A Unified Geospatial Clustering Framework to Identify Varying Density Clusters in E-Commerce Logistics

Arpit Tiwari, Bhavuk Singhal, Anshu Aditya, Aryan Tiwari, Shubham Jain, Debashis Mukherjee, Debdoot Mukherjee

Meesho, India

{arpit.tiwari1, bhavuk.singhal, anshu.aditya, aryan.tiwari1, shubham.jain1, debashis.mukherjee, debdoot.mukherjee}@meesho.com

Abstract

In e-commerce logistics, accurate geospatial clustering is essential for optimizing resource allocation, manpower planning, and delivery network design. However, existing density-based clustering approaches, particularly their reliance on heuristic parameter tuning, have been underexplored in datasets with significant density variations, limiting robustness and scalability. This study presents an unsupervised framework that extends DBSCAN by leveraging Gaussian Mixture Models (GMM). First, we propose a method that systematically identifies suitable clustering scales through statistical modeling. Second, the approach iteratively applies DBSCAN to extract clusters from dense to sparse regions, overcoming single-parameter limitations. Finally, we validate the method through large-scale offline experiments using data from over 200 last-mile dispatch centers (LMDC). The results demonstrate the framework's effectiveness in identifying heterogeneous geographic demand patterns and supporting workforce planning and operational benchmarking. This framework provides a scalable solution to a critical challenge in e-commerce logistics, offering a valuable reference for strategic and operational decision-making.

1 Introduction and Background

Geospatial clustering is essential in e-commerce logistics for turning raw location data into practical zones that guide workforce planning, routing, and performance benchmarking. It helps cut costs and optimize resources, especially in the last mile, which makes up 41–53% of logistics expenses (Wise Systems 2023; Business Insider 2025). Since labor accounts for 50–60% of these expenses (Elite Extra 2023), precise demand clustering is crucial. Conversely, inefficient clustering leads to overstaffing in low-density regions and under-staffing in high-density zones, increasing costs and degrading service. Clustering that accurately identifies delivery zones thus plays a critical role in reducing operational overhead, improving resource utilization, and maintaining service standards across hundreds of dispatch centers and millions of deliveries.

Delivery address datasets often show wide variation in point densities within the same region, which challenges standard clustering methods that assume uniform density. DBSCAN (Ester et al. 1996) is a popular density-based algorithm that uses fixed global parameters ε (neighborhood radius) and `MinPts` (minimum points per neighborhood) to define clusters. However, using a single global ε in heterogeneous datasets is problematic: small values miss sparse clusters and label them as noise, while large values merge distinct dense areas into one cluster. Several adaptive variants of DBSCAN have been proposed to mitigate this issue. ADBSCAN (Khan et al. 2018) introduces local ε_x for flexibility but struggles with high complexity and poor scalability. DBSCAN++ (Jang and Jiang 2019) uses subsampling for efficiency, yet still depends on a global bandwidth parameter. AMD-DBSCAN (Wang et al. 2022) auto-selects $(\varepsilon, \texttt{MinPts})$ pairs via k-NN distributions, improving mixed-density clustering but relies on heuristics without statistical rigor. MDBSCAN (Qian et al. 2024) adjusts parameters via relative density but assumes simple shapes and lacks real-world validation. AR-DBSCAN (Peng et al. 2025) uses reinforcement learning to adapt parameters across regions but adds complexity and requires specialized training. Overall, these methods face trade-offs between adaptability, scalability, and statistical soundness.

In this work, we present a scalable clustering method for datasets with highly variable densities, overcoming key limitations of existing approaches. Our approach combines GMMs with iterative DBSCAN to automatically identify clusters across density scales. Key contributions include:

1. A GMM-guided iterative DBSCAN framework that models neighborhood distances to extract clusters from dense to sparse regions.
2. We discuss the limitations of existing adaptive methods, highlighting issues in parameter selection, scalability, and real-world applicability.
3. We discuss the potential real-world impact of this approach across multiple use cases and outline a clear path to deployment. Our method has been validated on production-scale data from over 200 LMDC in India.

2 Related Works

Classical Density-Based Clustering: Density-based clustering methods like DBSCAN are effective for non-convex structures and noise filtering but falter with varying densities. OPTICS (Ankerst et al. 1999) and HDB-

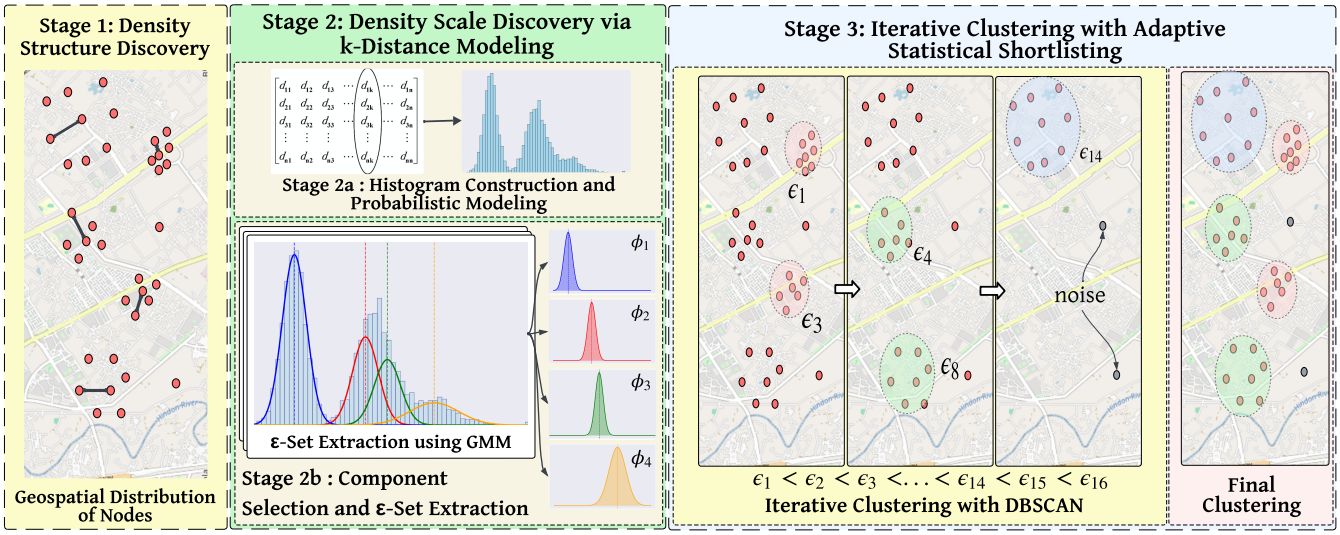

Figure 1: Proposed Framework

SCAN (Campello et al. 2015) address this via hierarchical density modeling, yet scalability and parameter sensitivity persist. To improve efficiency, methods leveraging approximate k-nearest neighbor graphs (Gan and Tao 2015) and fast indexing techniques (Weng, Gou, and Fan 2021) have been proposed, though the core assumption of uniform density still limits performance in heterogeneous settings.

Adaptive and Multi-Density DBSCAN Variants: To overcome DBSCAN's sensitivity to global parameters, adaptive variants estimate local density parameters. Early methods like VDBSCAN and ADBSCAN (Louhichi, Gzara, and Abdallah 2014; Khan et al. 2018) varied ε based on neighborhood traits, improving heterogeneity handling but relying on heuristics and struggling with noise. AMD-DBSCAN (Wang et al. 2022) introduced local GMMs for dynamic adjustment, yet still faced threshold sensitivity. Recent methods like MDBSCAN (Qian et al. 2024) and VDPC (Wang et al. 2023a) better manage density gradients but often incur high preprocessing costs and scalability issues in large-scale applications.

Learning-Based and Reinforcement Clustering: Recent advances in clustering replace heuristic rules with data-driven models for better adaptability to complex, high-dimensional data. Deep Embedded Clustering (DEC) (Xie, Girshick, and Farhadi 2016) and its extension VaDE (Jiang et al. 2016) improve latent representation learning but require predefined cluster counts and are sensitive to initialization. Semi-supervised methods like SDEC (Wang et al. 2023b) enhance control with limited supervision, while SSC (Elhamifar and Vidal 2013) offers theoretical robustness at the cost of scalability. Reinforcement learning approaches, such as AR-DBSCAN (Peng et al. 2025), improve adaptability but remain computationally intensive. Overall, learning-based clustering offers flexibility but faces challenges in real-time, large-scale applications due to complexity and infrastructure demands.

Statistical and Model-Based Clustering: Statistical clustering explicitly models the data-generating process, often using likelihood-based methods. GMMs, estimated via EM (McLachlan, Lee, and Rathnayake 2019), support soft assignments and underpin many classical pipelines, as detailed in (Bishop and Nasrabadi 2006). Model-based techniques (Fraley and Raftery 2002) use criteria like BIC or AIC to select cluster numbers and perform well under parametric assumptions but falter with non-elliptical or complex structures. Robustness issues, including sensitivity to initialization and outliers are highlighted in (Hennig 2004), posing challenges in noisy domains like logistics. Bayesian extensions offer more flexibility but are computationally demanding for large-scale or real-time use.

Scalability and Systems-Level Improvements: Many clustering algorithms struggle to scale due to costly neighborhood queries or iterative fitting. This limits their use in large-scale settings like logistics networks. To address this, research has focused on parallelizing DBSCAN, e.g., (Wang, Gu, and Shun 2020) use shared-memory models with spatial indexing, while others apply MapReduce for distributed computation (Kim et al. 2014). Though scalable, these often trade off accuracy. Fast variants like h-DBSCAN(Weng, Gou, and Fan 2021) use approximations to reduce runtime. For hierarchical clustering, parallel HDBSCAN* (Wang et al. 2021) leverages optimized MST algorithms. These advances show how system-level innovations are crucial for making clustering viable at scale.

E-Commerce and Last-Mile Logistics Applications: Clustering is widely used in logistics for delivery zone design, hub placement, and route planning. Most approaches still rely on traditional methods like K-Means, which assume uniformity and Euclidean geometry assumptions that often fail in uneven demand landscapes. Recent efforts include balancing delivery workloads (Wangwattanakool and Laesanklang 2024) , time-balanced clustering (Menchaca-Méndez et al. 2022) , and hybrid K-Means with p-Median for hub optimization (Rahman et al. 2025). While simple,

these methods struggle with noisy, multi-density data. More advanced models use graph-based clustering and agent-based simulations to capture complex dynamics (Zhang, Li, and Xiao 2024), (Calabrò et al. 2023), and (Fried and Goodchild 2023). Yet, density-based methods remain rarely adopted in large-scale e-commerce systems. Our work addresses this gap by integrating adaptive density estimation into a scalable system spanning hundreds of LMDCs.

3 Methodology

3.1 Problem Formulation

Let $\mathcal{X} = \{x_i \in \mathbb{R}^2 \mid i = 1, \ldots, n\}$ be geospatial points representing delivery addresses. The goal is to cluster $\mathcal{X}$ into $\mathcal{C} = \{C_1, \ldots, C_k\}$, where each $C_j \subset \mathcal{X}$ reflects areas of operational relevance, while outliers are treated as noise. A key difficulty is heterogeneous density: some regions are dense, others sparse. Formally,

$$\exists C_a, C_b \in \mathcal{C} \text{ such that } \rho(C_a) \gg \rho(C_b), \tag{1}$$

with $\rho(C) \approx \frac{|C|}{\text{area}(C)}$. We address this using an iterative method that combines GMMs with DBSCAN to capture clusters across density scales.

3.2 Conceptual Overview of the Proposed Framework

To overcome the limitations of single-parameter density-based clustering in heterogeneous geospatial datasets, we propose a three-stage framework that decouples density estimation from cluster formation:

- **Stage 1: Density Structure Discovery** – Compute k-nearest neighbor distances for each point to capture local density variation. The resulting univariate distribution reflects the multi-scale nature of the data.
- **Stage 2: Statistical Scale Extraction** – Fit a GMM to the k-distances distribution. Each Gaussian component corresponds to a density level.
- **Stage 3: Iterative Density Peeling** – Apply DBSCAN iteratively, starting with the smallest ε to extract clusters from the densest regions. After each iteration, clusters are removed and larger ε values are used to capture sparser regions. Statistically derived thresholds ensure only significant clusters are retained, filtering out noise and spurious groupings. Our end-to-end framework is shown in Figure 1.

3.3 Stage 1 — Density Structure Discovery

Let $\mathcal{X} = \{x_i \in \mathbb{R}^2 \mid i = 1, \ldots, n\}$ denote the set of geospatial points, and let $k =$ MinPts be the minimum number of neighbors required for a core point in DBSCAN. For each point $x_i \in \mathcal{X}$, we compute the Euclidean distance $d_k(x_i)$ to its k-th nearest neighbor $\text{NN}_k(x_i)$, i.e.,

$$d_k(x_i) = \|x_i - \text{NN}_k(x_i)\|_2. \tag{2}$$

The set of all such distances,

$$D_k = \{d_k(x_i) \mid x_i \in \mathcal{X}\}, \tag{3}$$

serves as a compact representation of local density variation. This one-dimensional transformation of the geospatial data is both computationally efficient and robust to irregular cluster geometries, making it well-suited for large-scale datasets. In our framework, the choice of MinPts is tied to dataset size: slabs of order counts (e.g., 0–1000, 1000–2000, ...) are mapped to increasing MinPts values (10, 20, ...). This heuristic discretization stabilizes performance across scales while keeping the computation tractable.The ordered vector D_k is the output of this stage and is used directly in Stage 2 to identify statistically significant density levels without relying on heuristic parameter tuning.

3.4 Stage 2 — Density Scale Discovery via k-Distance Modeling

Given the k-distance vector $D_k = \{d_k(x_i)\}_{i=1}^n$ from Stage 1, our objective is to extract a set of candidate neighborhood radii ε that capture the natural density levels present in the data. This process is divided into two sub-parts.

Histogram Construction and Probabilistic Modeling We treat the k-distance values as an empirical sample from an unknown probability distribution $p(x)$ and approximate $p(x)$ with a k-distance histogram H. In this context, lower k-distances indicate denser regions (smaller ε), while higher k-distances reflect sparser areas (larger ε). The height of each peak in the histogram denotes the number of points at that density level, taller peaks correspond to more points. To capture these multi-scale regimes, we fit a GMM:

$$p(x \mid \Theta) = \sum_{k=1}^{K} \phi_k \mathcal{N}(x \mid \mu_k, \sigma_k^2) \tag{4}$$

where $\Theta = \{\phi_k, \mu_k, \sigma_k\}_{k=1}^K$ are the mixing coefficients, means, and standard deviations. Here, μ_k represents the center of a density scale, σ_k its spread, and ϕ_k its relative significance. The GMM components are superimposed on H to visualize their contribution to $p(x)$.

Component Selection and ε-Set Extraction The number of mixture components K is unknown a priori. We estimate it using an Iterative GMM Component Selection procedure (Algorithm 1). Instead of sequentially increasing K, the process is distributed across worker machines, each assigned a candidate value up to $K_{\max}$, where $K_{\max}$ is a sufficiently large value. Every worker fits a GMM for its assigned K in parallel. After fitting, we evaluate the weight of the least significant component. Models where this weight falls below a predefined threshold $\phi_{\min}$ (e.g., 0.01) are deemed redundant, as added components contribute negligibly and lack statistical significance. From these parallel evaluations, we select the maximum feasible K that meets the significance criterion, ensuring larger redundant models are discarded while retaining the largest statistically valid number of components. Once K is fixed, the candidate ε-set is generated by systematically probing around each component's mean.

$$E = \text{sorted}\left(\bigcup_{k=1}^{K_{\max}} \{\mu_k + j\sigma_k \mid j \in \{0, 1, 2, 3\}\}\right) \tag{5}$$

These values span from the densest regions (μ_k) to progressively sparser neighborhoods ($\mu_k + 3\sigma_k$), ensuring coverage across all scales.

Algorithm 1: Iterative GMM Component Selection

Input: X (k-distance vector), $K_{\max}$, $\varphi_{\min}$
Output: K_{opt} (optimal number of components)

$K_{\text{opt}} \leftarrow K_{\max}$
for $K = 1$ **to** $K_{\max}$ **do**
  Fit GMM on X with K components
  Extract mixing coefficients: $\{\varphi_k\}$
  if $\min(\{\varphi_k\}) < \varphi_{\min}$ **then**
    $K_{\text{opt}} \leftarrow K - 1$
    break
return K_{opt}

3.5 Stage 3 — Iterative Clustering with Adaptive Statistical Shortlisting

With the ordered set of candidate neighborhood radii $E = \{\varepsilon_1 < \cdots < \varepsilon_m\}$ obtained from Stage 2, the framework applies an iterative density peeling strategy. At each iteration i, DBSCAN is applied to the current active dataset $D^{(i)}$, consisting of points not yet assigned to any cluster ($n = |D^{(i)}|$). The goal is to capture dense clusters first, using the smallest ε, and progressively reveal sparser ones as ε increases. For the current ε_i, we define the surviving set

$$Z_i = \{p \in D^{(i)} \mid \varepsilon_{i-1} \le k\text{-dist}(p)\}, \quad z_i = |Z_i|, \tag{6}$$

representing points whose local density is above the previous scale ε_{i-1}. Within Z_i, the subset that DBSCAN assigns to clusters at ε_i is

$$X_i = \{p \in D^{(i)} \mid \varepsilon_{i-1} \le k\text{-dist}(p) < \varepsilon_i\}, \quad x_i = |X_i|, \tag{7}$$

The base percentile is then given by

$$P_i = \frac{x_i}{z_i} \times 100, \tag{8}$$

quantifying the proportion of points in the current band that form clusters. To adaptively account for points outside the current band, let

$$y = n - z_i \tag{9}$$

be the number of remaining unclustered points in other bands. The corrected percentile is computed as

$$P_i^* = \frac{\frac{P_i \cdot z_i}{100} + (n - z_i)}{n} \times 100, \tag{10}$$

which can also be written as

$$P_i^* = [1 + \text{adj}_i] \times 100, \quad \text{adj}_i = \frac{z_i}{n}\left(\frac{P_i}{100} - 1\right). \tag{11}$$

This adjustment ensures that the acceptance threshold reflects the success rate within the current density band and the proportion of remaining points. Clusters produced at iteration i are sorted in descending order of size and accepted if they are not already assigned a cluster ID and their inclusion does not cause the cumulative accepted size to exceed

$$T_i = |D^{(i)}| \cdot \frac{P_i^*}{100}. \tag{12}$$

Accepted clusters are removed from $D^{(i)}$ before moving to the next ε. Once all radii have been processed, any remaining points are labeled as noise. This formulation allowing the method to remain fully data-driven and adaptable to multi-density geospatial structures. The complete iterative process is formally described in Algorithm 2.

Algorithm 2: Iterative Clustering

Input: D, $E = \{\varepsilon_1 < \cdots < \varepsilon_m\}$, MinPts
Output: Final set of clusters S

$D_{\text{active}} \leftarrow D$, $S \leftarrow \emptyset$, $\varepsilon_0 \leftarrow 0$
for $i = 1$ **to** m **do**
  Run DBSCAN on D_{active} with (ε_i, MinPts)
  Extract candidate clusters $\{C_1, C_2, \ldots, C_t\}$
  $n \leftarrow |D_{\text{active}}|$
  $Z_i \leftarrow \{\, p \in D_{\text{active}} \mid \varepsilon_{i-1} \le \text{k-dist}(p)\}$
  $z_i \leftarrow |Z_i|$
  $X_i \leftarrow \{\, p \in D_{\text{active}} \mid \varepsilon_{i-1} \le \text{k-dist}(p) < \varepsilon_i \,\}$
  $x_i \leftarrow |X_i|$
  if $z_i > 0$ **then**
    $P_i \leftarrow \dfrac{x_i}{z_i} \times 100$
  else
    $P_i \leftarrow 0$
  $P_i^* \leftarrow \dfrac{\frac{P_i\, z_i}{100} + (n - z_i)}{n} \times 100$
  $T_i \leftarrow n \cdot \dfrac{P_i^*}{100}$
  Sort $\{C_1, \ldots, C_t\}$ by size in descending order
  Accepted $\leftarrow \emptyset$, total $\leftarrow 0$
  for $j = 1$ **to** t **do**
    if *total* $+|C_j| \le T_i$ **then**
      Accepted $\leftarrow$ Accepted $\cup \{C_j\}$
      total $\leftarrow$ total $+|C_j|$
  $D_{\text{active}} \leftarrow D_{\text{active}} \setminus$ Accepted
  $S \leftarrow S \cup$ Accepted
Label remaining points in D_{active} as noise
return S

4 Experimental Setup

4.1 Dataset Description

We use real-world e-commerce logistics data from Meesho's national delivery network, sampling 200 LMDCs across the Indian states of Haryana and Maharashtra. These centers span both urban and rural geographies and exhibit diverse

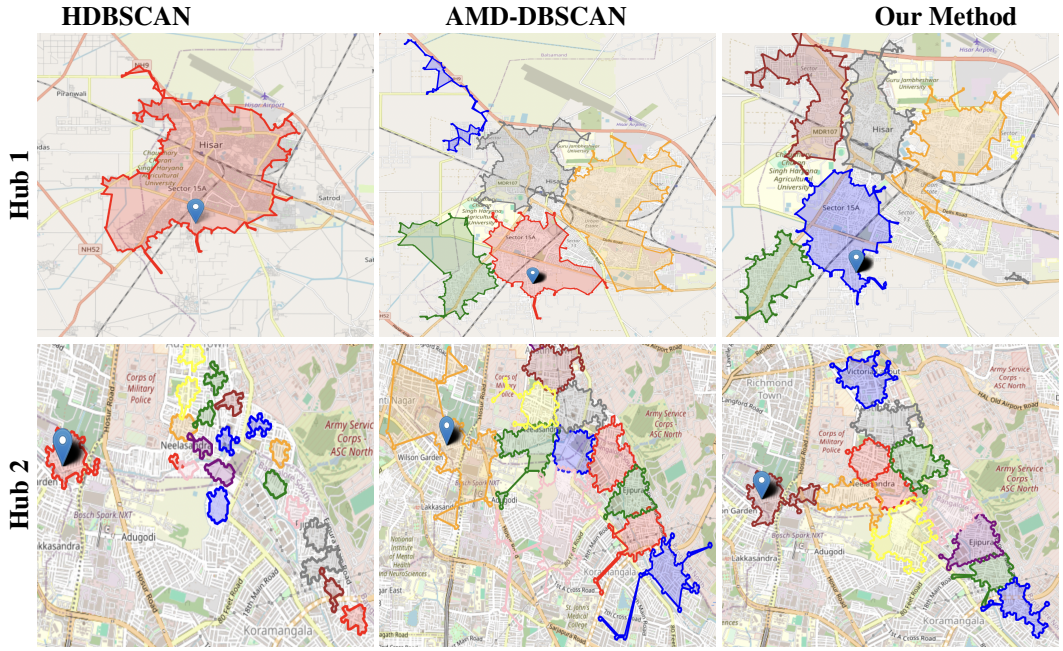

Figure 2: Visual Comparison of Clustering Results across DCs and Methods

geospatial density patterns. Table 1 summarizes the dataset statistics.

State	#DCs	#Orders
Haryana	110	∼6M
Maharashtra	95	∼4M

Table 1: Data Statistics

4.2 Baselines

We compare our method against twelve clustering baselines across four categories. Classical density-based methods include **DBSCAN** (fixed ε), **OPTICS** (reachability-based), and **HDBSCAN** (hierarchical density stability). Adaptive baselines include **ADBSCAN**, **AMD-DBSCAN**, **MDDBSCAN**, **VDBSCAN**, and **DBSCAN++**, each addressing density variation through local heuristics or statistical modeling. Statistical method include **GMM** that relies on Gaussian mixture assumptions with model selection. For contrast, we include **DEC**, a deep learning approach that clusters in embedded latent space. All baselines are implemented using public libraries with recommended parameter tuning.

5 Results and Discussion

We evaluate performance with three metrics: Silhouette Score (SS), where higher values mean better-defined clusters Davies–Bouldin Index (DBI), where lower values mean more compact, well-separated clusters; and Percentage of Unclustered Points (PUP), which measures the fraction of data labeled as noise or discarded. Table 2 compares our method against twelve baselines. Our method achieves the highest SS (0.67), the lowest DBI (0.72), and the smallest PUP (4.2%), outperforming DBSCAN (12.5%) and adaptive methods like AMD-DBSCAN (6.9%). While AMD-DBSCAN and HDBSCAN deliver strong cluster quality, they suffer higher noise in low-density regions. GMM-based methods and DEC show inconsistent results, especially in sparse or non-convex settings. Our method delivers quality and robustness across densities without manual tuning.

Method	SS ↑	DBI ↓	PUP ↓
DBSCAN	0.42	1.38	12.5%
OPTICS	0.45	1.29	10.8%
HDBSCAN	0.56	0.98	9.6%
ADBSCAN	0.47	1.21	11.2%
AMD-DBSCAN	0.61	0.84	6.9%
MDDBSCAN	0.58	0.92	8.1%
VDBSCAN	0.48	1.10	10.4%
DBSCAN++	0.50	1.05	9.9%
GMM	0.53	1.11	13.4%
DEC	0.46	1.27	14.3%
Ours	**0.72**	**0.72**	**4.2%**

Table 2: Clustering performance across methods.

To evaluate clustering quality in real-world geospatial settings, we compare HDBSCAN, AMD-DBSCAN, and our proposed method across two representative hubs, as shown in Figure 2. **Hub 1**, a semi-urban region with moderate density variation, shows that HDBSCAN merges distinct neighborhoods into a single large cluster, limiting use in route planning and manpower allocation. AMD-DBSCAN is more granular but produces irregular shapes that ignore natural

or infrastructural boundaries. In contrast, our method yields distinct, coherent clusters aligned with roads and neighborhood divisions, ensuring operationally meaningful segmentation. In **Hub 2**, with a dense irregular urban layout, differences are sharper. HDBSCAN over-fragments, generating many small clusters and leaving gaps. AMD-DBSCAN performs better but still shows boundary inconsistencies in transitional areas. Our method adapts well to complex density, producing compact, contiguous clusters that mirror locality structures, enhancing interpretability and enabling efficient delivery zone design without manual tuning.

6 Path to Deployment

6.1 Scalable Deployment via Distributed Computing

To ensure computational feasibility at production scale, the clustering framework was deployed on Apache Spark using a distributed pipeline, as illustrated in Figure 3, which consists of three main stages.

- **Distributed k-Distance Computation:** k-distances are computed in parallel across workers, with each worker processing a partition of the dataset and aggregating results centrally. This enables efficient scaling to tens of millions of points without incurring memory bottlenecks.
- **Parallelized GMM Component Selection:** Instead of sequentially fitting models for increasing values of,candidate K-values are distributed across workers (up to max. Each worker fits a GMM independently, while the driver node selects the minimum feasible K that satisfies statistical significance. This parallelization yields an order-of-magnitude speedup in model selection while preserving robustness.
- **Distributed Iterative DBSCAN:** Using derived ϵ-values, DBSCAN is executed via Spark's mapPartitions to parallelize clustering across LMDCs. In large-scale tests with 200 LMDCs, clusters were evenly distributed across workers, resulting in substantial runtime reductions and consistent speedups over sequential execution.
- **Hardware Configuration and Performance:** The framework was deployed on a multi-node Spark cluster comprising one driver (64 GB RAM, 16 vCPUs) and 20 worker nodes (32 GB RAM, 8 vCPUs each). This setup ensured stable execution, efficient resource utilization, and demonstrated production readiness

6.2 Pilot Plan and Business Impact

This framework will be piloted in Q4 2025 across select Tier-1 and Tier-2 cities and a phased pan-India(∼5000 LMDCs) rollout in the following quarters. The pilot will evaluate its impact on three logistics workflows: (i) Self-pickup spot allocation, (ii) Field Executive (FE) rate card selection, and (iii) FE manpower allocation. Offline tests show improvements in accessibility, cost efficiency, and workforce balance, which are key KPIs for the pilot.

- **Self-Pickup Spot Allocation:** We optimize pickup spot placement to ensure users have access within 500 m, reducing delivery overhead and improving accessibility. Across cities, coverage rose from 20% to 95%, demonstrating strong gains in efficiency and reach.

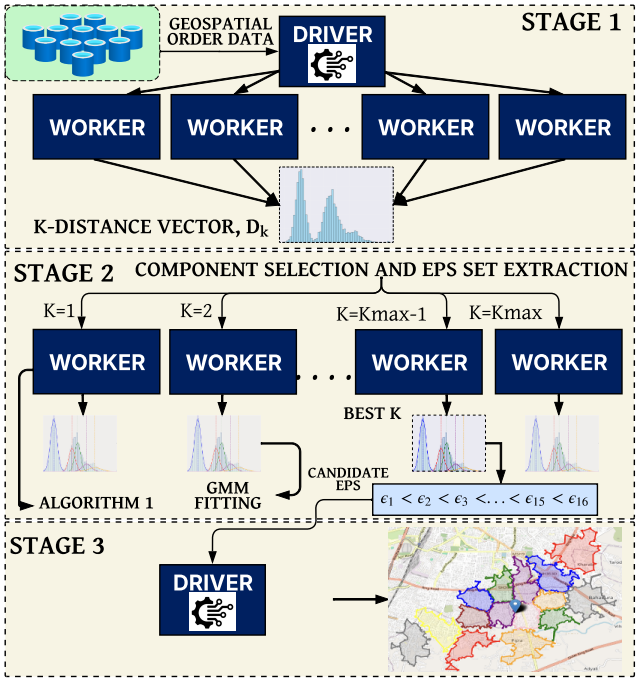

Figure 3: An overview of Distributed Computing

- **FE Rate Card Selection:** We propose a density-aware framework that adjusts payouts to reflect regional effort, unlike uniform per-order rates. Tested on 200 LMDCs, it cut delivery costs by 3.2% and RTOs by 0.1%, boosting efficiency and reliability.

- **FE Manpower Allocation:** We propose an adaptive FE manpower allocation framework that integrates cluster-driven demand with operational attributes to dynamically balance workforce supply and delivery needs. Across 200 DCs, offline experiments showed 21% sites in deficit (∼19% understaffing) and 79% in surplus (∼55% overstaffing). This approach improves balance, cost efficiency, and workforce utilization.

7 Conclusion & Future Work

This paper introduces a scalable clustering framework that integrates GMM with iterative DBSCAN to handle multi-density geospatial data. It outperforms classical and adaptive baselines in clustering quality, robustness, and geospatial coherence, aligning with real-world logistics needs. Validated across 200 dispatch centers, the method is being prepared for deployment in Meesho's logistics infrastructure. Future work includes automating MinPts selection, incorporating temporal demand shifts, road and SLA constraints, and linking clustering with routing and incentive systems.

References

Ankerst, M.; Breunig, M. M.; Kriegel, H.-P.; and Sander, J. 1999. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2): 49–60.

Bishop, C. M.; and Nasrabadi, N. M. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Business Insider. 2025. The supply chain's last mile is complex and expensive. https://www.businessinsider.com/ai-transforms-last-mile-delivery-predictive-analytics-route-optimization-2025-7.

Calabrò, G.; Le Pira, M.; Giuffrida, N.; Fazio, M.; Inturri, G.; and Ignaccolo, M. 2023. A spatial agent-based model of e-commerce last-mile logistics towards a delivery-oriented development. *Transportation Research Interdisciplinary Perspectives*, 21: 100895.

Campello, R. J.; Moulavi, D.; Zimek, A.; and Sander, J. 2015. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(1): 1–51.

Elhamifar, E.; and Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11): 2765–2781.

Elite Extra. 2023. Last Mile Delivery Costs: The Most Expensive Step in the Supply Chain. https://eliteextra.com/last-mile-delivery-costs-the-most-expensive-step-in-the-supply-chain/.

Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.; et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, 226–231.

Fraley, C.; and Raftery, A. E. 2002. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458): 611–631.

Fried, T.; and Goodchild, A. 2023. E-commerce and logistics sprawl: A spatial exploration of last-mile logistics platforms. *Journal of Transport Geography*, 112: 103692.

Gan, J.; and Tao, Y. 2015. DBSCAN revisited: Mis-claim, un-fixability, and approximation. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, 519–530.

Hennig, C. 2004. Breakdown points for maximum likelihood estimators of location–scale mixtures.

Jang, J.; and Jiang, H. 2019. DBSCAN++: Towards fast and scalable density clustering. In *International conference on machine learning*, 3019–3029. PMLR.

Jiang, Z.; Zheng, Y.; Tan, H.; Tang, B.; and Zhou, H. 2016. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*.

Khan, M. M. R.; Siddique, M. A. B.; Arif, R. B.; and Oishe, M. R. 2018. ADBSCAN: Adaptive density-based spatial clustering of applications with noise for identifying clusters with varying densities. In *2018 4th international conference on electrical engineering and information & communication technology (iCEEiCT)*, 107–111. IEEE.

Kim, Y.; Shim, K.; Kim, M.-S.; and Lee, J. S. 2014. DBCURE-MR: An efficient density-based clustering algorithm for large data using MapReduce. *Information Systems*, 42: 15–35.

Louhichi, S.; Gzara, M.; and Abdallah, H. B. 2014. A density based algorithm for discovering clusters with varied density. In *2014 world congress on computer applications and information systems (wccais)*, 1–6. IEEE.

McLachlan, G. J.; Lee, S. X.; and Rathnayake, S. I. 2019. Finite mixture models. *Annual review of statistics and its application*, 6(1): 355–378.

Menchaca-Méndez, A.; Montero, E.; Flores-Garrido, M.; and Miguel-Antonio, L. 2022. An algorithm to compute time-balanced clusters for the delivery logistics problem. *Engineering Applications of Artificial Intelligence*, 111: 104795.

Peng, H.; Huang, X.; Sun, S.; Zhang, R.; and Yu, P. S. 2025. Adaptive and robust DBSCAN with multi-agent reinforcement learning. *arXiv preprint arXiv:2505.04339*.

Qian, J.; Zhou, Y.; Han, X.; and Wang, Y. 2024. MDB-SCAN: A multi-density DBSCAN based on relative density. *Neurocomputing*, 576: 127329.

Rahman, M. A.; Basheer, M. A.; Khalid, Z.; Tahir, M.; and Uppal, M. 2025. Logistics hub location optimization: A k-means and p-median model hybrid approach using road network distances. *Transportation Research Procedia*, 84: 219–226.

Wang, Y.; Gu, Y.; and Shun, J. 2020. Theoretically-efficient and practical parallel DBSCAN. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2555–2571.

Wang, Y.; Wang, D.; Zhou, Y.; Zhang, X.; and Quek, C. 2023a. VDPC: Variational density peak clustering algorithm. *Information Sciences*, 621: 627–651.

Wang, Y.; Yu, S.; Gu, Y.; and Shun, J. 2021. Fast parallel algorithms for euclidean minimum spanning tree and hierarchical spatial clustering. In *Proceedings of the 2021 international conference on management of data*, 1982–1995.

Wang, Y.; Zou, J.; Wang, K.; Liu, C.; and Yuan, X. 2023b. Semi-supervised deep embedded clustering with pairwise constraints and subset allocation. *Neural Networks*, 164: 310–322.

Wang, Z.; Ye, Z.; Du, Y.; Mao, Y.; Liu, Y.; Wu, Z.; and Wang, J. 2022. AMD-DBSCAN: An Adaptive Multi-density DBSCAN for datasets of extremely variable density. In *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, 1–10. IEEE.

Wangwattanakool, J.; and Laesanklang, W. 2024. Delivery Zones Partitioning Considering Workload Balance Using Clustering Algorithm. In *14th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, SIMULTECH 2024*, 378–385. Science and Technology Publications, Lda.

Weng, S.; Gou, J.; and Fan, Z. 2021. h-DBSCAN: A simple fast DBSCAN algorithm for big data. In *Asian conference on machine learning*, 81–96. PMLR.

Wise Systems. 2023. Last Mile Statistics 2023: Insights, Innovations Trends. https://www.wisesystems.com/blog/last-mile-statistics-2023-insights-innovations-trends/.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, 478–487. PMLR.

Zhang, C.; Li, Y.; and Xiao, Z. 2024. Uncover the Dynamic Community Structure of Instant Delivery Network. *arXiv preprint arXiv:2411.10002*.