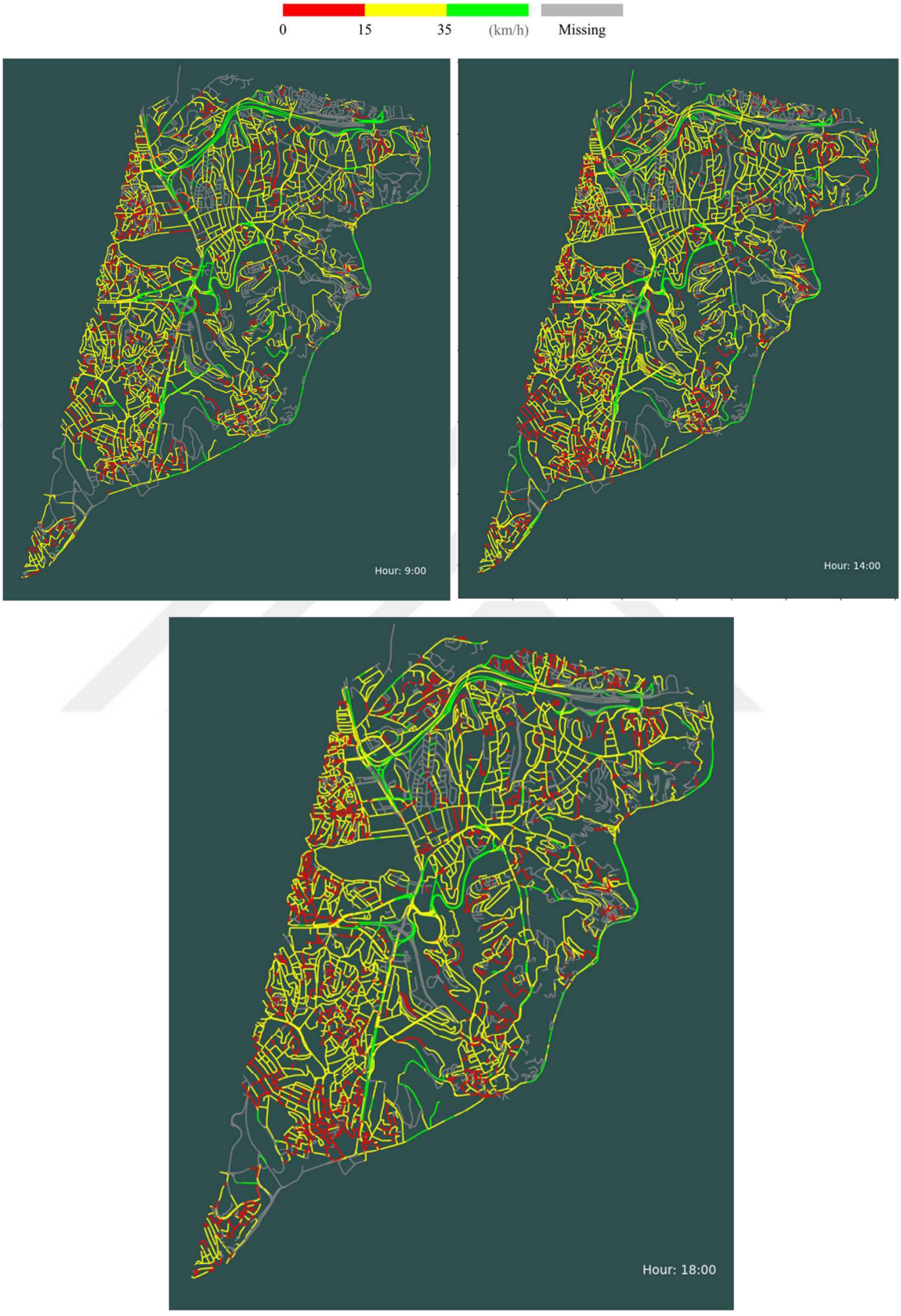

Figure 4.5. Beşiktaş road network - motorcycle speed profiles at (a) 09:00, (b) 14:00, (c) 18:00.

Table 4.1. Calibration of the travel interval parameters.

Time Interval	MAPE	MAE (minutes)
3 minutes	7.2%	0.31
4 minutes	6.1%	0.18
5 minutes	4.4%	0.12
10 minutes *	4.3%	0.11
15 minutes	4.8%	0.15
1 hour	10.3%	0.36

A grid search algorithm is applied to determine the most optimal parameters. The results suggest that the α parameter is 0.65, and the β parameter is 0.35. With this approach, it is managed to decrease the MAPE value by 9.6% and the MAE value by 7.2% as indicated in mented for accurate estimations.

Table 4.2. It shows that the combination of both the recent traffic information and the historical memory should be implemented for accurate estimations.

Table 4.2. Comparison of the results of travel time methods.

Method	**MAPE**	**MAE (minutes)**
Edge Travel Time	14.5%	0.40
Bubble Travel Time	15.9%	0.48
Combined Travel Time	13.5%	0.37

5. CONCLUSION

This study has proposed a methodology for estimating the traffic flow of powered two-wheelers (PTWs) in heterogeneous traffic using GPS data generated by motorcycles. The study successfully addressed five primary objectives: big data analysis of GPS-based trajectory data, identification of hotspot areas for motorcycles, investigation of temporal speed profiles, exploration of motorcycles' specific characteristics, and the development of a motorcycle-based estimation method for travel time.

The proposed methodology has demonstrated great potential in providing valuable insights into the behavior of PTWs in heterogeneous traffic environments. Through the analysis of big data derived from GPS-based trajectory data, researchers can effectively identify areas with high motorcycle density and pinpoint potential bottlenecks that affect travel times. Hexagonal locational shards called bubbles are an efficient way of storing historical speed values and delivering data in need of calculating travel times.

However, it is essential to acknowledge certain limitations. Firstly, the methodology's reliance on motorcycle data means that it may not be directly applicable to estimating traffic for other vehicle types, as their behaviors and characteristics may differ significantly. Furthermore, the accuracy of estimations may be compromised for new routes lacking historical data. This limitation underscores the need for continued research and development in this area. Future work should focus on addressing these limitations by exploring different and more complex combinations of historical information and recent data. Additionally, researchers could delve deeper into the factors affecting motorcycle travel times, such as road types and specifications, to enhance the accuracy of estimations.

Despite these limitations, the transferability of this methodology is noteworthy. It may be applied to other cities with typical GPS data structures, opening the door to broader applications in diverse urban settings. Moreover, it has the potential to be extended to other vehicle types that share a similar data structure, allowing for independent estimations. This flexibility is particularly valuable in situations where the interactions between multiple vehicle types are challenging to investigate.

In summary, the proposed methodology represents a transferable contribution to the field of traffic engineering. By harnessing various components such as raw GPS processing, spatial transformation, Kalman filtering, speed calculation, hexagonal locational shards (bubbles), bearing clustering, road network, map matching, and equation formulation, a comprehensive approach to traffic estimation is achieved. This methodology not only sheds light on the behavior of PTWs in heterogeneous traffic environments but also serves as a valuable resource for policymakers and researchers seeking to optimize transportation systems. Furthermore, experimenting with different parameters and applying this method to varied data sets across diverse locations can significantly enhance our understanding of its benefits, ultimately leading to a more robust and adaptable traffic management solution.

REFERENCES

1. Kaplan E.D. and Hegarty C., *Understanding GPS: Principles and Applications*, Artech House, Boston, 2017.

2. Carlier M., "Motorcycles and Cars Registered in Great Britain 2000-2022", **https://www.statista.com/statistics/312594/motorcycle-and-car-registrations-in-the-united-kingdom/**, accessed on December 10, 2023.

3. Perlot A., "Q1 2023 European Registrations of Motorcycles and Mopeds Return to Pre COVID Market Trends", **https://acem.eu/acem-statistical-release-q1-2023-european-registrations-of-motorcycles-and-mopeds-indicate-return-to-pre-covid-market-trends**, accessed on December 10, 2023.

4. Asaithambi G., V. Kanagaraj and T. Toledo, "Driving Behaviors: Models and Challenges for Non-Lane Based Mixed Traffic", *Transportation in Developing Economies,* Vol. 2, No. 2, pp. 1-16, 2016.

5. Asaithambi G. and S. Basheer, "Analysis and Modeling of Vehicle Following Behavior in Mixed Traffic Conditions", *Transportation Research Procedia,* Vol. 25, pp. 5094-5103, Shanghai, 2017.

6. Lee T. and K. Wong, "An Agent-Based Model for Queue Formation of Powered Two-Wheelers in Heterogeneous Traffic", *Physica A: Statistical Mechanics and its Applications,* Vol. 461, No. 1, pp. 199-216, 2016.

7. Hoang-Tung N. and T. L. Hoang, "Driving Behavior in Mixed Traffic Flow: A Novel Model for Assessing Bus Movement Considering the Interaction with Motorcyclists", *Journal of the International Association of Traffic and Safety Sciences Research,* Vol. 44, No. 2, pp. 125-131, 2020.

8. Wong K., T. Lee and Y. Chen, "Traffic Characteristics of Mixed Traffic Flows in Urban Arterials", *Asian Transport Studies,* Vol. 4, No. 2, pp. 379-391, 2016.

9. Amrutsamanvar R., B. Muthurajan and L. Vanajakshi, "Extraction and Analysis of Microscopic Traffic Data in Disordered Heterogeneous Traffic Conditions", *Transportation Letters,* Vol. 13, No. 1, pp. 1-20, 2021.

10. Kanagaraj V., G. Asaithambi, T. Toledo and T. Lee, "Trajectory Data and Flow Characteristics of Mixed Traffic", *Transportation Research Record,* Vol. 2491, No. 1, pp. 1-11, 2015.

11. Barmpounakis E. and N. Geroliminis, "On the New Era of Urban Traffic Monitoring with Massive Drone Data: The pNEUMA Large-Scale Field Experiment", *Transportation Research Part C: Emerging Technologies,* Vol. 111, No. 1, pp. 50-71, 2020.

12. Huang X., Y. Yin, S. Lim, G. Wang, B. Hu, J. Varadarajan, S. Zheng, A. Bulusu and R. Zimmermann, "Grab-posisi: An Extensive Real-Life Gps Trajectory Dataset in Southeast Asia", *Proceedings of the 3rd Association for Computing Machinery Special Interest Group on Spatial Information International Workshop on Prediction of Human Mobility,* pp. 1-10, Chicago, 2019.

13. Espié S., A. Boubezoul, S. Aupetit and S. Bouaziz, "Data Collection and Processing Tools for Naturalistic Study of Powered Two-Wheelers Users' Behaviours", *Accident Analysis & Prevention,* Vol. 58, No. 1, pp. 330-339, 2013.

14. Herrera J., D. Work, R. Herring, X. Ban, Q. Jacobson and A. Bayen, "Evaluation of Traffic Data Obtained via GPS-enabled Mobile Phones: The Mobile Century Field Experiment", *Transportation Research Part C: Emerging Technologies,* Vol. 18, No. 4, pp. 568-583, 2010.

15. Pham D., B. A. Mai Hoang, S. N. Thanh, H. Nguyen and V. Duong, "A Constructive Intelligent Transportation System for Urban Traffic Network in Developing Countries via GPS Data from Multiple Transportation Modes", *IEEE 18th International Conference on Intelligent Transportation Systems,* pp. 1729-1734, Gran Canaria, 2015.

16. Mai-Tan H., H. Pham-Nguyen and N. Long, "Mining Urban Traffic Condition from Crowd-Sourced Data", *SN Computer Science,* Vol. 1, No. 4, p. 225, 2020.

17. Kemajou A., R. Jaligot, M. Bosch and J. Chenal, "Assessing Motorcycle Taxi Activity in Cameroon using GPS Devices", *Journal of Transport Geography,* Vol. 79, No. 1, p. 102472, 2019.

18. Asvial M., G. Pandoyo and S. Arifin, "Internet of Things Solution for Motorcycle Riders to Overcome Traffic Jam in Jakarta Using EBkSP", *International Conference on Information and Communication Technology Convergence,* pp. 636-638, Jeju Island, 2018.

19. Babu F., P. Vortisch and T. Mathew, "Modelling of Motorcycle Movements in Mixed Traffic Conditions", *Proceedings of the Transportation Research Board 94th Annual Meeting,* p. 12, No. 15-4442, Washington DC, 2015.

20. Santos D., J. Saias, P. Quaresma and V. Nogueira, "Machine Learning Approaches to Traffic Accident Analysis and Hotspot Prediction", *Computers,* Vol. 10, No. 12, p. 157, 2021.

21. Koyama Y., T. Liang and T. Tanaka, "High-Precision GPS Measurement for Motorcycle Trajectory using Kalman Filter", *Sixth International Conference on Networked Sensing Systems,* pp. 1-4, Pittsburgh, 2009.

22. Cafiso S., A. Di Graziano, O. Giudice and G. Pappalardo, "Using GPS Data to Detect Critical Events in Motorcycle Rider Behaviour", *International Journal of Mobile Network Design and Innovation,* Vol. 5, No. 4, pp. 195-204, 2014.

23. Yalcin G. and H. Duzgun, "Spatial analysis of Two-Wheeled Vehicles Traffic Crashes: Osmaniye in Turkey", *KSCE Journal of Civil Engineering,* Vol. 19, No. 7, pp. 2225-2232, 2015.

24. Kalman R.E., "A New Approach to Linear Filtering and Prediction Problems", *Journal of Basic Engineering*, Vol. 82, No. 1, pp. 35-45, 1960.

25. Fukshansky L., "Revisiting the Hexagonal Lattice: on Optimal Lattice Circle Packing", *Elemente der Mathematik,* Vol. 66, No. 1, pp. 1-9, 2011.

26. Ciolac C., "Uber H3 API Examples on Urban Analytics in the City of Toulouse (France)", 2020, **https://github.com/uber/h3-py-notebooks/**, accessed on March 7, 2023.

27. Welch G. and G. Bishop, *An Introduction to the Kalman Filter*, Chapel Hill, North Carolina, 1995.

28. Becker A., "Kalman Filter from the Ground Up", **https:// kalmanfilter.net/**, accessed on October 19, 2023.

29. Minh C., K. Sano and S. Matsumoto, "The Speed, Flow and Headway Analyses of Motorcycle Traffic", *Journal of the Eastern Asia Society for Transportation Studies,* Vol. 6, No. 1, pp. 1496-1508, 2005.

30. Putra A. and A. Sukoco, "Analysis of Position Tracking using Motorcycle GPS Tracking", *4th International Conference on Engineering and Technology Development*, Lampung, pp. 453-461, 2017.

31. Khatri R., C. Cherry, S. Nambisan and L. Han, "Modeling Route Choice of Utilitarian Bikeshare Users with GPS Data", *Transportation Research Record,* Vol. 2587, No. 1, pp. 141-149, 2016.

32. Yang L., F. Zhang, M. Kwan, K. Wang, Z. Zuo, S. Xia, Z. Zhang and X. Zhao, "Space-Time Demand Cube for Spatial-Temporal Coverage Optimization Model of Shared Bicycle System: A Study using Big Bike GPS Data", *Journal of Transport Geography,* Vol. 88, No. 1, p. 102861, 2020.

33. Hulagu S. and H. Celikoglu, "Comparative Evaluation of Macro and Micro Approaches to Emission Modeling using GPS Data: A Case Study", *Transportation Research Procedia,* Vol. 52, No. 1, pp. 629-636, 2021.

34. Ibarra-Espinosa S., R. Ynoue, K. Ropkins, X. Zhang and E. de Freitas, "High Spatial and Temporal Resolution Vehicular Emissions in South-East Brazil with Traffic Data from Real-Time GPS and Travel Demand Models", *Atmospheric Environment,* Vol. 222, No. 1, p. 117136, 2020.

35. Barmpounakis E., E. Vlahogianni and J. Golias, "Intelligent Transportation Systems and Powered Two Wheelers Traffic", *IEEE Transactions on Intelligent Transportation Systems,* Vol. 17, No. 4, pp. 908-916, 2015.

36. Hidas P. and P. Wagner, "Review of Data Collection Methods for Microscopic Traffic Simulation", *Proceedings of 10th World Conference on Transport Research*, Istanbul, p. 125, 2004.