A Machine Learning Approach for Marginal Fulfillment Cost Estimation in Last Mile Delivery

by

Ali NALBANT

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Industrial Engineering

KOÇ ÜNİVERSİTESİ

July, 2023

A Machine Learning Approach for Marginal Fulfillment Cost Estimation in Last Mile Delivery

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Ali NALBANT

and have found that it is complete and satisfactory in all respects, and that any and all revisions required by the final examining committee have been made.

Committee Members:

Assist. Prof. Barış Yıldız (Advisor)

Prof. Fikri Karaesmen

Prof. Semra Ağralı

Date:

I dedicate this thesis to my dear family, who have always provided me unconditional love, support, and motivation throughout my academic career. I dedicate it to my mother, Akın Nalbant, whose unwavering faith in me has been the impetus behind my successes.

In addition, I honor the memory of those who tragically passed in the devastating earthquake that struck Turkey on February 6, 2023, by dedicating this thesis to their memory. My heart is with the affected families and communities, especially Hatay, where I was born and raised, severely affected by this tragedy.

ABSTRACT

A Machine Learning Approach for
Marginal Fulfillment Cost Estimation
in Last Mile Delivery
Ali NALBANT
Master of Science in Industrial Engineering
July, 2023

This thesis investigates a machine learning model to tackle the complex problem of accurately estimating the marginal fulfillment cost (MFC) for last-mile delivery for the online retail sector. Fast and accurate MFC estimation is essential for effective real-time management of online retail operations. However, due to the high computational complexity of the underlying vehicle routing problems (VRP), it is not possible to directly calculate the marginal delivery costs for real-time demand management and resource allocation decisions. To address this challenge, we propose a novel machine learning (ML) approach that eliminates the need to solve time-consuming VRP instances to calculate MFC and provides accurate predictions in milliseconds. We demonstrate the effectiveness of our proposed methodology through extensive numerical experiments with real-world delivery data. Our results show that the ML model outperforms the state-of-the-art location-based MFC estimation methods, improving decision-making capabilities for online retailers to reduce their costs, improve customer satisfaction, and reduce the negative externalities of delivery operations.

ÖZETÇE

Makine Öğrenmesi Temelli Sınırsal Teslimat Maliyeti Tahmini
Ali NALBANT
Endüstri Mühendisliği, Yüksek Lisans
Haziran 2023

Bu çalışma, çevrimiçi perakende sektöründe son adım teslimatı için marjinal yerine getirme maliyetinin (MFC) titizlikle tahmin edilmesine yönelik karmaşık problemin üstesinden gelmek amacıyla bir makine öğrenimi (ML) modeli araştırmaya odaklanmaktadır. Hızlı ve doğru marjinal MFC tahmini, çevrimiçi perakende operasyonlarının etkili gerçek zamanlı yönetimi için gerekmektedir. Ancak, temel araç rotalama problemlerinin (VRP) yüksek hesaplama karmaşıklığı nedeniyle, gerçek zamanlı talep yönetimi ve kaynak tahsisi kararları almak için marjinal teslimat maliyetlerini doğrudan hesaplamak pek de mümkün değildir. Bu çalışma, bahsi geçen zorluğun üstesinden gelmek ve MFC'yi yakınsamak için zaman alan VRP örneklerini çözme ihtiyacını ortadan kaldıran ve milisaniyeler içinde doğru tahminler sağlayan yeni bir makine öğrenimi yaklaşımı önermektedir. Önerdiğimiz metodolojinin etkinliği, e-market teslimat verileriyle yapılan kapsamlı sayısal çalışmalar ve literatürdeki çeşitli yaklaşımlar ile karşılaştırma yapılarak test edilmektedir. Sonuçlarımız, ML modelinin performansının en gelişmiş konum bilgisine dayalı MFC tahmin yöntemlerinden daha iyi performans gösterdiğini ve çevrimiçi perakendecilerin maliyetlerini düşürmek, müşteri memnuniyetini artırmak ve teslimat operasyonlarının olumsuz dışsallıklarını azaltmak için karar verme yeteneklerini geliştirebilecek potansiyeli olabileceğini göstermektedir.

ACKNOWLEDGMENTS

With profound gratitude and heartfelt appreciation, I extend my sincerest thanks to the exceptional individuals whose unwavering support and guidance have rendered the fruition of this thesis possible.

I am deeply indebted to my esteemed advisor, Barş Yıldız, whose invaluable assistance, unwavering direction, and boundless inspiration have been pivotal throughout the arduous journey of crafting this thesis. His profound expertise and meticulous guidance have indelibly shaped the caliber of my research, profoundly influencing its overall excellence and fostering my academic maturation.

My gratitude extends to the unwavering pillar of support that is my family. Their constant presence and genuine interest in my endeavors have given me a profound sense of fortune. I wish to express heartfelt appreciation to the remarkable friends who shared this transformative journey during my master's degree. Special mention goes to Atakan Yılmaz, with whom I shared the master's experience, both hailing from the esteemed Boğaziçi background. Gratitude is also extended to my esteemed colleagues Zeynep Çınar, Ali Şardağ, Damla Benli, and Bijan Bibak, whose exceptional support has been a guiding light throughout the thesis process. My dear friend, Bora Aktaş, also merits profound acknowledgment for his valuable contributions to this work.

Lastly, I extend my deepest appreciation to the esteemed jury members, Fikri Karaesmen and Semra Ağralı. Their profound insights and invaluable comments have significantly enriched this study and charted a course for future research endeavors. Their unwavering commitment to providing constructive feedback is cherished and deeply appreciated, and I consider myself fortunate to have had their esteemed presence during this academic pursuit.

TABLE OF CONTENTS

List of Tables ix

List of Figures x

Abbreviations xi

Chapter 1: Introduction 1

Chapter 2: Literature Review 5

2.1 Time Slot Management & Fulfillment Cost 5

2.2 New Delivery Mediums and & Fullfilment Cost 10

2.3 Machine Learning Applications in Routing Challanges 13

Chapter 3: Methodology 21

3.1 Problem Definition 21

3.2 Machine Learning Approach 22

3.2.1 Data labeling 23

3.2.2 Perspectives on Feature Characteristics 27

3.2.3 Feature & Model Selection 30

Chapter 4: Computational Studies 32

4.1 Data 32

4.2 Feature and model selection 33

4.2.1 Feature selection results 34

4.2.2 Model selection 37

4.3 Benchmarks approaches for MFC prediction 41

4.3.1 Benchmark - 1: Grid Approach 41

4.3.2 Benchmark - 2: Forward Value Function Approximation . . . 43
4.3.3 Benchmark - 3: Tour Patching Heuristic 45

Chapter 5: Conclusion **49**

Appendix A: Feature Lists **58**

LIST OF TABLES

3.1 Outline of the notation 22

4.1 Objective Function Parameter Values 32
4.2 Summary of the optimization runs for labeling 33
4.3 Features Selection Statistics 34
4.4 Selected Features 36
4.5 LazyPredict Model Results 38
4.6 FVFA and ERT Performances 45
4.7 Prediction Performance Compared to TPH 47

A.1 Selected Features from Literature 58
A.2 Spaital Features 59
A.3 ARCGIS-based Distance Features 60
A.4 ARCGIS-based Time Features 61

LIST OF FIGURES

1.1 Demand Management Scenarios 2

3.1 Methodology 23

3.2 Feature Classification 28

4.1 Feature-Base Performance Analysis 40

4.2 Feature Performance 41

4.3 Grid Approach Representation 42

4.4 Grid - Based Approach 42

4.5 An Instance (a) Original tour, (b) Removal Applied, (c) Re-Optimized 46

ABBREVIATIONS

DFJ	Dantzig-Fulkerson-Johnson
ERT	Extremely Randomized Trees
FVFA	Forward Value Function Approximation
MFC	Marginal Fulfillment Cost
MFCP	Marginal Fulfillment Cost Problem
MILP	Mixed-Integer Linear Programming
ML	Machine Learning
RMSE	Root Mean Squared Error
TSP	Traveling Salesman Problem
VRP	Vehicle Routing Problem

Chapter 1

INTRODUCTION

The e-grocery market has shown a steady growth trajectory, and the COVID-19 pandemic further fueled this expansion, changing consumer habits with a lasting impact on the post-pandemic world. Home deliveries have become an essential part of the grocery industry as online shopping becomes the new norm in the retail sector. Only in the US, the e-Commerce market size is expected to reach $250 billion within a span of two years [23]. Approximately 150 million Americans, or nearly half of the country's population, currently buy groceries online, and future projections indicate that this figure will rise much higher [24]. The catalytic forces of the COVID-19 pandemic, coupled with technological advancements, have fundamentally transformed the grocery shopping landscape in recent years [3].

While the world was transforming, the adaptation of e-grocery companies was slow and challenging. Numerous issues have arisen due to the surge in demand for online grocery shopping, including overloaded delivery schedules, canceled membership applications, and sporadic product shortages [49]. Due to these difficulties, disrupted global value chains, and supply limitations, prices have risen globally, contributing to food inflation.

While customers demand more flexibility in their choice of delivery times and tighter time windows, e-grocery businesses struggle to meet these arduous service quality standards without increasing operational costs [41]. Growing concerns about the negative environmental impact of last-mile delivery operations also put more pressure on e-grocers to increase efficiency in their delivery operations [10, 43, 14]. As a significant challenge, offering flexible and tight delivery times to customers results in a sparser distribution of delivery demand over space and time (at least for some specific locations and/or delivery time slots), which can significantly increase delivery costs [12]. To build some intuition about the critical importance of demand's spatial and temporal distribution over the delivery costs, we present two small toy scenarios in Figure 1.1.

In Figure 1.1.a, we show the tour of a delivery vehicle to serve from the depot (indicated as **D**) to six customers in the same delivery time slot. As shown in Figure 1.1.b, removing customer-6, for example, serving her by a crowd-shipper, can significantly shorten the vehicle's tour to reduce the costs and environmental impact. Similarly, Figures 1.1.c and 1.1.d show how demand management can help reduce delivery costs. In this scenario, customers 1, 4, and 6 demand their deliveries in the first time slot, and customers 2,3 and 5 prefer the second time slot, resulting in the red and blue delivery tours for the vehicle in the first and second time slots. We see in Figure 1.1.d that if customers 1 and 3 can be convinced to change their delivery time slots, the delivery vehicle can significantly shorten its tours.

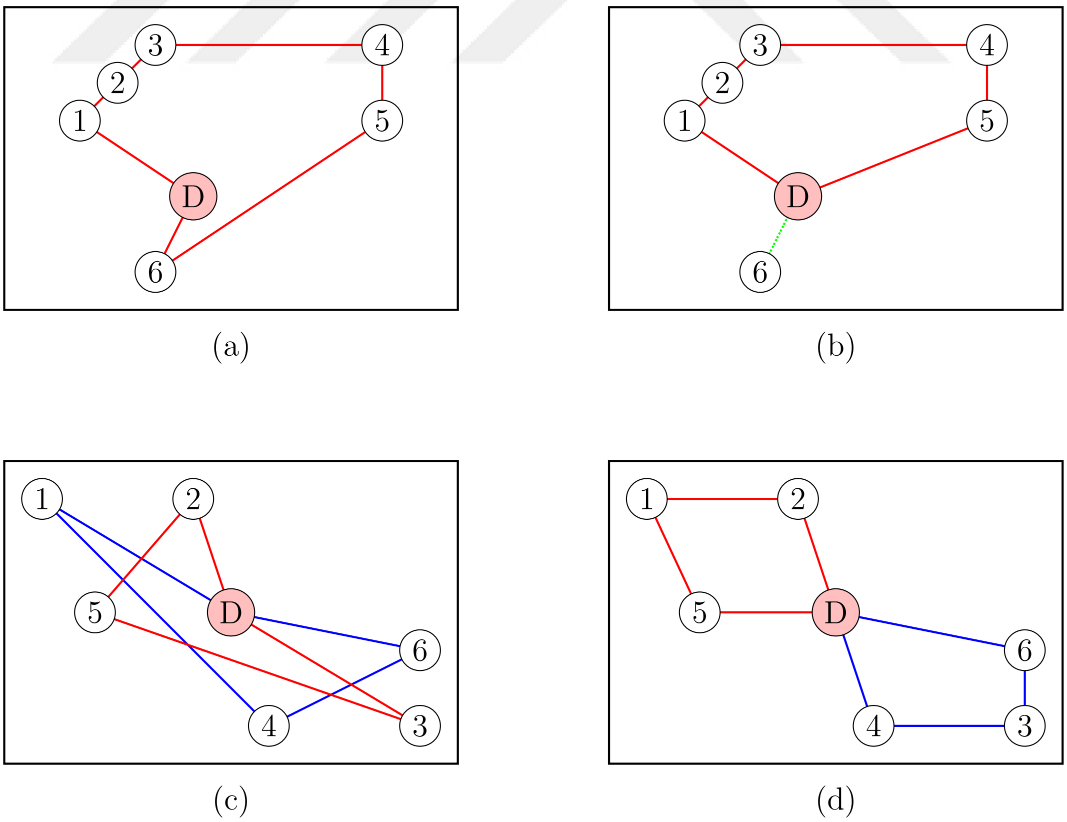


Figure 1.1: Demand Management Scenarios

Researchers and businesses look for novel demand management and outsourcing mechanisms to modify the temporal and spatial distribution of the demand to reduce the monetary and environmental costs of attended home deliveries [38, 15, 48]. One such interesting demand management mechanism that attracts growing interest is *time-slot management*, which aims to shape customers' delivery time preferences through discount labeling, green labeling, or displaying limited delivery time window options to the customers [1, 13, 37, 50, 54, 2]. For an effective time slot management it is crucial to understand the marginal fulfillment costs (MFC) of customers to optimize incentives that can achieve the intended cost reductions. As a significant challenge to successfully implement such strategies, e-grocers have rather limited time (typically not more than a few seconds) to engage with customers to decide on time-slot offering and their labels (i.e., prices or environmental impact indicators of available time slots) without full information about the additional demand that will arrive later. Similarly, novel approaches to extend the delivery mediums, such as delegating some deliveries to instore customers or occasional drivers as crowd-shippers [22] or promoting store pick-ups for online customers [36] require fast and accurate estimations and MFC to optimize the incentives to offer to crowd-shippers or customers. However, due to the complex nature of the underlying vehicle routing problems (VRP), brute force approaches to calculate/estimate MFC that require solving computationally challenging routing problems are not applicable in practice.

Despite the practical urgency, there is a notable gap in the literature regarding the fast and accurate calculation/estimation of MFC. This thesis aims to address this gap and propose a novel machine learning (ML) approach for MFC prediction that eliminates the need for solving complex VRP problem(s) during decision-making. With a novel data farming approach to create the training data by solving many VRP instances that are generated from the historical data, the suggested approach essentially moves the bulk of the computational effort to an offline preparation phase so that accurate MFC predictions can be obtained in milliseconds to use in the real-time systems management. To build an accurate ML model, we conduct extensive feature engineering, in which we consider features from literature (used in different VRP-related ML models) and propose new ones that we show to significantly contribute to the model's prediction power. We conduct extensive numerical studies with real-world data from a large e-commerce chain in Turkey to assess the performance of our approach, comparing it to various benchmarks. Our detailed tests with different feature sets to build ML models reveals interesting insights about the drivers of MFC that can be used in different applications.

The remainder of this thesis is structured as follows: Chapter 2 provides a comprehensive literature review, summarizing relevant studies on last-mile delivery and fulfillment cost estimation. This chapter also reveals ML studies for routing and last-mile-oriented problems. Chapter 3 presents the detailed methodology, outlining the steps to address the research problem. Chapter 4 presents the computational results, highlighting the performance and effectiveness of the ML model. Chapter 5 concludes our study by summarizing the essential findings and their significance in advancing the field of last-mile delivery and fulfillment cost estimation. It also presents potential future research directions and implications of the findings.

Chapter 2

LITERATURE REVIEW

In their recent survey on last-mile delivery concepts and models, Boysen et al. [12] introduce increasing volume, time pressure, sustainability, costs, and an aging workforce as the five major challenges in last-mile delivery. Except for the aging workforce, the fast (real-time) MFC prediction problem we focus on in this thesis intersects with all of these major challenges. The literature review is divided into two main sections relevant to our study. First, we look at studies on cost allocation and MFC approximation methods, which shed light on efficient cost and time-slot management. Second, we look into research on ML applications on the last mile and VRP. This research stream emphasizes machine learning as a potent decision support and prediction tool, demonstrating its efficacy in enhancing decision-making processes.

By integrating these research streams, offering a comprehensive understanding of fulfillment cost management, and proposing a novel ML strategy, our research seeks to add to the existing literature. In the following, we investigate related studies from application and methodology perspectives and clarify the relevance of our study within this framework.

2.1 *Time Slot Management* & *Fulfillment Cost*

As one of the pioneering studies in delivery time management, Campbell and Savelsbergh [16] concentrate on the difficulties that businesses operating under the consumer direct service model (direct engagement, bypassing intermediaries), particularly those engaged in grocery delivery, suffer due to demand uncertainty, tight delivery time windows, and low-profit margin goods. The study aims to understand how incentives might be used to change consumer behavior to reduce delivery costs.

Optimization models for two kinds of incentives, namely time slot incentives and wider slot incentives, are proposed and assessed through simulation experiments. The study also includes calculating the actual extra cost of making the delivery while taking various timetables and insertion points into account and assessing the cost of accepting an order in a brief span.

One of the recent studies that concentrate on the difficulties attended home delivery services face in providing narrow delivery time slots while limiting delivery costs is presented by Yang et al. [57]. They suggest a dynamic pricing approach to guide customers toward cost-effective schedules during the booking process. The researchers show how well their method calibrated pricing policies by estimating a customer choice model from historical data. They stress the significance of considering delivery costs, which are dynamically estimated when determining the incentives (discounts or charges) for each time slot. The authors introduce a heuristic method based on Campbell and Savelsberg's work [16] to orientate time slot pricing. Their strategy accepts orders and maintains a pool of schedules to improve the chances of finding viable alternatives. Simulation studies with actual data demonstrate that, in contrast to industry practices, foreseeing future delivery costs could significantly increase profitability. The study emphasizes the importance of dynamic pricing tactics in increasing the effectiveness of attended home delivery services, offering valuable information for businesses engaged in this industry.

In their article [20], Cleophas and Ehmke address two research questions. They first create an iterative solution strategy for value-based demand fulfillment to maximize predicted order value while considering transport capacities and minimizing transportation costs. Based on past data, they determine predicted delivery requests and pinpoint effective delivery times for scheduling delivery tours. Second, they combine revenue management and vehicle routing strategies to analyze the potential effects on expenses and revenues. Their simulation results show that precise demand forecasting is essential for increasing profits without incurring high transportation costs. However, as homogeneous demand restricts the efficacy of the suggested strategy, value-based fulfillment depends on accurate forecasting and an existing demand segmentation.

Online retailers using attended home delivery business models are explored by Klein et al. [31], highlighting the value of correct opportunity cost approximation for efficient dynamic time-slot pricing decisions. They propose a new approach based on the integration of mixed integer linear programming into the dynamic pricing framework. Their opportunity cost estimate considers the most recent customer data and an anticipated customer projection and is adjusted throughout the booking horizon. A dynamic seed-based approach approximates vehicle routes and delivery costs based on accepted customer locations. Their strategy consistently produces the maximum return, even when there are severe capacity restrictions, according to computational simulations. The study uses parallel computing and recurring opportunity cost recalculations to show their approach's practical applicability in real-time contexts.

Klein et al. [32] investigate the tactical issue of differentiated time slot pricing for e-grocers to achieve cost-effective delivery schedules and maximize overall profit. They aim to influence customer behavior when making time-slot selections. They develop a mixed-integer linear programming model for the issue, considering routing restrictions, and incorporating a non-parametric rank-based choice model to understand user behavior. They also suggest a cost anticipation approximation appropriate for real-world problem sizes. They argue the superiority of their model-based approximation over benchmark approaches used in the literature and real-world applications through an extensive computational study motivated by the real-world operations of their industry partner. Their findings demonstrate the usefulness of the approximation as a supplementary tool for actual pricing decisions made by e-grocers. The aforementioned studies highlight the critical importance of effective demand management and offer various mechanisms that depend on a reliable estimation of MFC. Our research distinguishes itself from these studies by its technical focus on MFC estimation. We aim to develop a fast and accurate MFC estimation method rather than studying models that use this information in real-time operations management to reduce delivery costs. Our study seeks a deeper understanding of the dynamics that shape the MFC and uses this understanding to build an effective ML model with an unparalleled prescience.

A notable improvement in estimating MFC is merging the region and time-slot based forward value approximation technique. Vinsensius et al. [54] show how dynamic incentive-oriented slot management can be effectively combined with the MFC to significantly reduce costs (up to 70%, according to their synthetic data) and improve the profitability of attending-home delivery. In contrast to conventional estimation techniques, Vinsensius et al. [54] propose a novel approximation method for evaluating MFC. This method uses a decomposition approach that simultaneously considers delivery slots and regions for more accurate calculations. The authors' strategy yields promising results in their extensive numerical experiments with synthetic data. Their research adds to earlier work on incentive mechanisms by Campbell et al.[16] and on approximating value function estimation by Yang et al. [57]. Vinsensius et al. [54] optimize the incentive menu and reduce overall delivery costs by combining approximate dynamic programming, a linear incentive function, an approximate value function, and estimated MFCs. In our study, we use the location-based prediction approach of Vinsensius et al. [54] as a benchmark to show the merits of a more sophisticated ML-based methodology for MFC estimation. We present a detailed analysis of the strengths and weaknesses of the suggested methodology and how our approach compares, considering the results of our extensive numerical experiments with real-world data in Chapter 4.

While financial incentives still matter, buyers also have environmental concerns about last-mile delivery. They prioritize environmentally friendly packaging, reduced waste production, and actions that reduce carbon emissions. This paradigm shift introduces a new tool of managing time slots for e-grocery; green labeling. It allows customers to select environmentally friendly delivery options. E-grocery providers can satisfy customer expectations and lay the groundwork for long-term customer loyalty and trust by adopting and prioritizing these labels.

Agatz, Fan, and Stam [2] focus on the influence of green labeling on last-mile delivery time slot selection and operational sustainability. The study focuses on how well green labels incentivize customers to select delivery time slots based on intrinsic motivation rather than financial incentives. The study was made up of two experiments and two simulation studies. The results show that green labels significantly impact customer decisions about delivery options, particularly for environmentally sensitive ones. Green labels successfully persuade customers to choose wider delivery time slots, even without price incentives. According to the study, green labels can be a viable strategy for retailers to control delivery window management and enhance last-mile operational performance. It is noteworthy to stress that using ad hoc methods to identify green labels may have limitations, whereas using a method that accurately predicts the environmental cost can significantly enhance the guidance and selection of green options, underscoring the accuracy and effectiveness of the prediction method used in such studies. In our context, a higher MFC denotes longer travel distances or times, while a lower MFC denotes more environmentally friendly options. Locating slots with lower MFC increases sustainability and reduces environmental impact. Eco-friendly practices are considered when MFC is incorporated into green labeling initiatives, improving sustainability and cost-effectiveness.

While numerous studies highlight the usefulness and financial advantages of MFC in last-mile logistics, there is extensive literature on the fair allocation of costs among the parties involved [55, 19]. These studies examine fair cost allocation using techniques such as the Shapley value [55, 34, 39], delving into cooperative game theory (CGT) ideas. These researches offer essential insights into cost-sharing mechanism optimization, ensuring fairness, and encouraging stakeholder collaboration during decision-making by addressing cost allocation. Understanding equitable cost distribution boosts last-mile logistics operations' overall effectiveness and sustainability. These studies aim to develop more effective algorithms for allocating transport costs between players, ultimately improving the efficiency of the last-mile delivery process. The Traveling Salesman Game (TSG) is a cooperative game in which the costs of a round trip are shared among several customers. One of the key solution concepts in cooperative game theory is the *Shapley* value, which provides a fair distribution of costs among players based on their marginal contributions. A recent study [34] proposed efficient methods for computing the Shapley value for TSG. The authors present two variants of TSG, one with a fixed order (predetermined sequence in which customers are serviced or requests are processed) of customer service and the other without, and demonstrate a method for efficiently computing the Shapley value for the first variant. They also propose a proxy approach that outperforms existing methods for the latter variant. Another study [6] addresses the challenge of computing the cost of serving each location in a single-vehicle transport setting using TSG. The authors propose and comprehensively evaluate various approaches, including an approximation method based on the Shapley value.

Calculating the Shapley value for the Traveling Salesman Problem (TSP) parties is computationally expensive due to the factorial nature of the Shapley. Especially for large datasets, the calculation can require substantial computational capacity. Therefore, researchers have proposed methods for estimating the Shapley value in TSP, including approximation algorithms and sampling techniques, to provide computationally efficient solutions for optimizing the routing process in last-mile logistics. Although Shapley shades a light an essential light on cost allocation, MFC and Shapley have intricate differences. First, Shapley divides the total cost so that parties in the sum equal the total. That is not the case for MFC. Its dynamic nature enables it to follow iterative changes in the set. It provides one-step (removing a customer from the system) changes. Nonetheless, considering the ordering dynamics, MFC still furnishes benefits for last-mile management. Furthermore, it can still be a support tool for portioning the total cost rather than simplistic bases [28], such as stand-alone costs [51] (cost incurred if someone is the only member of the set/group) in the set.

2.2 *New Delivery Mediums and* & *Fullfilment Cost*

In addition to studies on time slot management, there is also a growing interest in novel approaches that offer flexible outsourcing solutions for service provision. Crowdsourcing, a concept put forth by Howe [30], also refers to outsourcing value-creation activities to an undefined network of individuals and has profound implications for various industries, including logistics. Crowd logistics, a marketplace concept with information connectivity that matches supply and demand for logistics services with an external crowd, provides a ground-breaking answer to the last-mile issue [15]. Additionally, *Crowd Local Delivery* is a concept that can potentially improve last-mile parcel delivery by utilizing idle crowd resources to pick up and deliver packages locally using various transportation methods [17, 5]. According to Rougès and Montreuil [45] and Sampaio et al. [46], this strategy may result in more affordable, adaptable, and environmentally friendly deliveries. Furthermore, all parties involved in crowd-shipping may advantage from a wide range of compelling benefits [18]. Businesses can expand their service areas while drastically cutting the delivery cost [47]. Drivers can increase their income by including delivery tasks on their existing routes. Society benefits from reduced emissions and less traffic congestion. Additionally, crowd-shipping offers a versatile and effective solution for last-mile delivery by utilizing various vehicles, such as cars and bicycles.

Combining professional delivery services with occasional crowd shippers, Yıldız [58] presents an innovative ”courier-friendly” crowd-shipping model to facilitate effective express package deliveries in urban areas. The model incorporates transshipment points, which improve operational efficiency while maximizing occasional couriers' participation. The model also features a company-controlled backup delivery system to account for uncertainties in the crowd-sourced delivery capacity. Yıldız creates a dynamic programming method to successfully address the issue of real-time package routing within the crowd-shipment network. It incorporates Monte Carlo simulations to estimate shadow costs while considering several variables, including limited capacities and advance notice periods. In a more recent study, Ghaderi et al. [27] propose a novel approach to crowd-sourced last-mile delivery inspired by Physical Internet structure. They develop algorithms for locating parcel lockers, allocating delivery tasks, and assigning jobs to crowd-shippers. The experiments conducted demonstrate that the proposed algorithms provide (sub-)optimal results with less computational load. Furthermore, the study highlights the benefits of enabling joint delivery that includes two shippers through strategically located parcel lockers, improving delivery success rates by up to 5%.

Dayarian & Savelsberg [22] explores the use of crow-shipping in e-grocery deliveries by providing a sophisticated and innovative approach for same-day delivery. Offering a new perspective on extending the delivery mediums for e-grocery operations, the study reveals the potential advantages of crowd-shipping in maximizing service quality and operational costs through rigorous methodologies and insightful analyses. These results promote crowd-shipping as a cutting-edge solution in last-mile delivery, making a strong case for companies to adopt it to gain a competitive edge in the market by increasing availability and reducing costs simultaneously.

Although these approaches use the excess capacity of independent contractors and third-party logistics providers to optimize delivery routes and schedules, reduce delivery costs and improve operational efficiency; successful implementation requires careful consideration of factors such as capacity utilization and quality control. Moreover, bringing a third party into the picture introduces different risk factors. Additionally, couriers and delivery companies need effective pricing strategies to be successful. According to Roug‘es and Montreuil [45], there are five revenue models used by 26 CS companies, each with unique benefits and strategic applications depending on market segments and time of day. To choose a pricing strategy, companies need to analyze historical data to accurately determine MFCs in the short and long term. Outsourcing studies primarily focus on addressing the applicability and assessing the risk and benefits connected to Crowdshipper, a third-party logistics service provider. Thorough comprehension of the MFC is essential for successfully managing profitable operations and making informed bid decisions in the ever-changing world of outsourcing, customer-oriented utilizations, and drone or cargo bike deliveries. Analyzing and evaluating the viability of different customer sets allows businesses to tailor their offers and enhance their delivery services strategically. This knowledge is essential for optimizing profitability, ensuring effective resource allocation, and preserving market competitiveness. By utilizing this MFC knowledge, companies can optimize operations, provide outstanding customer experiences, and promote sustainable success in the dynamic last-mile delivery market. Our study promotes the incorporation of the MFC, which enables quick and effective integration with outsourced services when necessary.

2.3 Machine Learning Applications in Routing Challanges

ML techniques have been combined with analytical strategies to address complex decision problems in various real-world last-mile delivery operations. Bai et al. [7] provide an extensive review on this topic, arguing that the use of ML techniques in the logistics and transport industries can have a significant impact on solving complex operational problems on a larger scale that traditional approaches may not be able to handle effectively. The review discusses ML-assisted VRP modeling and optimization, emphasizing how ML can improve VRP modeling and algorithm performance for online and offline optimizations. The authors highlight the critical need for the ML and operations research communities to work together to develop novel approaches to deal with challenging logistics management problems that involve real-time decision-making in highly stochastic environments that involve intricate vehicle routing elements.

Although many VRP studies address stochasticity, they can only be used in small-scale or theoretical studies. They are rarely used in practical applications and are difficult to implement. A developing area of research combines data analytics, machine learning, and optimization methods to increase practicality. This hybrid approach closes the gap between theory and practical VRP implementation by addressing issues like simplifying assumptions, parameter estimation, and the practicality of solution algorithms. The VRP and its variants are NP-hard problems, so the applicability of exact algorithms is limited, particularly for large-scale problems. Exact methods attempt to find a (near-)optimal solution by treating the VRP as an integer or mixed-integer program. However, heuristic-based approaches are frequently preferred due to the size and complexity of real-world VRPs. Elshaer and Awad [25] claim that metaheuristics are used in more than 70% of the literature's solution methods. These metaheuristics use a variety of escape strategies from local optima, but they cannot ensure optimality [11]. These statements indicate a clear gap in the literature to introduce well-performing, generalizable ML-based models.

Exploring the potential of ML applications in the context of last-mile delivery has gained popularity in recent years [52]. Significant advances have been achieved in dynamic time slot management, travel time prediction, order assignment optimization, and arrival time estimation through ML methodologies. One such example is Liu et al.'s [35] work that attempted to investigate the use of delivery data to improve the punctuality of last-mile delivery services. They offer a framework that combines order-assignment optimization and travel-time predictors to model the driver's unpredictable and complex routing behavior. The study discusses tractable predictors and prediction models compatible with current optimization tools and focuses on the order-assignment problem. Additionally, they offer reformulated versions of the integrated models and suggest effective algorithms for problem-solving. The researchers run numerical tests on actual delivery data to show how well their suggested models perform when combined with travel-time predictors. They stress the importance of inferring behavioral aspects from operational data and point out that a large sample size does not make up for driver routing behavior that is not accurately described.

[53] focuses on adopting ML for dynamic time slot management in online grocery delivery. Based on current orders, the objective is to make real-time decisions about whether it can serve new customers during specific delivery timeslots. The associated *Vehicle Routing Problem with Time Windows* is challenging to solve using conventional methods because of time constraints and incomplete data. To assist in these decision-making processes, the researchers investigate ML techniques. Their experiments show that ML shows promise in determining the viability of customer insertions and outperforms conventional insertion heuristics in complicated routing problems. It introduces a novel planning environment for time slot management, where quick operational decisions must be made continuously within a reasonable time. They approach the problem as a classification task and train cutting-edge ML algorithms to assess the viability of servicing new clients during particular time windows. The computational studies demonstrate that ML outperforms conventional methods frequently used in practice or suggested in the literature regarding accuracy [2, 33]. Adopting the ML model improves capacity utilization, allowing the online grocer to serve more clients. The outcomes also indicate that the trained model can be generalized to routing issues of larger sizes since ML's quick computation times

enable scaling.

In a more recent study, Hildebrandt and Ulmer [29] investigate the estimation of arrival times in restaurant meal delivery and its effects on customer experience and the performance of the delivery system as a whole. They suggest two approaches: a supervised learning-based offline method and an online-offline strategy that combines online simulations with an offline approximation of delivery routing policy. According to computational experiments, both strategies perform similarly to a complete, near-optimal online simulation but take much less time to compute. By assisting customers with their choices and improving the delivery system, accurate arrival time predictions improve customer satisfaction and result in faster and fresher food delivery. The authors examine the results of telling customers their expected arrival times and draw several managerial conclusions. Accurate estimates reduce customer wait times and produce fresher food, boosting customer satisfaction. The trained solution remains precise as the company expands or demand shifts, making assumptions based on changing data or customer ordering patterns. Nevertheless, customers in remote areas might have to wait longer, which could impact what they order. The quantity of orders received depends on how far a restaurant is from densely populated areas. Order volume and accurate arrival time estimation are both factors that affect a restaurant's ability to predict preparation times. The choice of features for arrival time estimation is also covered in the paper, considering the trade-off between prediction error and feature dimensionality. Features include temporal, spatial, routing, and process-related factors. Due to the integration of these features, the model can foresee bundling and delays brought on by meal preparation and delivery timing. Considering the dynamic and uncertain nature of the issue, the suggested methods show promise in accurately predicting arrival times for restaurant meal deliveries. Future research directions recommended by the authors include dynamic pricing, customer choice models, location planning, fairness considerations, joint optimization of routing and assignment policies, and applying the online-offline framework to other pickup and delivery problems.

Despite the rise of studies focusing on using ML techniques for last-mile optimization, there is still a substantial gap in the performance analysis of these models.

Numerous studies lack thorough analyses and neglect to evaluate the models' performance in real-world scenarios. Furthermore, the feature sets used in these studies are often limited and leave out crucial elements that could affect the precision and efficacy of the optimization process. In our study, we conduct extensive numerical experiments with real-world data to validate the performance of our suggested approach and conduct a detailed survey to get a detailed list of features used in existing studies to contribute to filling this critical gap in the literature.

Despite the rapidly growing number of studies on the topic, feature engineering in VRP-based ML approaches still needs to be thoroughly explored [44]. Explicit and informative feature descriptors are essential for flourishing ML approaches in VRP. The full potential of machine learning for streamlining routing procedures and raising overall performance is hampered by this gap. The subsequent studies will provide information on the feature engineering component of the research, emphasizing the creation and application of feature descriptors.

VRP is one of the complex optimization problems for which ML is crucial in presenting alternate solutions. These solutions heavily rely on collecting and applying pertinent features and essential inputs for ML algorithms. Features give problem instances a structured representation, which helps ML algorithms comprehend and gain knowledge from the data. However, *"Literature of VRP descriptors is scarce"* [44]. Effective feature extractors are algorithms or methods that can locate and extract valuable and instructive features from unprocessed data. These extractors take the data's most noticeable features essential for completing a particular task or problem. Effective feature extractors are crucial in machine learning because they help convert raw input data into a format suitable for training and creating precise predictive models. Effective feature extractors enable us to describe the characteristics of each problem instance in the context of VRPs. ML algorithms can use this description as input to discover patterns, connections, and trends in the data. Combinatorial optimization problems have a considerable potential for improvement when assisted with feature-based ML approaches [44]. Researchers have seen encouraging results by applying this strategy to VRPs.

Mersmann et al. [40] investigate what makes 2-opt-based local search algorithms effective in solving the traveling salesperson problem (TSP). Despite the widespread use of 2-opt in practical applications, its effectiveness in addressing a specific problem is challenging to ascertain. To address this, the authors take a statistical approach and examine the characteristics of TSP instances that affect how effective using 2-opt is. To understand the elements influencing the effectiveness of 2-opt-based local search algorithms for resolving TSP, the paper investigates various feature groups, including centroid, minimum spanning tree (MST), angle, convex hull, nearest neighbor distance, mode, cluster, and distance to nearest neighbor features. These features offer information regarding edge statistics, mode distribution, clustering properties, nearest neighbor distances, centroid-based distances, MST characteristics, angle relationships, and convex hull properties.

Pihera and Musliu [42] presented a study that predicts the best-performing heuristic algorithm for the TSP. The authors employ machine learning techniques, which improve prediction accuracy compared to simple algorithm preference based on performance. The researchers suggest an ML technique that defines the problem instances using different attributes. These properties serve as inputs for the ML models and capture the distinctive characteristics of each TSP instance. The researchers assess the efficacy of various feature sets and ML techniques by training ML algorithms on a dataset containing over 2000 benchmark tasks. The outcomes show that the algorithm selection process's accuracy is increased by adding innovative characteristics, such as k-nearest neighbors (kNN), graph transformation features, and tour segments' features. Additionally, it introduces Cost Matrix, and Minimum Spanning Tree Clustering-based features that we also use in our study. Additionally, the study evaluates the effectiveness of several ML algorithms and emphasizes the advantages of feature selection methods.

Identifying optimal and non-optimal solutions is another research interest inspected [4]. In order to describe both the VRP instance and the VRP solution, several metrics are defined. The metrics are classified into two groups: solution and instance properties.

- The instance metrics (I1-I8) include the following: *the number of customers,* *the number of routes, the level of capacity utilization, the average distance between each pair of customers, the standard deviation of those distances, the average distance from customers to the depot, the standard deviation of those distances, and the standard deviation of the radians of customers along with depot.*

- The solution metrics (S1-S10) include characteristics like *the average number of intersections per customer, the longest distance between two connected customers per route, the average distance between the depot and directly connected customers, the average distance between routes (their centers of gravity), average width per route, an average span in radians per route, average compactness per route measured by width, average compactness per route measured by radian, average depth per route*, and others.

Using these metrics, the researchers could distinguish between optimal and non-optimal solutions with up to 90% accuracy by creating and categorizing 192,000 VRP solutions for various instances.

Liu and Shen [35] suggest using current optimization techniques such as integrating predictions with optimization, reformulating the mathematical model, and incorporating a branch and price algorithm to consider the service time uncertainty and enhance the overall on-time performance. The study's creation of prediction models for calculating the total travel time is crucial. These models are built without specific structural assumptions like the Continuum Approximation or the TSP. As an alternative, the authors develop predictors based on characteristics related to the number of locations, the visiting area, the distance, the dispersion, and their interactive terms. The authors test six strategies, including LASSO, ridge regression, elastic net, linear Support Vector Regression (SVR) with radial basis function (RBF) kernel, and random forest, to gauge how well the prediction models perform. These models' feature set consists of tractable features for predicting travel times and extra features set aside for evaluation. The authors use fivefold cross-validation to find the best hyperparameters for the models, enabling them to choose the best values for variables like shrinkage parameters in LASSO and ridge regression and the number of trees in the random forest.

Another study [21] aims to estimate the best penalization criterion for the Knowledge-Guided Local Search (KGLS) method in solving large-scale VRP. The penalization criterion is a measure used to evaluate and penalize solutions in the search process of KGLS. The authors hypothesize that the optimal penalization criterion depends on the specific VRP instance. The authors utilize ML techniques to achieve this and propose a classification task to predict the appropriate penalty criterion class for a given VRP instance. They consider many ML methods, including Random Forests, Support Vector Machines (SVM), and Genetic Programming. The authors extract features from the VRP instances to train the classification models. *The number of customers, the number of routes, the ratio of total demand to total capacity, the average distance between pairs of customers, the standard deviation of the average distance between customers, the average distance from all customers to the central depot, the standard deviation of the distance between all customers and the central depot, the standard deviation of the angle between customers and the depot, and the ratio between total demand and total capacity available* [21] are some of these features.

Deep learning techniques have been applied in recent years as an end-to-end method to find solutions to the high development time of exact and approximation algorithms for NP-hard combinatorial optimization problems and the high running time of exact solvers. However, using deep learning techniques has drawbacks, including representation, generalization, complex architectures, and interpretability of models for mathematical analysis. In a metaheuristics framework, ML can be used as a workaround to enhance the run-time performance of exact algorithms. Interesting research related to this subject is performed by Fitzpatrick et al. [26]. The authors use a precise Integer Programming approach after a pruning heuristic that uses ML as a pre-processing step. Their approach is employed to sparsify examples of the traditional TSP. The method specifies the graph and drastically reduces the number of decision variables by learning which edges in the underlying graph are unlikely to belong to an optimal solution and removing them. By combining insights from linear programming relaxation, cutting plane exploration, minimumweight spanning tree heuristics, and other analytical methods, the study harnesses the power of these techniques to generate a vast set of features. They show that, while maintaining the majority of the optimal tour edges, the method can reliably prune a significant portion of the variables in TSP instances.

The study by Rasku et al. [44] thoroughly classifies features and their relevance to VRP instances. It provides a helpful understanding and analysis of VRP instances, assisting in creating more efficient algorithms for resolving challenging vehicle routing issues. The authors concentrate on creating feature extractors to specify instances of VRP thoroughly. VRP can take many forms, and examples from the same source can differ significantly. Heuristic, metaheuristic, and hybrid algorithms currently used to solve VRP are sensitive to this variation and may exhibit inconsistent performance in new instances. Researchers frequently look for parameters available for customization to address this problem and improve the applicability of algorithms. However, finding the best parameter values can take time and extensive testing. The authors suggest using descriptors for problem classes and instances to make it possible to apply learning and adaptive techniques that take advantage of each instance's particular qualities. The study thoroughly categorizes the feature sets comprising node distribution features, MST features, Local search (LS) probing features, Branch-and-cut probing features, geometric features, Nearest neighborhood (NN) features, and VRP-oriented features. The study further improves the understanding of the feature sets by keeping track of the computation time for each instance's feature extraction. The authors create 386 features by adapting and recommending 76 feature extractors for VRPs.

Our study, in distinction to the research above, proposes two novel classification approaches for VRP descriptors. The first categorization provides a structured assignment of properties based on the representation of the feature, whether it relates to a node, set, or cluster settings. The second classification focuses on the data's origin, including travel time, distance, or geographic data such as latitude and longitude. A comprehensive feature set that improves the model's applicability is produced as a result of meticulous literature analysis to compare the computational effectiveness features gathered from the literature and newly introduced.

Chapter 3

METHODOLOGY

This section provides a detailed overview of the problem, the modeling framework, and the solution approach. The basic notation and the formal definition of the problem are presented in Section 3.1, and Section 3.2 describes the technical details of the ML approach we develop to estimate the MFC.

3.1 Problem Definition

Let N be the set of customers to be visited for deliveries from a depot n_0. We consider a complete graph $G = (\bar{N}, A)$, where the set of nodes is composed of the list of customers and the depot, i.e., $\bar{N} = N \cup \{n_o\}$. For an arc $(i, j) \in A$, the shortest path distance (on the real road network) is indicated by d_{ij}, and the travel time is denoted by t_{ij}. We consider a homogeneous fleet of delivery vehicles and denote the set of vehicles by K. For vehicles, the per kilometer fuel costs and the depreciation (maintenance) costs are given by α and β, respectively. The hourly salary of the drivers is denoted by γ. The service time for a customer $i \in N$ includes the time for the driver to park the vehicle, walk to the customer's door, complete the transaction, and return to the vehicle, which we denote by ω_i. Similarly, the service time of the vehicle in the depot at return is given by ω_{n_0}, which includes the time required to park the vehicle and complete the necessary bookkeeping procedures, if necessary. To be ready for the following delivery cycle, the vehicles are required to complete their tours in ϕ hours and incur a lateness cost τ per hour for failing to respect this time limit.

To simplify the technical discussions in the following parts, we define $N_i = N \setminus \{i\}$, for some $i \in N$, and for an arc $(i, j) \in A$ we define the cost as $c_{ij} = (\alpha + \beta)d_{ij} + \gamma(t_{ij} + \omega_j)$.

For a set of customers $V \subseteq N$, let $z(V)$ be the value of the optimal vehicle routes that minimize the total delivery cost composed of fuel, vehicle depreciation, personnel, and lateness costs. We define the marginal fulfillment cost C_i of a customer $i \in N$ as the difference between $z(N)$ and $z(N_i)$, i.e., $C_i = z(N) - z(N_i)$.

For a given instance (data) that is denoted by the tuple $\langle N, A, K, \tau, \phi \rangle$, the Marginal Fulfillment Cost Prediction Problem (MFCP) is to predict the marginal fulfillment costs C_i for $i \in N$.

The following section describes the ML approach we develop to address MFCP. For a quick reference, Table 3.1 presents a summary of the notation we use in the thesis.

Table 3.1: Outline of the notation

Notation	Definition
t_{ij}	Travel time from i to j
d_{ij}	Distance from i to j
α	Fuel cost per kilometer
β	Maintenance cost per kilometer
γ	Hourly Carrier Salary
ω	Service Time
τ	Lateness Cost per Hour
ϕ	Tour Time
N	All orders
n_0	Depot
$\bar{N}$	Orders & Depot
K	Fleet

3.2 Machine Learning Approach

The basic steps of our ML approach consist of data labeling, feature engineering, and feature and model selection, as illustrated in Figure 3.1. In the following subsections, we present the details for each of these steps.

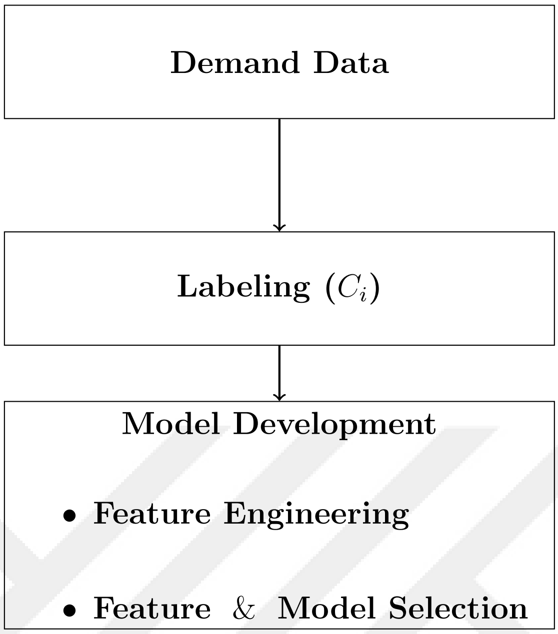


Figure 3.1: Methodology Flow Diagram

3.2.1 Data labeling

The data $\mathcal{D}$ we consider for our ML approach is a collection of MCFP instances that can be generated from historical data or a known distribution for delivery demand arrivals. To label the data, i.e., determine the marginal costs of customers $i \in N$ in an MCFP instance $\langle N, A, K, \tau, \phi \rangle$, we solve $|N| + 1$ vehicle problems to determine $z(N)$ and $z(N_i)$, for all $i \in N$. The data labeling procedure is summarized in Algorithm (1).

Algorithm 1 Data labelling

```
for each ⟨N, A, K, τ, φ⟩ in 𝒟 do
    Find z(N)
    for each customer i in N do
        Find z(N_i)
        C_i ← z(N)-z(N_i)
    end for
end for
```

To solve the vehicle routing problem for a set of customers $V \subseteq N$, i.e., to find $z(N)$ and $z(N_i)$, where N_i is $N \setminus \{i\}$ in Algorithm 1, we solve a mixed-integer program, which considers the following decision variables.

- $x_{ij}^k = \begin{cases} 1, & \text{if vehicle } k \in K \text{travels on arc } (i,j) \in A \\ 0, & \text{otherwise} \end{cases}$

- $y_i^k = \begin{cases} 1, & \text{if vehicle } k \in K \text{ visits customer } i \in V \\ 0, & \text{otherwise} \end{cases}$

- T_k : additional time vehicle k travels (lateness)

We define our mixed-integer programming formulation $\mathcal{P}$ as follows.

$$\min \quad \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k + \sum_{k \in K} T^k \tau \tag{3.1}$$

$$s.t. \quad \sum_{k \in K} y_i^k \geq 1 \qquad \forall i \in V \tag{3.2}$$

$$\sum_{j \in V} x_{0j}^k \leq 1 \qquad \forall k \in K \tag{3.3}$$

$$y_i^k \leq \sum_{(i,j) \in A} x_{ij}^k \qquad \forall i \in V, k \in K \tag{3.4}$$

$$\sum_{(i,j)\in A} x_{ij}^k - \sum_{(j,i)\in A} x_{ji}^k = 0 \qquad \forall i \in V, k \in K \tag{3.5}$$

$$\sum_{(i,j)\in A(S)} x_{ij}^k \leq |S| - 1 \qquad \forall k \in K, S \subseteq V \tag{3.6}$$

$$\sum_{j\in N} \sum_{(i,j)(j\neq 0)\in A} (\omega + t_{ij}) x_{ij}^k \leq \phi + T^k \qquad \forall k \in K \tag{3.7}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall (i,j) \in A, k \in K \tag{3.8}$$

$$y_i^k \in \{0,1\} \qquad \forall i \in V, k \in K \tag{3.9}$$

$$T^k \geq 0 \qquad \forall k \in K \tag{3.10}$$

The objective (3.1) is to minimize the total delivery cost composed of fuel, vehicle depreciation, personnel, and lateness costs. Constraints (3.2) guarantee that each customer is visited by a vehicle. Constraints (3.3) ensure that each vehicle is used only in one tour. Constraints (3.4) enforce that if a vehicle $k \in K$ visits a customer $i \in V$, the vehicle uses one of the incoming arcs of the customer i. Constraints (3.4) are the flow balance constraints for each vehicle at each customer location. Constraints (3.5) are the subtour elimination constraints we enforce for each vehicle. For each vehicle the lateness is tracked by the inequalities (3.6). Finally, variable domains are indicated in (3.8)-(3.10)

Note that, due to a high number of sub-tour elimination constraints (3.6) that grows exponentially with the number of customers $|V|$, $\mathcal{P}$ is not a compact formulation that can be directly solved for realistic problem instances. To address this issue, we employ a branch-and-cut approach where we start with a subset of subtour elimination constraints and add them to the model iteratively as needed.

Branch and cut approach to solve $\mathcal{P}$

In particular, when it comes to VRP or TSP, subtour elimination is crucial in solving optimization problems. Subtours, or subgraphs where a subset of nodes forms a cycle apart from the remainder of the tour, are what subtour elimination aims to avoid. However, there is an exponential growth in the number of restrictions needed as the number of nodes or components in the system rises. Therefore, explicitly including every potential subtour removal constraint in the optimization model becomes elusive.

Subtour elimination is a crucial step in the optimization process for the branch-and-cut algorithm. The approach iteratively checks the solution for subtours whenever a new and better solution is found using the mixed-integer linear programming (MILP) method. Constraint 3.6, *the Dantzig-Fulkerson-Johnson formulation* (DFJ), is added to the model to eliminate detected subtours. The branch-and-cut approach is characterized by this iterative procedure, which includes subtour detection, constraint incorporation, and cutting the subtour-included MILP solution.

Algorithm 2 Branch and Cut - SubtourElimination

for each $\widehat{MILP}$ encountered during $\mathcal{P}$ **do**
    Set $\hat{A} = \{(i, j) \in A : \hat{x}_{ij} > 0\}$ and $\bar{N} = \{i \in N : \hat{y}_i > 0\}$
    Find the set of connected components S from $\hat{G} = (N, \hat{A})$
    if S is not $\emptyset$ **then**
        for each customer s in S **do**
            for each vehicle k in K **do**
                Add the inequality $\sum_{(i,j)\in A(s)} x_{ij}^k \leq |S| - 1$ to $\mathcal{P}$
            end for
        end for
    end if
end for

The Subtour Elimination function in Algorithm 2 is activated when the solver encounters a new feasible solution during the branching process, denoted as $\widehat{MILP}$. Using $\bar{A}$, a set of connected components S is generated by analyzing the graph $\bar{G}$ emerged via encountered solution, and the subtours that exclude the depot (denoted by n_0) are detected. The set of nodes from each subtour s from the set S is denoted by N_s. To break the subtour that includes only the nodes of s, related DFJ constraint is implemented, forcing the number of edges in the subtour be less than the number of nodes, thus eliminating any potential cycles. Updated $\mathcal{P}$ continues the optimization process, and the algorithm repeats when a new $\widehat{MILP}$ is encountered. The optimization process ends when an optimal solution is found or the time limit is reached. By employing the modified strategy, we dynamically include subtour elimination constraints during the branching process instead of adding them all at, which is not applicable.

3.2.2 Perspectives on Feature Characteristics

In this part, we study the existing features in the literature proposed for ML based approaches to deal with various VRP related system management decisions and propose new ones that can capture important information that can be used to improve the MFC prediction.

The MFC of a customer depends not only on its individual properties but also on the properties of other customers that need to be served in the same delivery cycle. It is essential to consider both the unique characteristics of each customer location and the spatial relationships in their environment to analyze the spatial aspects of customers and extract valuable features. To gain insights into a customer's spatial positioning and relationships, capturing their properties, such as their proximity to the hub or nearest neighbor, is crucial. It is necessary to analyze the distance matrix statistics when defining the customer's set to identify patterns and variations in the spatial relationship of the set in which they exist. Furthermore, examining the neighborhood by building a subgraph using clustering techniques enables discovering spatial clusters and drawing the complete picture.

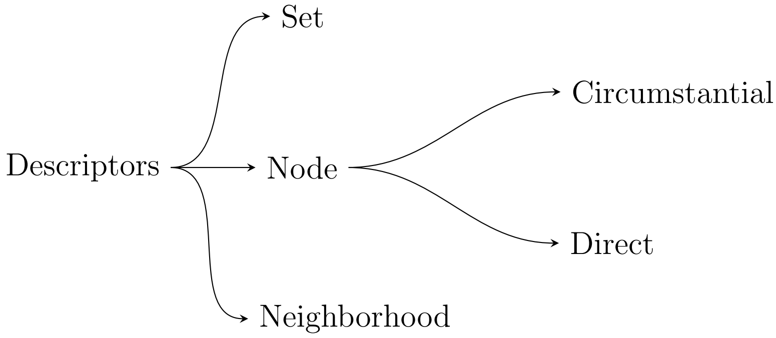


Figure 3.2: Feature Classification

Figure 3.2 presents defined feature bases for machine learning models to create a comprehensive instance picture for the customer. It is essential to consider both the unique characteristics of each customer location and the spatial relationships in their environment to analyze the spatial aspects of customers and extract valuable features. To gain insights into a customer's spatial positioning and relationships, capturing their properties, such as their proximity to the hub or nearest neighbor, is crucial. In order to identify patterns and variations in the spatial relationship of the set in which they exist, it is necessary to analyze the distance matrix statistics when defining the customer's set. Set-based features exhibit uniform characteristics across all customers within the set.

On the other hand, neighborhood features consider the interactions and dynamics between the nodes in their vicinity. Examining the neighborhood by building a subgraph using clustering techniques enables discovering spatial clusters and drawing the complete picture. Valuable insights can be gained by analyzing the pooled distribution of nodes within the same cluster or neighborhood. The average distance in the cluster, which determines the mean of the typical distance between nodes inside a particular cluster, illustrates a neighborhood feature. This function aids in the discovery of local interactions, dependencies, and patterns between neighboring nodes.

Two different methods can be used to examine node-based features further. The first method uses immediate data to make direct measurements, such as the node's distance from the hub or the average distance to all other customers. In the context of the overall dataset, these attributes offer clear insights regarding the node's spatial relationships. The second strategy is circumstantial and indirectly concentrates on the set's characteristics. These features entail examining the variations in values seen when a particular node is present to or absent in the set. These features provide a marginal and keen study of the node's impact on the set. They are referred to as *circumstantial* within the node-based feature category. They highlight the node's importance in the overall dynamics and characteristics.

This framework enables a thorough analysis of node characteristics at the individual level, node behavior at the neighborhood level, and overall set spatial distribution. Machine learning models can be improved by utilizing these insights, producing more accurate predictions and a deeper comprehension of the underlying patterns in the data. The features are divided into set-based, node-based, and cluster-based categories in tables of Appendix A, as the column named Base. Set-based traits are qualities that are the same for every set member. Direct and circumstantial node attributes of specific nodes are labeled as node-based features. Cluster-based features emphasize neighborhood characteristics obtained using clustering techniques.

To address the underlying data or tools used in the feature development process, we have provided an additional *Category* classification to the structural approach represented by the *Base* category. It provides a thorough framework for deriving valuable insights from the data beyond the structural *Base*. It helps with navigation and assistance throughout the feature analysis process. The class or category that each feature belongs to is described in the tables of Appendix A *Category* column. The various categories offer details on the data and methodology used to produce the features. The descriptions of each category are as follows:

- Literature: The papers cited in Section 2.3 of the thesis provided the features for this category. These characteristics are derived from the research results and methods detailed in the literature.

- Angle-Count-Coordinate: Latitude, longtitude, and fundamental set information are used to derive the features in this category. In order to offer insightful data, the features make use of the geometrical characteristics and spatial relationships between the customers and the depot.

- Time: The features in this category are based on actual road network trip times. They take into account the real transit times between the depot and customer locations and capture the temporal components of the issue.

- Distance: The base data for the features in this category is a distance matrix. These attributes provide information about the proximity and accessibility of the clients by quantifying the spatial separations between the depot and customer locations.

- Cluster: The K-nearest neighbors (KNN) algorithm is used to derive cluster-based features, which are then obtained. These features record the customer grouping and clustering behaviors, enabling customer cluster-based analysis and optimization.

Understanding the specific areas of the problem space that each feature concentrates on is made more accessible by classifying according to the base or category. Cluster-based features offer insights into individual properties and cluster relationships, while node-based features concentrate on specific customer attributes. Set-based features capture the overall spatial distribution. An in-depth analysis and comprehension of the underlying structure and characteristics of the problem domain are made possible by this thorough categorization. A full list of detailed feature descriptions can be seen in Appendix A.

3.2.3 Feature & Model Selection

To guarantee the reliability and robustness of selected features and eliminate the correlation between them, a 10-fold cross-validation technique is used. The final feature set consists of those features that take a non-zero value at least five folds. Afterward, the LazyPredict package fits the data into 34 machine-learning models with selected features. The ideal model for the MFC prediction is chosen through this thorough evaluation. The performance of the models is evaluated by taking into account performance metrics like R-Squared and RMSE. Gaining knowledge about the selected models' performance and the chosen features' predictive capability assist in the selection process.

Chapter 4

COMPUTATIONAL STUDIES

4.1 Data

In our numerical experiments, we use real-world data from a large grocery chain in Turkey that has extensive e-grocery operations. The data comprises 410 unique customer locations and a central depot. We have 520 MFCP instances in our data set, which includes five delivery vehicles and a delivery time slot width of one hour. In our experiments, we used cost parameters listed in Table 4.1 that were determined after consulting with the e-commerce specialists from the grocery chain that provided the data. During the labeling process, the service time for each customer is assumed to be the same. It is denoted as w below, whereas the depot's parameter, w_0 is neglected and assumed to be 0. These presumptions establish a consistent analytical framework.,

Table 4.1: Objective Function Parameter Values

Notation	Definition	Value
α	Fuel cost per kilometer	0.135 (€/km)
β	Maintenance cost per kilometer	0.11 (€/km) [9]
γ	Hourly Carrier Salary	6 (€/h)
ω	Service Time	2.5 (min)
τ	Lateness Cost per Hour	4.8 (€/h)
ϕ	Tour Time	60 (min)

We used 400 MFC instances from our data for training the ML model and spared 120 instances for testing. To label the data, we solved the routing model ($\mathcal{P}$) 16,692 times. We implemented the branch-and-cut algorithm in Python using Gurobi 10.0 to solve the respective mixed-integer programs. To complete the labeling phase in a reasonable time; for each $z(N)$ calculation process that is donated $\mathcal{P}$ where $\langle N, A, K, \tau, \phi \rangle \in \mathcal{D}$, the VRP optimization has a time limit of 9 minutes. The optimization for N ends if the difference between the linear relaxation of the problem and the best MILP solution obtained is less than 0.05 or the optimal solution is obtained. If not, the $z(N)$ calculation can last up to 72 minutes since up to 7 resume allowances are permitted. To speed up the solution procedure, we first solve the VRP instances for all the customers, which we refer to as the "original" tour, and then use this solution as a base to feed a feasible solution to the solver as a warm start to solve the problem when one of the customers are removed from the customers set, which we refer to as "removed" tour. The number of solved problem instances for original and removed tours and the respective average optimality gaps after the time limit is presented in Table 4.2. Each N_i (removed tour set) has two restart allowances and a high-quality starting solution, giving them at most 27 minutes. This stems from two motivations: first, the finite time constraints required by the actual market's limited time availability following the closing of a time slot which is reflected by the time limit, and second, the requirement to reduce solution gaps because the labeled data feeds into the machine learning model.

Table 4.2: Summary of the optimization runs for labeling

	Count		**Average Gap (%)**	
	Original	**Removed**	**Original**	**Removed**
Test data	120	3904	3.4	2.4
Train data	400	13078	3.4	2.7

4.2 Feature and model selection

In this part, we present the detailed results for feature and model selection and discuss important insights we derive from our numerical study.

4.2.1 Feature selection results

A rigorous feature selection procedure is carried out using 10-fold Lasso regression to pinpoint the features most pertinent to the VRP. This method made it possible to find characteristics that consistently greatly impacted the issue over multiple folds. The final set of features is chosen from those that obtained non-zero coefficients in at least five out of the ten folds. This method ensured that only the most significant and consistent features are kept for further analysis and model development. The study concentrated on the most instructive features using this feature selection method, which improved the precision and potency of the VRP optimization models.

Table 4.3: Features Selection Statistics

Category/Base	**Features Introduced**	**Features Selected**	**Selection Rate**
Literature	74	14	18.9%
Angle-Count-Coordinate	12	6	50.0%
Cluster	12	10	83.3%
Distance	24	5	20.8%
Time	29	7	24.1%
Set-Based	55	11	20.0%
Node-Based	84	21	25.0%
Cluster-Based	12	10	83.3%
Total	151	42	27.8%

Table 4.3 presents interesting statistics regarding the feature selection procedure, providing critical insights into the efficacy of various feature generation approaches for cluster-based feature selection. The results demonstrate that introducing new features in that category can significantly improve model performance. It is noteworthy that among all the feature categories, the computationally cheap features from the literature showed the lowest selection rate. Conversely, the highest selection number is also found for the Literature category, still indicating their relevance and influence on the VRP optimization. These results emphasize the significance of

carefully planning and choosing features that can capture the distinctive qualities and difficulties of the VRP.

Table 4.3 also shows that the count of node-based features equals the sum of cluster-based and set-based features. This emphasizes the importance of node-based features in accurately capturing the VRP problem's characteristics. Since none of the groups is eliminated after the Lasso method is applied, the analysis also showed that all input bases and categories are still required for the ML model. It shows that each input category makes a valuable contribution and helps the model to perform more accurately in prediction. The detailed inclusion of various feature types guarantees a comprehensive representation of the issue. It enables the model to utilize the varied information within these features.

Table 4.4: Selected Features

ID	Category	Note	Definition	Base
L1	Literature	*	The longest distance between the depot and customer locations	Node Based
L2	Literature		The longest distance between the depot and customer locations	Set Based
L3	Literature	*	Standard deviation of the angle between the customers and the depot (in radians)	Node Based
L4	Literature		Number of customers in the set deducted by one	Set Based
L5	Literature		Number of customers	Set Based
L6	Literature	*	Sum of all edge costs multiplied by 2/(N - 1)	Node Based
L7	Literature		Sum of all edge costs multiplied by 2/(N - 1)	Set Based
L8	Literature		The proportion of edges with distances shorter than the mean distance	Set Based
L9	Literature	*	The proportion of edges with distances shorter than the mean distance	Node Based
L10	Literature	*	Edge lengths of the convex hull - Mean	Node Based
L11	Literature		Edge lengths of the convex hull - Mean	Set Based
L12	Literature		Edge lengths of the convex hull - Min	Set Based
L13	Literature	*	Edge lengths of the convex hull - Max	Node Based
L14	Literature	*	Edge lengths of the convex hull - Median	Node Based
ACC1	Angle-Count-Coordinate		The angle made with the Depot	Node Based
ACC2	Angle-Count-Coordinate		Average angle in the customer set	Set Based
ACC3	Angle-Count-Coordinate		Absolute difference of customer angle and average angle of customer set	Node Based
ACC4	Angle-Count-Coordinate		Percentage of order number in +5, -5 degrees neighborhood	Node Based
ACC5	Angle-Count-Coordinate		The shortest distance of specified node to the line drawn between two pairs of other nodes in the set	Node Based
ACC6	Angle-Count-Coordinate	*	The area of the convex hull of the instance	Node Based
CL1	Cluster		Average angle of the assigned cluster to the depot	Cluster Based
CL2	Cluster		Percentage of orders that are assigned to the specified node's cluster	Cluster Based
CL3	Cluster		Absolute difference between average angle of the cluster and specified node angle	Cluster Based
CL4	Cluster		Angle standard deviation of the cluster	Cluster Based
CL5	Cluster		Maximum distance between two nodes inside the cluster	Cluster Based
CL6	Cluster	**	Average angle of the second cluster	Cluster Based
CL7	Cluster	**	Average angle of the assigned cluster to the depot	Cluster Based
CL8	Cluster	**	Absolute difference between average angle of the cluster and specified node angle	Cluster Based
CL9	Cluster	**	Angle standard deviation of the cluster	Cluster Based
CL10	Cluster	**	Maximum distance between two nodes inside the cluster	Cluster Based
D1	Distance		Total number of customers in this radius when the center is the specified customer (Radius: Hub centered distance covering half of the orders divided by three)	Node Based
D2	Distance		Harmonic Mean of the distances / Count of customers in the set	Node Based
D3	Distance	***	All nodes vote for each of its in-neighbors, and the node with the highest votes is elected iteratively. The voting ability of out-neighbors of elected nodes is decreased in subsequent turns	Node Based
D4	Distance	****	The sum of the reciprocal of the shortest path distances from all other nodes to the specified node	Node Based
D5	Distance	****	Number of nodes required to reach from depot to the specified node in a shortest path	Node Based
T1	Time		Maximum customer-depot distance in the customer set	Set Based
T2	Time		10% of the distance (0.1 quantile)	Set Based
T3	Time		90% of the distance (0.9 quantile)	Set Based
T4	Time		Harmonic Mean of the distances / Count of customers in the set	Node Based
T5	Time		The average distance of a node to all the other nodes	Node Based
T6	Time	***	All nodes vote for each of its in-neighbors, and the node with the highest votes is elected iteratively. The voting ability of out-neighbors of elected nodes is decreased in subsequent turns	Node Based
T7	Time	****	Number of nodes required to reach from depot to the specified node in a shortest path	Node Based

After the feature selection procedure, we determined 42 features in total to use in the ML model, which are listed in Table 4.4. When a particular node is excluded from the set, some node-based features show how sensitively the set-based features

[1]* The gap between the original set's feature value and the removed set's feature value

[2]** Cluster number is increased by one

[3]*** Base graph is minimum spanning tree

[4]**** Base graph is maximum spanning tree

react. The results consider the discrepancy between the feature value of the initial set and the feature value upon removal by including the provided footnotes (*). These features reflect how the characteristics of the set as a whole are affected by removing a node.

The study also investigated the effectiveness of various spanning tree base graphs for feature selection. According to the final features, the maximum spanning tree features might perform better, even though the literature has mainly focused on using the minimum spanning tree as the base graph. This rather intriguing result emphasizes the potential benefits of taking maximum spanning trees into account during feature selection procedures.

We employ a novel approach to incorporate cluster-based characteristics. First, a rough estimate of the necessary number of vehicles is determined and the estimated vehicle numbers are directly assigned to the corresponding cluster numbers. The predetermined vehicle count is then raised by one, and the cluster features are recalculated. Notably, both clusters are chosen for the final feature set, highlighting their importance in capturing the cluster-based characteristics. This novel method improves comprehension and analysis of the cluster-related and cluster-number-sensitive dataset attributes.

4.2.2 Model selection

Following the feature selection procedure, several models are trained using the Python LazyPredict package [8] and their performances are assessed. The results of LazyPredict are displayed in Table 4.5, allowing a detailed comparison of the predictive capabilities of several widely used ML models.

Table 4.5: LazyPredict Model Results

Model	Adjusted R-Squared	R-Squared	RMSE
ExtraTreesRegressor	**0.92**	**0.92**	**0.33**
HistGradientBoostingRegressor	0.91	0.91	0.36
XGBRegressor	0.91	0.91	0.36
GradientBoostingRegressor	0.9	0.9	0.37
BaggingRegressor	0.9	0.9	0.38
LGBMRegressor	0.9	0.9	0.37
RandomForestRegressor	0.9	0.9	0.38
MLPRegressor	0.89	0.89	0.4
LassoLarsIC	0.88	0.88	0.41
LinearRegression	0.88	0.88	0.41
BayesianRidge	0.88	0.88	0.41
ElasticNetCV	0.88	0.88	0.41
LassoCV	0.88	0.88	0.41
LassoLarsCV	0.88	0.88	0.41
LarsCV	0.88	0.88	0.41
KNeighborsRegressor	0.88	0.88	0.41
TransformedTargetRegressor	0.88	0.88	0.41
RidgeCV	0.88	0.88	0.41
Ridge	0.88	0.88	0.41
SGDRegressor	0.88	0.88	0.41
HuberRegressor	0.87	0.87	0.43
OrthogonalMatchingPursuitCV	0.87	0.87	0.43
LinearSVR	0.87	0.87	0.44
PassiveAggressiveRegressor	0.86	0.86	0.45
OrthogonalMatchingPursuit	0.85	0.85	0.46
DecisionTreeRegressor	0.84	0.85	0.47
TweedieRegressor	0.79	0.79	0.55
ExtraTreeRegressor	0.78	0.79	0.55
GammaRegressor	0.68	0.69	0.67
AdaBoostRegressor	0.67	0.67	0.69
KernelRidge	0.59	0.59	0.76
ElasticNet	0.54	0.54	0.81
SVR	0.54	0.54	0.81
NuSVR	0.53	0.54	0.81

Extremely Randomized Trees (ERT) is a type of ensemble learning used in the ExtraTreesRegressor model, which excelled in this study. The ERT, or ExtraTrees, creates multiple decision trees using random split feature selection. The ExtraTreesRegressor model outperformed the other 33 candidate ML models in the evaluation carried out using Lazypredict [8]. Its adjusted R-squared value of 0.92 and its R-squared value of 0.92 demonstrated the model's strong explanatory and predictive abilities. Notably, the RMSE, the baseline comparison for estimation performance, showed that the ExtraTreesRegressor model outperformed (see Table 4.5) all other candidate models.

The relative contribution of the generated features to the model's predictive power (feature importance) is examined using Shapley values [29]. Shapley values are a method for allocating rewards based on how much each party contributed to the final payout. Although the concept originates in coalitional game theory, we follow Hildebrandt and Ulmer [29] to use the same concept in understanding the feature importances. In our context, the parties stand for feature bases or categories. The total payout denotes the RMSE compared to the average of the train data output, which is a single value fit to all the data. Using the Shapley values, one can evaluate how each feature affects the model's predictive ability. To show the relative importance of the Shapley values, they are standardized. Set-based analysis and category-based analysis are both used. Given the permutational nature of Shapley values, only the literature and other aspects are compared in the category-based analysis. The findings provide insight into the individual contributions of the feature groups and their relative importance in increasing the prediction performance of the model, which are indicated by the normalized exact Shapley values.

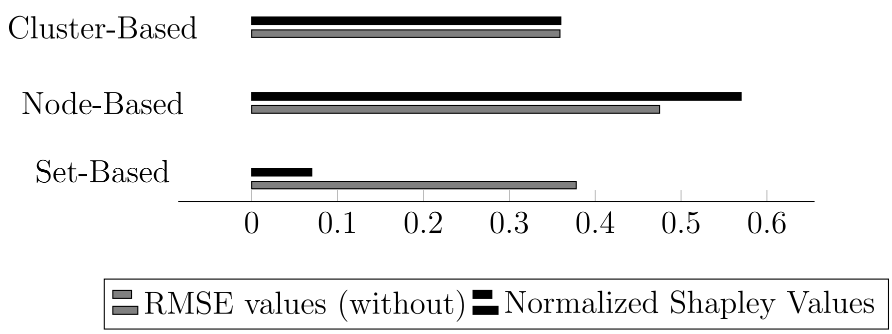


Figure 4.1: Feature-Base Performance Analysis

As we see in Figure 4.1, according to the Shapley values, node-based features have the highest contribution to the model's predictive power. Furthermore, it indicates that RMSE approaches 0.5, excluding node-based features. This result should not be surprising, given that the MFC analysis focuses primarily on node-oriented characteristics. The prominence of node-based features underlines their essential contribution to enhancing the model's predictive ability and relevance in capturing key features. The vitality of the cluster-based approach is yet another critical finding of this study. The importance given to the customer's neighborhood as opposed to the overall spatial distribution of the set, as shown by the Shapley values in Figure 4.1, presents a critical insight. The analysis shows that understanding the customer's immediate surroundings is more crucial in explaining their MFC than considering the set in which customers are included. This interpretation provides insightful information for decision-making, enabling businesses and organizations to create more specialized plans that address customers' unique preferences and requirements in their vicinity. This result highlights the importance of considering neighborhood dynamics when examining consumer behavior in spatial contexts.

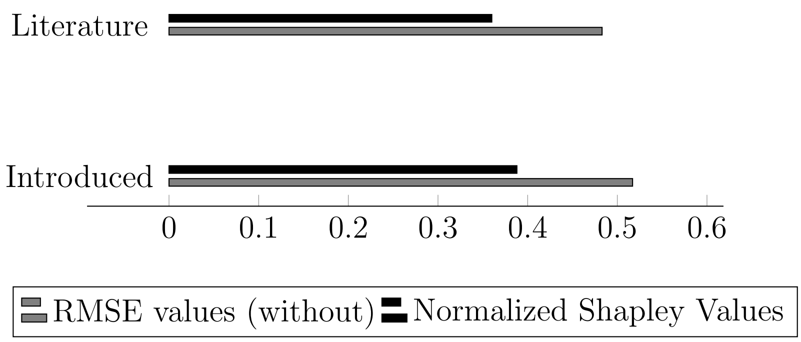


Figure 4.2: Feature Performance

The effectiveness of the new features we introduce in this study compared to those used in ML-based approaches in the literature [4, 21, 26, 35, 42, 44] is clarified by a visual representation in Figure 4.2. The benchmark category in this analysis consists of computationally affordable features used in earlier research. From looking at the Shapley values given to each feature, it is clear that the new features proposed in this thesis significantly contribute to the model's predictive power.

4.3 Benchmarks approaches for MFC prediction

In this part, we present the benchmarks we use to better assess the performance of the suggested ML model.

4.3.1 Benchmark - 1: Grid Approach

A service region can be divided into smaller sub-regions or cells using the grid approach. Each cell in the region corresponds to a particular area, enabling an approximation for MFC. By assigning an MFC value to each cell, the method offers insights into the anticipated MFC for orders from that specific region, considering the historical data or previously encountered instances. In that way, each slot's customer distribution will be indirectly considered. Figure 4.3 shows a grid approach to dividing a region into equal subregions. To ensure systematic coverage of the entire service area, each sub-region is represented by a grid cell.

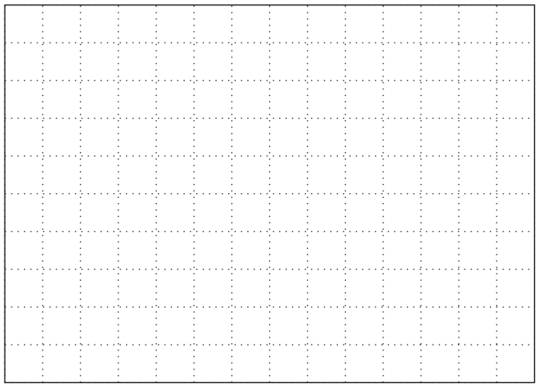


Figure 4.3: Grid Approach Representation

To estimate MFC under ideal circumstances, the grid benchmark emphasizes the significance of segmenting the service area into subregions. To accurately estimate MFC values, we want to test the pertinence of merely region-based division and evaluate how well it can account for variations in spatial costs. By analyzing the results and the data, we aim to demonstrate whether this approach provides a meaningful and reliable approximation of MFC.

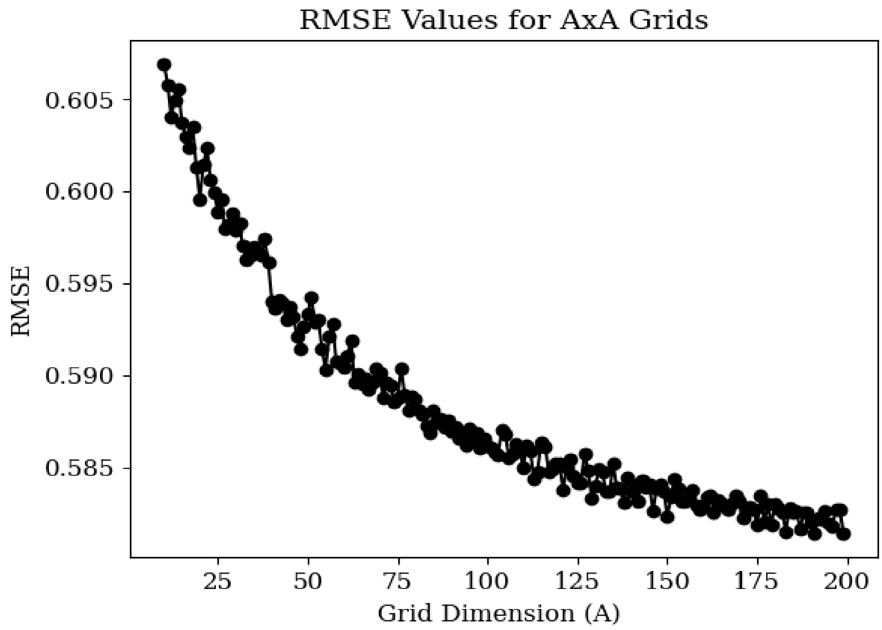


Figure 4.4: Grid - Based Approach

Despite its benefits in identifying areas with higher or lower MFC values, the grid approach has drawbacks. The connection between a customer's location within a cell and their MFC value needs to be more concise for a more accurate prediction. MFC value is influenced by complex dynamics involving numerous factors, such as the spatial distribution of the full set rather than being solely determined by the portion of the service region a customer belongs to.

Comparing the effectiveness of the machine-learning model to the grid-based method, the outcomes show how effective the ML model is at estimating MFCs. Figure 4.4 shows a considerable difference in performance between our machine-learning model and the grid-based technique for estimating MFCs. Despite dividing the area into 40,000 pieces, the grid technique regularly produces an RMSE above 0.5, although it is within sample error. In contrast, the ML model obtains a significantly lower RMSE of 0.332 on the test data. This demonstrates the model's excellent predictive ability, successfully capturing the complex patterns and variables affecting MFCs in e-grocery home delivery.

The grid-based method of calculating the MFCs for delivery merely considers the location of the order. However, this strategy ignores essential elements that significantly impact the MFCs. Features that specify the spatial distribution of customer set properties and the time of day might cause significant variations in MFC, even for the same customer. As a result, relying solely on the grid-based method may result in biased and incorrect estimates. In contrast to the grid-based approach, the ML model considers several factors, such as the spatial distribution of customer sets and network-based analysis. Our ML model provides a more thorough and accurate estimation of MFCs by considering these extra factors.

4.3.2 Benchmark - 2: Forward Value Function Approximation

Forward Value Function Approximation (FVFA) entails training a rough value function using a stochastic gradient descent algorithm. This method aims to capture the dynamics and underlying patterns of the value function, facilitating insightful analysis and decision-making in the given situation. Although there are pronounced differences in the functional form of the approximative value function and the discretization method, this approach is similar to that of Yang and Strauss [56]. In addition, as discussed in Vinsensius et al. [54], the modified version applies discretization to insert incentivizing and slot choice management. The primary goal of this study is to calculate the MFC of combinations using the dimensions of the grid and slots. This method incorporates the slot dimension as a distinct factor, which sets it apart from conventional grid-based techniques, essentially adding a temporal dimension to it. The approximation can thus capture the distributional differences related to different slots. The available training data is used to train the model using optimization results and MFCs. It is crucial to remember that FVFA is a benchmark for performance analysis investigation, with our ML model conforming as the foundation. It can be used to compare the performance of our model to this standard by estimating the MFCs of the grid and slot-based combinations. This benchmark study helps us assess the efficacy of the ML strategy. It measures how accurately our machine-learning model estimates MFCs.

Aside from the abovementioned method, it is vital to recognize that FVFA could perform worse than the ML model. This is because the ML model stresses precision in its forecasts. In contrast, incentive-oriented FVFA [54] mainly depends on relative MFC estimates of the consumers. The efficiency of FVFA may therefore be constrained in collecting fine-grained details and subtle fluctuations. However, it offers valuable insights into assessing the MFC of the grid and slot-based combinations. The ML model, on the other hand, uses sophisticated algorithms and data-driven methods to improve estimating process performance overall and enhance accuracy.

Although Visensius et. al's [54] study showed a notable increase in profit when using FVFA, it mainly concentrated on estimating relative costs rather than emphasizing precision. Table 4.6 offers a thorough overview for assessing the performance based on slot divisions. Except for slot 8, the FVFA approach's performance cannot outperform the ML model's performance. The ML model outperformed FVFA regarding precision and fine-grained predictions because it could capture subtle variations in MFCs based on a more flexible approach.

Table 4.6: FVFA and ERT Performances

Slots	Count	FVFA RMSE	ERT RMSE
1	437	2.072	0.171
2	489	0.445	0.172
3	497	2.289	0.239
4	352	0.624	0.207
5	351	0.558	0.213
6	600	1.298	0.398
7	598	0.410	0.175
8	580	0.437	0.630
Overall	3904	1.245	0.332

These results suggest that even though FVFA may have succeeded in raising overall profit [54], it may bring trouble matching the accuracy and performance attained by the ML model. The ML model's strategy, which uses cutting-edge ML techniques and incorporates a comprehensive understanding of the data, is more flourishing in capturing the complexities and nuances of the current issue.

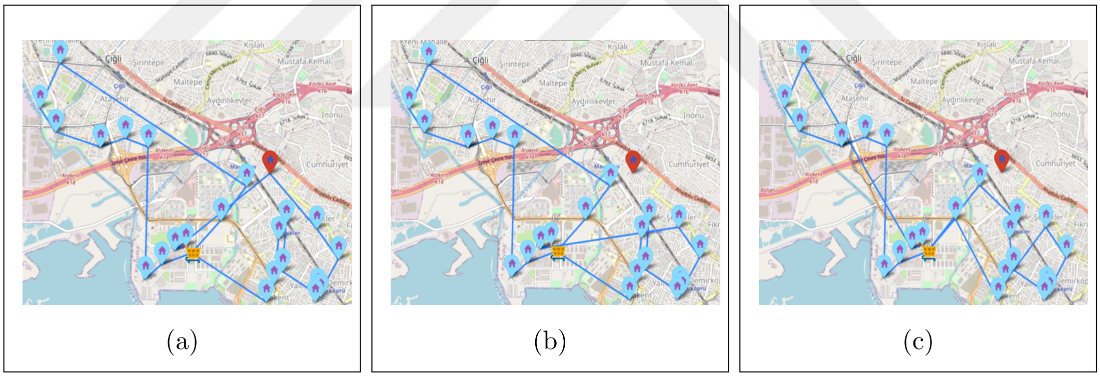
4.3.3 Benchmark - 3: Tour Patching Heuristic

Tour Patching Heuristic (TPH) can be introduced as an optimization method used in VRPs to repair current tours after bypassing particular nodes. An initial tour in a VRP is frequently made based on a group of customers and vehicles. However, there might be circumstances where it becomes necessary to remove a visitor from the tour for various causes, like cancellations, delays, or shifts in demand.

In order to find the best solution for the updated customer set, the tour must be re-optimized because each customer removed from the tour reduces the number of remaining customers by one. This would require carrying out n optimizations, each of which would take the removal of a different customer into account. For the

VRP with n-1 customers, TPH present a workable alternative solution by directly connecting the prior and subsequent nodes of the removed customer with an edge. This approach may result in suboptimal solutions because removing a customer can significantly alter the optimal routes. It is crucial to remember that TPH does not ensure the discovery of the global optimum but rather deliver satisfactory results in a reasonable amount of time.

Figure 4.5 provides a good illustration of the utility and constraints of TPH. Each customer is included in the initial ideal tour shown in (a). The TPH must be used to repair the tour when a specific customer is eliminated. The resulting solution, depicted in (b), creates an edge connecting the removed customer's preceding and subsequent nodes. It's critical to remember that the repaired solution in (b) is only sometimes the best option. The removal of a customer can cause significant adjustments in the initial optimal route.



(a) (b) (c)

Figure 4.5: An Instance

(a) Original tour, (b) Removal Applied, (c) Re-Optimized

The remaining clients in (b) can be re-optimized, as shown in (c), to enhance the solution further. The example in Figure 4.5 highlights the trade-off between efficiency and optimality in TPH. While these heuristics offer quick fixes by eliminating particular nodes, they do not ensure the overall optimum. Making precise projections of the MFC value based on TPH is only sometimes valid, which may bring risk and profit loss when making managemental decisions.

The TPH is one of the most challenging standards to meet in this study. It presents a significant challenge to the ML model because it only uses one approximation that is based on a slightly altered version of previously optimized models. The TPH presents a reliable tool to estimate the costs of removed tours and account for the actual initial tour cost when calculating the MFC. Removing a node from the tour offers the chance to investigate additional and perhaps better VRP routes. This permits potential improvements but implies that the TPH may underestimate the real performance gap but never overestimate it.

These concerns illustrate how challenging it is to outperform the TPH in this situation. The ML model faces a considerable challenge due to the inclusion of accurate tour prices and a powerful heuristic approach to estimate eliminated tour costs. The TPH is a challenging benchmark to surpass since it takes advantage of the inherent properties of the problem and the opportunity for route optimization.

Table 4.7: Prediction Performance Compared to TPH

Interval	Instance Count	Extra-Trees (RMSE)	TPH (RMSE)
[0,0.5)	2369	0.163	**0.060**
[0.5,1)	1168	**0.224**	0.227
[1,3)	23	**0.901**	1.110
[3,5)	17	**1.475**	1.756
[5,∞)	29	**0.604**	1.215
Overall	3904	**0.332**	0.379

However, the ML model surpassed the benchmark the TPH set. The ML model outperformed the TPH in all intervals except for orders that promise incredibly low MFCs, as observed in table 4.7. It highlights the ML model's ability to produce better results in MFCs and shows how well it performs across various scenarios.

Another output that further affirms the ML model's superiority is the overall RMSE. The RMSE for the ML model is 0.332, while the RMSE for the TPH is 0.379. It shows that the ML model has improved accuracy and precision because its predictions are closer to the actual values. The ML model demonstrates its efficiency in addressing the difficulties of last-mile logistics management by surpassing the heuristic removal benchmark. It is an invaluable tool for logistics companies looking to optimize their operations and provide superior customer experiences because of its capacity to optimize the last mile and increase efficiency.

Chapter 5

CONCLUSION

In this study, we looked at how ML may be used in last-mile logistics and evaluated its effectiveness using several benchmarks. It has been determined through thorough analysis and evaluation that ML demonstrates superior agility and adaptability, making it a promising solution for handling various characteristics, mainly when dealing with geographic variations. The fact that ML has consistently outperformed all the benchmarks used in this study also shows its merits.

The ability of ML to manage the complexity and dynamic nature of the delivery process is one of the technology's main advantages in last-mile logistics. Numerous factors, including customer preferences, current traffic conditions, and delivery window considerations, can be considered by ML algorithms as they analyze and process enormous amounts of data. Due to their inherent flexibility, ML models can update and improve performance while adjusting to each geographical area's particulars. Last-mile logistics providers can increase operational efficiency, boost customer satisfaction, and realize cost savings by utilizing ML's capacity to dynamically adjust and respond to changing circumstances.

The superior performance of ML over conventional methods and benchmarks can be attributed to its ability to recognize and take advantage of complex patterns and correlations in the data. In order to make predictions and decisions that are more accurate, ML models are excellent at spotting minor correlations between different variables. Advanced algorithms and methodologies allow ML models to discover essential insights that are previously concealed or disregarded, improving slot management's efficiency and efficacy. This aptitude is beneficial when dealing with the difficulties given by last-mile delivery climate, where variables like road congestion, a wide range of consumer preferences, and time limits significantly impact the success of delivery operations.

The thorough feature generation and literature review carried out in this study highlighted the significance of challenging conventional feature generation methods in VRP. Our study reveals important insights into the complexity of VRPs and the importance of incorporating various features in the optimization process by looking at a wide range of factors and variables that affect last-mile logistics. The results imply that better performance and better solutions in VRPs can result from a more comprehensive and data-driven approach to feature generation. This emphasizes the need for additional study and investigation into novel feature-generation approaches that can adequately capture the nuances of the last-mile delivery process.

Future research on managerially applied ML in last-mile logistics is quite promising. Simulation studies can offer a valuable method for assessing how ML-based slot management affects profit growth. Testing various ML algorithms, optimization schemes, and decision-making techniques is possible by generating virtual scenarios that mimic real-world circumstances. Researchers and practitioners can methodically assess the possible advantages and difficulties of using ML to slot management through simulation-based trials. This method thoroughly evaluates the influence on critical performance parameters, including delivery times, resource usage, and overall profitability. These research findings can help decision-makers apply ML solutions that are catered to their particular operational needs, resulting in more intelligent and successful slot management tactics.

BIBLIOGRAPHY

[1] Niels Agatz, Ann Campbell, Moritz Fleischmann, and Martin Savelsbergh. Time slot management in attended home delivery. *Transportation Science*, 45(3):435–449, 2011.

[2] Niels Agatz, Yingjie Fan, and Daan Stam. The impact of green labels on time slot choice and operational sustainability. *Production and Operations Management*, 30(7):2285–2303, 2021.

[3] Mohammad Affan Akhtar and Md Rashid Farooqi. Online consumer's purchasing behavior towards e-grocery shopping-a critical review. 2022.

[4] Florian Arnold and Kenneth Sörensen. What makes a vrp solution good? the generation of problem-specific knowledge for heuristics. *Computers & Operations Research*, 106:280–288, 2019.

[5] Alp M Arslan, Niels Agatz, Leo Kroon, and Rob Zuidwijk. Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers. *Transportation Science*, 53(1):222–235, 2019.

[6] Haris Aziz, Casey Cahan, Charles Gretton, Philip Kilby, Nicholas Mattei, and Toby Walsh. A study of proxies for shapley allocations of transport costs. *Journal of Artificial Intelligence Research*, 56:573–611, 2016.

[7] Ruibin Bai, Xinan Chen, Zhi-Long Chen, Tianxiang Cui, Shuhui Gong, Wentao He, Xiaoping Jiang, Huan Jin, Jiahuan Jin, Graham Kendall, et al. Analytics and machine learning in vehicle routing research. *International Journal of Production Research*, 61(1):4–30, 2023.

[8] Germán Omar Barrionuevo, Sergio Ríos, Stewart W Williams, and Jorge Andrés Ramos-Grez. Comparative evaluation of machine learning regressors for the layer geometry prediction in wire arc additive manufacturing. In

2021 IEEE 12th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT), pages 186–190. IEEE, 2021.

[9] Hussein Basma, Felipe Rodríguez, Julia Hildermeier, and Andreas Jahn. Electrifying last-mile delivery: A total cost of ownership comparison of battery-electric and diesel trucks in europe. 2022.

[10] Elena Belavina, Karan Girotra, and Ashish Kabra. Online grocery retail: Revenue models and environmental impact. *Management Science*, 63(6):1781–1799, 2017.

[11] Ilhem Boussaïd, Julien Lepagnot, and Patrick Siarry. A survey on optimization metaheuristics. *Information sciences*, 237:82–117, 2013.

[12] Kenneth K Boyer, Andrea M Prud'homme, and Wenming Chung. The last mile challenge: evaluating the effects of customer density and delivery window patterns. *Journal of business logistics*, 30(1):185–201, 2009.

[13] Bruno P Bruck, Jean-François Cordeau, and Manuel Iori. A practical time slot management and routing problem for attended home services. *Omega*, 81:208–219, 2018.

[14] Heleen Buldeo Rai, Sara Verlinde, and Cathy Macharis. The "next day, free delivery" myth unravelled: Possibilities for sustainable last mile transport in an omnichannel environment. *International Journal of Retail & Distribution Management*, 47(1):39–54, 2019.

[15] Heleen Buldeo Rai, Sara Verlinde, Jan Merckx, and Cathy Macharis. Crowd logistics: an opportunity for more sustainable urban freight transport? *European Transport Research Review*, 9(3):1–13, 2017.

[16] Ann Melissa Campbell and Martin Savelsbergh. Incentive schemes for attended home delivery services. *Transportation science*, 40(3):327–341, 2006.

[17] Valentina Carbone, Aurélien Rouquet, and Christine Roussat. The rise of crowd logistics: a new way to co-create logistics value. *Journal of Business Logistics*, 38(4):238–252, 2017.

[18] Cheng Chen, Emrah Demir, Yuan Huang, and Rongzu Qiu. The adoption of self-driving delivery robots in last mile logistics. *Transportation research part E: logistics and transportation review*, 146:102214, 2021.

[19] Francesco Ciardiello, Andrea Genovese, Shucheng Luo, and Antonino Sgalambro. A game-theoretic multi-stakeholder model for cost allocation in urban consolidation centres. *Annals of Operations Research*, pages 1–24, 2021.

[20] Catherine Cleophas and Jan Fabian Ehmke. When are deliveries profitable? considering order value and transport capacity in demand fulfillment for last-mile deliveries in metropolitan areas. *Business & Information Systems Engineering*, 6:153–163, 2014.

[21] Joao Guilherme Cavalcanti Costa, Yi Mei, and Menzjie Zhang. Learning penalisation criterion of guided local search for large scale vehicle routing problem. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2021.

[22] Iman Dayarian and Martin Savelsbergh. Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. *Production and Operations Management*, 29(9):2153–2174, 2020.

[23] Statista Research Department. Statistica, 2021. Date accessed: 15.09.2021.

[24] Statista Research Department. Statistica, 2022. Date accessed: 07.06.2023.

[25] Raafat Elshaer and Hadeer Awad. A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers & Industrial Engineering*, 140:106242, 2020.

[26] James Fitzpatrick, Deepak Ajwani, and Paula Carroll. Learning to sparsify travelling salesman problem instances. In *Integration of Constraint Program-*

ming, Artificial Intelligence, and Operations Research: 18th International Conference, CPAIOR 2021, Vienna, Austria, July 5–8, 2021, Proceedings 18, pages 410–426. Springer, 2021.

[27] Hadi Ghaderi, Lele Zhang, Pei-Wei Tsai, and Jihoon Woo. Crowdsourced last-mile delivery with parcel lockers. *International Journal of Production Economics*, 251:108549, 2022.

[28] Mario Guajardo and Mikael Rönnqvist. A review on cost allocation methods in collaborative transportation. *International transactions in operational research*, 23(3):371–392, 2016.

[29] Florentin D Hildebrandt and Marlin W Ulmer. Supervised learning for arrival time estimations in restaurant meal delivery. *Transportation Science*, 56(4):1058–1084, 2022.

[30] Jeff Howe et al. The rise of crowdsourcing. *Wired magazine*, 14(6):176–183, 2006.

[31] Robert Klein, Jochen Mackert, Michael Neugebauer, and Claudius Steinhardt. A model-based approximation of opportunity cost for dynamic pricing in attended home delivery. *OR Spectrum*, 40:969–996, 2018.

[32] Robert Klein, Michael Neugebauer, Dimitri Ratkovitch, and Claudius Steinhardt. Differentiated time slot pricing under routing considerations in attended home delivery. *Transportation Science*, 53(1):236–255, 2019.

[33] Charlotte Köhler, Jan Fabian Ehmke, and Ann Melissa Campbell. Flexible time window management for attended home deliveries. *Omega*, 91:102023, 2020.

[34] Chaya Levinger, Noam Hazon, and Amos Azaria. Efficient computation and estimation of the shapley value for traveling salesman games. *IEEE Access*, 9:129119–129129, 2021.

[35] Sheng Liu, Long He, and Zuo-Jun Max Shen. On-time last-mile delivery: Order assignment with travel-time predictors. *Management Science*, 67(7):4095–4119, 2021.

[36] Bart L MacCarthy, Lina Zhang, and Luc Muyldermans. Best performance frontiers for buy-online-pickup-in-store order fulfilment. *International Journal of Production Economics*, 211:251–264, 2019.

[37] Jochen Mackert. Choice-based dynamic time slot management in attended home delivery. *Computers & Industrial Engineering*, 129:333–345, 2019.

[38] Jeffrey Marshall. Extending the supply chain: How cutting-edge companies bridge the critical last mile into customers' homes. *Financial Executive*, 20(7):15–16, 2004.

[39] Aykhan Masimli. Shapley value for shortest path routing in dynamic networks. 2023.

[40] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Markus Wagner, Jakob Bossek, and Frank Neumann. A novel feature-based approach to characterize algorithm performance for the traveling salesperson problem. *Annals of Mathematics and Artificial Intelligence*, 69:151–182, 2013.

[41] Maida Napolitano. Omni-channel distribution: moving at the speed of "now. *Logistics management (Highlands Ranch, Colo.: 2002)*, 52(6), 2013.

[42] Josef Pihera and Nysret Musliu. Application of machine learning to algorithm selection for tsp. In *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, pages 47–54. IEEE, 2014.

[43] Wei Qi, Lefei Li, Sheng Liu, and Zuo-Jun Max Shen. Shared mobility for last-mile delivery: Design, operational prescriptions, and environmental impact. *Manufacturing & Service Operations Management*, 20(4):737–751, 2018.

[44] Jussi Rasku, Tommi Kärkkäinen, and Nysret Musliu. Feature extractors for describing vehicle routing problem instances. In *OASICS*. Dagstuhl Publishing, 2016.

[45] Jean-François Rougès and Benoit Montreuil. Crowdsourcing delivery: New interconnected business models to reinvent delivery. In *1st international physical internet conference*, volume 1, pages 1–19. Québec City (Canada) IPIC, 2014.

[46] Afonso Sampaio, Martin Savelsbergh, Lucas Veelenturf, and Tom Van Woensel. Crowd-based city logistics. In *Sustainable transportation and smart logistics*, pages 381–400. Elsevier, 2019.

[47] Arianna Seghezzi, Riccardo Mangiaracina, Angela Tumino, and Alessandro Perego. 'pony express' crowdsourcing logistics for last-mile delivery in b2c e-commerce: an economic analysis. *International Journal of Logistics Research and Applications*, 24(5):456–472, 2021.

[48] Michele D Simoni, Edoardo Marcucci, Valerio Gatta, and Christian G Claudel. Potential last-mile impacts of crowdshipping services: A simulation-based evaluation. *Transportation*, 47(4):1933–1954, 2020.

[49] Sube Singh, Ramesh Kumar, Rohit Panchal, and Manoj Kumar Tiwari. Impact of covid-19 on logistics systems and disruptions in food supply chain. *International journal of production research*, 59(7):1993–2008, 2021.

[50] Arne Strauss, Nalan Gülpınar, and Yijun Zheng. Dynamic pricing of flexible time slots for attended home delivery. *European Journal of Operational Research*, 294(3):1022–1041, 2021.

[51] Lei Sun, Atul Rangarajan, Mark H Karwan, and Jose M Pinto. Transportation cost allocation on a fixed route. *Computers & Industrial Engineering*, 83:61–73, 2015.

[52] Oliver C Turner, Brian Knight, Aleksandra Zuraw, Geert Litjens, and Daniel G Rudmann. Mini review: the last mile—opportunities and challenges for machine

learning in digital toxicologic pathology. *Toxicologic pathology*, 49(4):714–719, 2021.

[53] Liana van der Hagen, Niels Agatz, Remy Spliet, Thomas R Visser, and Leendert Kok. Machine learning–based feasibility checks for dynamic time slot management. *Transportation Science*, 2022.

[54] Albert Vinsensius, Yuan Wang, Ek Peng Chew, and Loo Hay Lee. Dynamic incentive mechanism for delivery slot management in e-commerce attended home delivery. *Transportation Science*, 54(3):567–587, 2020.

[55] Fei Yang, Ying Dai, and Zu-Jun Ma. A cooperative rich vehicle routing problem in the last-mile logistics industry in rural areas. *Transportation Research Part E: Logistics and Transportation Review*, 141:102024, 2020.

[56] Xinan Yang and Arne K Strauss. An approximate dynamic programming approach to attended home delivery management. *European Journal of Operational Research*, 263(3):935–945, 2017.

[57] Xinan Yang, Arne K Strauss, Christine SM Currie, and Richard Eglese. Choice-based demand management and vehicle routing in e-fulfillment. *Transportation science*, 50(2):473–488, 2016.

[58] Barış Yıldız. Express package routing problem with occasional couriers. *Transportation Research Part C: Emerging Technologies*, 123:102994, 2021.

Appendix A

FEATURE LISTS

Table A.1: Selected Features from Literature

Category	Definition	Base
Literature	The average distance between the depot and customer locations	Set Based
Literature	The shortest distance between the depot and customer locations	Set Based
Literature	The longest distance between the depot and customer locations	Set Based
Literature	The area of the smallest rectangle covering the customer locations	Set Based
Literature	The maximum latitudinal difference between a pair of customer locations (including the depot)	Set Based
Literature	The maximum longitudinal difference between a pair of customer locations (including the depot)	Set Based
Literature	The standard deviation of the latitudinal differences between the depot and customer locations	Set Based
Literature	The standard deviation of the longitudinal differences between the depot and customer locations	Set Based
Literature	The average latitudinal difference between a pair of customer locations	Set Based
Literature	The average longitudinal difference between a pair of customer locations	Set Based
Literature	Standard deviation of the angle between the customers and the depot (in radians)	Set Based
Literature	Number of customers in the set deducted by one	Set Based
Literature	Number of customers	Set Based
Literature	Standard deviation of the distance between all customer to the central depot	Set Based
Literature	Standard deviation of the distance between customers' pairs	Set Based
Literature	Average distance between customers' pairs	Set Based
Literature	Cost Matrix statistics - mean	Set Based
Literature	Cost Matrix statistics - skew	Set Based
Literature	Cost Matrix statistics - standard deviation	Set Based
Literature	Radius - mean distance from node to centroid	Set Based
Literature	Maximum distance of the nodes from the centroid	Set Based
Literature	Minimum distance of the nodes from the centroid	Set Based
Literature	Sum of all edge costs multiplied by 2/(N - 1)	Set Based
Literature	Median of edge values	Set Based
Literature	The proportion of edges with distances shorter than the mean distance	Set Based
Literature	Distance of nodes to hull contour - Mean	Set Based
Literature	Distance of nodes to hull contour - Min	Set Based
Literature	Distance of nodes to hull contour - Max	Set Based
Literature	Distance of nodes to hull contour - Median	Set Based
Literature	Distance of nodes to hull contour - standard deviation	Set Based
Literature	Edge lengths of the convex hull - Mean	Set Based
Literature	Edge lengths of the convex hull - Min	Set Based
Literature	Edge lengths of the convex hull - Max	Set Based
Literature	Edge lengths of the convex hull - Median	Set Based
Literature	Edge lengths of the convex hull - standard deviation	Set Based
Literature	Area of convex hull (after normalization to a unit square)	Set Based
Literature	Fraction of nodes on the hull	Set Based
Literature	The area of the convex hull of the instance	Set Based

[1]* The gap between the original set's feature value and the removed set's value is also introduced for all of the features defined in table A.1

Table A.2: Spaital Features

Category	Note	Definition	Base
Angle-Count-Coordinate		Average angle in the customer set	Set Based
Angle-Count-Coordinate		Number of cars required if each vehicle were to satisfy on average 11 orders	Set Based
Angle-Count-Coordinate		The area of the convex hull of the instance when customer is removed	Set Based
Angle-Count-Coordinate		The angle made with the Depot	Node Based
Angle-Count-Coordinate		Absolute difference of customer angle and average angle of customer set	Node Based
Angle-Count-Coordinate		Number of orders within +5, -5 degrees of the customer	Node Based
Angle-Count-Coordinate		Percentage of order number in +5, -5 degrees neighborhood	Node Based
Angle-Count-Coordinate		”Haversine” distance to the average lat-long information (Customer center of mass)	Node Based
Angle-Count-Coordinate		The shortest distance of specified node to the line drawn between two pairs of other nodes in the set	Node Based
Angle-Count-Coordinate		Corner points according to latitude longitude extremes	Node Based
Angle-Count-Coordinate		Being a hull defining customer or not	Node Based
Angle-Count-Coordinate	*	The area of the convex hull of the instance	Node Based
Cluster		The Cluster ID the node assigned	Cluster Based
Cluster		Average angle of the assigned cluster to the depot	Cluster Based
Cluster		Percentage of orders that are assigned to the specified node's cluster	Cluster Based
Cluster		Absolute difference between average angle of the cluster and specified node angle	Cluster Based
Cluster		Angle standard deviation of the cluster	Cluster Based
Cluster		Maximum distance between two nodes inside the cluster	Cluster Based
Cluster	**	Which cluster the node assigned	Cluster Based
Cluster	**	Average angle of the second cluster	Cluster Based
Cluster	**	Average angle of the assigned cluster to the depot	Cluster Based
Cluster	**	Absolute difference between average angle of the cluster and specified node angle	Cluster Based
Cluster	**	Angle standard deviation of the cluster	Cluster Based
Cluster	**	Maximum distance between two nodes inside the cluster	Cluster Based

[1]* The gap between the original set's feature value and the removed set's feature value

[2]** Cluster number is increased by one

Table A.3: ARCGIS-based Distance Features

Category	Note	Definition	Base
Distance		10% of the distance (0.1 quantile)	Set Based
Distance		50% of the distance (0.5 quantile)	Set Based
Distance		90% of the distance (0.9 quantile)	Set Based
Distance		Median of all distances to Hub	Set Based
Distance		Absolute difference between calculated Median and Mean values	Set Based
Distance		The distance between customer and the depot	Node Based
Distance		Total number of customers in this radius when center is the specified customer (Radius: Hub centered distance covering half of the orders divided by three)	Node Based
Distance		Percentage representation of " Total number of customers in this radius when center is the spesified customer (Radius : Hub centered distance covering half of the orders divided by three)"	Node Based
Distance		1/sum of distances to other locations from the current customer	Node Based
Distance		Harmonic Mean of the distances / Count of customers in the set	Node Based
Distance		Total of alpha (0.7) to the power of distances to other locations from the current customer	Node Based
Distance		The average distance of a node to all the other nodes	Node Based
Distance	***	The degree centrality of the node	Node Based
Distance	***	The betweenness centrality of the node	Node Based
Distance	***	The sum of the reciprocal of the shortest path distances from all other nodes to the specified node	Node Based
Distance	***	The centrality for a node based on the centrality of its neighbors. It is a generalization of the eigenvector centrality	Node Based
Distance	***	All nodes vote for each of its in-neighbors and the node with the highest votes is elected iteratively. The voting ability of out-neighbors of elected nodes is decreased in subsequent turns	Node Based
Distance	***	Number of nodes required to reach from depot to the specified node in a shortest path	Node Based
Distance	****	The degree centrality of the node	Node Based
Distance	****	The betweenness centrality of the node	Node Based
Distance	****	The sum of the reciprocal of the shortest path distances from all other nodes to the specified node	Node Based
Distance	****	The centrality for a node based on the centrality of its neighbors. It is a generalization of the eigenvector centrality	Node Based
Distance	****	All nodes vote for each of its in-neighbors and the node with the highest votes is elected iteratively. The voting ability of out-neighbors of elected nodes is decreased in subsequent turns	Node Based
Distance	****	Number of nodes required to reach from depot to the specified node in a shortest path	Node Based

[1]*** Base graph is minimum spanning tree

[2]**** Base graph is maximum spanning tree

Table A.4: ARCGIS-based Time Features

Category	Note	Definition	Base
Time		Absolute difference between calculated Median and Mean values of the distance to depot	
Time		Maximum customer-depot distance in the customer set	Set Based
Time		Min distance to the Hub (Hub-Node)	Set Based
Time		10% of the distance (0.1 quantile)	Set Based
Time		50% of the distance (0.5 quantile)	Set Based
Time		90% of the distance (0.9 quantile)	Set Based
Time		Median of the all distances to Hub	Set Based
Time		Mean of the all distances to Hub	Set Based
Time		Sum all the values in distance matrix and divide it by (N*N)	Set Based
Time		Sum all the values in distance matrix and divide it by (N*N-1)	Set Based
Time		The distance between customer and the depot	Node Based
Time		Total number of customers in this radius when center is the specified customer (Radius: Hub centered distance covering half of the orders divided by three)	Node Based
Time		Percentage representation of " Total number of customers in this radius when center is the spesified customer (Radius : Hub centered distance covering half of the orders divided by three)"	Node Based
Time		1/sum of distances to other locations from the current customer	Node Based
Time		Harmonic Mean of the distances / Count of customers in the set	Node Based
Time		Total of alpha (0.7) to the power of distances to other locations from the current customer	Node Based
Time		The average distance of a node to all the other nodes	Node Based
Time	***	The degree centrality of the node	Node Based
Time	***	The betweenness centrality of the node	Node Based
Time	***	The sum of the reciprocal of the shortest path distances from all other nodes to the specified node	Node Based
Time	***	The centrality for a node based on the centrality of its neighbors. It is a generalization of the eigenvector centrality	Node Based
Time	***	All nodes vote for each of its in-neighbours and the node with the highest votes is elected iteratively. The voting ability of out-neighbors of elected nodes is decreased in subsequent turns	Node Based
Time	***	Number of nodes required to reach from depot to the specified node in a shortest path	Node Based
Time	****	The degree centrality of the node	Node Based
Time	****	The betweenness centrality of the node	Node Based
Time	****	The sum of the reciprocal of the shortest path distances from all other nodes to the specified node	Node Based
Time	****	The centrality for a node based on the centrality of its neighbors. It is a generalization of the eigenvector centrality	Node Based
Time	****	All nodes vote for each of its in-neighbours and the node with the highest votes is elected iteratively. The voting ability of out-neighbors of elected nodes is decreased in subsequent turns	Node Based
Time	****	Number of nodes required to reach from depot to the specified node in a shortest path	Node Based

[1]*** Base graph is minimum spanning tree

[2]**** Base graph is maximum spanning tree