A Short-Term Predict-Then-Cluster Framework for Meal Delivery Services

Cheng, Jingyi; Sharif Azadeh, Shadi

DOI
10.1007/s42421-025-00140-6

Publication date
2025

Document Version
Final published version

Published in
Data Science for Transportation

Citation (APA)
Cheng, J., & Sharif Azadeh, S. (2025). A Short-Term Predict-Then-Cluster Framework for Meal Delivery Services. *Data Science for Transportation*, *7*(3), Article 26. https://doi.org/10.1007/s42421-025-00140-6

Data Science for Transportation (2025) 7:26
https://doi.org/10.1007/s42421-025-00140-6

RESEARCH

A Short-Term Predict-Then-Cluster Framework for Meal Delivery Services

Jingyi Cheng[1] · **Shadi Sharif Azadeh**[1]

Received: 21 October 2025 / Revised: 23 October 2025 / Accepted: 24 October 2025

Abstract
Micro-delivery services offer promising solutions for on-demand city logistics, but their success relies on efficient real-time delivery operations and fleet management. On-demand meal delivery platforms seek to optimize real-time operations based on anticipatory insights into city-wide demand distributions. To address these needs, this study proposes a short-term predict-then-cluster framework for on-demand meal delivery services. In the forecasting stage, point and distributional predictions are generated using multivariate features, including temporal, contextual, and lagged-dependent features to capture complex demand dynamics. In the clustering stage, we propose two methods: Constrained K-Means Clustering (CKMC) and Contiguity Constrained Hierarchical Clustering with Iterative Constraint Enforcement (CCHC-ICE). These approaches form dynamic, geographically coherent clusters based on predicted demand, while accommodating user-defined operational constraints. Case studies on European and Taiwanese datasets demonstrate that lagged-dependent ensemble learning models perform robustly under sparse, zero-inflated demand conditions, whereas deep learning models such as LSTM excel in denser data regimes. Furthermore, results from the European case study highlight that incorporating distributional forecasts effectively captures demand uncertainty, thereby enhancing the quality of clustering outcomes and operational decision-making. By integrating demand uncertainty and operational constraints, the proposed framework delivers forward-looking, actionable insights for optimizing real-time meal delivery operations. The approach is adaptable to other on-demand platform-based city logistics and passenger mobility services, contributing to more sustainable and efficient urban operations.

Keywords Short-term demand forecasting · On-demand services · Predict-then-cluster · Contiguity constrained clustering · Non-parametric distributional predictions

Introduction

“Click and pay, soon food is at the doorway" has become an urban lifestyle nowadays, driving the growth of the meal delivery industry into a global market exceeding 150 billion dollars [1]. The COVID-19 pandemic further accelerated demand due to restaurant closures and government-imposed restrictions. In response, more restaurants joined delivery platforms [1]. However, competition remains fierce among on-demand meal delivery (ODMD) platforms, such as DoorDash, Uber Eats, JustEat, and Grubhub, making it crucial for these platforms to deliver exceptional customer experiences. Speed and reliability, particularly in meeting promised delivery times, are critical factors for maintaining customer satisfaction and loyalty Liu and Florkowski (2018); Koay et al. (2022).

The operational flow of ODMD platforms involves multiple interdependent steps. In practice, once an order is placed on the platform, its details are sent to the restaurant for preparation. Meanwhile, a nearby courier receives the task and travels to the restaurant for pick-up. Once the order is ready, the assigned courier will have it delivered to the customer. Key metrics for success in this process are speed, availability, and punctuality. Performance hinges on speed, availability, and punctuality: customers seek short order-to-delivery times, while restaurants expect freshness upon arrival Dai et al. (2020); Ulmer et al. (2021). To sustain fulfillment rates and service quality, platforms must operate with high time efficiency, ensuring optimal fleet management to minimize

✉ Shadi Sharif Azadeh
s.sharifazadeh@tudelft.nl

Jingyi Cheng
j.cheng-1@tudelft.nl

[1] Transport and Planning, Delft University of Technology, Stevinweg 1, 2628CN Delft, Zuid Holland, The Netherlands

Published online: 06 November 2025

unnecessary costs or resource wastage. However, the operation of ODMD services is intrinsically complex due to highly dynamic and uncertain demand patterns. Orders arrive stochastically, each with unique specifications. These orders may originate from any restaurant in the service network. And the customers have varying preferences for delivery times, as some prefer to receive their order as soon as possible, while others may specify a desired delivery time in the future.

Rather than relying solely on reactive, myopic adjustments based on current demand, ODMD platforms benefit significantly from short-term demand predictions to inform proactive real-time operations over the fleet and demand management. Accurate demand forecasting helps platforms uphold user experience and optimize fleet utilization over time. Recent research has explored advanced machine learning and deep learning approaches for spatial–temporal demand forecasting Crivellari et al. (2022); Yu et al. (2023) and probabilistic predictions Liang et al. (2023). Ensemble-learning models also showed promising forecasting performance in recent ODMD study Hess et al. (2021).

ODMD services face the critical challenge of managing a large service network with limited resources. Beyond accurate demand forecasting, clustering is essential for managing the complexity of operations in large urban areas. By revealing emerging hot- and coldspots across neighborhoods, it provides actionable spatial units for prioritizing rebalancing and relocations Caggiani et al. (2017); Chen et al. (2016). Grouping zones with similar demand into clusters also streamlines downstream optimization, supporting facility location planningLiu et al. (2022), route optimization Prajapati et al. (2023), and fleet rebalancing Lv et al. (2020); Caggiani et al. (2017).

Real-time implementation faces several challenges. Meal-delivery demand is driven by seasonality, weather, holidays, and events, with spatial patterns that shift hour from hour. Forecasting must be both computationally efficient and accurate at high frequency. As prediction intervals become shorter to support frequent updates and spatial units become finer to offer granular information, the signal-to-noise ratio drops, making it harder for forecasting models to distinguish patterns from randomness. In addition, static clusters based on historical data fail to offer timely insights to support dynamic operations. To enhance efficiency of dynamic operations, platforms need a dynamic, forecast-driven clustered view of the network that reflects anticipated demand. Operationally viable clustering must account for both demand and spatial similarity while enforcing policy constraints (e.g., contiguity and cluster size for order bundling) to ensure feasibility and efficiency.

Addressing these challenges, we present a predict-then-cluster framework that turns short-term demand forecasts into dynamic, operations-ready hotspot and cold-spot clusters. The framework consists of two integrated stages. In forecasting, short-term ODMD demand predictions are generated for the next-interval leveraging multivariate features, including temporal, contextual, and recent demand observations. Three model families are considered: classical time-series models (SARIMA, SARIMAX, TBATS), deep learning predictor Long Short-Term Memory (LSTM), tree-based ensemble-learning (EL) predictors Random Forest (RF), eXtreme Gradient Boosting (XGBoost), and Quantile Regression Forest (QRF) for distributional forecasts. Beyond these benchmarks, we introduce lagged-dependent ensemble-learning (LD-EL) variants (LD-RF, LD-XGBoost, and LD-QRF) that augment inputs with recent-demand lags to capture local sequential dynamics and improve short-term forecasting accuracy. In clustering, forecasts are translated into forward-looking clusters that respect spatial structure of zones and operational requirements. For this stage, Constrained K-Means Clustering (CKMC) and Contiguity Constrained Hierarchical Clustering with Iterative Constraint Enforcement (CCHC-ICE) are proposed as constrained clustering solutions. CKMC jointly accounts for anticipated demand similarity and spatial proximity, but its clusters may be noncontiguous and thus unsuitable for contiguity-critical tasks (e.g., order bundling). To address this, CCHC-ICE optimizes demand similarity while simultaneously enforcing geospatial contiguity and other user-defined constraints, yielding deployment-ready clusters for downstream operations.

We demonstrate the predict-then-cluster framework on a case study using synthesized order transactions from a major meal-delivery platform in a European metropolis, where the demand time-series is highly sparse. To assess model performance across data regimes, we also evaluate the forecasting models on a Taiwanese case with higher demand volume with markedly lower sparsity. We evaluate forecasting benchmarks on predictive accuracy and computational efficiency across different data regimes. Overall, LD-QRF is the top-performing predictor. Results indicate that adding lagged-demand features consistently boosts EL predictors for both point and distributional forecasts. With identical inputs, LSTM performs best on denser series (Taiwanese case), whereas LD-QRF excels on sparse, noisy series (European case). This underscores the advantage of LD-EL models for sparse, flat-tailed zonal demand series, which are common in short-term ODMD forecasting. In our predict-then-cluster experiments, we generate clusters using CKMC and CCHC-ICE based on either deterministic point forecasts or distributional (quantile) forecasts of short-term demand. To evaluate clustering performance, we compare the clusters formed from predicted demand against those derived from actual demand data. Across both clustering methods, results consistently show that incorporating quantile-based demand predictions leads to more accurate

clustering outcomes. This suggests that explicitly accounting for zonal demand uncertainty enables the generation of predictive clusters that more closely reflect real-world demand patterns, thereby enhancing the reliability of the clustering process. To facilitate applications and enhance reproducibility, we share the coding implementation, example datasets of the empirical case studies, and a comprehensive documentation with this publication.

The key contributions of our study are summarized as follows:

- First, we introduce a short-term predict-then-cluster framework that translate short-term demand predictions into dynamic clusters, offering a frequently updated view of evolving demand across the network. These operationally viable clusters enable faster and more actionable real-time operations in meal-delivery operations. Our experiments further highlight that incorporating distributional demand estimates significantly improves clustering performance, underscoring the importance of quantifying demand uncertainty in practical applications.
- Second, we evaluate various predictive models under both sparse and dense data regimes using case studies from a European metropolis and Taiwan. These studies offer practical insights into model selection: LD-EL models demonstrate robustness in handling sparse, zero-inflated data, while LSTM models perform competitively on denser time-series. Our analysis provide data-regime-specific guidance on model accuracy, computational efficiency, and deployment suitability.
- Finally, we introduce CCHC-ICE, an adaptive clustering approach that dynamically forms operationally viable zone clusters based on demand forecasts. It groups zones with similar projected demand while ensuring spatial contiguity and adherence to user-defined operational constraints. CCHC-ICE offers flexible integration of various constraints, enabling customized clustering solutions tailored to diverse operational scenarios.

The remainder of this paper is structured as follows: Sect. 2 reviews related literature on demand forecasting and clustering for on-demand services. Section 3 formulates the predict-then-cluster problem and introduces the case studies. Section 4 details the forecasting models and clustering algorithms. Section 5 presents the experimental design, results, and implications. Section 6 offers concluding remarks.

Literature Review

Demand forecasting has been an important area of interest in both fields of forecasting and operation research for various applications Song et al. (2019); Suganthi and Samuel (2012); Ghalehkhondabi et al. (2019); Fildes et al. (2019). In Sect. 2.1, we outline why spatially and temporally high-resolution forecasts are valuable for on-demand delivery operations and the practical challenges they introduce. And we motivate clustering as an intermediary layer that translates granular predictions into operationally manageable units. Section 2.2 reviews short-term forecasting studies for on-demand. Section 2.3 surveys integrated existing frameworks that couple forecasting with clustering for decision support. Finally, in Sect. 2.4, we address the research gaps.

Challenges in Operating On-Demand Meal Delivery Services with Demand Predictions

Real-time operations, such as fleet rebalancing and order-matching, are central to the efficiency of on-demand meal delivery services. These operations are highly time-sensitive due to the limited lead time inherent in last-mile delivery. If the prediction horizon exceeds the operational completion time and is updated infrequently, the resulting demand forecasts may become outdated and unreliable, compromising optimization outcomes. Additionally, real-time rebalancing and order-matching strategies that rely on specific locations may not fully leverage forecasted demand information for a larger service area. This indicates that demand forecasts should process a comparable temporal and spatial granularity to the target operations. Effective and forward-looking decision-making for these real-time operations requires location-specific, up-to-date demand predictions alongside accurate fleet information. Compared to forecasts over broader areas or longer horizons, fine-grained temporal and spatial forecasting provides more detailed and timely insights, supporting dynamic optimization in real-time operations. However, achieving this granularity means forecasts must be generated rapidly to remain actionable.

The design of a high-resolution, fast-computing short-term demand forecasting algorithm for a meal delivery platform should address several challenges. First, demand patterns are often highly stochastic over short intervals in limited-sized service areas, resulting in noisy and intermittent data. Demand time-series may exhibit high data sparsity, making it particularly susceptible to randomness, especially when the expected demand volumes are relatively low. Second, meal delivery demand typically exhibits complex seasonality, with patterns influenced by seasonal factors such as time of year, day of the week, and hour of the day. These patterns vary significantly across different areas within a city Hess et al. (2021); Yu et al. (2023); Liang et al. (2023); Crivellari et al. (2022). For instance, business districts may see peak lunch-hour demand for sandwiches and coffee on weekdays, while residential areas may experience increased demand for pizza in the evenings or on weekends. Furthermore, forecasting models must also be resilient to demand

fluctuations caused by unforeseen events Hess et al. (2021); Liang et al. (2023). Unforeseen circumstances, such as sudden system failures or promotional events in specific areas, can significantly affect demand Liang et al. (2023). Beyond resilience, these models must be computationally efficient to support real-time predictions and adaptable to the evolving operational landscape of meal delivery platforms. As platforms expand to new service areas and adjust their networks over time, forecasting methods must be generalizable, easy to update, and capable of performing well with limited observations Hess et al. (2021). Maintaining zone-specific forecasting models offers additional flexibility, enabling independent updates without retraining the entire system or affecting predictions for other zones.

Beyond forecasting, optimizing operations over numerous small service zones is computationally demanding. Finer spatial units reduce aggregation and increase operational precision, but at the cost of scalability. In large cities like Shanghai or New York, platforms often oversee hundreds of zones within their service networks. As managing each zone individually is inefficient, it is a common strategy to group similar zones into clusters to streamline the planning process Prajapati et al. (2023). Clustering similar zones reduces computational demands, allowing platforms to optimize operations sequentially at the cluster level rather than for individual zones. Crucially, these clusters must be adaptively updated to reflect evolving demand dynamics driven by factors, such as weather, holidays, and special events. Continuous updates ensure that clusters represent the current demand landscape while preserving both computational efficiency and decision quality.

These reviewed challenges are not unique to meal delivery services but are shared by other on-demand urban mobility and logistics services. As such, investigating a holistic predict-then-cluster framework offers broader applicability to similar challenges across various on-demand service domains.

Short-Term Forecasting for On-Demand Services

Valuable insights can be drawn from short-term forecasting studies conducted for other on-demand services, such as ride-hailing and shared micro-mobility Saadi et al. (2017); Qian et al. (2020). Saadi et al. Saadi et al. (2017) investigate the performance of different machine learning models, including SVM, RF, GBM, and ANN-based regressions, using features related to pricing, weather, and traffic. Their findings highlight that ensemble-learning models, particularly GBM, achieve the highest prediction accuracy. Qian et al. Qian et al. (2020) focus on 15-min passenger demand forecasting for New York taxi services and propose a boosting Gaussian conditional random field model capable of generating robust point estimates and probabilistic predictions. Similarly, Noland Noland (2021) applies RF to predict daily shared e-scooter and bike usage, considering the effects of weather, special events, and holidays.

In the domain of on-demand meal delivery services, research has primarily focused on fulfillment cycle time forecasting Zhu et al. (2020) and arrival time estimation Hildebrandt and Ulmer (2021), but there is growing interest in short-term demand forecasting. Hess et al. Hess et al. (2021) analyze hourly demand forecasting using data from an urban delivery platform in France. Their study reveals that exponential smoothing models perform better when extensive training data are available, while machine learning models excel with limited training data. Crivellari et al. Crivellari et al. (2022) introduce the Multi-target CNN-LSTM, a deep learning model for simultaneous next-hour demand forecasting across multiple service areas, updated every 15 min in their experiments. Building on this work, Yu et al. Yu et al. (2023) propose an attention-based convolutional LSTM model to capture the inter-location correlations and spatial variation among various meal delivery service areas in the city. Liang et al. Liang et al. (2023) further extend the field by proposing a multi-task learning framework for distributional demand predictions. Their approach assumes a Poisson distribution for the order arrival process, estimating the associated parameters to quantify demand uncertainty.

Combined Prediction and Clustering Framework for On-Demand Services

Integrating clustering with time-series demand data and geographical information across multiple service areas has received significant attention in the recent literature. Many studies explore the advantages of the cluster-then-predict framework to improve demand forecasting performance for on-demand mobility servicesChen et al. (2016); Davis et al. (2016); Feng et al. (2018); Kim (2021). This hierarchical approach reduces the randomness in demand observations by aggregating demand from similar areas at a cluster level. Consequently, the demand predicted by a cluster-based forecasting model tends to be more robust than modeling individual areas with sparse demand. Chen et al. Chen et al. (2016) propose a dynamic cluster-based forecasting approach to predict the over-demand probability for clustered areas in the city. They introduce a clustering method that considers geographical proximity, historical demand patterns of service areas, as well as the ongoing special events at the predicted time step. Kim Kim (2021) introduces a spatial contiguity-constrained hierarchical clustering method to generate clusters for bike-sharing traffic prediction. This study defines two grids to be spatially contiguous if the shortest path between them does not pass through any other grids with bike-sharing stations.

Importantly, the outcomes from clustering can be used to generate insights into the demand landscape within the service network over time Liu et al. (2019); Caggiani et al. (2017). Liu et al. Liu et al. (2019) use K-Means clustering with historical demand patterns and location data to identify which districts were historically more likely to have supply surplus and deficits at different periods for a ride-hailing service. To minimize the duration of time when vehicles are unavailable, Caggiani et al. propose a dynamic clustering method to assist vehicle relocations Caggiani et al. (2017). Their approach jointly considers the geographical proximity, availability of vehicles, and received demand of service areas as clustering inputs. They show that the efficiency of fleet relocation is enhanced by utilizing dynamic zoning instead of static zoning. Focusing on the real-time operations of large-scale ride-hailing services, Alisoltani et al. propose to address the challenge of computational time and solution quality trade-off with clustering Alisoltani et al. (2020, 2022). In their solutions, clusters are generated to pre-group the most compatible trips together. Dispatching or matching optimization is then performed for each individual cluster, achieving major reduction in computational time while reserving near-optimal solutions Alisoltani et al. (2020, 2022).

Although dynamically clustering entities based on forecasts is still emerging in last-mile logistics, analogous ideas have been explored in other fields. Su et al. adopt a "predict-and-cluster" strategy for unsupervised action recognition, where future-frame prediction encourages the latent space to self-organize into meaningful motion clusters Su et al. (2020). Barrio-Hernández et al. cluster predicted protein structures from AlphaFold to systematically organize the structural space of the protein universe, enabling discovery of new functional and evolutionary relationships Barrio-Hernandez et al. (2023). In the logistics domain, Zhang et al. apply a differentiable constrained K-Means model optimized end-to-end with a graph-based demand predictor to form courier assignment zones Zhang et al. (2024). However, their formulation does not strictly guarantee spatial contiguity among zones.

A wide range of clustering methods have been proposed for spatial and regionalization problems, yet their suitability for dynamic clustering applications in meal delivery closely depends on the level of spatial and operational constraint required. Unconstrained methods, such as K-Means, DBSCAN, or spectral clustering, generate compact clusters in feature space but disregard spatial contiguity and operational feasibility. making them unsuitable for service zoning where geographical coherence is essential. Constrained extensions introduce partial control through mechanisms like pairwise constraints (e.g., COP-KMeans) or spatially constrained regionalization approaches (e.g., SKATER, AZP, Max-p regions), which enforce geographic connectivity or cluster-size thresholds Basu et al. (2008); Duque et al. (2012). However, it remains unclear whether existing clustering methods can be extended to simultaneously enforce multiple spatial and operational constraints while preserving similarity structures in the feature space Gançarski et al. (2020). Moreover, the generation of predictive clusters based on distributional demand inputs (e.g., forecast quantiles) remains largely unexplored. In addition, a few studies have addressed the computational efficiency challenges associated with high-frequency re-clustering needed for real-time or rolling operational applications Guo (2008).

Research Gaps

Upon reviewing the existing literature, we have identified several research gaps in demand forecasting for on-demand meal delivery. First, the current literature on short-term demand forecasting is limited, particularly for prediction horizons and update frequencies of less than an hour. To the best of our knowledge, most studies focus on the hourly demand forecasting problem for ODMD Yu et al. (2023); Crivellari et al. (2022); Liang et al. (2023); Hess et al. (2021). Demand forecasting for intervals shorter than an hour remains largely unexplored. Second, a few efforts have been made to estimate the uncertainty of demand. The quantification of demand uncertainty has been proven to be the key to the success of operation policies by recent literature on on-demand mobility services Lei et al. (2020); Huang et al. (2022); Guo et al. (2023). Liang et al. Liang et al. (2023) introduce a probabilistic prediction approach based on the Poisson assumption for the arrival of orders. However, this assumption might not fully capture real-life demand services, especially when demand changes during a period due to exogenous factors not considered in the model. As suggested by Lei et al. Lei and Wasserman (2014), it could be beneficial to explore nonparametric prediction models to address this limitation. Finally, despite the increasing attention to hierarchical "cluster-then-predict" frameworks for demand forecasting, a few studies have investigated "predict-then-cluster" approaches for dynamic cluster generation. Real-time operations based on dynamic zoning have been proven to increase resource management efficiency, i.e., microdelivery fleet management. Caggiani et al. (2017); Alisoltani et al. (2020, 2022). Operations such as fleet rebalancing require lead time before taking effect. Thus, using demand predictions to generate forward-looking dynamic zoning further improves the operational efficiency of the system in both short- and long-terms. Additionally, geographical contiguity-constrained clustering methods remain underexplored in the context of on-demand meal delivery services. Real-time operational efficiency can be improved by identifying dynamic clusters that satisfy both contiguity and user-specified constraints based on predicted

short-term demand distributions across the city. This gap highlights the need for methodologies that integrate predictive demand models with flexible, geographically coherent clustering frameworks to support the unique challenges of on-demand delivery services.

Problem Description

In this section, we outline the research questions and introduce the datasets used for our empirical case studies. In Sect. 3.1, we provide the formulation of our research problems, followed by the definitions of key terminologies in Sect. 3.2. Finally, Sect. 3.3 provides an overview of our selected empirical case studies.

Research Problem Formulation

In this study, we propose a predict-then-cluster framework designed to accurately forecast demand across multiple service zones within a city and subsequently group zones with similar demand profiles. This approach enables the identification of future demand coldspots and hotspots in the service network of a meal delivery platform.

Figure 1 illustrates the overall workflow of the proposed framework. It leverages real-time multivariate inputs, such as date-time, weather, and order information, to capture recent demand patterns. Data from multiple sources are processed into zone-wise data arrays, which are then fed into pretrained predictive models for short-term demand forecasting. The predicted demand, along with spatial information for each zone, serves as input for dynamic clustering. The resulting clusters reveal areas with similar anticipated demand levels, providing actionable insights to support real-time operational decisions, such as reallocating delivery couriers to zones with high expected demand.

To summarize, the proposed framework consists of two components: i) short-term demand forecasting and ii) dynamic clustering. The forecasting step aims to efficiently generate accurate short-term demand predictions using the latest demand observations and available contextual information with computationally efficient predictor. These demand predictions then serve as inputs for the clustering step, where service zones with similar predicted demands are grouped based on operational requirements, such as geographical contiguity or location proximity. The resulting clusters are designed to enhance delivery efficiency by meeting user-defined operational constraints. Moreover, the dynamic clusters generated from predicted demand should closely resemble those formed using actual demand, thereby validating the robustness and accuracy of the forecasting model.

Definitions

We consider a platform's service region to be divided into local service zones. In practice, these service zones are typically represented as equal-shaped grids within a city. Based on functionality, a service zone z_i is called a ***pick-up zone*** if users can place orders from any restaurant within this zone, meaning that couriers may pick up orders from z_i. On the other hand, a service zone is a ***destination zone*** if households in this zone can be served by the platform. That is, this service zone can be listed as the destination of a meal delivery task. Forecasting tasks mainly concern the pick-up zones in this study. Hence, we define a set of pick-up zones $Z = \{z_1, z_2, \cdots, z_N\}$, where N is the total number of pick-up zones.

As discussed in Sect. 2.1, to facilitate real-time operations, the meal delivery platform should generate frequently updated short-term demand forecasts. In this study, we adopt a 15-min forecasting horizon and update predictions every 15 min. Prior to model fitting, historical orders are aggregated into zone-level time-series at 15-min resolution. Assuming the platform updates demand predictions at 0, 15, 30, and 45 min past the hour (business hours only), we construct demand time-series $[y_t^i, y_{t-1}^i, \cdots, y_1^i]$ where y_t^i is the number of orders in the preceding 15-min interval. The prediction target is the next interval's demand y_{t+1}^i defined as the count of orders arriving in the subsequent 15-min window for zone i.

Demand forecasting models are trained to specifically predict for each pick-up zone using both global and zone-specific inputs. Global features X_t share the same value across all zones at time t, whereas zone features X_t^i vary by zone i. The next-interval prediction for zone i is

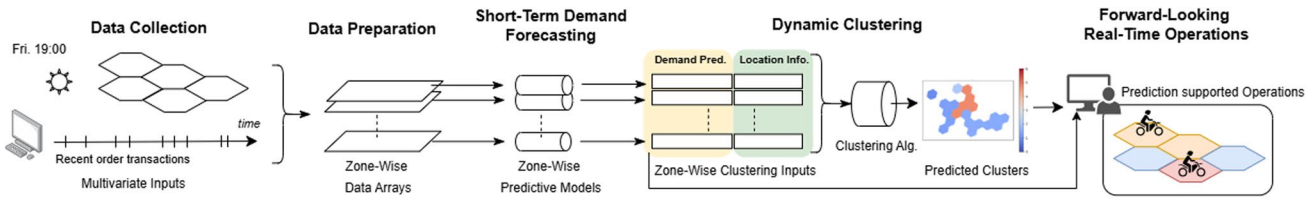

Fig. 1 Workflow of the predict-then-cluster framework for supporting real-time operations in meal delivery services.

$$\hat{y}^i_{t+1} = f^i(X_t, X^i_t)$$

, if a separate forecaster for each pick-up zone. When a single global predictor is used (e.g., an LSTM shared across zones), the zone index drops from the model. Depending on the algorithm, the predictor produces either a point forecast or a full predictive distribution (e.g., quantiles), the latter enabling explicit uncertainty assessment.

Demand predictions $\hat{Y}_{t+1} = \{\hat{y}^1_{t+1}, \hat{y}^2_{t+1}, \cdots, \hat{y}^N_{t+1}\}$ are gathered as input to the clustering step. Depending on the requirements for the geographical proximity within clusters which are imposed by the target operations defined by the platform, we identify three possible cases:

1. **No Geographical Proximity Required:** In this scenario, clustering can be performed solely based on the predicted demand of zones. A straightforward method is to classify zones using a simple thresholding approach to their predicted demand.
2. **Geographical Proximity Required:** Many real-time operations of on-demand services can benefit from regional insights (as discussed in Sect. 2.1). To support these operations, clusters are formed by jointly considering predicted demand similarity and geographical proximity of zones.
3. **Geographical Contiguity Required:** When contiguous service areas are needed (e.g., for order bundling), the clustering imposes a contiguity constraint which ensures that zones within the same cluster are contiguous.

The proposed clustering methods target the last two scenarios requiring either geographical proximity or contiguity. At time interval t, the zones are partitioned into K_t clusters: $\Omega_t = \{\omega_1, \cdots, \omega_{K_t}\}$, where each cluster ω_k contains a set of pick-up zones assigned to it.

Empirical Case Studies

To assess the performance of our proposed demand forecasting and clustering methods, we analyze the model's performance using meal delivery demand data from two empirical case studies. Before modeling, exploratory data analysis (EDA) is performed to uncover the underlying patterns and reveal the potential data characteristics. Below, we introduce the background of these case studies and highlight key insights from EDA.

European Use Case

The first case study features transaction-level data from our industry partner, a leading meal delivery platform in Europe. The dataset contains 188,162 simulated order records derived from operational logs in a large European metropolis, spanning April 1^{st}, 2020 to September 14^{th}, 2020. Orders are mapped to 20 pick-up zones and 50 destination zones. Each record includes the timestamp of order placement, and the H3 cell IDs for pick-up and delivery. The H3 hexagon grid unit is at resolution 8 with an average cell area of $\approx 0.737 km^2$ Uber: H3 (2018). The platform operates daily from 10:30 to 21:30, allowing aggregation into 44 consecutive 15-min intervals per day based on the placement time. Orders are accepted only when the restaurant lies within a designated pick-up zone and the destination address lies within a serviced zone. Although raw data cannot be shared, we provide a synthetic dataset generated via the anonymization procedure in Appendix A.1. Additional descriptive analysis for the European (EU) use case is provided in Appendix A.2.

The average daily order volume ranges from 800 to 1400 and exhibits a clear weekly cycle with relatively lower demand volumes from Monday to Thursday, and generally higher volumes from Friday to Sunday. To examine intraday patterns, we compare mean orders per business hour, grouping days into weekdays (Monday–Thursday) and weekends (Friday–Sunday). Friday is treated as weekend given typical dinner surges. As shown in Fig. 2, demand follows a two-peak profile at lunch ($\approx$ 12:00) and dinner ($\approx$ 18:00) on both groups, with the dinner peak substantially larger. The lunch peak is only modestly above the mid-afternoon levels. Overall levels are higher on weekends, particularly during the evening hours, consistent with leisure-driven ordering. Beyond weekends, holidays also affect meal-delivery demand Tong et al. (2020); Chen (2020). Between April 1^{st} and September 14^{th}, 2020, five of eight holidays recorded order volumes above the weekday 80th–percentile level. Our findings suggest that holidays have a positive influence on the demand for food delivery in the city.

We next examine demand aggregated at 15-min resolution. Figure 3 maps average orders for the full day and for lunch (11:00–14:00) and dinner (17:00–21:00) windows. A central pick-up zone consistently exhibits the highest demand. Spatial patterns differ across periods. Several zones show modest activity at lunch but a pronounced surge at dinner, indicating time-of-day heterogeneity in demand hotspots. To examine the sparsity of demand data in a demand series aggregated at 15-min intervals, we analyze the sparsity ratio, which shows the proportion of observations with zero values. In the European use case, zeros account for an average of 47.0% of total observations in the 15-min aggregated demand series, with a maximum sparsity ratio of 67.2% among the pick-up zones.

Weather conditions, such as rain and snow precipitation, often influence demand for meal delivery Yao et al. (2023). In the European case, light precipitation is associated with

a modest uplift in orders relative to dry conditions, whereas heavy precipitation corresponds to lower volumes.

Taiwanese Use Case

Our second case study uses the open-source Taiwan dataset released by Delivery Hero, a global meal delivery service provider Assylbekov et al. (2023). The platform has operated in Taiwan since 2012 and averages over 22,000 orders per day across a service area of 600 km^2 Assylbekov et al. (2023). The dataset has been anonymized in advance by the platform to ensure privacy. Each order transaction in the data includes the pick-up and drop-off locations, the placement hour and day-of-the-week information. Unlike the European case, calendar date information is not provided. The pick-up and drop-off locations are encoded into zonal addresses via geohash at a precision level of 5. There are 25 pick-up zones in this dataset, each covering a squared area of approximately 24 km^2. The 3-month historical record contains 1.64 million orders from the first 76 days used for training and 0.36 million orders from the last 14 days for testing,

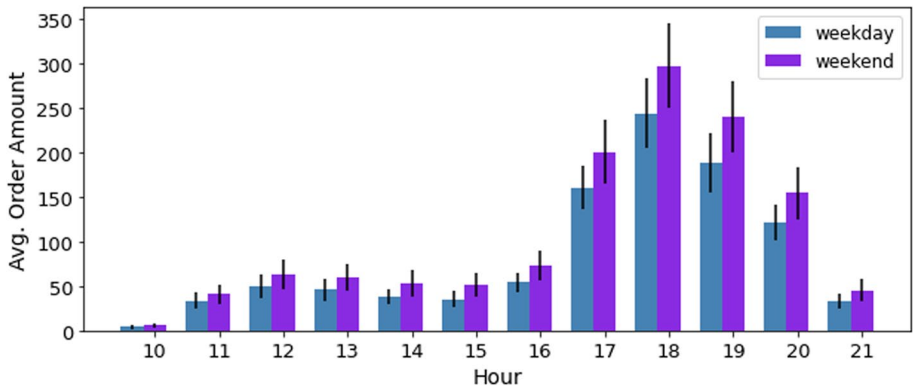

Fig. 2 The average number of orders received in the European city during different hours on weekdays versus weekends, from April 1^{st}, 2020 to September 14^{th}, 2020

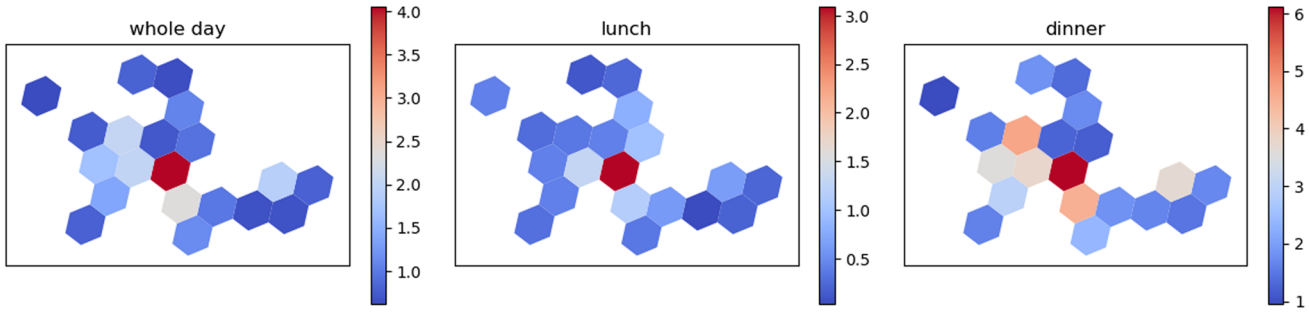

Fig. 3 The heatmap visualization of the number of orders received by different pick-up zones from the European use case per 15-min intervals, averaged over three time periods: full day, lunch (11:00–14:00), and dinner (17:00–21:00). The heatmaps use a color scale to indicate the density of orders, with hotter colors representing higher zonal order volumes

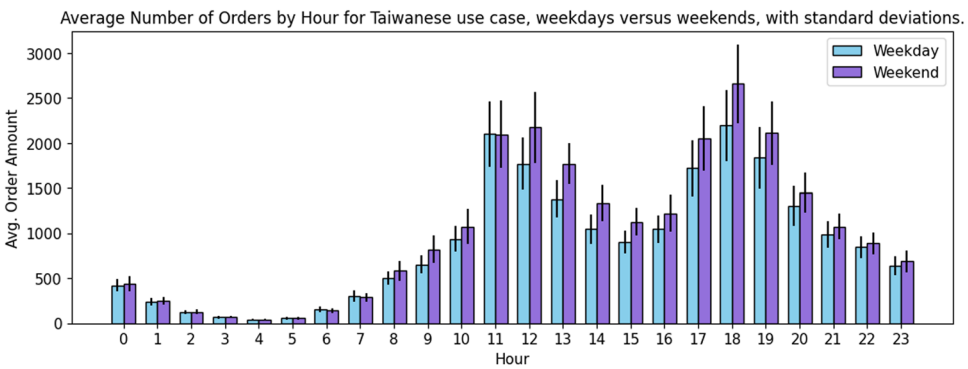

Fig. 4 The average number of orders received in Taiwan during different hours on weekdays versus weekends was calculated using a dataset gathered over a 3-month period. Error bars represent the standard deviation per hour across different days

following the guideline from Assylbekov et al. Assylbekov et al. (2023).

Splitting the order data into weekdays and weekends as before, Fig. 4 presents the average number of orders received per hour. With the delivery service operating 24 h a day, the order volume exhibits a two-peak pattern at lunch and dinner times. Lunch and dinner volumes are broadly comparable across weekdays and weekends, with a modest weekend uplift around 18:00. As in the European case, weekend levels are generally higher. Additionally, significant intra-hour variability is observed during peak times, with narrower dispersion observed during off-peak hours.

Figure 5 maps average demand per 15-min interval by pick-up zone for the full day, lunch, and dinner periods. Relative demand rankings are largely stable across lunch and dinner, though several central zones show a pronounced dinner surge.

Additionally, the average number of orders received per 15 min is significantly higher than the European case, because each Taiwanese zone covers a larger area. Consequently, the sparsity ratios per 15-min aggregated demand series are much lower in the Taiwanese use case, with only 18.3% of total observations being zero on average across zones.

The Short-Term Predict-Then-Cluster Framework

This section presents the short-term predict-then-cluster framework for ODMD. Section 4.1 details the forecasting models, and Sect. 4.2 describes the constrained clustering approaches used to form dynamic clusters from predicted demand.

Short-Term Demand Forecasting Methods

We consider four families of models for short-term demand forecasting within the predict-then-cluster framework: (i) classical time-series methods, (ii) tree-based ensemble learners, (iii) deep learning models, and (iv) naive baselines commonly used in practice. We first present the selected deterministic and distributional forecasting benchmarks considered in Sects. 4.1.1 and 4.1.2. Then, we introduce a lagged-dependent extension for ensemble models in Sect. 4.1.3.

Short-Term Demand Forecasting benchmarks

Tree-based ensemble-learning (EL) methods, such as random forest regression (RF) Breiman (2001) and eXtreme Gradient Boosting (XGBoost) Chen and Guestrin (2016), are strong point-forecasting baselines for hourly demand in meal delivery Hess et al. (2021) and other short-term demand forecasting applications He et al. (2020); Albrecht et al. (2021); Saadi et al. (2017). They are robust, efficient, and comparatively interpretable, and often outperform neural networks on structured, short-term forecasting tasks Vairagade et al. (2019); Lu et al. (2019). A key advantage in our setting is resilience to sparsity and zero inflation. EL models achieve this by recursively partitioning the feature space, trees separate “no-demand" and “demand" regimes and capture context-specific patterns without fitting a single global parametric surface Breiman (2001). The bagging/boosting further stabilize predictions via model averaging, yielding strong performance on intermittent series Makridakis et al. (2020, 2022). In addition, tree ensembles provide transparent, rule-based decision paths that facilitate auditability and feature attribution Molnar (2025); Carvalho et al. (2019); Ribeiro et al. (2016), which is important in transport and logistics applications Makridakis et al. (2020); Bertsimas and Kallus (2020). For these reasons, we adopt EL models, RF and XGBoost, as deterministic demand forecasting benchmarks.

Next to the EL predictors, we include widely used time-series models, SARIMA, SARIMAX, and TBATS, which have demonstrated state-of-the-art performance in demand forecasting tasks across various domains Kaffash et al. (2021); Kumar and Vanajakshi (2015); Shuai et al. (2022); Nosal and Miranda-Moreno (2014); Chiang et al. (2011). To

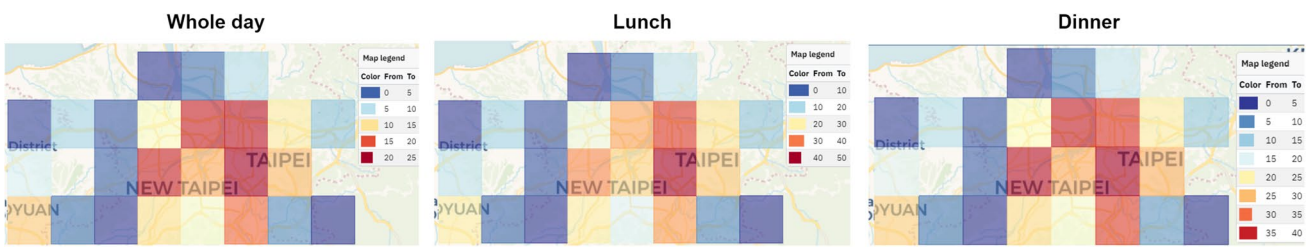

Fig. 5 The heatmap visualization of the number of orders received by different pick-up zones from the Taiwanese use case per 15-min intervals, averaged over three time periods: whole day, during lunch (11:00–14:00), and dinner (17:00–21:00) times. The heatmaps use a color scale to indicate the density of orders, with warmer colors representing higher order volumes

benchmark against a deep learning alternative, we implemented a multivariate Long Short-Term Memory (LSTM) network as a deterministic predictor for short-term demand Hochreiter and Schmidhuber (1997). The model utilizes the same input features as the ensemble-learning predictors and is trained as a global model to forecast demand across all zones. In practice, many meal delivery platforms do not yet implement demand forecasting. A naive alternative is to use the most recent observation as the prediction for the next time interval. This approach, referred to as the **myopic** predictor, is also included as a naive benchmark for deterministic demand forecasting.

Distributional Short-Term Demand Forecasting Benchmarks

Distributional demand predictions offer two key advantages. First, they allow the use of the median of the predicted distribution as a point estimate. Unlike traditional regression methods that default to predicting the mean, median-based predictions are less influenced by extreme values or outliers, resulting in more robust performance. Second, distributional predictions quantify demand uncertainty, providing critical insights that enable platforms to optimize operational policies more effectively. To leverage these benefits, we apply quantile regression forest (QRF), a distributional forecasting variant of RF introduced by Meinshausen Meinshausen and Ridgeway (2006). QRF generates robust point predictions while also capturing uncertainty through demand quantiles. It has demonstrated strong performance in various applications, including short-term electricity demand forecasting Xing et al. (2020) and online grocery retailing demand Ulrich et al. (2021).

As a naive benchmark, we include the **Seasonal Quantile (Seasonal)** predictor to evaluate the distributional forecasting performance of QRF methods. The Seasonal predictor segments historical demand data by hour and day of the week to create time-dependent empirical distributions for each pick-up zone. Quantile predictions are then generated based on these seasonally conditioned distributions, providing a baseline for evaluating the effectiveness of the proposed methods.

Lagged-Dependent Ensemble-Learning Predictors

While seasonal variables can be extracted from historical data, short-term demand fluctuations are often influenced by transient and unobserved factors, such as restaurant promotions and major sporting events. These factors may not be explicitly captured in contextual inputs, yet they can significantly affect demand levels. To address this challenge, we propose enriching multivariate ensemble-learning (EL) predictors with lagged-dependent (LD) features, specifically recent-demand observations. These LD features serve as implicit proxies for unobserved short-term influences, complementing contextual variables and improving model responsiveness to real-time dynamics.

The utility of past observations is well established in traditional time-series models. For instance, time-series autoregressive (AR) models capture linear dependencies between the current target variable based on its recent values. Subsequence time-series clustering algorithm is also introduced to identify recurring patterns in time-series segments using sliding windows and similarity-based clustering techniques Aghabozorgi et al. (2015).

Inspired by these approaches, we incorporate four lagged terms, $y_t, y_{t-1}, y_{t-2}, y_{t-3}$, as local demand dynamic signals to improve predictions. This leads to the development of LD-extended ensemble models (LD-EL): LD-RF (Random Forest), LD-XGBoost, and LD-QRF (Quantile Random Forest). The LD-EL models offers several advantages. First, it enables the model to leverage recent-demand trends without requiring complex preprocessing or additional data collection. Second, unlike subsequence clustering, it avoids the need for separate clustering algorithms, simplifying implementation. Finally, compared to parametric model (e.g., SARIMAX and LSTM), LD-EL models retain the nonlinear, robust, and computationally efficient properties of decision-tree-based ensemble methods.

Dynamic Predict-Then-Cluster Framework for Demand Cold and Hotspots Forecasting

We propose a dynamic predict-then-cluster framework that converts short-term demand forecasts into spatially coherent cold-/hotspot clusters across a meal-delivery network. The resulting clusters provide a real-time map of anticipated demand, adapting to short-term fluctuations and supporting operational decisions.

Within this framework, we introduce two clustering approaches. Constrained K-Means (CKMC) groups similar zones into larger, cohesive units for managerial operations by enforcing a minimal cluster-size constraint. This allows operators to dynamically define operational areas and compare predicted demand across the city. However, its limitation lies in potentially forming geographically disconnected clusters, which can hinder efficiency in scenarios requiring physical connectivity.

Along with cluster-specific conditions such as the number of clusters and cluster sizes, geographical contiguity is often a critical requirement for supporting operations. This condition ensures connectivity within each cluster, guaranteeing that all zones from the same cluster are directly accessible to one another. However, while methods like CKMC consider geographical proximity, they do not guarantee such contiguity. As a result, zones with similar demand levels may

be grouped together even if natural barriers, such as road networks, waterways, or mountainous regions, render them inaccessible from one another. This lack of contiguity can hinder applications requiring coherent and seamless cluster configurations, such as order batching or route planning, where continuous and logical path connectivity is essential. To address these limitations, we introduce an adaptive Contiguity Constrained Hierarchical Clustering with Iterative Constraint Enforcement (CCHC-ICE) framework designed for dynamic predicted demand cluster generation under operational requirements. Accepting short-term predictions as input, this approach integrates an iterative constraint enforcement mechanism with a contiguity-constrained hierarchical clustering algorithm, ensuring that resulting clusters satisfy both future demand similarity and user-specified criteria regarding geographical contiguity and other properties.

Constrained K-Means Clustering

By measuring the similarities among clusters with both location and predicted demand information as inputs, CKMC helps monitor the spatial distribution of over-demand/under-supplied areas. CKMC utilizes K-Means clustering to group zones into clusters. This approach partitions a high-dimensional space spanned by data points into clusters by identifying representative centroids for each cluster. The algorithm assigns each data point to the nearest centroid, aiming to minimize the within-cluster variance. K-Means assumes the clusters to be approximately spherical and evenly sized.

In CKMC, the geographical proximity between two pick-up zones is quantified by the distance between their respective grid centers. Each zone includes the latitude and longitude of its grid center as static geographical features. In contrast, the predicted demand features for clustering are dynamic, which are generated from the forecasting step for the upcoming interval of all zones. The type of predicted demand features used for clustering depends on the forecasting method employed in the prediction stage of the predict-then-cluster framework. Importantly, demand uncertainty can also be incorporated into the clustering process. For example, multiple quantile predictions generated by QRF/LD-QRF can be jointly used to describe the future demand distribution.

CKMC measures similarity with a weighted Euclidean distance. Zones that are closer in feature space are considered more similar. For zones i and j with feature vectors x_i and x_j, the distance measure is

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{d} w_k (x_{i,k} - x_{j,k})^2}, \tag{1}$$

where the user-predefined weights w_k control the relation importance of each feature.

For the predict-then-cluster experiment with QRF/LD-QRF, we consider applying 25th, 50th, and 75th quantile predictions of each pick-up zone to measure the predicted demand similarity. The interquartile range (25–75th quantiles) offers a stable measure of dispersion under sparse, zero-inflated demand, being less sensitive to noise than tail quantiles. While our framework defaults to these quartiles for reliable clustering, it remains flexible, allowing alternative quantiles to be incorporated based on specific operational priorities. For instance, if a use case prioritizes tail risk (e.g., ambulance management), additional or asymmetric quantiles (e.g., 80–95th quantiles) can be incorporated with appropriate weights for distance calculation without changing the pipeline.

Equal weights are allocated to each geographical and predicted demand feature. The distance measure can be further simplified as

$$d(x_i, x_j) = \sqrt{(lat._i - lat._j)^2 + (lng._i - lng._j)^2 + \sum_{q=0.25,0.50,0.75} (\hat{y}_i^q - \hat{y}_j^q)^2}.$$

Operator users of the predict-then-cluster framework can also finetune the hyperparameters, including the set of quantile predictions and the weights, based on the operational needs in specific use cases. For experiments using point forecasts only, we triple the demand weight to keep location's relative influence comparable to the quantile setup:

$$d(x_i, x_j) = \sqrt{(lat._i - lat._j)^2 + (lng._i - lng._j)^2 + 3 \times (\hat{y}_i - \hat{y}_j)^2}.$$

All features are normalized prior to clustering.

Using $d(x_i, x_j)$, we compute the Silhouette coefficient to compare within-cluster cohesion to between-cluster separation Rousseeuw (1987). Higher average Silhouette values (range $[-1, 1]$) indicate better structure and help select the number of clusters K.

Contiguity Constrained Hierarchical Clustering with Iterative Constraint Enforcement

Contiguity Constrained Hierarchical Clustering

Contiguity Constrained Hierarchical Clustering (CCHC) extends traditional hierarchical clustering by incorporating contiguity constraints, ensuring that clusters meet specific requirements for geographical connectivity Guénard and Legendre (2022). The process of agglomerative hierarchical clustering can be illustrated by a dendrogram. It begins with each data point as an individual cluster. At each iteration, the pair of clusters with the highest similarity (or shortest

distance) are merged. This process continues until only one cluster remains or a stopping criterion is met Murtagh and Contreras (2012). In CCHC, additional constraints ensure that merging two clusters does not violate contiguity requirement, i.e., zones within a cluster must remain directly reachable without stepping out the cluster Guénard and Legendre (2022).

The incorporation of contiguity constraints within hierarchical clustering has been explored in the literature. Guénard and Legendre Guénard and Legendre (2022) provide a general CCHC framework via a Lance–Williams scheme that enforces contiguity through an adjacency matrix updated with dissimilarities. ClustGeo Chavent et al. (2018) balances attribute and geographic distances using two dissimilarity matrices, yielding flexible, soft spatial constraints. Côme Côme (2024) adopts a Bayesian view, incorporating priors (e.g., expected clusters) to obtain MAP partitions and dendrograms that encode spatial/probabilistic structure. REDCAP Guo (2008) overcomes static, first-order contiguity limits by dynamically updating contiguity after each merge, enforcing full-order spatial coherence throughout agglomeration.

Iterative Constraint Enforcement

Existing models for CCHC primarily focus on enforcing geographical contiguity during the merging process, often overlooking other critical cluster characteristics. Most frameworks Guénard and Legendre (2022); Chavent et al. (2018) rely on static adjacency matrices, and prioritize minimizing dissimilarity and maintaining spatial adjacency. While REDCAP Guo (2008) supports dynamic contiguity updates, its scope is primarily focused on spatial applications, limiting its versatility for other operational or domain-specific requirements.

To address these limitations, we propose a CCHC framework with an Iterative Constraint Enforcement (ICE) mechanism embedded in the agglomerative hierarchical clustering process. This mechanism dynamically verifies and enforces not only contiguity but also user-defined constraints, such as size, shape, or attribute homogeneity, at each step of the clustering process. By integrating these considerations, the proposed framework ensures that the resulting clusters are operationally viable, geographically coherent, and aligned with application-specific requirements.

In the CCHC-ICE framework, merging conditions between clusters are dynamically updated to account for both geographical contiguity and user-defined constraints, based on the current composition of each cluster. A pair of clusters is eligible for merging only if the resulting merged cluster satisfies all specified constraints. This ensures that, at each step, only pairs of clusters meeting the criteria are considered, maintaining adherence to constraints throughout the clustering process. The framework enforces these constraints by iteratively updating a constrained distance matrix D_z, which measures the zone-wise dissimilarities under constraints. The calculation of D_z involves updates to two symmetric binary matrices C_p and C_e at each iteration.

Constraint condition matrix C_p specifies whether two zones, respectively, belong to a pair of feasible contiguous clusters under all constraints. Upon initialization of the algorithm, a predefined symmetric binary matrix is provided to represent the adjacency relationships among zones, where two zones are considered adjacent if they share a common border on the map. Following the definition in Guénard and Legendre Guénard and Legendre (2022), geographical contiguity between two clusters is established if at least one zone from one cluster is adjacent to a zone in the other cluster. Additionally, for two clusters to be considered qualified for merging, they should also meet the user-specific constraints. For instance, if maximal cluster size is defined, the combined size of two clusters shall be under this threshold. The algorithm defines $C_p[i,j] = 1$, if zones i and j belong to a pair of clusters qualified for merging under constraints, otherwise $C_p[i,j] = 0$.

To accelerate clustering computations in the iterative framework, the matrix C_e is updated at each step to identify whether a pair of zones currently belong to the same cluster. For a pair of zones i and j from the same cluster, $C_e[i,j] = 1$; otherwise, $C_e[i,j] = 0$. This effectively transfers the clustering outcomes from the previous iteration to the current one, allowing the algorithm to bypass calculations from previous agglomerative hierarchical clustering steps and thereby accelerating the process.

Given zones i and j, a distance of zero is assigned if they already belong to the same cluster. Elsewise, the zone-wise distance $D_z[i,j]$ follows the regular distance measure $d(x_i, x_j)$ from Equation (1) with each weight parameter $w_k = 1$. If their respective clusters should not be merged in the next step according to the constraints, the zone-wise distance is set to $+\infty$. The calculation of $D_z[i,j]$ is summarized below

$$D_z[i,j] = \begin{cases} 0 & C_e[i,j] = 1 \text{ and } C_p[i,j] = 0, \\ d(x_i, x_j) & C_e[i,j] = 0 \text{ and } C_p[i,j] = 1, \\ +\infty & C_e[i,j] = 0 \text{ and } C_p[i,j] = 0. \end{cases} \tag{2}$$

At each iteration, D_z is updated and input into a standard Agglomerative Hierarchical Clustering (AHC) algorithm Murtagh and Contreras (2012). At each iteration, clusters with the smallest cluster-wise distance $\bar{d}$ are merged. The cluster-wise distance is defined as the average distance between all pairs of zones in the two clusters, consistent with standard AHC criteria Murtagh and Contreras (2012).

Additionally, a violation detector is incorporated to verify the contiguity conditions of the resulting clusters at the end of each iteration. If a violation is detected, the clustering process terminates, and the last valid set of clusters is returned.

Example operational constraints integration

Tailoring to the case study conducted in this research, we showcase the integration of three types of user-specific constraints into the CCHC-ICE framework: a cluster dissimilarity threshold, a minimum number of clusters, and a maximum cluster size. To limit the merging of dissimilar clusters, a cluster dissimilarity threshold D_{max} can be included. If the cluster-wise distance in the next merging step exceeds this threshold, the clustering process terminates, and the results from the previous iteration are returned. This ensures that clusters remain homogeneous in terms of their dissimilarity measures. The minimum number of clusters, $K_{\min}$, is useful for operations when it is necessary to maintain a certain level of granularity in the clustering output, such as in applications requiring a minimum number of operational units. The iterative clustering process of CCHC-ICE terminates when the current number of clusters reaches $K_{\min}$. Finally, limiting the maximum cluster size is essential where overly large clusters may hinder operation efficiency. For instance, in the case of meal delivery and other last-mile logistic services, clusters as operation units should be within manageable size Given a maximum cluster size of $s_{\max}$, the merge between two clusters is only feasible if the sum of their cluster size does not exceed $s_{\max}$. Pseudocodes 1 and 2 in Appendix D provide a detailed outline of the implementation for the specified constraints within the CCHC-ICE framework.

Figure 6 illustrates an example cluster merging iteration in the CCHC-ICE algorithm. subplot (1) displays the adjacency relationships for zone a. Subplot (2) shows that cluster ① (green) is contiguous with clusters ② (yellow) and ③ (orange), forming feasible pairs for merging based on contiguity constraints. Subplots (2) and (3) together depict a single iteration of agglomerative hierarchical clustering (AHC), where clusters are merged in order of increasing cluster-wise distances among feasible, contiguous pairs. This one-step merging happens for the pair exhibiting the smallest cluster-wise distance. Subplot (4) shows an example an example where the merge between clusters ① (green) and ③ (orange) is seen infeasible due to the maximum cluster-size constraint being exceeded.

Experimental Design, Results, and Implications

This section presents the design and computational results of the short-term predict-then-cluster experiments for on-demand meal delivery case studies. Section 5.1 details the

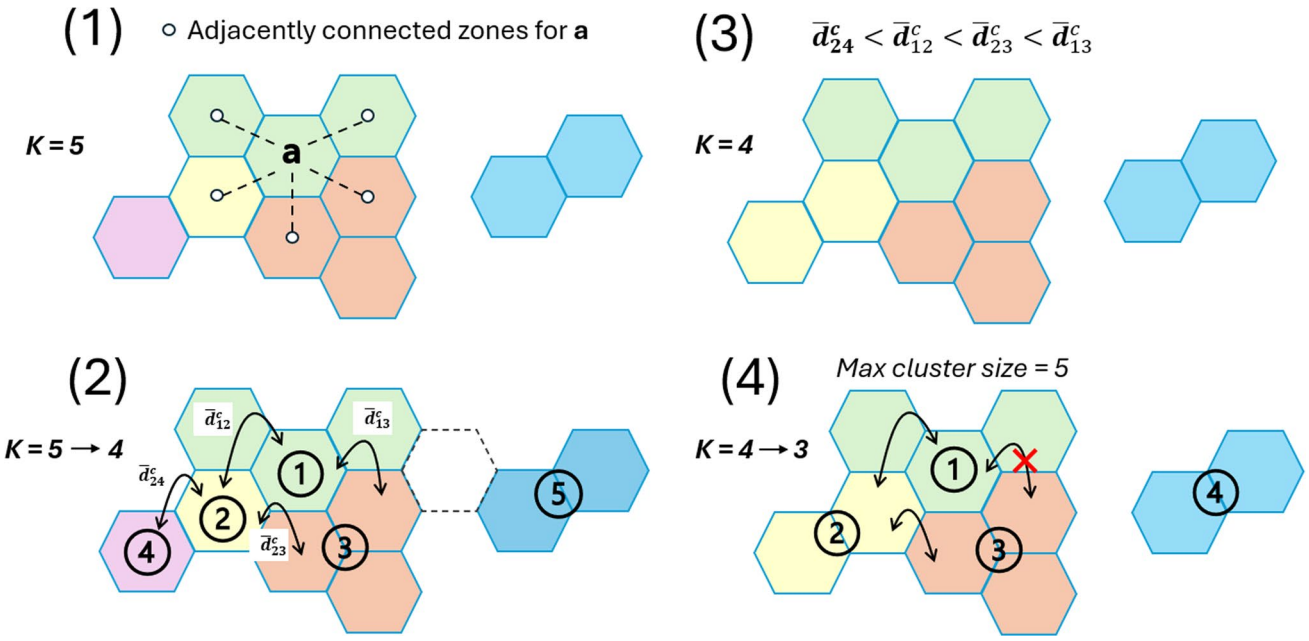

Fig. 6 Illustration of the iterative clustering process in the Contiguity Constrained Hierarchical Clustering with Iterative Constraint Enforcement (CCHC-ICE) algorithm: K is the number of clusters at the current iteration. *Max cluster size* refers to the maximum number of zones within a cluster specified in the CCHC-ICE algorithm. $\bar{d}^c_{mn}$ is the average cluster-wise distance between clusters m and n. Arrow → denotes the computation process for one-step merging within the agglomerative iterative clustering algorithm

experiment design, including the model training and prediction generation process, input features, and evaluation scheme of the experiments, as well as the computational efficiency of our selected forecasting and clustering algorithms. Section 5.2 presents the performance of deterministic and distributional demand forecasting for the European and Taiwanese use cases. Finally, Sect. 5.3 presents the results of dynamic clustering using demand predictions.

Design of Case Studies

In this section, we first detail the input features selected for demand forecasting. We then provide an overview of the model training and prediction generation processes, followed by an explanation of the evaluation schemes used for both demand forecasting and clustering experiments.

Features for Short-Term Demand Forecasting

Based on the analysis from Sect. 3, we propose to utilize three types of input features for the short-term demand forecasting of on-demand meal delivery services. These are the temporal features, weather features, and lagged-dependent features which are the previous observations of demand.

Meal-delivery demand exhibits complex seasonal patterns, as analyzed in Sect. 3.3. These patterns include daily peaks during lunch and dinner hours, higher weekend demand on a weekly scale, and spikes during national holidays. Motivated by these findings, we incorporate temporal features for forecasting, including the day of the week, the hour of the day, and a binary indicator for national holidays. These temporal features are extracted for the specific time interval being predicted.

Weather conditions affect customers' demand for meal delivery services Liu et al. (2021); Chen (2020). In this study, we utilize historical weather data from the open-source climate database of the National Centers for Environmental Information National Centers for Environmental Information (2023). The weather is described using three hourly numerical variables: average temperature (C), precipitation amount (mm), and wind speed (m/s). It is assumed that weather conditions remain uniform across the city during the same hour.

As shown in Fig. 7, the short-term demand forecasting algorithms discussed in Sect. 4.1 can be categorized based on the types of input features they employed. Note that the lagged-dependent features for each pick-up zone are extracted from their demand series as explained in Sect. 4.1.3. Time-series benchmarks Myopic, SARIMA, and TBATS rely on historical demand observations only for forecasting. RF, XGBoost, QRF, and seasonal predictors utilize contextual information, including the weather and temporal features. SARIMAX jointly considers past demand observations and contextual inputs, including weather features and holiday information. Only lagged-dependent ensemble-learning models (LD-RF, LD-XGBoost, LD-QRF) and LSTM predict using all three types of features.

As historical date information from the Taiwanese use case is unavailable, only seasonal features (hour and day of the week) and lagged-dependent features are applied for demand forecasting. Two seasonal features are available for the none lagged-dependent models in this use case; therefore, we propose using historical averaged demand based on the hour and day-of-the-week information as point forecasting benchmark ('Seasonal Average'), and the seasonal approach introduced in Sect. 4.1 as the distribution forecasting benchmark.

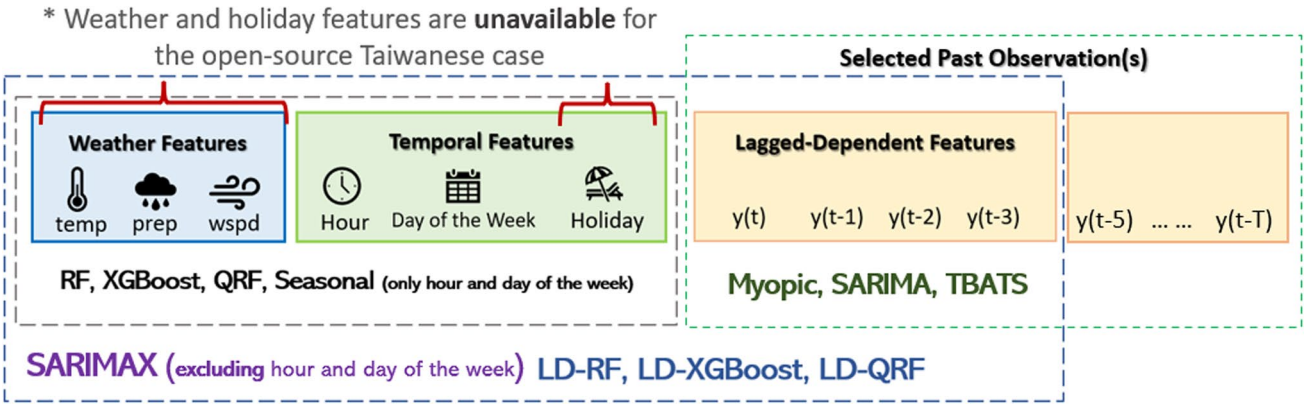

Fig. 7 Selected features for short-term demand forecasting models. Abbreviations: temperature (temp), precipitation (prep), and wind speed (wspd)

Model Training and Prediction Generation

For the European case, the original dataset contains orders gathered over 22 weeks. For each pick-up zone, we use the first 21 weeks (6468 observations) to train the model and the final week (308 observations) to test. Utilizing the date-time details of order transaction data, we can retrieve the historical hourly weather features and generate temporal features. These features are then matched to each data point in the training and testing sets. For the Taiwanese case, we follow the dataset's predefined split with the first 76 days for training and the last 14 days for testing (i.e., 7296 training and 1344 testing observations for each zone). As in the European case study, we pre-process these orders to create the target demand variable and lagged-dependent input features, along with the hour and day-of-the-week features based on the order placement time.

Except for LSTM, zone-specific predictors are trained per pick-up zone to capture its unique demand dynamics, using that zone s training data. For each zone-specific predictor, we applied time-aware tenfold cross-validation, which preserves the temporal order of data by avoiding any leakage of future information into the past. In fold k, the training set contains all observations on the time-series up to time t_k, and the validation set includes data from the immediate subsequent period $[t_k, t+k+1)$. As the fold index k increases, both the training and validation sets move forward in time. This setup ensures that model evaluation reflects real-world forecasting conditions, where future values are not available during training. Therefore, it helps to identify hyperparameters that generalize well to unseen future data. The global LSTM benchmark predictor contains a single recurrent layer of LSTM cells with a tunable number of hidden units, followed by a dense output layer with linear activation. Training minimizes the RMSE loss. Hyperparameter tuning is performed for the learning rate, size of hidden units, and dropout rate, following the time-aware cross-validation procedure. The list of hyperparameters considered for tuning and their candidate values are detailed in Table 6 in Appendix B.

The parameter selection for TBATS is done automatically by the algorithm. For SARIMA and SARIMAX, only one seasonal periodicity can be selected due to algorithmic limitations. For computational complexity concerns, we use a daily seasonal periodicity with 44 and 95 observations for the European and Taiwanese use cases, respectively. The optimal orders and parameters for SARIMA and SARIMAX are selected via a grid-search process.

Predictions are generated for each test interval without retraining. In the European case, we evaluate the benchmark predictors and the lagged-dependent ensemble-learning (LD-EL) models introduced in Sect. 4.1. For the Taiwanese case, contextual ensemble-learning benchmarks are omitted due to missing weather and holiday features. For QRF and LD-QRF, the point forecast is the conditional median ($q = 0.5$). Providing distributional inputs for sequential clustering, QRF and LD-QRF produce quantiles $q \in \{0.25, 0.5, 0.75\}$ at each time interval.

Following short-term demand forecasting, the predicted zonal demands per interval are embedded as dynamic inputs to the sequential clustering procedure, while zone geography (e.g., zone centroid locations, zone-wise adjacency) serves as a static input. At each interval, CKMC or CCHC-ICE produces generates clusters that satisfy operational constraints and reflect the predicted short-term demand. In this study, clustering experiments are conducted only for the European use case, as the pick-up zones in the Taiwanese use case are too large to form local clusters that effectively support microdelivery operations. The CKMC algorithm selects the optimal number of clusters K_t between 3 and 6 based on the highest mean silhouette coefficient, for each time interval. The implementation of CCHC-ICE algorithm in the European case study applies the contiguity and example operational constraints defined in Sect. 4.2.2. Here, we require the final outcome containing no less than 3 clusters ($K_{min} = 3$) while limiting each cluster to a maximum of 9 zones ($s_{max} = 9$). The dissimilarity threshold within each cluster is set to 9 ($D_{max} = 9$).

Evaluation Scheme

To evaluate point demand forecasting performance, we use common metrics including Mean Absolute Error (MAE) and Root-Mean-Squared Error (RMSE) to assess the magnitude of errors, and Root-Mean-Squared Logarithmic Error (RMSLE) to measure relative errors. Temporal stability of predictions is analyzed using the standard deviation of residuals (Resid. std.), calculated as the difference between actual and predicted values.

As a strictly proper scoring rule, CRPS evaluates how close a forecast CDF is to the empirical CDF induced by the observed scalar y Gneiting and Raftery (2007). For distributional demand forecasting, we report the mean CRPS (MCPRS) per zone as the time-average of CRPS across T forecasting intervals

$$MCRPS^i = \frac{\sum_{t=1}^{T} \mathrm{CRPS}_t^i(\hat{F}_t, y_t)}{T}, \tag{3}$$

with lower values indicating better distributional accuracy for zone i.

For each metric V, the summarized evaluation metric $\bar{V}$ is computed as the average value of V across all pick-up zones, defined as $\bar{V} = \frac{1}{N}\sum_i^N V^i$. The computation of selected metric V^i follows the descriptions in E. To assess spatial stability, we also report the standard deviation of metric values across

pick-up zones. Additionally, to evaluate overfitting, we provide in-sample forecast performance of ensemble-learning models using the average MAE.

For the forecasting experiment of each case study, we compute the absolute errors of each model as the absolute differences between the actual and predicted demand values for each zone, at each predicted time interval on the testing set. To assess whether the accuracy gains are statistically meaningful, we carry out pairwise comparisons of models' residuals using two complementary tests. The Mann–Whitney *U* test is used to determine if there is a significant difference in distribution between groups of residuals, while Welch s t test assesses the hypothesis of equal means between the two groups of residuals with unknown variances. Because p values alone do not convey practical importance and are sensitive to sample size, we also report effect sizes matched to each test: Cohen's d for Welch's t tests, and rank-biserial correlation for Mann–Whitney tests. All tests are reported under 95% confidence interval.

The predictive cluster insight generated from the short-term predict-then-cluster framework can inform key strategic decisions, such as determining the required fleet size for each cluster or prioritizing fleet relocation to high-demand areas, thereby reducing the need for micro-level management of individual zones. More importantly, these predicted clusters should reflect the shifting spatial demand dynamics across the city to improve the efficiency of real-time operations. Therefore, we focus on evaluating how well the forward-looking clusters, generated using predicted demand, align with the ideal scenario where clusters are generated based on actual demand.

To assess how the quality of demand predictions affects the clustering outcomes, we compute the difference between the predicted and actual within-cluster median demand for each pick-up zone. The demand level of each cluster is represented by the median predicted demand among all grids belonging to that cluster. Based on the resulting cluster assignments, we then calculate the absolute difference between the actual and predicted cluster medians, obtained from clustering results using actual versus predicted demand inputs. This metric provides an interpretable measure of how consistent the predicted clusters are with the true demand structure. The differences between actual and predicted within-cluster median demand are evaluated using metrics, such as MAE, RMSE, RMSLE, and Resid. std. This evaluation captures the combined effect of forecasting accuracy and its downstream influence on clustering validity. In essence, it quantifies how closely the clustering configuration generated from predicted demand approximates the configuration that would have been obtained if actual demand were known in advance. Smaller deviations indicate that the forecasting and clustering components are well aligned, producing clusters that remain stable and operationally meaningful under real demand realizations.

Model Computational Efficiency

Computational efficiency is a key consideration in model selection. On-demand service providers need scalable models to support their extensive service network and favor models that can quickly generate predictive insights within real-time operations. We report the average model training and prediction time for demand forecasting and clustering algorithms. All computational experiments are performed using four concurrent processes on a cloud service with one CPU and 64 GB memory. For each algorithm, we report the training time averaged among the zone-specific demand predictors. The one-step prediction generation time is measured by averaging the computation time among the predictions for the whole testing set. Likewise, we report the one-step prediction time as the average among all zone-specific models. An algorithm with low training and prediction generation times is preferred for computational efficiency.

Table 1 presents the training and prediction generating times of various demand forecasting algorithms for the European use case. SARIMA and SARIMAX models show significantly higher training time than the others. The training time of ensemble-learning and deep learning methods are significantly less. As expected, the training time becomes slightly longer for the ensemble-learning models when lagged dependent variables are included. QRF and LD-QRF store observation values directly on regression tree leaves, unlike RF and LD-RF, which apply statistical transformations. In our experiments, we found that this QRF model design slightly reduces training time compared to RF models, but results in significantly larger model file sizes.

For result generation, all short-term demand forecasting models are capable of producing one-step-ahead predictions

Table 1 The model training and one-step demand prediction generating time for the experiments in European case study

Model	SARIMA	SARIMAX	TBATS	LSTM	RF	LD-RF	XGB	LD-XGB	QRF	LD-QRF
Training (*sec*)	304.0	583.6	80.1	158.2	104.0	155.7	22.8	35.6	90.1	117.0
Predicting (10^{-3}*sec*)	74.2	75.6	0.4	0.75	7.5	9.1	1.4	1.9	9.3	9.5

sec seconds, *LD* lagged-dependent, *XGB* XGBoost

*The global LSTM predictor is trained until convergence, achieved within 20 epochs with a batch size of 64

within an average computation time of 0.1 s. The comparison of computational efficiency for the lagged-dependent predictors in the Taiwanese use case is consistent with the European use case. For the dynamic clustering algorithms, the average computation time for CKMC approach is about 1 s on average, while the average for the proposed CCHC-ICE algorithm is within 0.1 s.

Short-Term Point and Distributional Demand Forecasting

This subsection presents the point and distributional forecasting performance for selected short-term demand forecasting models in the European and Taiwanses case studies.

Point Forecasting in European Case Study

The predictive performance of selected models for European case study is summarized in Table 2. Among the time-series benchmarks, SARIMAX has obtained the highest prediction accuracy and best performance stability. However, the performance gain of SARIMAX compared to SARIMA is marginal, suggesting that the additional contextual information may not have been effectively leveraged by SARIMAX. The lagged-dependent variants (LD-RF, LD-XGB, LD-QRF) consistently improve over their base EL counterparts, and all LD-EL models clearly outperform the classical time-series baselines. Among all predictors, LD-QRF obtained the lowest MAE. Compared to the Myopic benchmark, the LD-QRF model is capable of reducing average MAE and RMSLE by 26.9% and 55.0%, respectively.

The European case study represents a highly sparse demand setting, where both the temporal (15-min) and spatial (zone-level) resolutions are fine-grained, and most target time series are dominated by zeros. Under such conditions, the LD-EL models perform comparably to or better than LSTM benchmark, across all evaluation metrics. In particular, LD-QRF achieves a lower MAE than the LSTM benchmark on both the training and testing sets, indicating its good generalization capability and high fitness to the training data. It suggests that, given the same set of input features (recent-demand lags, temporal, and contextual features), the lagged-dependent ensemble learners can effectively capture the short-term temporal dependencies that LSTM is designed to model, while remaining robust to the sparse and zero-inflated demand patterns characteristic of this dataset. The nonparametric structure of LD-EL models enables them to partition the input space adaptively, capturing both no-demand and demand regimes. The parametric neural networks, however, tend to over-smooth rare but important demand bursts in sparse and flat-tailed data.

According to the statistical tests, all learned models outperform the Myopic baseline by a highly significant margin. Observation-level effect sizes are small, with Cohen's d valuing from 0.14−0.26, and rank-biserial from 0.11−0.40, which is expected as the absolute errors comparison is done at per observation level with substantial within-group variance. Classical time-series baselines (SARIMA/SARIMAX/TBATS) are significantly worse than Myopic. Between EL models and their lag-designed variants, LD-RF and LD-XGBoost show modest but statistically reliable improvements over their EL counterparts, while LDQRF is statistically indistinguishable from QRF. Among all models, LDQRF is the top performer and is significantly more accurate than LSTM (Cohen's d $\approx$ 0.114 and rank-biserial $\approx$ 0.158).

Beyond the overall accuracy summarized from all the pick-up zones, we are also interested in analyzing the spatial variance in performance to inspect whether a model performs better or worse in particular areas in the city than

Table 2 European case study: the 1-week point forecast results of various models

Category	Model	In-Sam. MAE	MAE	RMSE	RMSLE	Resid std.
Benchmarks	Myopic	-	1.045 (0.405)	1.554 (0.499)	0.401 (0.062)	1.554 (0.499)
	SARIMA	-	1.176 (0.599)	1.550 (0.675)	0.411 (0.118)	1.424 (0.518)
	SARIMAX	-	1.119 (0.469)	1.454 (0.564)	0.391 (0.095)	1.384 (0.503)
	TBATS	-	1.228 (1.077)	1.582 (1.176)	0.585 (0.339)	1.542 (1.147)
	LSTM	0.817	0.816 (0.274)	**1.131 (0.349)**	**0.295 (0.040)**	**1.124 (0.347)**
	RF	0.785	0.863 (0.279)	1.153 (0.361)	0.308 (0.043)	1.147 (0.359)
	XGB	0.819	0.887 (0.314)	1.186 (0.393)	0.322 (0.055)	1.140 (0.359)
	QRF	0.757	0.768 (0.294)	1.197 (0.339)	0.319 (0.039)	1.169 (0.344)
LD-EL Models	LD-RF	0.752	0.848 (0.279)	1.141 (0.354)	0.304 (0.043)	1.139 (0.354)
	LD-XGB	0.815	0.860 (0.318)	1.169 (0.396)	0.311 (0.057)	1.128 (0.342)
	LD-QRF	**0.745**	**0.764 (0.297)**	1.195 (0.337)	0.319 (0.039)	1.168 (0.346)

The standard deviations of the metric value among pick-up zone models are reported between parentheses. The lowest values for each metric are marked in bold

EL Decision-tree-based ensemble-learning methods, *XGB* XGBoost, *In-Sam*. In-Sample

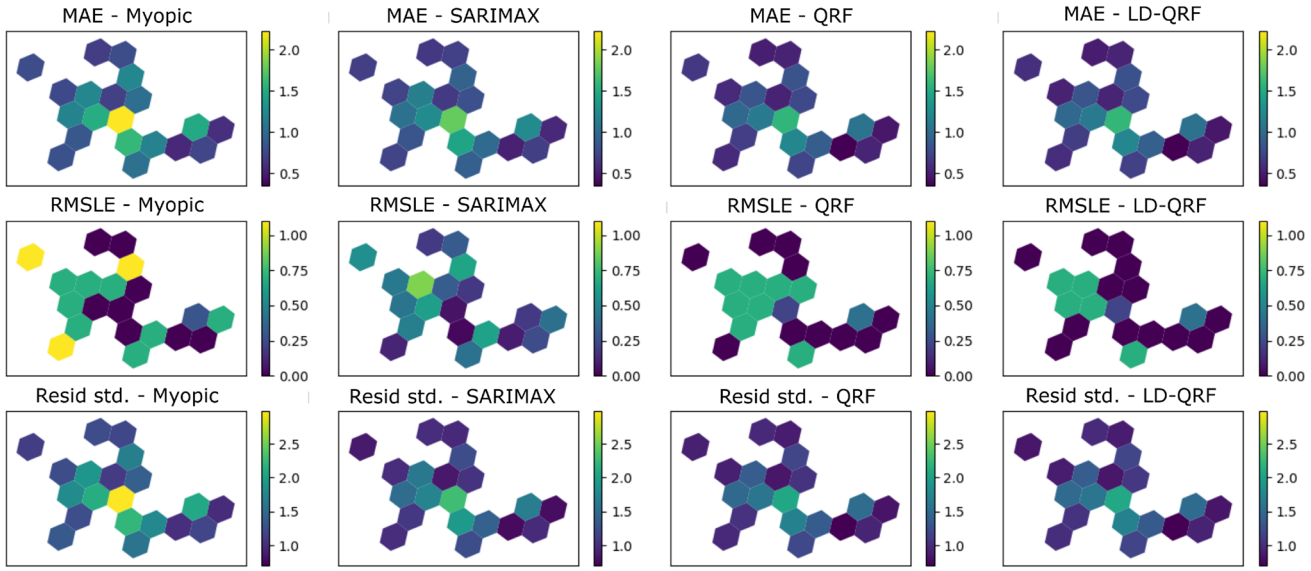

Fig. 8 European case study: heatmap visualization of the pick-up zone-wise metrics MAE, RMSLE, and standard errors of residuals (Resid. std) for the point predictions generated by Myopic, SARIMAX, QRF, and LDQRF, assessed across all predicted time intervals. The heatmaps employ a color scale to show the relative metrics values, with lighter colors denoting higher values

the others. In Fig. 8, we visualize and compare the zone-wise MAE, RMSLE, and standard deviations of residuals from Myopic, SARIMAX, QRF, and LDQRF. Overall, the metric values are higher for the pick-up zones with higher demand. These zones are located in the city center, which has been previously highlighted in Fig. 3. Comparing QRF to SARIMA, we observe an improvement in MAE across the map, indicating higher accuracy in point forecasting overall. However, the RMSLEs of QRF are higher in the center grids compared to SARIMAX's, while the relative errors of QRF are lower at the outer-center areas of the city. This implies that QRF slightly struggles in the forecasting for high-demand zones, which may present more variations in short-term demand. For the LD-QRF models, there is a significant reduction in RMSLEs in three of the central pick-up zones compared to QRF, highlighting the benefit of incorporating lagged dependencies for capturing demand patterns more accurately in high-demand areas.

To enhance the interpretability of the trained short-term demand forecasting models, we perform a post hoc analysis using SHapley Additive exPlanations (SHAP)Lundberg and Lee (2017), to quantify feature importance. SHAP values provide a unified measure of each feature's contribution to the model's predictions, evaluated over the training instances. A feature is considered important by the model if its mean absolute SHAP value is comparatively higher. These values are non-negative, with larger values indicating greater impact on prediction outcomes. Given that SHAP has been optimally designed for classic tree models like RF and XGBoost Lundberg and Lee (2017), we calculate the absolute SHAP values for these tree-based ensemble-learning models, including their lagged-dependent variants (LD-RF and LD-XGBoost). Their mean absolute SHAP values are evaluated for each zonal demand predictor across 20 pick-up zones in the European case study.

Figure 9 compares the mean absolute SHAP values SHAP of the features averaged over the all zonal predictors for RF and XGBoost models on the left sub-figure, and evaluates those for LD-RF and LD-XGBoost on its right sub-figure. For all models, hour-of-the-day feature has the highest importance among all, meaning that the daily demand pattern effects dominate in forecasting modeling. In contrast, weather and holiday features are seen less influential by the models. In the SHAP analysis for lagged-dependent models, lagged-demand features emerge as the second most important contributors to predictions. While the temporal features still dominate the predictions made by LD-XGBoost, LD-RF distributes greater importance to the lagged-demand features. It is important to note that sparsely varying features tend to have lower SHAP, because their contributions are averaged over many 'inactive' observations. In our data, precipitation and holiday are heavily zero-inflated, meaning that there were few observations from rainy or holiday times. Accordingly, their mean absolute SHAP values and the cross-zone variances are relatively small, indicating limited average influence for the EL predictor. Future analysis should explore whether accuracy gains can be obtained by fitting separate models for rainy vs. dry and holiday vs. non-holiday periods.

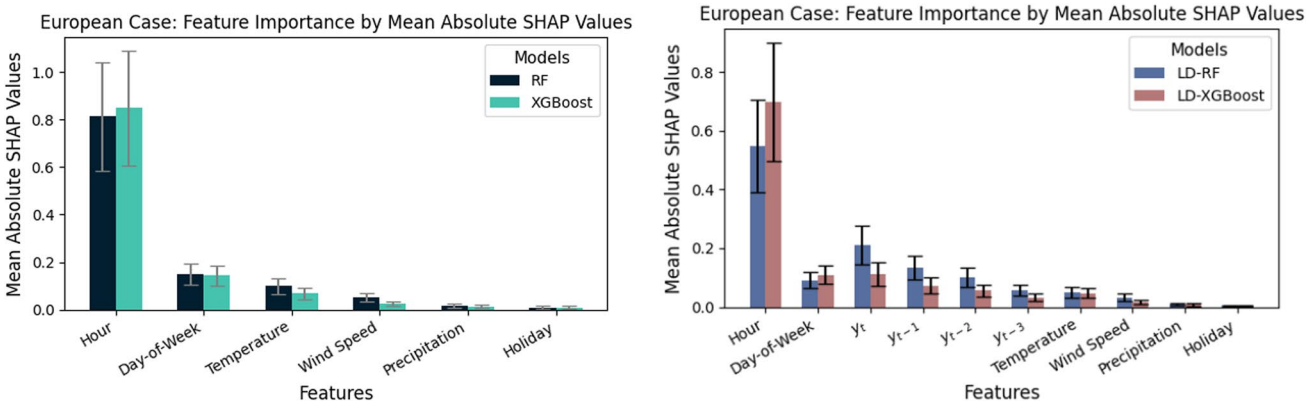

Fig. 9 European case study: mean absolute SHAP values from the feature importance analysis. The error bars denotes the range of 0.25× standard deviation among all zones. The left sub-figure compares feature importance among benchmark models (RF/QRF and XGBoost). The right sub-figure illustrates feature importance for lagged-dependent models (LD-RF/LD-QRF and LD-XGBoost), which incorporate lagged demand features (y_t, y_{t-1}, y_{t-2}, and y_{t-3})

Table 3 Taiwanese case study: the 14-day ahead point forecast results of various models

Category	Model	In-Sam. MAE	MAE	RMSE	RMSLE	Resid std.
Benchmarks	**Myopic**	-	3.192 (1.439)	4.567 (1.969)	0.407 (0.033)	4.567 (1.969)
	Seasonal Avg.	-	2.819 (1.474)	4.187 (2.183)	0.314 (0.018)	3.843 (1.880)
	TBATS	-	3.026 (1.674)	4.109 (2.166)	0.415 (0.094)	3.909 (1.954)
	LSTM	2.127	**2.409 (1.070)**	**3.356 (1.468)**	0.310 (0.020)	**3.346 (1.466)**
LD-EL Models	**LD-RF**	**1.771**	2.551 (1.167)	3.696 (1.672)	0.309 (0.024)	3.645 (1.636)
	LD-XGB	2.126	2.490 (1.134)	3.593 (1.614)	**0.304 (0.022)**	3.537 (1.576)
	LD-QRF	1.879	2.558 (1.169)	3.709 (1.680)	0.309 (0.024)	3.658 (1.646)

The standard deviations of the metric value among pick-up zone models are reported in parentheses. The lowest values for each metric are marked in bold

Point Forecasting in Taiwanese Case Study

The short-term demand point forecasting results for the Taiwanese use case are presented in Table 3. In the Taiwanese case study, the spatial units are larger, resulting in more aggregated 15-min demand data compared to the sparse zone-level structure in the European case. The local meal delivery service also operates 24/7, producing a continuous time-series without the discontinuities observed at daily boundaries in the European dataset. Moreover, no contextual features such as holidays or weather are available, leaving only historical demand and temporal indicators as predictive inputs. Due to these reasons, the values of all metrics are higher in the Taiwanese use case compared to the European case. Among all predictors, LSTM achieves the highest accuracy, measured by MAE, RMSE, and resid std. metrics. It indicates LSTM's advantage in learning smooth temporal dependencies when the demand data are less sparse. Among the LD-EL models, LD-XGBoost performs most comparably to the LSTM, achieving competitive accuracy despite its simpler architecture. Meanwhile, LD-RF and LD-QRF show lower in-sample MAE, indicating a tighter fit to the training data and a slight tendency toward overfitting in this setting, where the absence of contextual features limits the model s ability to generalize beyond temporal patterns alone.

Similar to Fig. 8 for the European case study, Fig. 10 presents a comparative analysis of forecasting errors for pick-up demand across zones in a Taiwanese city. The heatmaps visualize three error metrics (MAE, RMSLE, and Resid. std.) for four selected models Myopic, Seasonal Average, LD-XGBoost, and LD-QRF. Relating to zonal demand intensity heatmap showed in Fig. 5, the lagged-dependent models (LD-XGBoost and LD-QRF) consistently reduce MAE and Resid. std. across high-demand zones compared to Myopic and Seasonal Average benchmarks. This improvement indicates that the inclusion of lagged features enhances predictive performance for high-demand areas , where smoother temporal patterns provide reliable signals. In contrast, low- and moderate-demand areas are dominated by zeros and irregular fluctuations, limiting the usefulness of lagged inputs. In such cases, models rely more on stable

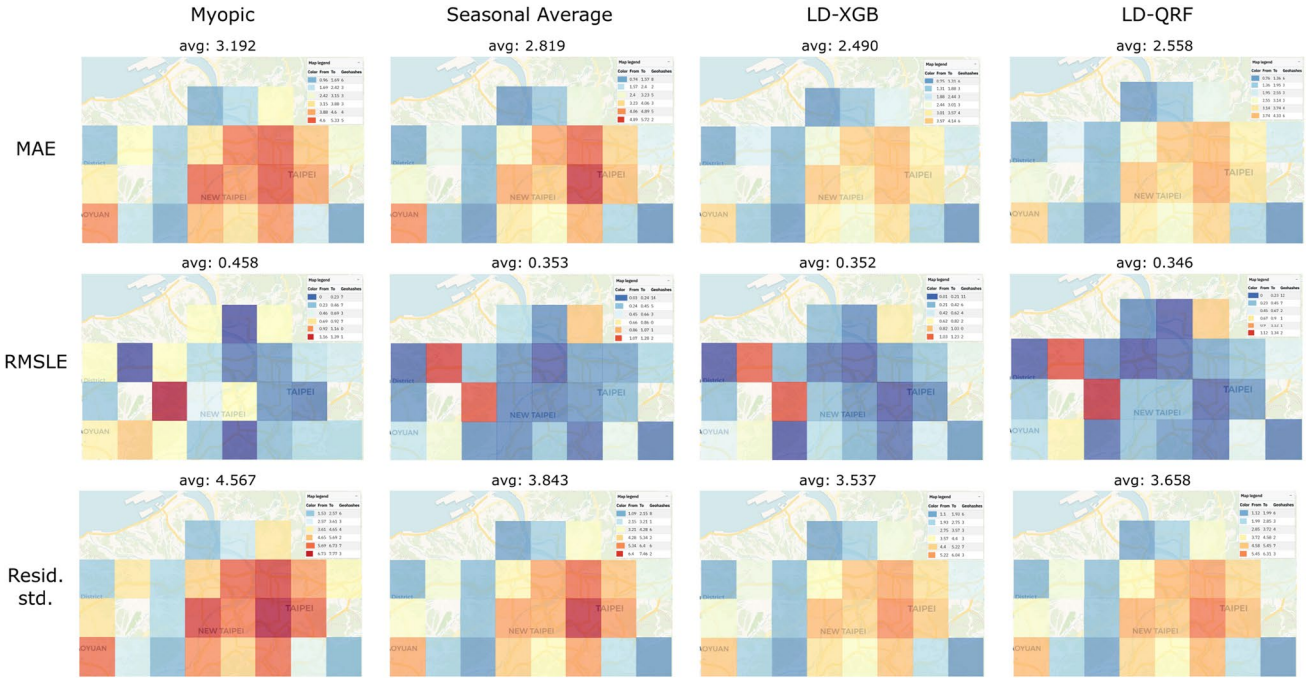

Fig. 10 Taiwanese case study: heatmap visualization of the pick-up zone-wise metrics MAE, RMSLE, and standard errors of residuals (Resid. std.) for the point predictions generated by Myopic, Seasonal Average, LD-XGBoost and LD-QRF, assessed across all predicted time intervals. The heatmaps employ a color scale to show the relative metrics values, with hotter (closer to red) colors denoting higher values

temporal indicators and contextual features. This reflects the signal-to-noise characteristics influencing ensemble-learning model performance across different demand levels. In addition, for RMSLE, the highest error values consistently appear at two moderate-demand zones on the left side of the map. This suggests that demand patterns of these zones are less predictable, when only temporal and demand lags are considered. For this particular zone, the Myopic baseline achieves the lowest RMSLE, indicating that one-step persistence captures the local dynamics better than models that emphasize seasonal structure..

As in the European case study, we performed statistical tests on the prediction residuals to compare the significance of performance differences between predictors. Versus the Myopic baseline, all learning-based predictors (LD-QRF, LD-RF, LD-XGB, LSTM) are significantly more accurate, with Cohen's d valuing from 0.24 to 0.27, and rank-biserial from 0.29−0.31. Statistical tests indicate that LSTM significantly outperforms the LDEL predictors. However, the effect sizes are small (Cohen's d ranging from 0.04 to 0.07, and rank-biserial from 0.008 to 0.033), meaning that the accuracy gain at per-observation level is marginal.

With mean absolute SHAP values, we assess feature importance for temporal and lagged inputs in the Taiwan case. Averaged across 25 zonal models, Fig. 11 shows a

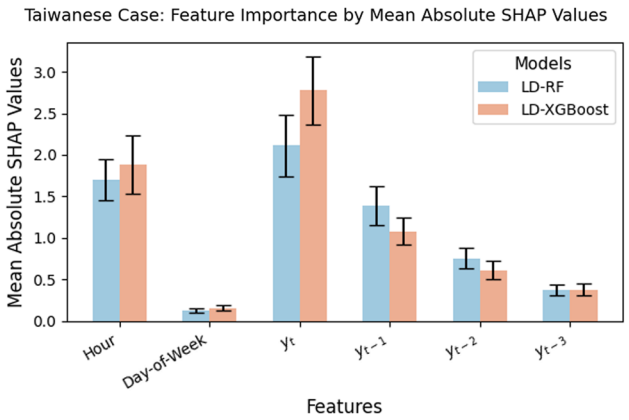

Fig. 11 Taiwanese case study: mean absolute SHAP values from the feature importance analysis for LD-RF and LD-XGBoost. The error bars denotes the range of 0.25× standard deviation among all zones.

monotonic decay in importance across lags: y_t is most influential, followed by y_{t-1}, y_{t-2}, and y_{t-3}. Temporal seasonal effects (hour, day-of-week) are secondary. The error bars ($\pm 0.25 \times$ std.) suggest moderate variation across zonal predictors, which decreases in the order of feature importance ranking. Compared with the European case,

lagged features carry relatively more weight here, plausibly because weather and holiday inputs are unavailable. The models, therefore, rely more on local autoregressive information to describe the short-term demand signals. LD-XGBoost places slightly greater emphasis on recent lags than LD-RF, consistent with its stronger focus on near-term dynamics (also observed at the European case in Fig. 9).

Distributional Forecasting Performance

We evaluate the distributional forecasting capability of short-term distributional demand predictors QRF and its lagged-dependent extension LD-QRF using MCRPS and compare against the Seasonal Average benchmark. For the European use case, we evaluate model performance by comparing training on partial data (the most recent 4 weeks) versus the full historical dataset. This setup simulates an early deployment scenario, where the model must generate reliable distributional forecasts despite limited observational data.

The results are presented in Table 4.

Overall, LD-QRF demonstrates the best fit to the historical demand distribution within the training sets among the distributional predictors. In the European case, when full training data are available, all three models show comparable performance in distributional forecasting on the testing set. However, under data scarcity (i.e., when only partial data are available), LD-QRF achieves the lowest MCRPS on both training and testing sets. This suggests that its lagged-dependent architecture more effectively captures autoregressive signals, resulting in more sample-efficient and robust distributional forecasts. The superior performance of LD-QRF predictors is more pronounced in the Taiwanese case, where demand patterns exhibit higher values and larger variations. Compared to the Seasonal benchmark, the average MCRPS for LD-QRF on the training set is reduced by 34.2% (from 1.416 to 0.931) and on the testing set by 17.2% (from 1.728 to 1.430). These results highlight LD-QRF s ability to effectively learn underlying demand distributions, even when training data are limited. The model's advantage in distributional forecasting becomes more significant as demand series exhibit higher magnitudes and greater variability.

Table 4 The mean continuous ranked probability score (MCRPS) of probabilistic predictions given by benchmark seasonal method, QRF, and LD-QRF models for both European and Taiwanese case studies

Category	Model	Training	Testing
European use case	Seasonal	0.513 (0.201)	0.571 (0.220)
Partial (4-week)	QRF	0.484 (0.183)	0.564 (0.215)
Training data	LD-QRF	**0.423 (0.140)**	**0.560 (0.214)**
European Use Case	Seasonal	0.559 (0.216)	**0.554 (0.202)**
Full (21-week)	QRF	0.515 (0.199)	0.554 (0.205)
Training data	LD-QRF	**0.456 (0.170)**	0.555 (0.207)
Taiwanese Use Case	Seasonal	1.416 (0.563)	1.728 (0.742)
	LD-QRF	**0.931 (0.275)**	**1.430 (0.504)**

The lowest values for each metric are marked in bold

To assess model fit and potential over- or under-fitting, both training and testing MCRPS scores are reported. While the Seasonal benchmark performs comparably to QRF/LD-QRF on test data in the European case, its poorer training score suggests limited learning of demand patterns. The small train-test gap reflects smoothing effects rather than true distributional accuracy, highlighting the value of more expressive models.

Practical Implications: Feature Design and Model Choice for Operational Forecasting

The superior point forecasting accuracy of LSTM and LD-EL models underscores the pivotal role of lagged-dependent features in short-term meal-delivery demand prediction. Augmenting contextual seasonality (hour-of-day, day-of-week) with recent-demand observations significantly improves accuracy by capturing rapid intra-day fluctuations that context alone cannot explain. Post hoc SHAP analysis for EL models reinforces this finding: across both European and Taiwanese case studies, lag features rank immediately after hour-of-day in importance, and their contribution becomes even more critical when auxiliary signals (e.g., weather, holidays) are unavailable. This highlights lagged demand as a robust, always-on proxy for short-term dynamics. From a policy perspective, these findings highlight the importance of integrating real-time demand signals into forecasting pipelines. Doing so enhances model predictive accuracy, particularly under conditions of limited contextual information. Improved demand forecasts enable platforms to make more informed decisions during periods of dynamic fluctuations, ultimately boosting operational efficiency and overall system performance.

For distributional forecasting, QRF delivers calibrated quantiles without imposing a parametric error model, lowering the risk of misspecification when the true demand distribution is complex or unknown. This nonparametric advantage often makes QRF more reliable than neural probabilistic predictors, particularly in settings with irregular demand or limited observations where the underlying distribution cannot be easily inferred. This capability is particularly valuable for downstream tasks such as clustering, which require quantile inputs. However, accurate quantile estimation, especially in the tails, remains data-intensive. Platforms should prioritize consistent historical data collection and retention to ensure reliable distributional forecasts.

Table 5 European case study: 1-week dynamic predict-then-cluster results of constrained K-means clustering (CKMC) and contiguity-constrained hierarchical clustering with iterative constraint enforcement (CCHC-ICE) with predicted demand inputs from SARIMA, LSTM, QRF, and LD-QRF

		CKMC		CCHC-ICE	
		MAE	RMSE	MAE	RMSE
Point Pred.	SARIMA	1.133 (0.544)	1.459 (0.609)	1.144 (0.621)	1.515 (0.729)
	LSTM	0.759 (0.237)	**1.072 (0.317)**	0.827 (0.294)	**1.144 (0.393)**
	QRF	0.694 (0.289)	1.122 (0.343)	0.763 (0.339)	1.190 (0.439)
	LD-QRF	0.699 (0.287)	1.122 (0.331)	0.760 (0.339)	1.191 (0.428)
Quantile Pred.	QRF	0.693 (0.288)	1.137 (0.345)	0.762 (0.342)	1.198 (0.434)
	LD-QRF	**0.687 (0.281)**	1.123 (0.335)	**0.749 (0.340)**	1.187 (0.422)

The lowest metric values in each category are marked in bold

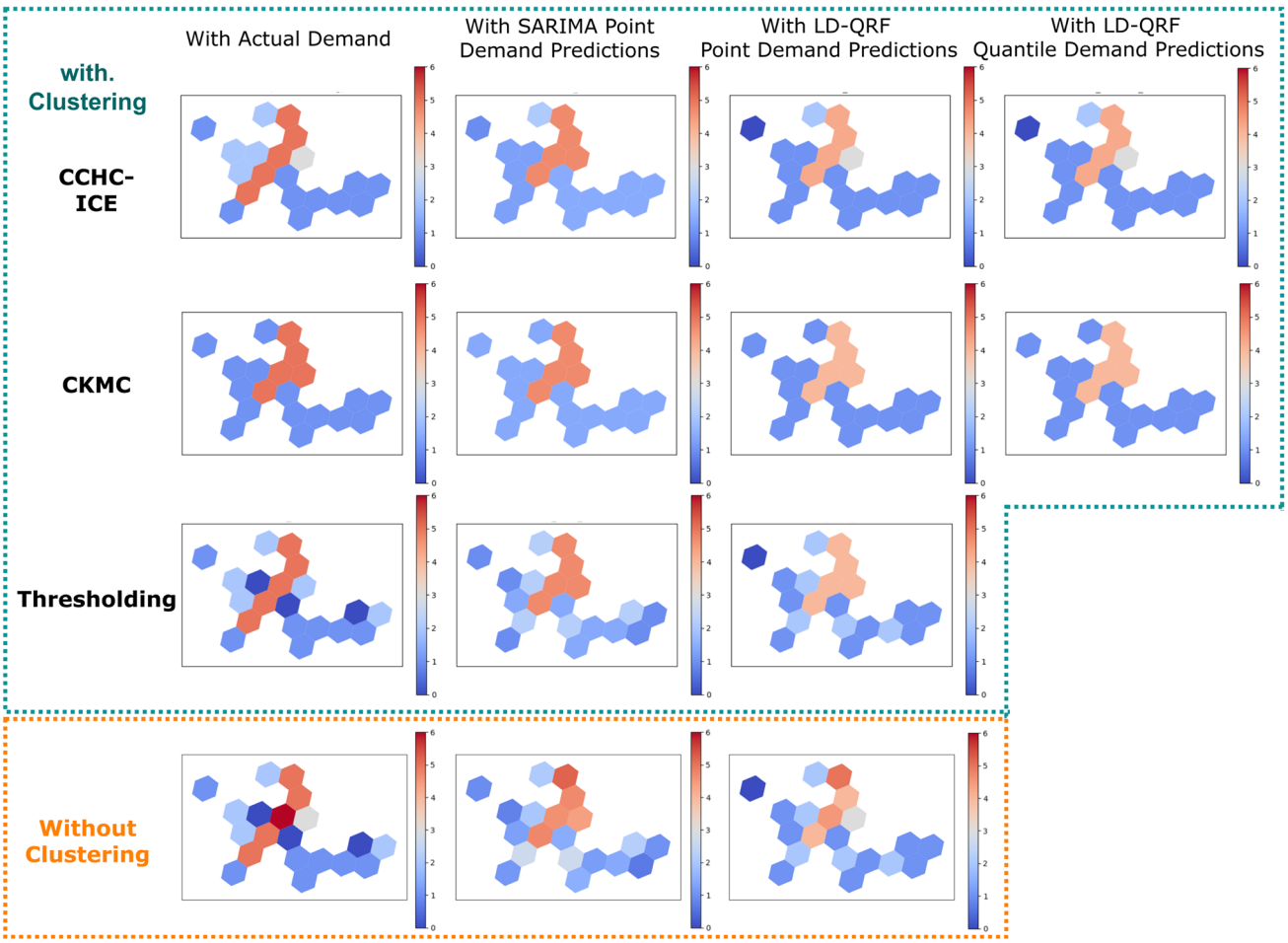

Fig. 12 European case study: heatmap visualization of demand and the corresponding clustering outcomes for 19:00–19:15 on Monday, September 14th. Clusters are derived from three different approaches: Thresholding (ignoring location), CKMC (considering geographical proximity), and CCHC-ICE (enforcing geographical contiguity). Clusters are produced, respectively, with actual demand, SARIMA point predictions, and LD-QRF median and quantile predictions for CKMC and CCHC-ICE methods. The color scale represents demand density, with warmer colors indicating higher order volumes

For model selection, our findings strongly favor LD-EL models in most operational contexts. Compared to classical time-series baselines such as SARIMA/SARIMAX, LD-EL models offer superior scalability, faster retraining, and inherent handling of interacting dual seasonalities (daily and weekly) without requiring bespoke specification. These advantages translate into consistently better point and distributional forecasts, making LD-EL well suited for real-time deployment. By contrast, the LSTM baseline performs competitively on denser, more autocorrelated data

(Taiwanese use case) but underperforms LD-EL on sparser, noisier data (European use case). This pattern aligns with LSTM s dependence on abundant, informative sequences and LD-EL s robustness under weaker signal-to-noise conditions Kourentzes (2013); Rahmani et al. (2023). From a practical perspective, this suggests a data-driven guideline: when demand traces are sparse or irregular, LD-EL is preferable; when sequences are rich and stable, a well-tuned LSTM can be viable. Nevertheless, LD-EL often remains simpler to maintain and scale operationally, making it an attractive default choice in heterogeneous environments.

Prior work has extensively optimized on-demand mobility and logistics operations, including fleet sizing Xue et al. (2021), fleet reallocation Lei et al. (2020), and order matching Chen et al. (2019). However, many of these studies either implicitly predict demand within a dynamic optimization framework without evaluating its accuracy, or rely on the assumption of constant demand arrival rates. Meanwhile, short-term demand forecasting has proven effective for guiding forward-looking policies Grahn et al. (2021). To make this connection concrete in meal delivery context, we include a focused simulation that explicitly consumes 15-min demand forecasts to drive an idle-fleet rebalancing policy. The experimental setup and result analysis is detailed in Appendix F. Using the European case, we compare a prediction-informed relocation policy which steers idle couriers toward zones with higher 15-min expected demand. The prediction-informed relocation policy is compared against a rule-based myopic policy that ignores forecasts. Results show that integrating demand predictions significantly enhances platform efficiency by proactive steering of idle couriers toward the areas with higher demand expectations. The forecast-aware policy reduces average delivery times by over 10%, underscoring that high-frequency (per-15-min) predictions are actionable for real-time rebalancing. Beyond the idle-fleet rebalancing example, operators of meal delivery and other on-demand services can leverage short-term demand predictions to develop forward-looking policies, enhancing resource allocation efficiency and enabling proactive decision-making driven by predicted demand insights.

Short-Term Predict-Then-Cluster in the European Case Study

In the dynamic predict-then-cluster framework, we combine the geographical information and the predicted demand information of pick-up zones as inputs to the proposed CKMC and CCHC-ICE clustering approaches to generated predicted demand hot- and cold-spot insights of the service network. Since the zone units in the Taiwanese use case are too coarse for the intended clustering operations, we focus on the European case for the predict-then-cluster experiment. We use point forecasts from LSTM, QRF, and LD-QRF (the top short-term performers) and quantile forecasts from QRF/LD-QRF to supply demand-uncertainty inputs for clustering.

To quantify the predict-then-cluster performance, we measure the difference between predicted and actual

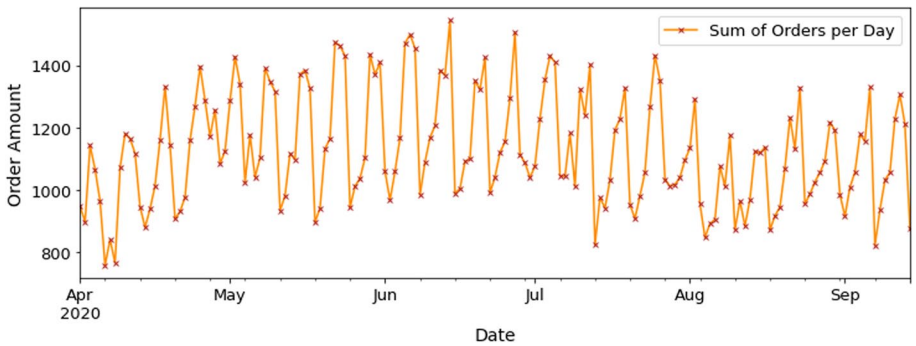

Fig. 13 The daily total number of orders received in the European city from the case study

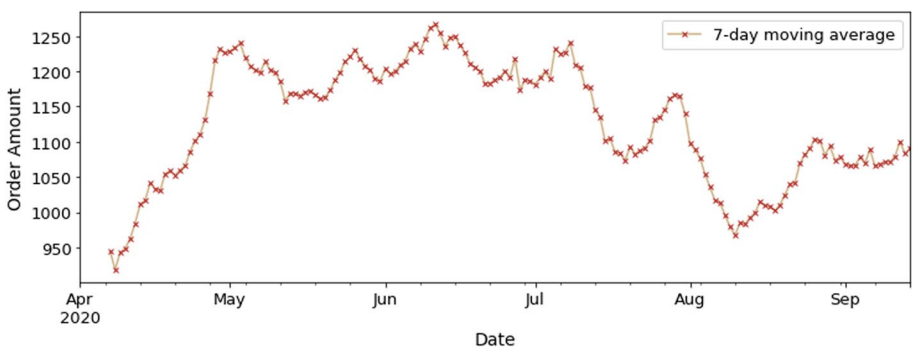

Fig. 14 The 7-day moving-average series of the number of orders received daily in the city

Table 6 The candidate hyperparameter values for demand forecasting ensemble-learning models

RF & LD-RF & QRF & LD-QRF		XGBoost & LD-XGBoost	
n_estimators	50, 75, ..., 175, 200	*n_estimators*	50, 75, ..., 175, 200
max_features	'auto', 'sqrt'	learning_rate	0.1, 0.15, 0.2, 0.25, 0.3
max_depth	3,4,5,6,7	max_depth	3,4,5,6,7
min_samples_split	4,6,8,10	subsample	0.5, 0.75, 1.0
min_samples_leaf	2,3,4,5,10		

We refer to the documentation of model packages by scikit-learn Pedregosa et al. (2011) for an elaborate explanation of the hyperparameters

within-cluster median demand value for each pick-up zone, as detailed in Sect. 5.1. Table 5 summarizes the dynamic predict-then-cluster performance for CKMC and CCHC-ICE, evaluated with MAE and RMSE. Using LD-QRF quantile inputs yields the lowest MAE for both clustering methods, indicating closer alignment with the clusters generated with actual demand. When comparing clustering approaches, CKMC receives marginally lower MAE and RMSE values compared to CCHC-ICE is expected, as CCHC-ICE clusters are required to satisfy more constraints. Compared to the time-series benchmark SARIMA, the inclusion of LD-QRF quantile forecasts has enhanced the performance by 39.4% and 34.5% for CKMC and CCHC-ICE, respectively, measured by MAE.

A key observation is the consistent improvement brought by quantile-based predictions across both clustering strategies. For example, LD-QRF reduces MAE from 0.699 (point prediction inputs) to 0.687 (quantile prediction inputs) under CKMC, and from 0.760 to 0.749 under CCHC-ICE. This reduction demonstrates that incorporating distributional forecasts enhances predict-then-cluster performance, moving closer to the ideal scenario where clustering is based on actual demand.

Overall, the predict-then-cluster performance for both clustering methods reflects the forecasting accuracy trends reported in Tables 2 and 4. Notably, the zone-averaged MAE improvement of LD-QRF over LSTM becomes more pronounced in the clustering context. When quantile-based demand predictions are incorporated, the accuracy gain increases from 6.81% (in pure forecasting evaluation) to 10.48% and 10.41% for CKMC and CCHC-ICE, respectively. This highlights that the LD-EL distributional predictor LD-QRF not only enhances predictive accuracy, but also significantly improves downstream clustering quality under operational constraints.

We conducted a sensitivity analysis in the European case study to assess how the predict-then-cluster stage responds to changes in user-defined operational criteria, including the minimum number of cluster, maximum cluster size, dissimilarity threshold, and importance weights for distance calculation. Holding demand predictions fixed, we varied each criterion within a practical range and evaluated performance. Across settings, the predict-then-cluster performance remains robust, indicating that the framework is flexible and robust to reasonable constraint choices. In practice, we recommend users to carefully select the values of these constraints to reflect downstream operational needs (e.g., dispatch workload and geographic contiguity).

For the deployment of predict-then-cluster framework, the choice between CKMC and CCHC-ICE is application-driven. For exploratory analytics or simulation studies where spatial contiguity is not critical, CKMC offers a simple and effective approach for achieving low aggregation error. However, for operational settings that require compliance with spatial contiguity, service boundaries, or workforce balance constraints, CCHC-ICE is the appropriate choice, as it ensures operationally valid and interpretable zones.

Figure 12 depicts visualized dynamic clustering outcomes for the 19:00 19:15 interval on Monday, 14 September (European case). In this example, we apply three representative dynamic clustering approaches (CCHC-ICE, CKMC, and Thresholding) using actual or predicted demand, including the point predictions from SARIMA, LD-QRF, and quantile predictions from LD-QRF. The non-clustered panels display raw demand heatmaps for reference. In each heatmap, every hexagon (pick-up zone) is colored by the within-cluster median demand (actual or predicted) of its assigned cluster, so zones in the same cluster share a color. Cooler shades (closer to blue) indicate lower within-cluster median demand, and warmer shades (closer to red) indicate higher demand. Clusters with similar medians may appear with similar colors.

Thresholding is a straightforward clustering approach that categorizes pick-up zones based solely on demand, without considering geographical proximity. In this approach, clusters are formed by grouping zones according to the percentiles of predicted or actual demand values. For each heatmap visualized for clusters under thresholding approach, four clusters are formed, representing zones with demand levels in the highest 75% of demand, 50–75% demand, 25–50% demand, and the lowest 25% of demand.

In the example, thresholding identifies the highest demand cluster near the city center using demand predictions. These

methods are better suited for operations requiring only local comparative insights into demand. For instance, meal delivery platforms can use these outcomes to develop fleet rebalancing strategies by locally prioritizing where to deploy its active couriers among several adjacent pick-up zones based on their comparative future demand level. However, the clusters generated by thresholding do not provide a spatially consistent view of the city, limiting their effectiveness for coordinating city-wide fleet rebalancing.

CKMC generates clusters by considering geographic proximity and constraining the minimum cluster size to 3. The optimal number of clusters, ranging from 2 to 6, is automatically selected based on the highest mean silhouette score for each clustering iteration. Using regardless the actual or predicted zonal demand as inputs, each CKMC heatmap is divided into the same two major clusters, highlighting the city's high- and low-demand regions. Notably, the minimum cluster-size constraint does not always benefit the creation of cohesive service neighborhoods. Compared to the case without clustering, some zones within the same cluster are geographically distant from each other and could be more appropriately separated into different clusters.

Unlike thresholding, CKMC preserves geographical proximity among zones within the same cluster while also considering predicted demand levels. This makes CKMC more suitable for proactive operational strategies where regional consistency is needed. For instance, fleet rebalancing can direct idle couriers toward the direction of high-demand regions in a coordinated fashion.

CCHC-ICE clustering optimizes demand similarity while enforcing geographical contiguity and user-defined operational constraints (in this case: at least 3 clusters, no more than 9 zones per cluster, and a dissimilarity threshold of 9), through iterative constraint enforcement. In most cases, CCHC-ICE identifies a central high-demand cluster surrounded by medium-demand clusters. When applied to predicted demand, the CCHC-ICE also accurately detects the medium-demand zone on the right (highlighted in gray). Compared to CKMC and thresholding methods, CCHC-ICE produces more informative and constraint-consistent outputs, effectively highlighting relatively lower predicted demand in specific pick-up zones compared to their high-demand surroundings.

Operationally, CCHC-ICE delivers contiguous clusters that rigorously adhere to predefined constraints, enabling platforms to anticipate future demand landscapes with greater granularity and reliability. Its flexibility in defining user-specified conditions makes it suitable for scenarios requiring contiguity and tailored cluster configurations. In real-time management, these dynamically generated clusters can serve as operational units to enhance the scalability of optimization tasks, such as order bundling and delivery routing, ultimately improving delivery efficiency. For example, a courier can be assigned a route that consolidates multiple orders within a cluster with high internal accessibility. Conversely, if zones within a cluster lack connectivity, inefficiencies may arise. These applications underscore the importance of integrating accessibility considerations into clustering strategies.

Conclusion

To maintain resilience in service quality, meal delivery platforms must prioritize the timely fulfillment of user orders despite the complexities of stochastic order arrivals and fluctuating demand patterns influenced by factors, such as weather, holidays, and time of day. These challenges are compounded by the dynamic movement of couriers, which often leads to imbalances in supply and demand across the service network. Efficient real-time operations are essential for addressing these imbalances, and generating accurate short-term demand predictions is a critical first step. Moreover, the computational complexity of optimizing operations in real time often presents a bottleneck. By clustering service zones based on predicted demand, the complexity of decision-making can be significantly reduced, enabling faster and more efficient operational strategies.

To address these challenges, this study proposes a computationally efficient short-term predict-then-cluster framework, which produces high-quality demand forecasts for each service zone and generates dynamic cluster insights to identify future demand hotspots. The demand forecasting component introduces LD-EL models to generate accurate and robust point forecasts and distributional predictions without underlying demand distribution assumptions. These models leverage local demand lags alongside temporal and contextual features to jointly capture complex demand patterns. The dynamic clustering component introduces two constrained clustering approaches, CKMC and CCHC-ICE, to group service zones based on predicted demand while satisfying geographical and operational constraints. Experiments are conducted over the empirical order data from a European and a Taiwanese city, where we identified unique temporal and spatial demand patterns in meal delivery services of these cities. Forecasting experiments evaluate the predictive performance of various models across sparse (European) and dense (Taiwanese) data regimes. Results indicate that LD-EL models achieve high accuracy and robustness, particularly under sparse and zero-inflated demand conditions. Distributional predictions further improved clustering outcomes by incorporating demand uncertainties, enhancing the robustness of the clustering process.

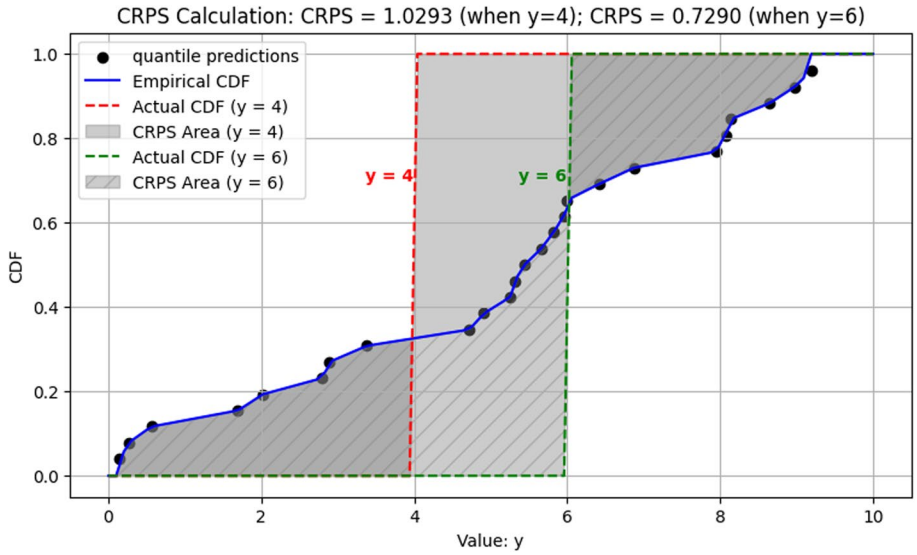

Fig. 15 Visualized demonstration of CRPS calculation for a group of 25 sample quantile predictions (corresponding to quantile values $q = 0.04, 0.08, \cdots 0.96$) and two different target values, $y = 4$ and $y = 6$

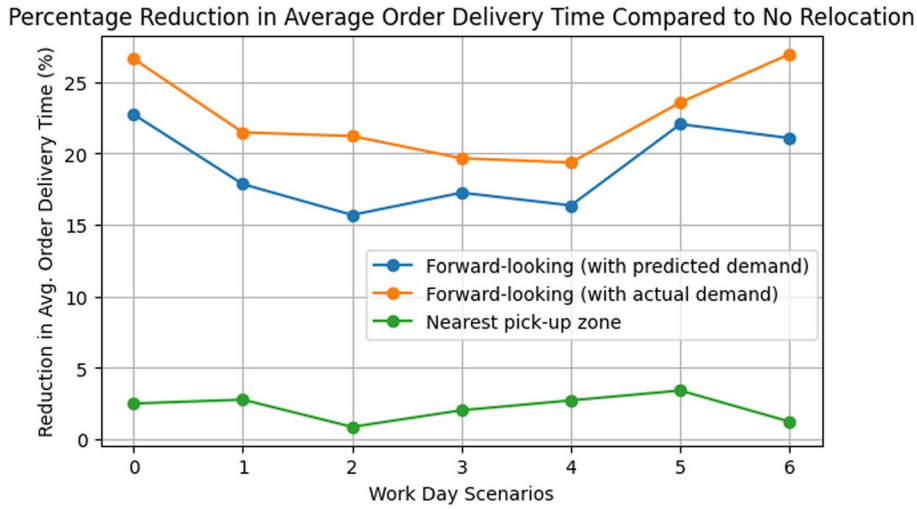

Fig. 16 Impact of different relocation policies on the reduction of average order delivery time. The figure shows the percentage reductions in the average delivery time per order compared to no relocation for three policies: forward-looking idle courier relocation with predicted demand, forward-looking relocation with actual demand, and nearest pick-up zone relocation. Results are shown for seven different workday scenarios using historical order data from the European case study, with 100 simulations conducted for each scenario

From a managerial perspective, the findings of this study provide valuable guidance for selecting forecasting models tailored to the availability of training data, computational resources, and specific operational requirements. The analysis emphasizes the importance of incorporating temporal and contextual features to enhance short-term forecasting accuracy, effectively capturing periodic demand patterns and recent variations. The integration of distributional predictions addresses demand uncertainties, offering robust tools to support stochastic optimization in real-time decision-making. By dynamically clustering service zones, the predict-then-cluster framework delivers actionable insights into near-future demand dynamics. Furthermore, its adaptability also makes it applicable to other on-demand urban logistics and passenger mobility services, contributing to more sustainable and efficient city operations.

Future work should examine how short-term probabilistic demand forecasts and dynamically generated, contiguity-aware clusters can be coupled with real-time decision making. A natural next step is to feed predictive clusters into sequential optimization (e.g., dispatching, rebalancing, and routing) and evaluate end-to-end performance gains. This integration also opens the door to learning clustering hyperparameters from operational outcomes (e.g., via bilevel or decision-focused tuning) rather than fixing them a priori. Nevertheless, for downstream operations that require uncertainty estimates with guaranteed coverage across varying demand conditions, Conformalized Quantile Regression (CQR) Romano et al. (2019) can be further integrated to recalibrate the prediction intervals derived from QRF-based forecasts. We also note that recent deep learning architectures, such as Transformers Zerveas et al. (2021) and DeepAR Salinas et al. (2020), can be extended to produce probabilistic forecasts by modeling conditional distributions directly, for example through parametric likelihood functions or quantile output heads. Especially, when the data are dense

and exhibit regular temporal patterns, such models could serve as effective alternatives in the forecasting stage of the predict-then-cluster framework. Hence, future research can also extend the predict-then-cluster use case to on-demand application with denser time-series patterns, exploring the usage of advance deep learning predictors to capture spatial and temporal dependencies in these complex demand series.

Appendix A: Supplementary Material for the European Use Case

A.1 Data Anonymization Process for Synthetic Data Generation

To ensure the anonymity of the dataset while preserving its utility for reproducible research, a data anonymization process is applied to the original meal delivery demand data to generate a synthetic dataset for the European use case. The process consists of two parts: spatial transformation and demand value standardization. Together, these steps safeguard sensitive information, ensuring that neither spatial nor demand-related details can be used to deduce the identity of the city or the specific operational data of the platform, while still enabling meaningful analysis of the anonymized data.

First, spatial transformation is performed over the hexagonal zones (defined by the H3 geospatial indexing system) to obscure the exact geographical location of the studied European city. Specifically, the center coordinates of all hexagonal zones were uniformly shifted by a fixed offset. This transformation preserves the relative spatial relationships between zones while ensuring that the original city's location cannot be identified from the data.

Second, to protect the confidentiality of demand patterns, the order demand values, which are aggregated into 15-min intervals for each pick-up zone in the forecasting experiments, were standardized. The standardization process involves adjusting the values to have a mean of zero and a standard deviation of one for each pick-up zone. This ensures that temporal demand trends remain comparable across zones, while the absolute demand volumes cannot be traced back to the original data.

A.2 Supplementary Data Analysis

Figure 13 shows the daily total number of orders received in this European city from April 1^{st}, 2020 to September 14^{th}, 2020. The visualized daily demand time-series indicates a recurrent weekly demand pattern. To further inspect potential trends in the time-series, we plot the 7-day moving-average series of the daily order data with the dates as the x-axis in Fig. 14. The moving averages are calculated as

$$MA_t = \frac{1}{7}\sum_{i=t-6}^{t} x_i, \qquad t = 7, 8, \ldots, T,$$

where t is the index of the date, which starts from 7 and ends at $T = 166$, the last day covered by the data. Figure 14 shows a clear upward trend from the beginning of April to the start of May, after which the moving averages stay at a comparatively constant level between May and July before the dive-and-rise patterns between July and September. The moving-average plot suggests a potentially significant trend in the daily order time-series. These shifts in demand may be caused by the COVID-19-related measures implemented during the time. We cannot yet conclude a yearly seasonality pattern exists for the demand time-series, since the available data only cover a timeline for less than half of a year. A more extended time-series is needed to account for potential annual seasonal patterns in the future.

Based on the analysis of the dual-seasonal pattern discussed in Sect. 3.3, we can further decompose each business day into 5 periods: breakfast (10:30–11:00), lunch (11:00–14:00), afternoon (14:00–17:00), dinner (17:00–21:00), and night (21:00–21:30). During breakfast time, there is an average of 0.14 orders received per 15 min with a standard deviation (std) of 0.26. An average of 0.64 (std: 0.73) orders were received during 15-min intervals for lunch times, 0.68 (std: 0.62) for afternoon times, 2.58 (std: 1.47) for dinner times, and 1.0 (std: 0.86) for night times, respectively.

Appendix B: Candidate Hyperparameters Values for Demand Forecasting Model Tuning

The hyperparameter values we try for the model tuning of ensemble-learning predictors are listed in Table 6. In all zone-specific ensemble-learning models, hyperparameters are selected by minimizing MAE under a tenfold expanding, forward-chaining cross-validation scheme that preserves temporal order and avoids leakage. For each model's search space, we run a randomized grid search with a fixed budget of 50 trials per model per grid. During tuning of QRF/LD-QRF, point forecasts are computed as the mean of terminal-node leaves to define the MAE objective. Because models are trained independently per zone, the optimal settings are zone-dependent. To ensure reproducibility without overloading the manuscript, we release the full per-zone hyperparameter choices and cross-validation scores as CSV files in the code repository.

We tuned the LSTM hyperparameters using Optuna with a tenfold time-aware cross-validation scheme to prevent temporal leakage. In each trial, Optuna sampled a learning rate from a log-uniform prior (10^{-4} - 10^{-2}), a hidden size from {16, 32, 64}, and a dropout rate from [0.0, 0.3]. For each fold, we trained the LSTM for up to 3 epochs with a batch size of 64 and recorded the validation loss. The trial's objective value was the mean validation loss (MAE) across the tenfold, and the study minimized this quantity over 10 trials to select the configuration with the best average out-of-sample performance. The final selected learning rate is 0.0025, hidden size is 64, and dropout rate is 0.005.

Appendix C: Python Packages Used for Model Implementation

For model implementation, we use package pmdarima Smith et al. (2017) for SARIMA and SARIMAX models, a package Skorupa (2020) written by Skorupa for TBATS model, package scikit-learn Pedregosa et al. (2011) for RF and XGBoost model, package quantile-forest quantile-forest. GitHub (2023) for QRF model, and package pytorch-forecasting Beitner (2020) for LSTM.

Appendix D: Pseudocode Implementation of the CCHC-ICE Framework with Example User-Specified Constraints

This appendix presents the pseudocode for the Contiguity Constrained Hierarchical Clustering with Iterative Constraint Enforcement (CCHC-ICE) framework introduced in Sect. 4.2.2. Three kinds of user-specified constraints are introduced in this example implementation: the cluster dissimilarity threshold, minimum number of clusters, and maximum cluster size. Algorithm 1 outlines the main iterative clustering process, explaining the inputs and outputs of the framework. Algorithm 2 provides the helper functions necessary for enforcing geographical contiguity and user-specified constraints, updating adjacency relationships, and managing cluster operations. Together, these pseudocodes detail the implementation of the CCHC-ICE framework, demonstrating how constraints are dynamically enforced during the clustering process, as applied in the European case study. This serves as a guide for implementation and replication.

Algorithm 1 Contiguity constrained hierarchical clustering with iterative constraint enforcement (CCHC-ICE)

Standard Input:

- **Predicted Demand Features:** $\mathbf{X}_t$, an $N \times M$ matrix, where N is the number of zones and M the length of predicted demand inputs, at current time step t;
- **Zone-wise Adjacency Matrix:** $\mathbf{C}_z$, geographical adjacency among zones;
- **Feature Importance Vector*:** w, an $M \times 1$ weight vector where w_j correspond to the feature importance for feature j for the calculation of similarity matrix (only when $M \geq 2$).

User Specified Constraints Input:

- **Minimum Number of Clusters:** $K_{\min}$, a hyperparameter which indicates the minimum number of clusters. Hierarchical clustering stops when $K \leq K_{\min}$;
- **Maximum Size per Cluster :** $s_{\max}$, a hyperparameter that controls the maximum number of pickup zones that can be clustered together;
- **Cluster-distance Threshold :** $D_{\max}$, the maximum distance value between clusters to be merged.

Output:

- **Cluster labels:** L, an $N \times 1$ vector recording the final labels for all zones;
- **Dictionary of clusters:** $\Omega = \{\omega_1, \omega_2, \cdots, \omega_K\}$, where each cluster element ω contains all the pickup zones belonging to this cluster, i.e., with label l.

function INITIALIZE HELPER VARIABLES(N):
    $\Omega \leftarrow \{\omega_1, \omega_2, \cdots, \omega_N\}$, where each pickup zone is an individual cluster initially.
    $K \leftarrow N$, the initial number of cluster is the number of zones.
    Set *contiguity violation* $\leftarrow$ False
    return Ω, K, *contiguity violation*
end function

function CCHC-ICE($\mathbf{X}_t, \mathbf{C}_z, K_{\min}, s_{\max}$)
    Ω, K, *contiguity violation* $\leftarrow$ INITIALIZE HELPER VARIABLES(N)
    while $K \geq K_{\min}$ and *contiguity violation* = False **do**
        $\mathbf{D}_s \leftarrow$ CONSTRAINED DISTANCE MATRIX UPDATE($\mathbf{C}_z, \Omega, \mathbf{X}_t$)
        $\Omega' \leftarrow$ AGGLOMERATIVE HIERARCHICAL CLUSTERING(()$\mathbf{D}_s, K, D_{\max}$)
        contiguity violation $\leftarrow$ CHECK-CONTIGUITY-CONDITIONS(()$\Omega', \mathbf{C}_e, \mathbf{C}_p$)
        if *contiguity violation* = True **or** $\Omega = \Omega'$ **then**
            break
        else
            $\Omega \leftarrow \Omega'$
            $K \leftarrow K - 1$
        end if
    end while
    return Ω
end function

Algorithm 2 Helper functions for CCHC-ICE

function WITHIN CLUSTER CONNECTIVITY(Ω):
    Initialize zone connectivity matrix: $\mathbf{C}_e \leftarrow$ zero-matrix of shape $N \times N$
    for $i \leftarrow 0$ **to** N **do**
        for $j \leftarrow 0$ **to** N **do**
            if $i \neq j$ and zone i and zone j in the same cluster according to Ω **then**
                $\mathbf{C}_e[i, j] \leftarrow 1$
            else
                $\mathbf{C}_e[i, j] \leftarrow 0$
            end if
        end for
    end for
    return $\mathbf{C}_e$
end function

function CLUSTER ADJACENCY CONSTRAINED CONNECTIVITY($\mathbf{C}_z$, Ω, s_{max}):
    Initialize zone connectivity matrix: $\mathbf{C}_p \leftarrow$ zero-matrix of shape $N \times N$
    $\Psi = [\psi_1, \cdots, \psi_\omega]$ where ψ_w is the list of feasible contiguous clusters for a cluster c_w, based on the zone-wise adjacency matrix $\mathbf{C}_z$ and maximum cluster size constraint s_{max}.
    for i in 0 to N **do** $c_w \leftarrow$ current cluster of zone i
        for j in 0 to N **do**
            if $i \neq j$ and zone j belongs to a contiguous cluster defined in ψ_w **then**
                $\mathbf{C}_p[i, j] \leftarrow 1$
            else
                $\mathbf{C}_p[i, j] \leftarrow 0$
            end if
        end for
    end for
    return $\mathbf{C}_p$
end function

function CONSTRAINED DISTANCE MATRIX UPDATE($\mathbf{C}_z, \Omega, \mathbf{X}_t$):
    Initialize contiguity constrained distance matrix: $\mathbf{D}_z \leftarrow N \times N$ zero-matrix
    $\mathbf{C}_e$(Existing connection matrix) $\leftarrow$ WITHIN CLUSTER CONNECTIVITY(Ω)
    $\mathbf{C}_p$(Potential connection matrix) $\leftarrow$ CLUSTER ADJACENCY CONSTRAINED CONNECTIVITY($\mathbf{C}_z$, Ω)
    for i in 0 to N **do**
        for j in 0 to N **do**
            if $\mathbf{C}_e[i, j] = 1$ **then**
                # if connection is reserved since two zones already in the one cluster,
                $\mathbf{D}_z[i, j] \leftarrow 0$
            else if $\mathbf{C}_p[i, j] = 1$ **then**
                # if connection is possible since they belong to two adjacent clusters,
                $\mathbf{D}_z[i, j] \leftarrow$ *distance*$(\mathbf{X}_t[i, :], \mathbf{X}_t[j, :])$, according to Eq.(1).
            else
                # if the connection doesn't satisfy contiguity constraints,
                $\mathbf{D}_z[i, j] \leftarrow +\infty$ 44
            end if
        end for
    end for
    return $\mathbf{D}_z$
end function

Appendix E: Evaluation Metrics Formulation

E.1 Point Forecasting Metric: MAE, RMSE, and RMSLE

Given the total number of prediction equals to T for each pick-up zone i, the actual order demand vector y^i, and corresponding predicted demand vector $\hat{y}^i$, the mean absolute error for pick-up zone i is calculated as

$$\text{MAE}^i = \frac{\sum_{t=1}^{T} |y_t^i - \hat{y}_t^i|}{T}. \tag{E1}$$

The root mean squared error is calculated as

$$\text{RMSE}^i = \sqrt{\frac{\sum_{t=1}^{T} (y_t^i - \hat{y}_t^i)^2}{T}}. \tag{E2}$$

Although Mean Absolute Percentage Error (MAPE) is a common metric for other demand forecasting problems, it is not suitable for our case. Our data analysis shows that it is common for many pick-up zones to receive no orders during several 15-min time windows during the off-peak hours, which means that there are many zero values on the demand time-series. Hence, using MAPE directly will encounter the divided-by-zero problem. We utilize an alternative approach, Root-Mean-Square Logarithmic Error (RMSLE), as a relative error measure that does not suffer the zero-demand problem. It is formulated as

$$\text{RMSLE}^i = \sqrt{\frac{\sum_{t=1}^{T} (log(\hat{y}_t^i + 1) - log(y_t^i + 1))^2}{T}}. \tag{E3}$$

RMSLE is a relative error measure that does not penalize large residuals when both predicted and actual values are large. Moreover, it penalizes underestimates more than overestimates. This property is suitable for cases where having extra inventory or supply is preferable to failing to meet the demand. This approach has also been applied to other demand forecasting studies of on-demand shared mobility services Qiao et al. (2021); Jiménez-Bravo et al. (2021).

E.2 Distributional Forecasting Metric: MCRPS

CRPS quantifies the discrepancy between the predicted cumulative distribution function (CDF) resulting from the forecast and the 'actual' CDF. The 'actual' empirical CDF of the scalar observation y is simply represented as a step function $\mathbf{1}(\hat{y} \geq y)$ that returns 1 where x is greater or equal to the observation and returns 0 otherwise. Following the instantaneous form formulation in Gneiting and Raftery Gneiting and Raftery (2007), CRPS is defined as:

$$\text{CRPS}(F, y) = \int_{\mathbb{R}} \left[F(\hat{y}) - \mathbf{1}(\hat{y} \geq y)\right]^2 d\hat{y}. \tag{E4}$$

By taking the integral over the squared difference between the target and estimated distributions, CRPS values are non-negative, ranging from 0 to $+\infty$. In our study, nonparametric forecasting method QRF only provides quantile predictions for a given quantile value $q_\kappa \in [0, 1]$, instead of a continuous distribution. Therefore, an empirical CDF $\hat{F}(\hat{y})$ needs to be estimated from a collection of quantile predictions $Q(q_\kappa)$ corresponds to quantiles $q_1, q_2, \cdots, q_K$

$$\hat{F}(\hat{y}) = \begin{cases} 0, & \text{if } \hat{y} \leq Q(q_1), \\ q_\kappa + \frac{(\hat{y} - Q(q_\kappa))}{(Q(q_{\kappa+1}) - Q(q_\kappa))} \cdot (q_{\kappa+1} - q_\kappa), & \text{if } Q(q_\kappa) \leq x \leq Q(q_{\kappa+1}), \\ 1, & \text{if } \hat{y} \geq Q(q_K). \end{cases} \tag{E5}$$

Figure 15 demonstrates how CRPS is estimated by measuring the discrepancy between the estimated empirical CDF, which is fitted from a group of quantile predictions $Q(q_\kappa)$, and the actual empirical CDF generated by a step function from a scalar observation y. In this study, we select $K = 9$ quantile predictions corresponding to quantile values $q = 0.1, 0.2, \cdots, 0.9$ to calculate $\text{CRPS}_t^i(\hat{F}_t, y_t)$ each pick-up zone i for time interval t. We only consider a limited number of quantiles for the evaluation, because both QRF and LD-QRF generate quantile predictions from historical observations, while the historical demand values typically fall within a narrow range for most time windows. The mean CRPS (MCPRS) for zone i will be computed as the average among all the forecasting time steps

$$MCRPS^i = \frac{\sum_{t=1}^{T} \text{CRPS}_t^i(\hat{F}_t, y_t)}{T}. \tag{E6}$$

A lower MCRPS value indicates that the quantile predictions are closer to the underlying actual demand distribution in general, therefore suggesting a better distributional forecasting performance by the model.

Appendix F: Simulation Study on Utilizing Short-Term Demand Forecasting to Enhance Meal Delivery System Efficiency

In line with our discussion on the potential implementation of the proposed demand forecasting model and its policy-making benefits, we demonstrate how high-quality short-term demand predictions can support real-time operations of on-demand meal delivery services through a simulation experiment. Specifically, we introduce an idle-fleet rebalancing strategy to relocate idle couriers toward under-supplied regions in the future based on short-term demand predictions

of the service network. As an extension of the European case study, a simulation study is designed for the city and performed with the historical order data as input. The goal is to assess how the overall delivery efficiency is impacted by the implementation of this prediction-informed policy. The remainder of this section is organized as follows: F.1 describes the design of the meal delivery fleet simulator, F.2 defines the forward-looking idle courier relocation policy, and F.3 presents the detailed specification of simulations and analyzes the results.

F.1 Design of the Meal Delivery Simulations

In this section, we describe the design of a simulator that models the movement of an active fleet within a meal delivery service network over the course of a day. The simulator dynamically assigns delivery and relocation tasks to couriers while tracking orders, tasks, and courier statuses. Performance indicators are embedded to assess the impact of different real-time idle-fleet rebalancing strategies on overall delivery efficiency.

The meal delivery service network of a city is represented by a set of individual zones in the simulation, denoted as $Z : \{z_1, z_2, \ldots, z_M\}$. As described in Sect. 3.1, each meal delivery order is associated with a pick-up zone and a destination zone, representing the restaurant and household locations for the order.

The simulator begins by initializing a fleet of J couriers ($C : \{c_1, c_2, \ldots, c_J\}$). Couriers are categorized as either idle or busy. An idle courier is available for a delivery or relocation assignment, while a busy courier is engaged in an ongoing task. Upon completing a task, a courier becomes idle at the destination location and is available for a new assignment. Throughout the simulation, the number of idle couriers in each zone varies.

Orders arrive in the system dynamically during the simulation. An order o only arrives in the system upon arrival time t_a^o, and is unknown to the meal delivery platform beforehand. The pick-up and drop-off zone locations of the order are denoted as z_p^o and z_d^o. Orders are assumed to be ready for pick-up from the restaurant as soon as they arrive in the platform.

At each time step, the simulation updates its dynamics. First, the simulator goes through the assignment status of busy couriers, releases who just completed the tasks at this time step, and updates them to be idle. Next, the simulator checks whether one or more orders have arrived to the system. The simulator then checks for new orders, assigning each to the closest available courier from the order's pick-up location. Upon assignment, the delivery time of this task is registered as the sum of the pick-up travel time from this courier's current location to the pick-up zone, delivery travel time from the pick-up zone to the destination zone, plus the service time for picking up and delivering the order. If no courier is available for assignment, the order is rejected. Finally, the system looks for couriers who have stayed idle for a consecutive period beyond a predefined threshold. These couriers may be assigned to a relocation task following the suggestions from the idle-fleet rebalancing algorithm.

F.2 Idle Courier Relocation Policies

Relocating idle couriers is essential for maintaining an efficient fleet, minimizing downtime, and balancing supply and demand across urban areas. A classic approach used by meal delivery platforms involves relocating idle couriers to the nearest pick-up zone with restaurants. However, this method often ignores the overall supply and demand dynamics of the network, which can lead to fleet imbalances across the city and reduced efficiency.

With short-term demand predictions, a forward-looking relocation policy can be implemented to improve fleet efficiency. This policy takes into account both idle courier supply and short-term demand forecasts to anticipate supply–demand imbalances across the network. And it addresses them by proactively moving idle couriers to areas expected to have higher demand.

The forward-looking relocation policy contains two parts: local supply–demand imbalance inspection and relocation assignment. The anticipated future supply–demand deficit $\hat{\Delta}(z)$ is calculated as $\hat{\Delta}(z) = s(z) - \hat{d}(z)$, where $s(z)$ is the current supply of idle couriers at zone z and $\hat{d}(z)$ its predicted demand for the next time interval. For an idle courier ready for relocation, $\hat{\Delta}(z)$ is first computed for the candidate zones Z, which include the courier's current zone and its neighboring adjacent zones. The zone from Z with the highest estimated future supply–demand deficit is selected. If this is the courier's current zone, they remain in place. Otherwise, the courier is assigned to relocate to the selected zone.

F.3 Specifications and Results of Simulation Experiment

We define seven scenarios, one for each day (Monday–Sunday) in the testing week of the European use case. In scenario d, the simulator replays the actual order transactions from day d, including timestamps and locations information. All other parameters, such as fleet size, routing logic, and operational rules, are held constant across scenarios. This setup ensures that the only varying factor between scenarios is the empirical demand trace specific to each day. As a result, any observed performance differences can be directly attributed to day-specific demand dynamics, allowing for a controlled and interpretable evaluation of system behavior under varying real-world conditions.

The simulation adopts the service network topology from the European use case, consisting of a total of 50 zones. Of these, 20 zones can serve as pick-up zones, while all zones can function as destination zones. The simulation updates at each time step, corresponding to 1 min of the day. The simulation's start and end times match the operating hours of the meal delivery service in the European use case. Orders arrive within the simulation according to their actual recorded times in the historical transaction data, ensuring realistic demand dynamics. Couriers are assumed to be active for the entire duration of each simulated day. To simplify the design, the travel times between any two zones in the network are calculated based on the distance derived from a grid traversal function provided by the geo-indexing package H3 for the hexagonal zones defined in the use case Uber: H3 (2018). The travel time between adjacent zones is computed using the distance between the centers of the adjacent grids and the local average biking speed. The service times for pick-up at restaurants and delivery at households are assumed to be both 3 min, for each order. The fleet size J is fixed at 30 couriers for each simulation, where the courier is initialized at a randomly selected zone as their starting location. Finally, the courier's idle duration threshold for suggesting relocation is set to 5 min.

To assess the impact of the *forward-looking* idle courier relocation policy on system performance, we include two benchmarks: *nearest pick-up zone* and *forward-looking (with actual demand)* idle courier relocation policies. The first benchmark is the myopic policy that relocates idle couriers to the nearest pick-up zone. By comparing to this benchmark, we aim to compare the effectiveness of incorporating predictive insights into fleet rebalancing. The second benchmark replaces the predicted demand $\hat{d}(z)$ by the actual future demand $d(z)$ in the anticipated supply–demand deficit $\hat{\Delta}(z)$ calculations. The inclusion of this benchmark helps to evaluate how forecasting errors affect the rebalancing efficiency.

Delivery speed is a key performance indicator for on-demand meal delivery services. Short delivery times are crucial for maintaining customer satisfaction. We conducted simulations for each workday scenario to evaluate the effectiveness of three different idle courier relocation policies in reducing average order delivery times, compared to no relocation. Each policy was tested with 100 simulations per scenario to ensure consistency and reliability of results. Figure 16 shows that both forward-looking policies consistently achieved significantly greater reductions in average delivery time per order compared to the nearest pick-up zone benchmark. The forward-looking policy with demand predictions performed almost as well as the policy using actual demand. These results highlight that accurate demand anticipation is crucial for minimizing delivery times, with even predicted demand providing significant efficiency gains. In contrast, the traditional heuristic approach of nearest pick-up zone relocation led to only minor improvements. In addition, the simulation outcomes suggest that average order rejection rates remained consistent regardless of the relocation policy applied, given the same fleet size. The simulation results underscore the value of proactive and forward-looking policy-making with short-term demand predictions for on-demand meal delivery services.

Author Contributions J.C developed the methods, coded, implemented, validated, and wrote both original and revised manuscripts. S.S.A conceptualized the idea, methodological developments, validation, supervision, writing the original, and the revised version.

Data Availability To facilitate applications and enhance reproducibility, we share the coding implementation, example datasets of the empirical case studies, and a comprehensive documentation with this publication via GitHub Repository: https://github.com/RinaPiggy/predict-then-cluster-meal-delivery

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Aghabozorgi S, Shirkhorshidi AS, Wah TY (2015) Time-series clustering-a decade review. Inf Syst 53:16–38

Ahuja K (2021) Ordering in: The rapid evolution of food delivery. https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/ordering-in-the-rapid-evolution-of-food-delivery. Accessed 23 Nov 2021

Albrecht T, Rausch TM, Derra ND (2021) Call me maybe: methods and practical implementation of artificial intelligence in call center arrivals' forecasting. J Bus Res 123:267–278

Alisoltani N, Zargayouna M, Leclercq L (2020) A sequential clustering method for the taxi-dispatching problem considering traffic dynamics. IEEE Intell Transp Syst Mag 12(4):169–181

Alisoltani N, Ameli M, Zargayouna M, Leclercq L (2022) Space-time clustering-based method to optimize shareability in real-time ridesharing. PLoS ONE 17(1):0262499

Assylbekov Y, Bali R, Bovard L, Klaue C (2023) Delivery hero recommendation dataset: a novel dataset for benchmarking recommendation algorithms. Proceedings of the 17th ACM conference on recommender systems. ACM Digital Library, New York, pp 1042–1044

Barrio-Hernandez I, Yeo J, Jänes J, Mirdita M, Gilchrist CL, Wein T, Varadi M, Velankar S, Beltrao P, Steinegger M (2023) Clustering predicted structures at the scale of the known protein universe. Nature 622(7983):637–645

Basu S, Davidson I, Wagstaff K (2008) Constrained clustering: advances in algorithms, theory, and applications. Chapman and Hall/CRC, Boca Raton, FL

Beitner J, Contributors (2020) PyTorch forecasting: time series forecasting with PyTorch. [Online; accessed 2025-10-20]. https://pytorch-forecasting.readthedocs.io/

Bertsimas D, Kallus N (2020) From predictive to prescriptive analytics. Manage Sci 66(3):1025–1044

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Caggiani L, Camporeale R, Ottomanelli M (2017) A dynamic clustering method for relocation process in free-floating vehicle sharing systems. Transport Res Proc 27:278–285

Carvalho DV, Pereira EM, Cardoso JS (2019) Machine learning interpretability: a survey on methods and metrics. Electronics 8(8):832
Chavent M, Kuentz-Simonet V, Labenne A, Saracco J (2018) Clustgeo: an R package for hierarchical clustering with spatial constraints. Comput Stat 33(4):1799–1822
Chen J (2020) The influence of environmental factors on college students' online catering ordering. E3S web of conferences, vol 145. EDP Sciences, Les Ulis, p 01012
Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. Proceedings of the 22nd Acm Sigkdd international conference on knowledge discovery and data mining. ACM Digital Library, New York, pp 785–794
Chen L, Zhang D, Wang L, Yang D, Ma X, Li S, Wu Z, Pan G, Nguyen T-M-T, Jakubowicz J (2016) Dynamic cluster-based over-demand prediction in bike sharing systems. Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing. ACM Digital Library, New York, pp 841–852
Chen Y, Qian Y, Yao Y, Wu Z, Li R, Zhou Y, Hu H, Xu Y (2019) Can sophisticated dispatching strategy acquired by reinforcement learning?—a case study in dynamic courier dispatching system. arXiv preprint arXiv:1903.02716
Chiang W-C, Russell RA, Urban TL (2011) Forecasting ridership for a metropolitan transit authority. Transport Res Part A Policy Practice 45(7):696–705
Côme E (2024) Bayesian contiguity constrained clustering: spanning trees and dendrograms. Stat Comput 34(2):64
Crivellari A, Beinat E, Caetano S, Seydoux A, Cardoso T (2022) Multi-target cnn-lstm regressor for predicting urban distribution of short-term food delivery demand. J Bus Res 144:844–853
Dai H, Ge L, Liu Y (2020) Information matters: an empirical study of the efficiency of on-demand services. Inf Syst Front 22(4):815–827
Davis N, Raina G, Jagannathan K (2016) A multi-level clustering approach for forecasting taxi travel demand. 2016 IEEE 19th international conference on intelligent transportation systems (ITSC). IEEE, Geneva, pp 223–228
Duque JC, Anselin L, Rey SJ (2012) The max-p-regions problem. J Reg Sci 52(3):397–419
Feng S, Chen H, Du C, Li J, Jing N (2018) A hierarchical demand prediction method with station clustering for bike sharing system. 2018 IEEE third international conference on data science in cyberspace (DSC). IEEE, Geneva, pp 829–836
Fildes R, Ma S, Kolassa S (2019) Retail forecasting: research and practice. Int J Forecast 38(4):1283–1318
Gançarski P, Dao T-B-H, Crémilleux B, Forestier G, Lampert T (2020) Constrained clustering: current and new trends. A guided tour of artificial intelligence research: volume II: AI algorithms. Springer, Cham, Switzerland, pp 447–484
Ghalehkhondabi I, Ardjmand E, Young WA, Weckman GR (2019) A review of demand forecasting models and methodological developments within tourism and passenger transportation industry. J Tourism Futures 5(1):75–93
Gneiting T, Raftery AE (2007) Strictly proper scoring rules, prediction and estimation. J Am Stat Assoc 102(477):359–378
Grahn R, Qian S, Hendrickson C (2021) Improving the performance of first-and last-mile mobility services through transit coordination, real-time demand prediction, advanced reservations, and trip prioritization. Transport Res Part C Emerg Technol 133:103430
Guénard G, Legendre P (2022) Hierarchical clustering with contiguity constraint in R. J Stat Softw 103:1–26
Guo D (2008) Regionalization with dynamically constrained agglomerative clustering and partitioning (redcap). Int J Geogr Inf Sci 22(7):801–823
Guo Z, Yu B, Shan W, Yao B (2023) Data-driven robust optimization for contextual vehicle rebalancing in on-demand ride services under demand uncertainty. Transport Res Part C Emerg Technol 154:104244
He F, Zhou J, Mo L, Feng K, Liu G, He Z (2020) Day-ahead short-term load probability density forecasting method with a decomposition-based quantile regression forest. Appl Energy 262:114396
Hess A, Spinler S, Winkenbach M (2021) Real-time demand forecasting for an urban delivery platform. Transport Res Part E Logist Transport Rev 145:102147
Hildebrandt FD, Ulmer MW (2021) Supervised learning for arrival time estimations in restaurant meal delivery. Transport Sci 56(4):1058–1084
Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
Huang W, Huang W, Jian S (2022) One-way carsharing service design under demand uncertainty: a service reliability-based two-stage stochastic program approach. Transport Res Part E Logist Transport Rev 159:102624
Jiménez-Bravo DM, Murciego ÁL, Crocker P, Leithardt VRQ, De Paz JF (2021) Can user data improve bike sharing systems demand forecasting? 2021 telecoms conference (ConfTELE). IEEE, Geneva, pp 1–6
Kaffash S, Nguyen AT, Zhu J (2021) Big data algorithms and applications in intelligent transportation system: a review and bibliometric analysis. Int J Prod Econ 231:107868
Kim K (2021) Spatial contiguity-constrained hierarchical clustering for traffic prediction in bike sharing systems. IEEE Trans Intell Transp Syst 23(6):5754–5764
Koay KY, Cheah CW, Chang YX (2022) A model of online food delivery service quality, customer satisfaction and customer loyalty: a combination of pls-sem and NCA approaches. Brit Food J 124(12):4516–4532
Kourentzes N (2013) Intermittent demand forecasts with neural networks. Int J Prod Econ 143(1):198–206
Kumar SV, Vanajakshi L (2015) Short-term traffic flow prediction using seasonal Arima model with limited input data. Eur Transp Res Rev 7(3):1–9
Lei J, Wasserman L (2014) Distribution-free prediction bands for non-parametric regression. J R Stat Soc Ser B Stat Methodol 76(1):71–96
Lei Z, Qian X, Ukkusuri SV (2020) Efficient proactive vehicle relocation for on-demand mobility service with recurrent neural networks. Transport Res Part C Emerg Technol 117:102678
Liang J, Ke J, Wang H, Ye H, Tang JA (2023) Poisson-based distribution learning framework for short-term prediction of food delivery demand ranges. IEEE Trans Intell Transport Syst 24(12):14556–14569
Liu TL, Krishnakumari P, Cats O (2019) Exploring demand patterns of a ride-sourcing service using spatial and temporal clustering. 2019 6th international conference on models and technologies for intelligent transportation systems (MT-ITS). IEEE, Geneva, pp 1–9
Liu D, Wang W, Zhao Y (2021) Effect of weather on online food ordering. Kybernetes 51(1):165–209
Liu J, Chen W, Yang J, Xiong H, Chen C (2022) Iterative prediction-and-optimization for e-logistics distribution network design. Inform J Comput 34(2):769–789
Liu W, Florkowski WJ (2018) Online meal delivery services: perception of service quality and delivery speed among Chinese consumers. Technical report
Lu S, Li Q, Bai L, Wang R (2019) Performance predictions of ground source heat pump system based on random forest and back propagation neural network models. Energy Convers Manage 197:111864
Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. Proceedings of the 31st international conference on

neural information processing systems. Curran Associates Inc., New York, United States, pp 4765–4774
Lv C, Zhang C, Lian K, Ren Y, Meng L (2020) A hybrid algorithm for the static bike-sharing re-positioning problem based on an effective clustering strategy. Transport Res Part B Methodol 140:1–21
Makridakis S, Spiliotis E, Assimakopoulos V (2020) The m4 competition: 100,000 time series and 61 forecasting methods. Int J Forecast 36(1):54–74
Makridakis S, Spiliotis E, Assimakopoulos V (2022) M5 accuracy competition: results, findings, and conclusions. Int J Forecast 38(4):1346–1364
Meinshausen N, Ridgeway G (2006) Quantile regression forests. J Mach Learn Res. 7(6)
Molnar C (2025) Interpretable machine learning, 3rd edn. https://christophm.github.io/interpretable-ml-book
Murtagh F, Contreras P (2012) Algorithms for hierarchical clustering: an overview. Wiley Interdiscipl Rev Data Min Knowl Discov 2(1):86–97
National Centers for Environmental Information, climate data online portal. Accessed 30 Sept 2023. https://www.ncei.noaa.gov/cdo-web/
Noland RB (2021) Scootin'in the rain: does weather affect micromobility? Transport Res Part A Policy Pract 149:114–123
Nosal T, Miranda-Moreno LF (2014) The effect of weather on the use of north American bicycle facilities: a multi-city analysis using automatic counts. Transport Res Part A Policy Pract 66:213–225
Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
Prajapati D, Harish AR, Daultani Y, Singh H, Pratap S (2023) A clustering based routing heuristic for last-mile logistics in fresh food e-commerce. Glob Bus Rev 24(1):7–20
Qian X, Ukkusuri SV, Yang C, Yan F (2020) Short-term demand forecasting for on-demand mobility service. IEEE Trans Intell Transport Syst 23(2):1019–1029
Qiao S, Han N, Huang J, Yue K, Mao R, Shu H, He Q, Wu X (2021) A dynamic convolutional neural network based shared-bike demand forecasting model. ACM Trans Intell Syst Technol (TIST) 12(6):1–24
Quantile-forest (2023) GitHub
Rahmani S, Baghbani A, Bouguila N, Patterson Z (2023) Graph neural networks for intelligent transportation systems: a survey. IEEE Trans Intell Transp Syst 24(8):8846–8885
Ribeiro MT, Singh S, Guestrin C (2016) "why should i trust you?" explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM Digital Library, New York, pp 1135–1144
Romano Y, Patterson E, Candes E (2019) Conformalized quantile regression. Adv Neural Inform Process Syst. https://doi.org/10.48550/arXiv.1905.03222
Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
Saadi I, Wong M, Farooq B, Teller J, Cools M (2017) An investigation into machine learning approaches for forecasting spatio-temporal demand in ride-hailing service. arXiv preprint arXiv:1703.02433
Salinas D, Flunkert V, Gasthaus J, Januschowski T (2020) Deepar: probabilistic forecasting with autoregressive recurrent networks. Int J Forecast 36(3):1181–1191
Shuai C, Shan J, Bai J, Lee J, He M, Ouyang X (2022) Relationship analysis of short-term origin-destination prediction performance and spatiotemporal characteristics in urban rail transit. Transport Res Part A Policy Pract 164:206–223
Skorupa G (2020) Python implementation of TBATS. GitHub
Smith TG, et al.: pmdarima: ARIMA estimators for Python. [Online; accessed ¡today¿] (2017–). http://www.alkaline-ml.com/pmdarima
Song H, Qiu RT, Park J (2019) A review of research on tourism demand forecasting: launching the annals of tourism research curated collection on tourism demand forecasting. Ann Tour Res 75:338–362
Su K, Liu X, Shlizerman E (2020) Predict & cluster: unsupervised skeleton based action recognition. Proceedings of the IEEE/CVF conference on computer vision and pattern Recognition. IEEE, Geneva, pp 9631–9640
Suganthi L, Samuel AA (2012) Energy models for demand forecasting—a review. Renew Sustain Energy Rev 16(2):1223–1240
Tong T, Dai H, Xiao Q, Yan N (2020) Will dynamic pricing outperform? theoretical analysis and empirical evidence from o2o on-demand food service market. Int J Prod Econ 219:375–385
Uber: H3: Uber's Hexagonal Hierarchical Spatial Index (2018) https://www.uber.com/en-NL/blog/h3/ Accessed 2023-06-03
Ulmer MW, Thomas BW, Campbell AM, Woyak N (2021) The restaurant meal delivery problem: dynamic pickup and delivery with deadlines and random ready times. Transp Sci 55(1):75–100
Ulrich M, Jahnke H, Langrock R, Pesch R, Senge R (2021) Distributional regression for demand forecasting in e-grocery. Eur J Oper Res 294(3):831–842
Vairagade N, Logofatu D, Leon F, Muharemi F (2019) Demand forecasting using random forest and artificial neural network for supply chain management. Computational collective intelligence: 11th international conference, ICCCI 2019, Hendaye, France, September 4–6, 2019, Proceedings, Part I 11. Springer, Cham, pp 328–339
Xing Y, Zhang S, Wen P, Shao L, Rouyendegh BD (2020) Load prediction in short-term implementing the multivariate quantile regression. Energy 196:117035
Xue G, Wang Z, Wang G (2021) Optimization of rider scheduling for a food delivery service in o2o business. J Adv Transp 2021:1–15
Yao W, Zhao H, Liu L (2023) Weather and time factors impact on online food delivery sales: a comparative analysis of three Chinese cities. Theoret Appl Climatol 153(3):1425–1438
Yu X, Lan A, Mao H (2023) Short-term demand prediction for on-demand food delivery with attention-based convolutional LSTM. Systems 11(10):485
Zerveas G, Jayaraman S, Patel D, Bhamidipaty A, Eickhoff C (2021) A transformer-based framework for multivariate time series representation learning. Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. ACM Digital Library, New York, pp 2114–2124
Zhang J, Shan E, Wu L, Yin J, Yang L, Gao Z (2024) An end-to-end predict-then-optimize clustering method for stochastic assignment problems. IEEE Trans Intell Transp Syst 25(9):12605–12620
Zhu L, Yu W, Zhou K, Wang X, Feng W, Wang P, Chen N, Lee P (2020) Order fulfillment cycle time estimation for on-demand food delivery. Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. ACM Digital Library, New York, pp 2571–2580