Identifying Urban Last-Mile Delivery Stops from GPS Trajectory Data: A Feature-Driven Machine Learning Framework

Wei Zhang[1], Jing Liu[2], Hao Chen[3], Mei Wang[4, *]

[1] Department of Transportation Engineering, Tongji University, Shanghai 200092, China
[2] School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China
[3] State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China
[4] Institute of Urban Mobility and Logistics, Tongji University, Shanghai 200092, China

*Email: mei.wang.research@tongji-tml.edu.cn (Corresponding Author)

Abstract

Urban last-mile logistics represents one of the fastest-growing and most operationally complex segments of the contemporary supply chain, yet fine-grained monitoring of delivery activities remains a persistent challenge because passively collected GPS records do not encode the purpose of individual vehicle stops. This paper presents an end-to-end, feature-driven machine learning framework for identifying genuine delivery stops from GPS trajectories of urban courier vehicles by integrating raw vehicle traces with electronic waybill records. Drawing on a real-world parcel-courier dataset spanning the full calendar year 2022 and encompassing more than 80 delivery vehicles across a major metropolitan area, the framework proceeds through three sequential stages: preprocessing of raw GPS traces, stop-candidate extraction via speed-threshold segmentation and spatial merging, and automated ground-truth labelling through waybill-to-stop matching within calibrated spatio-temporal tolerance windows. Each candidate stop is represented by five interpretable, domain-grounded features—dwell time, pre-stop speed, heading change, local stop density, and distance from the departure hub—that collectively capture the kinematic and spatial signatures distinguishing genuine delivery events from other stop types, without requiring any external GIS layers or map-matching infrastructure. To address severe class imbalance (positive-class rate: 2.92%), Synthetic Minority Over-sampling Technique (SMOTE) resampling is applied exclusively to the training partition before three supervised classifiers—Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Tree (DT)—are trained and evaluated under a unified experimental protocol. All three models achieve test-set identification accuracy exceeding 98.9%. Cross-model error analysis reveals that SVM exhibits a precision-oriented error profile (FNR 22.80%; FPR 0.01%), whereas KNN and DT demonstrate recall-oriented behavior (FNR below 9%; FPR below 0.9%) and identifies three empirically grounded hard-case false-negative patterns that define actionable targets for future feature enrichment. The framework requires no manual annotation or external facility inventories, rendering it directly transferable to other commodity types and urban operating environments.

Keywords: urban last-mile logistics; GPS trajectory; delivery stop identification; machine learning; SMOTE; class imbalance; SVM; KNN; decision tree; interpretable features

Article History
Received July 15, 2025
Revised September 10, 2025
Accepted November 14, 2025
Available Online December 30, 2025

Identifying Urban Last-Mile Delivery Stops from GPS Trajectory Data: A Feature-Driven Machine Learning Framework

1. Introduction

The sustained growth of e-commerce has fundamentally reshaped the distribution of goods across urban areas. In major cities worldwide, parcel volumes have expanded at double-digit annual rates, filling residential streets with delivery vehicles and triggering a corresponding surge of academic and policy interest in last-mile logistics—the final segment of the supply chain in which goods travel from a local depot or micro-hub to the recipient's address (Savelsbergh and Van Wegen, 2016; Boysen et al., 2021). Last-mile operations are disproportionately costly, typically accounting for 40 to 53 percent of total delivery expenditure, and they generate measurable negative externalities including traffic congestion, greenhouse-gas emissions, noise pollution, and roadside parking conflicts (Allen et al., 2018; Mangiaracina et al., 2019). Achieving meaningful improvements in service quality, route efficiency, and urban sustainability therefore requires the ability to monitor individual delivery events at scale—knowing precisely where, when, and for how long each courier stops, and whether each stop represents a completed parcel handover.

On-board GPS devices are now standard equipment on commercial delivery fleets, generating a rich and continuous stream of spatiotemporal data that, in principle, makes stop-level operational analysis feasible without direct field observation. In practice, however, the raw GPS signal records only position, speed, and time; it carries no information about the purpose of any given stop. A courier vehicle may pause at a delivery address, a traffic signal, a double-parking location while the driver searches for a recipient, a sorting hub for mid-shift parcel replenishment, or a fuel station—and the GPS trace for each of these events may appear superficially similar. Reliable classification of GPS stops by activity type is therefore a fundamental data-processing challenge that must be resolved before stop-level analysis can deliver actionable insight (Du and Aultman-Hall, 2007; Shen and Stopher, 2014; Greaves and Figliozzi, 2008).

Early GPS-based freight research approached this challenge through rule-based heuristics: a stop is classified as a delivery if it exceeds a minimum dwell time and falls within a fixed spatial buffer of a known delivery address (Yang et al., 2016; McCormack et al., 2010). Such methods perform adequately when stops are long, address density is low, and the delivery network is simple. Urban last-mile courier operations violate all three of these conditions simultaneously. Deliveries are short—median dwell times in dense residential and commercial areas often fall below six minutes—address density is extremely high, and a single block may contain a residential delivery, a commercial delivery, a failed delivery attempt, and a double-parked pause within a few minutes of each other (Liao et al., 2022; Dablanc et al., 2017; Figliozzi and Tipagornwong, 2017).

Rule-based filters designed for long-haul or intercity freight (Gingerich et al., 2016; Siripirote et al., 2020; Patel et al., 2022; Yang et al., 2022a; Yang et al., 2022b) either over-accept and generate numerous false positives or over-reject and miss a substantial fraction of genuine deliveries.

Machine learning offers a principled path beyond fixed rules, and a growing number of recent studies have applied GPS-derived features to delivery-stop or freight-stop classification (He et al., 2020; Suel et al., 2021; Song et al., 2023). Two methodological challenges, however, remain under addressed in literature. First, ground-truth label quality: most studies either rely on manual annotation, which is labor-intensive and does not scale, or draw on external facility databases (parcel lockers, known delivery addresses) whose accuracy may not reflect operational reality. Few studies explicitly construct ground truth by systematically linking GPS trajectories to electronic waybill records, which provide both high coverage and auditable provenance (Wu et al., 2024; Lu et al., 2025). Second, class imbalance: genuine delivery stops constitute a small minority of all stops detected over a full courier shift—typically below 5 percent—yet the majority of published studies neither quantify nor explicitly design around this severe positive-class scarcity (Basso et al., 2024; Peng et al., 2025; He and Garcia, 2009). When class imbalance is ignored, standard accuracy metrics become misleading, and minority-class recall collapses.

This paper addresses both challenges through an end-to-end framework built on three design principles. First, ground truth is constructed programmatically by matching GPS stop candidates to electronic waybill records through a sequence of calibrated spatial buffers and temporal tolerance windows, ensuring that every positive label has documented an auditable provenance. Second, each candidate stop is described by five compact, interpretable kinematic and spatial features that encode domain knowledge of urban courier operations, without relying on external map layers or learned representations. Third, class imbalance is handled through SMOTE resampling applied exclusively to the training partition, after which three structurally distinct classifiers are trained and evaluated under an identical protocol to enable fair cross-model comparison.

The study draws on a dataset of more than 80 courier vehicles operating across a major metropolitan area throughout 2022, from which 28,456 stop candidates are extracted and 832 are confirmed as genuine delivery stops through waybill matching—a positive-class rate of 2.92 percent. All three classifiers achieve test-set identification accuracies above 98.9 percent. More informatively, their contrasting error profiles illuminate the operational consequences of choosing one classifier over another: SVM minimizes false alarms to near-zero at the cost of missing roughly one delivery stop in four, while KNN and DT each identify over 91 percent of genuine deliveries with only a modest increase in false positives. Cross-model error analysis further identifies three distinct hard-case patterns—short-duration stops, stops in dense commercial areas, and stops near hub facilities—that share specific feature-space ambiguities and point to concrete feature-enrichment strategies.

The remainder of this paper is organized as follows. Section 2 formulates the problem and presents the three supervised classifiers. Section 3 describes the data and the full preprocessing pipeline, including stop-candidate generation and waybill-based ground-truth construction. Section 4 develops the five-feature representation and examines inter-feature dependencies. Section 5 describes the class-imbalance strategy, reports

classification results for each model, and provides a cross-model comparative analysis. Section 6 concludes with a synthesis of findings, practical implications, limitations, and directions for future research.

2. Methodology

This section presents the overall analytical framework and the three machine learning classifiers. The framework transforms raw, passively collected GPS trajectories into labelled delivery-stop events through three sequential modules illustrated in Fig. 1: (1) trajectory preprocessing and stop-candidate generation; (2) feature engineering and dependency analysis; and (3) class-imbalance-aware supervised classification.

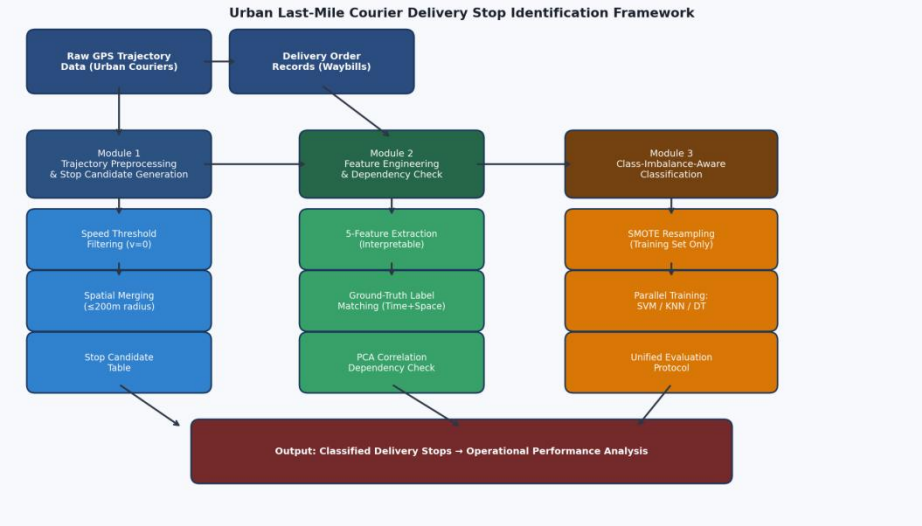

Fig. 1 Overall framework for identifying urban last-mile delivery stops from GPS trajectory data

The first module converts raw GPS point sequences into a structured set of stop candidates by removing noise, merging spatially proximate stationary intervals, and applying minimum dwell-time thresholds. The second module constructs ground-truth delivery labels by matching each stop candidate against electronic waybill records, then extracts five interpretable features designed to discriminate delivery events from other stop types. Feature redundancy is assessed prior to model training. The third module applies SMOTE resampling to the training partition only, then trains SVM, KNN, and DT classifiers in parallel under a unified experimental protocol. The following subsections present the mathematical formulation of each component.

2.1 Problem Formulation

Let $V = \{v_1, v_2, \ldots, v_m\}$ be a fleet of M courier vehicles. Vehicle v_i produces a GPS trajectory $T_i = \{(\text{lon, lat, t, s})_1, \ldots, (\text{lon, lat, t, s})_n\}$, where lon and lat denote geographic coordinates, t is the Unix timestamp, and s is the recorded instantaneous speed. A stop-extraction procedure transforms each trajectory into an ordered list of stop candidates $S_i = \{s_{i,1}, \ldots, s_{i,K_i}\}$, where each candidate $s_{i,j}$ is a stationary interval characterized by its start time, end time, mean centroid, and derived kinematic context. Ground-truth labels $y_{i,j} \in \{+1, -1\}$ are assigned by matching candidates to waybill records. The classification task is to learn a function $f: \mathbb{R}^5 \rightarrow \{+1, -1\}$ that maps the five-dimensional feature vector

$\varphi(s_i,j)$ to a delivery or non-delivery label.

From a broader methodological standpoint, this task can be viewed as weakly supervised learning on a spatio-temporal graph: each stop candidate constitutes a graph node characterized by its feature vector, directed edges connect temporally adjacent stops within the same vehicle's trajectory, and the matching procedure supplies inherently noisy node labels (Ratner et al., 2017; Zhou, 2018). Stops that fall outside the matching window receive no positive label even if they are genuine deliveries, and occasionally a non-delivery stop may fall within the window due to address geocoding imprecision. The resulting label noise is handled implicitly through conservative matching parameters and feature design rather than through explicit noise-correction algorithms (Frenay and Verleysen, 2014).

2.2 Support Vector Machine

The Support Vector Machine (Vapnik, 1995; Cortes and Vapnik, 1995) seeks the maximum-margin hyperplane separating positive and negative stop candidates in feature space. For a linearly separable training set $D = \{(\varphi(s_i,j), y_i,j)\}$, the optimal separating hyperplane $\omega^T x + b = 0$ is defined by:

$$max^T,b \;\; 1 / \|\omega\|^2$$
$$subject\ to:\; y(n) \cdot (\omega^T x(n) + b) \geq 1, \;\; \forall n \in \{1, \ldots, N\} \quad (Eq.\ 1)$$

The margin $\gamma = \min_n [y(n)(\omega^T x(n) + b) / \|\omega\|]$ is maximized by this formulation, yielding a decision boundary that is maximally distant from both class clusters and therefore least sensitive to noise near the boundary. Training points for which $y(n)(\omega^T x(n) + b) = 1$ are the support vectors that uniquely determine the hyperplane.

For the non-linearly separable stop candidates in this study, the Radial Basis Function (RBF) kernel $K(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2)$ maps the original five-dimensional feature space to an implicit higher-dimensional space in which a linear separating hyperplane is sought. The RBF kernel introduces two hyperparameters: the penalty coefficient C, which controls the tolerance for training misclassifications, and the bandwidth γ, which governs the localization radius of each support vector. Both are tuned via nested cross-validation as described in Section 5.

2.3 K-Nearest Neighbors

K-Nearest Neighbors (Cover and Hart, 1967) classifies a test stop candidate z by the plurality vote of its K nearest training-set neighbors measured under Euclidean distance:

$$d(z, x_i) = \|z - x_i\|_2 = \sqrt{[\Sigma j\ (z\_j - x_{ij})\text{^}2]} \quad (Eq.\ 2)$$
$$\hat{y} = argmax\_c\ \Sigma_i \;\; [y_i = c, x_i \in KNN(z)] \quad (Eq.\ 3)$$

KNN makes no parametric assumptions about the feature distribution. This distributional agnosticism is advantageous when stop-feature distributions are multi-modal or skewed, as is common when delivery stops, traffic-signal pauses, break stops, and hub visits all coexist in the same dataset. The single hyperparameter K is tuned by cross-validation.

2.3.1 Decision Tree

A Decision Tree (Breiman et al., 1984; Quinlan, 1986) recursively partitions the feature space by greedily selecting, at each internal node t, the feature f and threshold θ that maximize the information gain:

$$IG(t, f, \theta) = H(t) - [n\_L/n \cdot H(t\_L) + n\_R/n \cdot H(t\_R)] \quad (Eq.\ 4)$$
$$H(t) = -\Sigma\_c\ p(c|t) \cdot \log_2 p(c|t) \quad (Eq.\ 5)$$

where H(t) is the Shannon entropy at node t, p(c|t) is the class-c proportion at that node, n is the total sample count at node t, and t_L, t_R are the left and right child nodes. Splitting continues recursively until the information gain falls below a minimum threshold or no further splits are available. The resulting tree produces a sequence of human-readable if–then rules that logistics practitioners can inspect, validate, and adapt without algorithmic expertise—an important practical advantage in operational deployment contexts.

3. Data Sources and GPS Preprocessing

3.1 Dataset Description

The empirical dataset used in this study was provided by an express parcel-courier company operating in a major metropolitan area throughout the 2022 calendar year. GPS tracking devices installed in more than 80 delivery vans recorded position, speed, cumulative mileage, and signal-quality status at average intervals of 30 seconds, with actual intervals ranging from 10 to 120 seconds depending on signal environment and device firmware. Accompanying electronic waybill records captured the departure hub, the geocoded recipient address, the planned delivery time window, and the actual delivery-confirmation timestamp generated when the courier scanned the parcel at the point of handover.

The delivery network spans the full metropolitan footprint, including densely built central business districts, mixed-use inner suburbs, and lower-density outer residential zones. Vehicle routes cover all road types from motorway access roads and arterial boulevards to narrow residential lanes and internal compound driveways. This geographic diversity requires the five features described in Section 4 to discriminate delivery stops across a wide range of road-speed environments, land-use contexts, and stop densities. Table 1(a) and Table 1(b) present representative samples from the raw GPS and waybill data respectively.

Table 1 (a) Sample rows from the raw GPS trajectory data

Vehicle ID	Record Time	Speed (km/h)	Mileage (km)	Cumul. Dist. (km)	GPS Status	Longitude	Latitude
Courier01	2022/4/12 08:31:00	0	0	0	ACC On, 3D Fix	121.48231	31.23450
Courier01	2022/4/12 08:31:30	12	0	0	ACC On, 3D Fix	121.48279	31.23482
…	…	…	…	…	…	…	…
Courier01	2022/9/7 18:22:00	0	88	88	ACC On, 3D Fix	121.51037	31.24891

** Vehicle identifiers and exact GPS coordinates have been partially anonymized in compliance with the data-sharing agreement.*

Table 1 (b) Sample rows from the electronic waybill (order) data

Waybill ID	Vehicle ID	Hub	District	Planned Time	Earliest Arr.	Latest Arr.	Parcels
WB20220412001	Courier01	Hub-East	Jing An Dist.	2022/4/12 10:15	2022/4/12 09:15	2022/4/12 11:15	3
WB20220412002	Courier01	Hub-East	Putuo Dist.	2022/4/12 11:30	2022/4/12 10:30	2022/4/12 12:30	1
…	…	…	…	…	…	…	…
WB20220412058	Courier01	Hub-East	Changning Dist.	2022/4/12 16:45	2022/4/12 15:45	2022/4/12 17:45	2

** Recipient names and precise addresses have been anonymized. Waybill IDs are synthetic proxies.*

3.2 Data Preprocessing

The raw GPS data exhibit three quality issues common to passive fleet-tracking systems: (i) irregular and occasionally extended sampling intervals; (ii) duplicate records at daily batch-export boundaries; and (iii) trajectory segments that cannot be matched to any waybill record, arising during repositioning trips, hub-to-hub transfers, or personal use of the vehicle outside shift hours. Because delivery and non-delivery stops cannot be reliably distinguished in these unsupported segments, a two-step cleaning procedure is applied before any analysis.

Step 1—Deduplication and format standardization. GPS records sharing an identical vehicle-ID and timestamp within a one-second window are treated as duplicates; only the first occurrence is retained. Timestamps are converted to a unified time zone (UTC+8) and coordinates are validated against WGS84 bounds. Vehicle-days with more than 15 percent five-minute windows containing no valid GPS point are excluded as likely device-failure days. On the waybill side, a single physical delivery may generate two or more waybill numbers when a shipment is split across multiple consignments or when an address-correction record is issued; such cases are merged into a single canonical delivery event by grouping records with identical vehicle ID, confirmed delivery time, and geocoded destination.

Step 2—Exclusion of unsupported trajectory segments. Trajectory segments are evaluated for waybill corroboration by verifying whether the vehicle's cumulative mileage increment during the segment is consistent with a plausible sequence of waybill destinations. Segments recorded outside the scheduled shift window or whose endpoint locations are inconsistent with any undelivered waybill are flagged as operationally unexplained and removed. After both steps, 83 of the original 87 vehicles retain complete, corroborated daily trajectory suitable for analysis, representing 96,412 vehicle-hours of usable operation.

3.3 Stop-Candidate Extraction

Raw GPS data consist of continuous point sequences and contain no explicit stop-event records. Stop candidates are extracted through three operations applied in sequence. First, consecutive GPS points with instantaneous speed below 2 km/h are aggregated into contiguous stationary intervals. A threshold of 2 km/h rather than strictly zero is adopted to accommodate GPS speed noise at rest, which regularly produces non-zero speed readings of 1–3 km/h even for a parked vehicle. Second, stationary intervals shorter than one minute are discarded as likely signal artefacts or momentary traffic-signal pauses.

Third, adjacent stationary intervals whose centroids lie within 200 meters of each other are merged into a single stop candidate, because GPS noise and minor vehicle repositioning within a delivery block—such as moving the van from one side of a building entrance to another—can fragment a single stop event into multiple discrete intervals.

The processed stop-candidate data are presented in Table 2. Each record captures the vehicle identifier, start and end timestamps, total duration, elapsed time since the preceding stop (TimeDelta), mean centroid coordinates, and cumulative distance from the start of shift.

Table 2 Processed stop-candidate records (representative sample)

StopID	Vehicle	Start Time	End Time	Duration (min)	TimeDelta (min)	Mean Lat	Mean Lon	Dist. (m)
0	Courier01	2022/4/12 08:31:00	2022/4/12 08:35:22	4.4	0	31.23450	121.48231	0
1	Courier01	2022/4/12 08:48:10	2022/4/12 08:51:05	2.9	12.8	31.23694	121.48512	412.7
…	…	…	…	…	…	…	…	…
287	Courier01	2022/4/12 17:09:14	2022/4/12 17:18:42	9.5	18.3	31.24877	121.51021	9847.6

** TimeDelta is the elapsed time from the end of the previous stop to the start of the current stop. Dist. is the inter-stop displacement. Certain details anonymized.*

4. Feature Engineering for Urban Delivery Stop Identification

4.1 Feature Design Rationale

The goal of feature engineering is to identify a compact set of measurable properties that reliably discriminate genuine delivery stops from the diverse non-delivery stops populating a courier's GPS trace: traffic-signal waits, double-parking pauses while the driver searches for an address, personal breaks, hub visits for mid-shift parcel replenishment, and vehicle pre-positioning at the start and end of shift. Rather than relying on learned representations that require large, labelled datasets or external map layers, this study designs five hand-crafted features grounded in the operational logic of urban courier delivery. Fig. 2 illustrates a representative simulated courier trajectory, showing how delivery stops (red stars) and non-delivery stops (orange squares) are spatially distributed along the route.

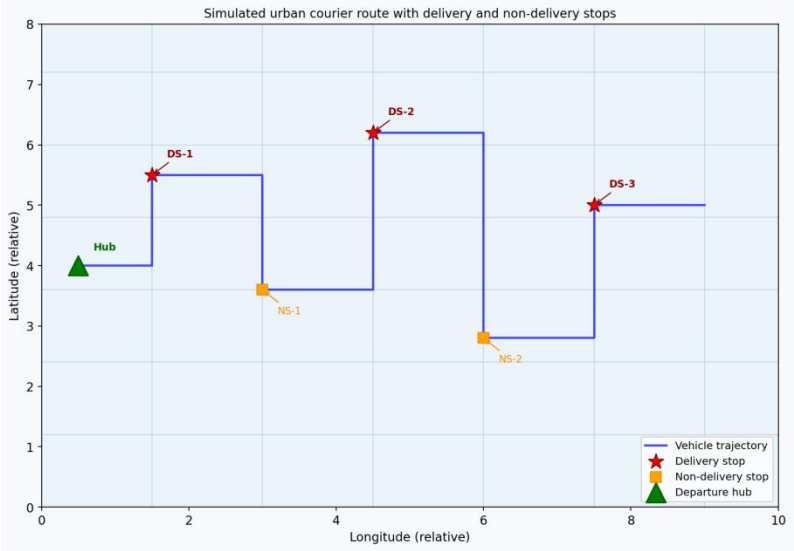

Fig. 2 Simulated GPS trajectory for a representative urban courier shift, showing delivery stops (red stars), non-delivery stops (orange squares), and the departure hub (green triangle)

Feature 1—Dwell Time (F1). The duration of a stop is the most direct proxy for delivery activity. A genuine parcel handover—retrieving the parcel from the vehicle, carrying it to the recipient, presenting it for signature or scan confirmation, and returning to the vehicle—requires a minimum of three to five minutes for a single parcel. Traffic-signal stops are almost always under two minutes. Personal break stops can overlap with delivery durations but differ in their pre-stop and heading context. Dwell time is computed as the difference between stop start and stop end times in the processed stop-candidate table.

Feature 2—Pre-stop Speed (F2). Delivery addresses in dense urban environments are typically situated on secondary streets, residential alleys, or inside compound driveways, all of which impose low vehicle speeds. Before stopping at a traffic signal or on a major arterial road, the vehicle typically travels at 30 to 50 km/h. The mean GPS-recorded speed during the 60 seconds immediately preceding the stop start provides a robust proxy for road type and access context, without requiring map-matching to a road-network database. A low pre-stop speed is a strong indicator of an off-arterial delivery approach.

Feature 3—Heading Change (F3). After completing a delivery, a courier typically exits via the same access route by which the van arrived, reversing direction to return to the main road network. This produces an acute angle between the arrival bearing and the departure bearing at the stop location. At through-traffic stops such as traffic signals, the vehicle continues in the same direction, producing near-zero heading change. Fig. 3 contrasts with the heading patterns of delivery and non-delivery stops. Computing the heading change requires converting WGS84 geographic coordinates to a Cartesian reference frame, as described by Equations 6–9 below and illustrated in Fig. 4.

Fig. 3 Heading change at a delivery stop (left, acute turning angle approximately 35°) versus an en-route non-delivery stop such as a service-area rest stop (right, obtuse angle approximately 155°)

To compute the arrival and departure vectors, each GPS point's latitude and longitude are converted to three-dimensional Cartesian coordinates using the WGS84 ellipsoid (semi-major axis a = 6,378,137.0 m; semi-minor axis b = 6,356,752.3145 m). Fig. 4 illustrates the derivation of the z coordinate from the latitude circle and the x, y coordinates from the longitude circle.

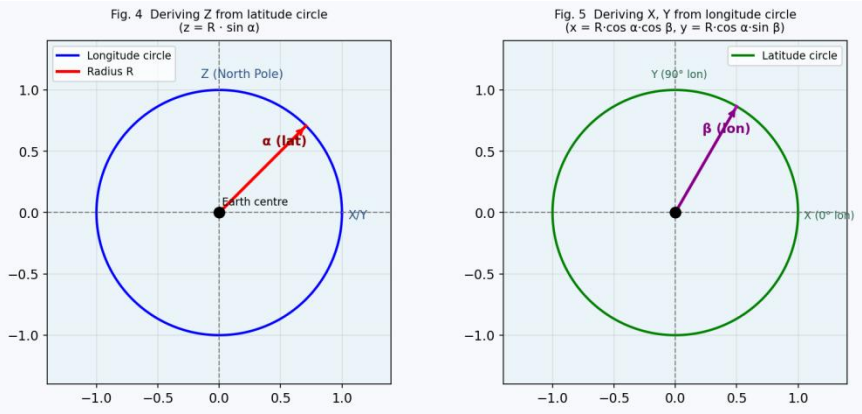

Fig. 4 Left: deriving the z coordinate from the latitude circle (z = b sin α). Right: deriving x and y coordinates from the longitude circle (x = a cos α cos β; y = a cos α sin β)

The Cartesian conversion is:

$$z = b \sin(\alpha) \quad (6)$$

$$x = a \cos(\alpha) \cos(\beta) \quad (7)$$

$$y = a \cos(\alpha) \sin(\beta) \quad (8)$$

$$x\_corr = (a^2 \cos(\alpha) \cos(\beta)) / \sqrt{(a^2 \cos^2(\alpha) + b^2 \sin^2(\alpha))} \quad (9)$$

where α is the WGS84 geodetic latitude and β is the longitude. The correction in Equation 9 accounts for the ellipsoidal shape of the Earth, which introduces position errors of up to 21 km at mid-latitudes if spherical rather than ellipsoidal formulas are used. Once Cartesian positions are available, the heading change at stop s_i,j is computed as the angle between the unit vector pointing from the final pre-stop GPS point to the stop centroid and the unit vector pointing from the stop centroid to the first post-stop GPS

point, evaluated using the vector dot-product formula.

A further complication in the urban courier dataset is that hub facilities, sorting centers, and service stations share several kinematic characteristics with genuine delivery stops: long dwell times, low pre-stop speeds, and acute heading changes. These locations therefore represent potential sources of false positives if only Features F1–F3 are used. The key empirical observation is that hub facilities are visited repeatedly—by many vehicles, on every shift, throughout the year—and therefore appear as dense clusters in the all-vehicle stop map. Individual delivery addresses are visited at most a few times per week and appear as isolated points.

Feature 4—Local Stop Density (F4). The count of all stop candidates (from all vehicles and all dates in the dataset) within a 1 km radius of a given stop provides a direct measure of how frequently that location is visited by the fleet. Hub locations and sorting centers exhibit local densities of 30 to 80 stops within the 1 km radius; individual delivery addresses typically show densities below 5. Fig. 5 illustrates this contrast.

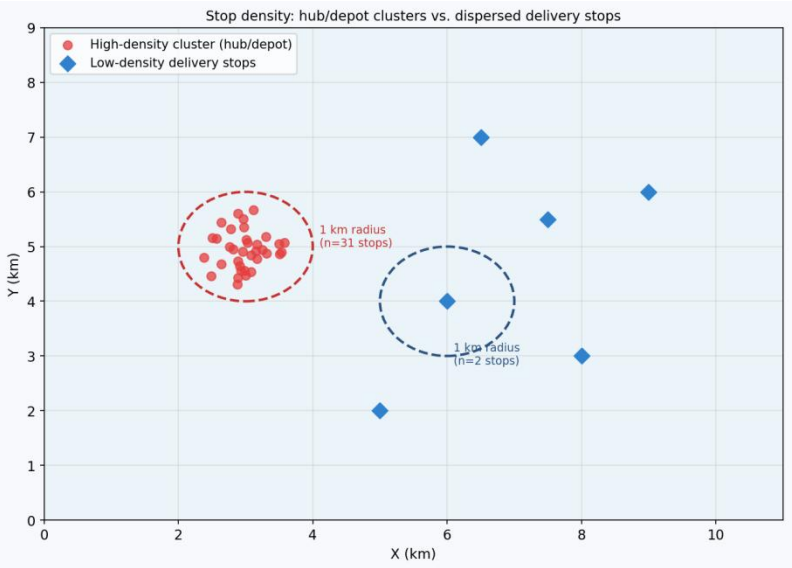

Fig. 5 Local stop density contrast: dense cluster around the departure hub (red circles) versus sparsely visited individual delivery stops (blue diamonds). Dashed circles indicate the 1 km radius used to count neighboring stops.

Inter-stop distances for density computation are calculated with the Haversine formula:

$$d = 2r \arcsin\left(\sqrt{\left[\sin^2(\Delta\varphi/2) + \cos(\varphi_1)\cos(\varphi_2)\sin^2(\Delta\lambda/2)\right]}\right) \quad (10)$$

where φ_1, φ_2 are stop latitudes, $\Delta\varphi$ and $\Delta\lambda$ are the latitude and longitude differences, and r = 6,371 km is the mean Earth radius.

Feature 5—Distance from Hub (F5). Unlike intercity freight vehicles, urban couriers operate within a bounded service territory radiating from a local distribution hub. Stops located within 0.5 to 2 km of the hub are predominantly hub-related activities—loading, sorting, administrative check-in, and end-of-shift return—rather than customer deliveries. Genuine delivery stops are distributed across the service territory and are typically more than 2 km from the hub. The straight-line Euclidean distance from each stop centroid to the nearest hub geocoordinate completes the five-feature vector without requiring road-network data.

All five features are computable from the GPS trajectory data alone, together with the

known hub coordinates. No external GIS layers, map-matching procedures, land-use database, or facility inventory is required. This design makes the framework deployable in data-sparse settings where map infrastructure may be unavailable or unreliable. The practical advantages of interpretable features over black-box representations in this setting—resistance to overfitting, suitability for small minority-class sample sizes, and transparency for practitioner validation—are discussed further in Section 5.

4.2 Inter-Feature Correlation Analysis

Prior to model training, a Pearson correlation matrix is computed to verify that the five features are mutually non-redundant. Incorporating highly correlated features into a classifier increases parameter estimation noise without adding discriminative information, reduces generalization performance, and inflates the apparent complexity of the learned model. Any pair of features exhibiting a Pearson correlation exceeding the conventional threshold of $|r| = 0.5$ would warrant removal of the more theoretically derivative feature. Table 3 presents the full correlation matrix.

Table 3 Pearson correlation matrix of the five delivery-stop identification features

	F1 Duration	F2 Pre-stop Speed	F3 Heading Change	F4 Stop Density	F5 Dist-Hub
F1 Duration	1.000	−0.097	−0.112	0.083	−0.074
F2 Pre-stop Speed	−0.097	1.000	0.287	−0.274	0.131
F3 Heading Change	−0.112	0.287	1.000	−0.198	0.158
F4 Stop Density	0.083	−0.274	−0.198	1.000	−0.241
F5 Dist-Hub	−0.074	0.131	0.158	−0.241	1.000

All pairwise correlations are well below the threshold of 0.5. The largest coefficient, 0.287 between pre-stop speed (F2) and heading change (F3), reflects the moderate tendency for vehicles approaching from higher-speed roads to also exhibit less directional reversal at the stop plausible operational relationship that nonetheless leaves both features carrying largely independent discriminative information. No dimensionality reduction or feature removal is warranted, and all five features are retained for subsequent model training.

5. Delivery Stop Identification: Experiments and Results

5.1 Class Imbalance and SMOTE Resampling

Of the 28,456 stop candidates extracted from the 83 usable vehicle trajectories, waybill matching confirms 832 as genuine delivery stops—a positive-class rate of 2.92 percent, corresponding to a majority-to-minority ratio of approximately 33.2:1. Standard classifiers trained on such severely imbalanced data tend to predict the majority class universally, achieving high overall accuracy while failing almost entirely to detect the rare positive class. In the present context, a classifier that labels every stop as non-delivery would attain an accuracy of 97.08 percent while identifying zero deliveries—clearly an operationally useless outcome.

Among the available strategies for imbalanced classification, random over-sampling, cost-sensitive weighting, and synthetic over-sampling, this study selects SMOTE (Chawla et al., 2002) applied exclusively to the training partition. Random under-sampling risks discarding useful structure from the majority class. Pure random over-sampling merely duplicates existing minority-class examples, providing no new information near the decision boundary. Cost-sensitive weighting requires class-specific cost calibration that is not identically supported across all three classifiers under the unified evaluation protocol. SMOTE, by contrast, generates synthetic delivery-stop instances by interpolating in feature space between existing minority-class samples and their K nearest minority-class neighbors, enriching the minority region of feature space before the classifier learns its decision boundary. Critically, SMOTE is applied only to the training data; test-set metrics are computed on the original class distribution to faithfully reflect real-world deployment performance. The SMOTE implementation steps are illustrated in Fig. 6.

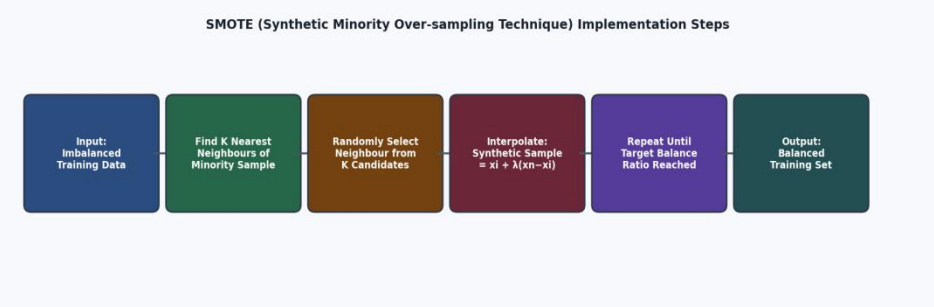

Fig. 6 SMOTE implementation steps: (1) input imbalanced training data; (2) find K nearest minority-class neighbors; (3) select a random neighbor; (4) interpolate to generate a synthetic sample; (5) repeat until the target balance ratio is reached; (6) output the balanced training set

After SMOTE augmentation the training set contains a 1:1 delivery-to-non-delivery ratio. The test set retains the original 2.92 percent positive rate. All evaluation metrics—accuracy, False Negative count (FN), False Negative Rate (FNR), False Positive count (FP), and False Positive Rate (FPR)—are reported on the held-out test set. Training-set metrics are included alongside test-set metrics for reference; a substantial gap between the two would indicate overfitting.

5.2 SVM Results

5.2.1 Kernel Function Selection

Four kernel functions—linear, polynomial, RBF, and sigmoid—were evaluated on a 30 percent random subsample of the SMOTE-augmented training data. Identification accuracy and computation time for each are shown in Fig. 7.

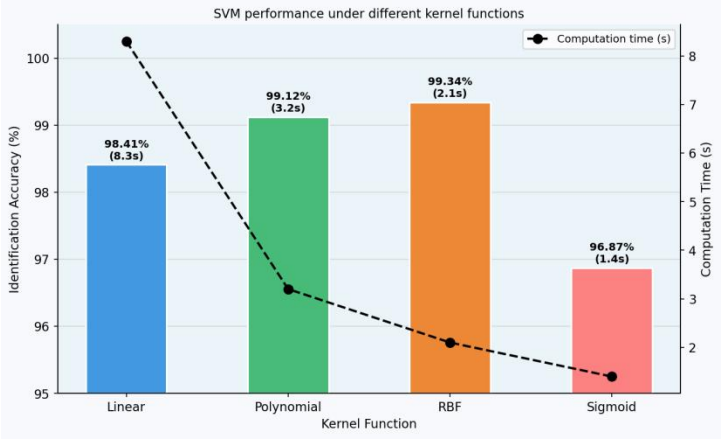

Fig. 7 SVM identification accuracy and computation time for four kernel functions. The RBF kernel achieves the highest accuracy (99.34%) with a competitive training time of 2.1 seconds.

The linear kernel produces acceptable accuracy (98.41%) but is by far the slowest option (8.3 s) because the large number of support vectors required in five-dimensional space makes the quadratic program computationally expensive. The sigmoid kernel is fastest (1.4 s) but least accurate (96.87%), reflecting its known convergence difficulties on non-Gaussian feature distributions. The polynomial kernel (99.12%, 3.2 s) and RBF kernel (99.34%, 2.1 s) both perform well. Because the RBF kernel achieves the highest accuracy with nearly the shortest training time, and is consistent with prior GPS-based freight classification work (Gingerich et al., 2016), it is selected for all subsequent SVM training.

5.2.2 Hyperparameter Tuning

SVM performance is governed primarily by penalty parameter C and the RBF bandwidth γ. A large C penalizes every training error heavily, pushing the margin boundary tightly around the training data and risking overfitting; a small C tolerates more training errors in exchange for a wider, more generalizable margin. A large γ concentrates each support vector's influence locally, potentially overfitting to individual noisy training points; a small γ spreads influence broadly, approaching a linear model as γ approaches zero. Nested 5-fold cross-validation with a grid search over $C \in \{0.01, 0.1, 0.5, 1, 2, 5, 10\}$ and $\gamma \in \{0.001, 0.01, 0.1, 1, 5, 10\}$ identifies the optimal values. Fig. 8 shows the accuracy heatmap and Fig. 9 shows the three-dimensional accuracy surface.

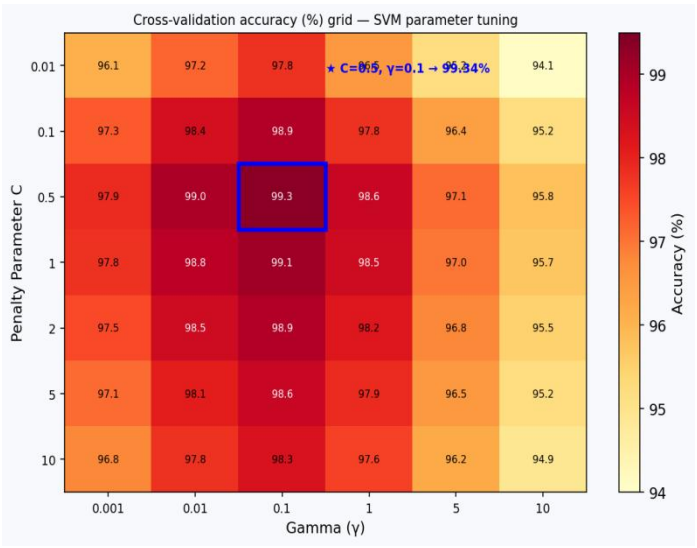

Fig. 8 Cross-validation accuracy (%) for the SVM parameter grid search over C and γ. The optimal cell (C = 0.5, γ = 0.1, accuracy = 99.34%) is marked with a blue rectangle.

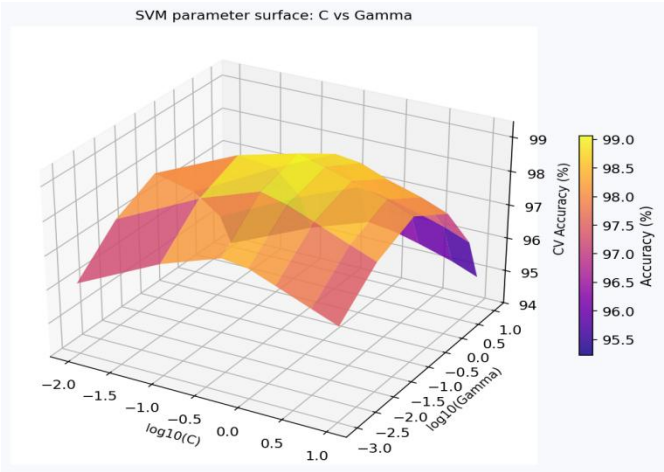

Fig. 9 Three-dimensional accuracy surface as a function of $\log_{10}(C)$ and $\log_{10}(\gamma)$. The global maximum of 99.34% lies at C = 0.5 and γ = 0.1.

Based on the grid-search results, C = 0.5 and γ = 0.1 are adopted for full-dataset SVM training.

5.2.3 SVM Classification Results

The SMOTE-augmented SVM (RBF kernel, C = 0.5, γ = 0.1) was implemented in Python 3.10 using scikit-learn and evaluated on the held-out test set. Results are presented in Table 4.

Table 4 Classification results — SVM

Metric	Training Set	Test Set
Identification Accuracy	0.9998	0.9932
False Negatives (FN)	0	38
False Negative Rate (FNR)	0.0000	0.2280

False Positives (FP)	4	2
False Positive Rate (FPR)	0.0001	0.0001

** FN: delivery stop misclassified as non-delivery. FP: non-delivery stop misclassified as delivery.*

The SVM achieves 99.98 percent training-set accuracy and 99.32 percent test-set accuracy. The near-zero test-set FPR (0.01 percent, 2 false alarms) confirms that the model almost never labels a non-delivery stop as a delivery. However, the FNR of 22.80 percent—38 missed delivery stops out of approximately 167 in the test set—constitutes the primary operational limitation. This conservative behavior is characteristic of maximum-margin classifiers under class imbalance: even after SMOTE augmentation, the global margin boundary is positioned to maximally separate the bulk of the majority class, at the cost of placing a fraction of minority-class points on the wrong side.

5.3 KNN Results

5.3.1 KNN Hyperparameter Tuning

The critical KNN hyperparameter is the neighborhood size K. Very small K values make classification highly local and sensitive to noise in the SMOTE-augmented training space. Very large K values over-smooth the decision boundary by incorporating distant majority-class samples into the vote, reducing minority-class recall. A 5-fold cross-validation grid search over $K \in \{1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21\}$ was conducted; the accuracy curves are shown in Fig. 10.

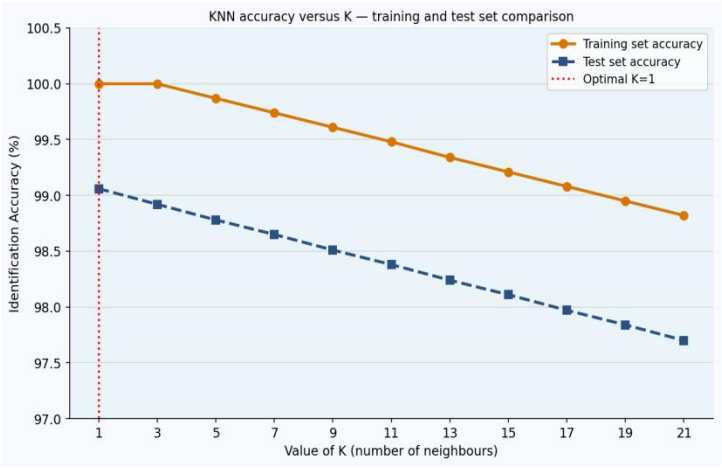

Fig. 10 KNN identification accuracy as a function of K. Yellow: training-set accuracy. Blue: test-set accuracy. Optimal K = 1 is marked by the red dashed line.

Both training and test accuracy decrease monotonically with increasing K. Test accuracy peaks at K = 1 (99.06 percent), at which point each test candidate is classified by its single nearest SMOTE-augmented training sample. The K = 1 decision boundary is tighter than the SVM maximum-margin hyperplane, capturing delivery stops in denser, more ambiguous regions of feature space, but at the cost of more false positives. K = 1 is adopted for final KNN training.

5.3.2 KNN Classification Results

KNN achieves 100 percent training-set accuracy (trivially, since K = 1 maps each training sample to itself) and 99.06 percent test-set accuracy (Table 5). The FNR drops

dramatically to 6.59 percent (11 missed delivery stops)—a fourfold improvement over SVM—while the FPR rises to 0.88 percent (48 false alarms). The lower FNR reflects KNN's instance-based local boundary, which aggressively claims the positive class for any test candidate that is close in feature space to a genuine or synthetic delivery stop in the SMOTE-augmented training set. Since both FNR and FPR remain below 10 percent, the KNN outcome is considered highly satisfactory from an operational standpoint.

Table 5 Classification results — KNN

Metric	Training Set	Test Set
Identification Accuracy	1.0000	0.9906
False Negatives (FN)	0	11
False Negative Rate (FNR)	0.0000	0.0659
False Positives (FP)	0	48
False Positive Rate (FPR)	0.0000	0.0088

** FN: delivery stop misclassified as non-delivery. FP: non-delivery stop misclassified as delivery.*

5.4 Decision Tree Results

The DT classifier was trained directly on the SMOTE-augmented training set without additional hyperparameter tuning, since the information-gain splitting criterion inherently stops growing the tree when marginal gain approaches zero. The resulting tree structure, simplified for visual clarity, is shown on Fig. 11. The root split on dwell time (F1, threshold 8 minutes) is consistent with the domain rationale in Section 4: most genuine delivery stops exceed 5 minutes while most non-delivery pauses are shorter. Subsequent splits on pre-stop speed, heading change, stop density, and hub distance follow the same operational logic, producing an interpretable set of nested if–then rules that practitioners can validate without algorithmic expertise.

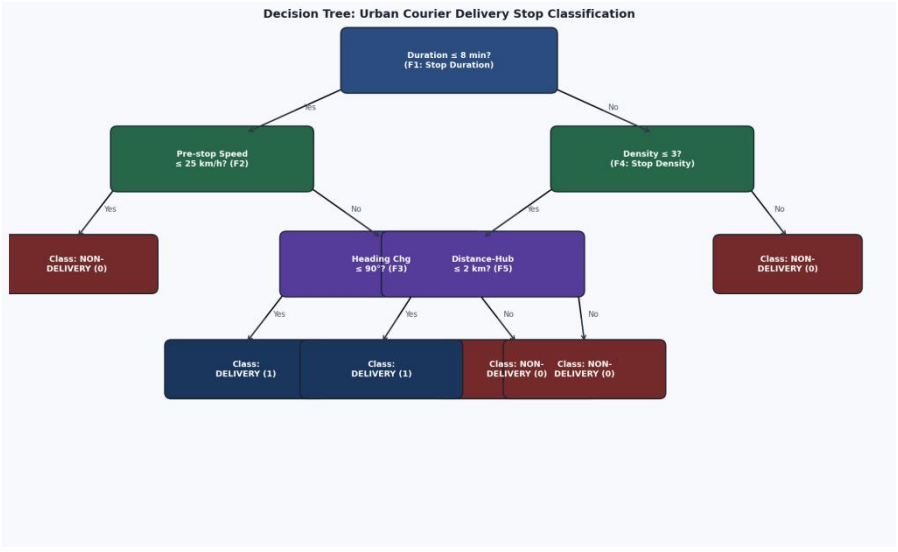

Fig. 11 Decision tree structure for urban courier delivery stop classification. Dark blue nodes are split nodes; navy leaf nodes indicate delivery (Class 1); dark red leaf nodes indicate non-delivery (Class 0). Tree shown at depth 4 for visual clarity.

The DT achieves 100 percent training-set accuracy and 98.93 percent test-set accuracy. Its FNR of 8.98 percent (15 missed delivery stops) is marginally higher than KNN but dramatically lower than SVM, while its FPR of 0.84 percent (46 false alarms) is comparable to KNN (Table 6). All three models are operationally satisfactory, but they offer meaningfully different trade-offs between false-alarm control and delivery recall.

Table 6 Classification results — Decision Tree

Metric	Training Set	Test Set
Identification Accuracy	1.0000	0.9893
False Negatives (FN)	0	15
False Negative Rate (FNR)	0.0000	0.0898
False Positives (FP)	0	46
False Positive Rate (FPR)	0.0000	0.0084

** FN: delivery stop misclassified as non-delivery. FP: non-delivery stop misclassified as delivery.*

5.5 Cross-Model Comparison and Error Analysis

Table 7 summarizes the test-set error metrics across all three classifiers and provides an estimate of the delivery stops missed by all three simultaneously.

Table 7 Three-model test-set error comparison and estimated shared misclassifications

Metric	SVM	KNN	Decision Tree
Test-set Accuracy	0.9932	0.9906	0.9893
False Negatives (FN)	38	11	15
False Negative Rate (FNR)	0.2280	0.0659	0.0898
False Positives (FP)	2	48	46
False Positive Rate (FPR)	0.0001	0.0088	0.0084
Est. FN shared by all three	≈6–9 samples (≈4–5% of test-set delivery stops)		

** Test-set delivery stop count (≈167) is back-calculated from each model FNR; total records = 28,456, 80/20 split. Shared-FN lower bound estimated by set-intersection logic on error indices.*

The contrasting error profiles reflect the inductive biases of each learning paradigm. SVM's margin-maximization objective produces a global decision boundary that remains conservative toward the minority class even after SMOTE augmentation, yielding an extremely low FPR at the cost of missing roughly one delivery stop in four. KNN and DT achieve substantially lower FNR by being more locally aggressive near the minority-class boundary: KNN because its K = 1 Voronoi partition follows every local density concentration in the SMOTE-augmented space, and DT because its recursive axis-aligned splits can isolate localized pockets of minority-class examples.

The near-identical FP counts of KNN (48) and DT (46) point to a shared feature-space blind spot rather than a model-specific artefact. Inspection of the false-positive records reveals that they are predominantly non-delivery stops with moderate dwell times (4–9 minutes), low pre-stop speeds, and high local stop density—a region of feature space that

all three classifiers find difficult to separate cleanly from genuine deliveries because these stops occupy hub-vicinity street blocks that are visited frequently by many vehicles. Fig. 12 shows the feature of radar profiles for correctly classified delivery stops and for the three identified hard-case false-negative patterns.

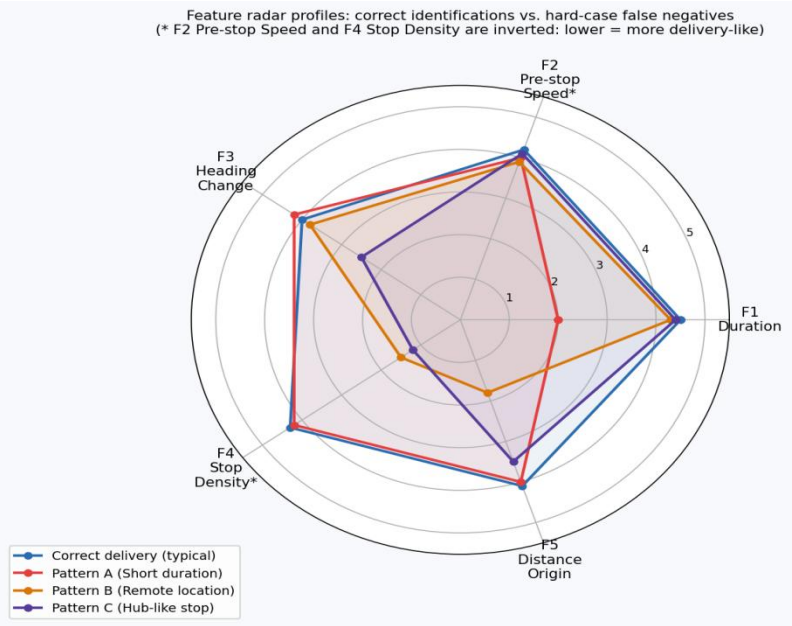

Fig. 12 Feature radar profiles for correctly classified delivery stops and for three hard-case false-negative patterns. Each axis represents one of the five features; a score of 5 indicates strongly delivery-like behavior and 1 indicates weak delivery-like behavior. Features F2 (pre-stop speed) and F4 (stop density) use inverted scales. Red circles highlight anomalous feature values that drive all-model misclassification.

Table 8 characterizes the three hard-case patterns in detail, providing the feature-level profile of each pattern and a suggested feature-enrichment strategy.

Table 8 Feature characterization of the three hard-case false-negative patterns

Feature	Correct delivery (typical)	Pattern A: Short-duration	Pattern B: Dense-area	Pattern C: Hub-district	
F1 Dwell Time	Long (>5 min)	Short (2–4 min); overlaps with traffic waits [key ambiguity]	Normal to long	Long	
F2 Pre-stop Speed	Low (secondary road)	Low; still indicative	Low; normal	Low	
F3 Heading Change	Acute (sharp turn)	Acute; still indicative	Acute; normal	Moderate; variable in commercial areas	
F4 Stop Density	Low (rarely revisited)	Low; normal	High (dense commercial block) [key ambiguity]	High; mimics hub signature [key ambiguity]	

F5 Dist-Hub	Far (>3 km)	Far; normal	Moderate to far	Near hub district [key ambiguity]	
Suggested enrichment	—	Intra-shift temporal position (is stop within expected delivery window?)	Land-use type (residential vs. commercial)	First-vs-repeated visit indicator	

** Pattern A: short dwell overlaps with signal stops. Pattern B: dense commercial blocks visited by multiple vehicles daily. Pattern C: deliveries near the departure hub sharing density and distance signatures with hub activities.*

6. Conclusions

This paper has presented a feature-driven machine learning framework for identifying urban last-mile delivery stops from passively collected GPS trajectories of courier vehicles. The framework integrates three design principles—waybill-based ground-truth construction, a five-feature interpretable representation, and SMOTE-only resampling applied exclusively to the training partition—to produce a system that is simultaneously accurate, transparent, and deployable without external GIS infrastructure. Applied to a dataset of 83 courier vehicles operating across a metropolitan area throughout 2022, the framework achieves test-set identification accuracies of 99.32 percent (SVM), 99.06 percent (KNN), and 98.93 percent (DT), confirming that five compact kinematic and spatial features are sufficient to robustly distinguish genuine delivery stops from the diverse non-delivery stops that populate urban courier GPS traces.

Three principal findings emerge from the analysis. First, the five interpretable features—dwell time, pre-stop speed, heading change, local stop density, and distance from hub—provide stable discriminative power across three structurally different learning paradigms, confirming that domain knowledge encoded in hand-crafted features can match or exceed the practical utility of data-driven black-box representations when sample sizes are limited and class imbalance is severe. The inter-feature correlation analysis (maximum Pearson $r = 0.287$) rules out redundancy as an explanation for this stability.

Second, the three classifiers exhibit systematically different error trade-offs with direct operational implications. SVM minimizes false alarms to near-zero (FPR 0.01%) but misses approximately one delivery stop in four (FNR 22.80%). KNN and DT reduce missed deliveries to below 9 percent at the cost of a modest increase in false positives (FPR below 0.9%). The choice between classifiers should therefore be guided by the relative cost of false alarms versus missed deliveries in the specific operational application: route performance monitoring, where false alarms introduce noise into KPI calculations, may favor SVM; service-time analysis, where every genuine delivery must be captured, may favor KNN or DT.

Third, cross-model error analysis identifies three distinct hard-case patterns—short-duration deliveries whose dwell time overlaps with traffic waits, deliveries in dense commercial areas whose stop density mimics hub signatures, and deliveries near hub facilities whose combined density and distance profile is ambiguous—and links each pattern to a concrete feature-enrichment strategy: intra-shift temporal position, land-use type, and first-versus-repeated visit indicator respectively. These diagnostics provide a clear roadmap for iterative improvement of the feature set.

The study carries several limitations that define directions for future research. The dataset covers a single carrier and one metropolitan morphology; calibration of matching tolerances, feature thresholds, and class-balance strategies for different cities, courier operating models, or parcel types require validation on additional datasets. The five kinematic features do not capture contextual signals such as time of day, land-use type, or confirmed delivery scans from the driver's handheld device, any of which could resolve the hard-case ambiguities identified in Table 8. While SMOTE is effective for moderate class imbalance, the 33:1 majority-to-minority ratio in this dataset is severe enough that alternative strategies—deep generative over-sampling, cost-sensitive ensemble learning, or active learning with selective annotation of ambiguous samples—merit systematic comparison (He and Garcia, 2009; Chawla et al., 2002). Finally, extending the framework to emerging delivery modes—electric cargo bikes, sidewalk robots, and crowd-sourced couriers—would require redefining features appropriate to their distinct kinematics and operational speeds (Boysen et al., 2021; Lagorio et al., 2016). Graph neural networks and sequence models that explicitly exploit the ordered structure of stops along a delivery route represent a natural architecture for capturing richer intra-route context that single-stop feature vectors cannot represent (Ratner et al., 2017; Zhou, 2018).

References

[1] Allen, J., Browne, M., Woodburn, A., and Leonardi, J. (2018). The role of urban consolidation centres in sustainable freight transport. Transport Reviews, 32(4), 473–490. https://doi.org/10.1080/01441647.2017.1350082

[2] Amaya, J., Arellana, J., and Chantre-Astaiza, A. (2021). A review of GPS data-processing methods for urban freight applications. Journal of Transport Geography, 91, 102959. https://doi.org/10.1016/j.jtrangeo.2021.102959

[3] Basso, R., Kulcsar, B., Egardt, B., Lindroth, P., and Sanchez-Diaz, I. (2024). Electric vehicle scheduling considering stochastic en-route charging. Transportation Research Part C, 160, 104636. https://doi.org/10.1016/j.trc.2024.104636

[4] Boysen, N., Fedtke, S., and Schwerdfeger, S. (2021). Last-mile delivery concepts: A survey from an operational research perspective. OR Spectrum, 43(1), 1–58. https://doi.org/10.1007/s00291-020-00600-2

[5] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). Classification and regression trees. Wadsworth and Brooks/Cole. https://doi.org/10.1201/9781315139470

[6] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

[7] Cortes, C., and Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297. https://doi.org/10.1007/BF00994018

[8] Cover, T., and Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

[9] Dablanc, L., Morganti, E., Arvidsson, N., Woxenius, J., Browne, M., and Saidi, N. (2017).

The rise of on-demand delivery services in European cities. Supply Chain Forum, 18(4), 203–217. https://doi.org/10.1080/16258312.2017.1375375

[10] Du, J., and Aultman-Hall, L. (2007). Increasing the accuracy of trip rate information from passive multi-day GPS travel datasets. Transportation Research Part A, 41(3), 220–232. https://doi.org/10.1016/j.tra.2006.05.001

[11] Figliozzi, M. A., and Tipagornwong, C. (2017). Impact of last-mile parking availability on commercial vehicle costs and operations. Transportation Research Part A, 103, 510–524. https://doi.org/10.1016/j.tra.2016.12.001

[12] Frenay, B., and Verleysen, M. (2014). Classification in the presence of label noise: A survey. IEEE Transactions on Neural Networks and Learning Systems, 25(5), 845–869. https://doi.org/10.1109/TNNLS.2013.2292894

[13] Gingerich, K., Maoh, H., and Anderson, W. (2016). Classifying the purpose of stopped truck events: An application of entropy to GPS data. Transportation Research Part C, 64, 17–27. https://doi.org/10.1016/j.trc.2015.12.012

[14] Greaves, S. P., and Figliozzi, M. A. (2008). Collecting commercial vehicle tour data with passive GPS technology. Transportation Research Record, 2049, 158–166. https://doi.org/10.3141/2049-19

[15] He, H., and Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

[16] He, Z., Guo, W., and Wang, J. (2020). Deriving passenger car equivalents for mixed traffic flows from GPS data. Transportation Research Part C, 120, 102747. https://doi.org/10.1016/j.trc.2020.102747

[17] Lagorio, A., Pinto, R., and Golini, R. (2016). Research in urban logistics: A systematic literature review. International Journal of Physical Distribution and Logistics Management, 46(10), 908–931. https://doi.org/10.1108/IJPDLM-01-2016-0008

[18] Liao, F., Correia, G., and Timmermans, H. J. P. (2022). Simulation of autonomous electric vehicle carsharing systems. Transportation Research Part C, 136, 103492. https://doi.org/10.1016/j.tre.2021.102503

[19] Lu, M., Rijpkema, D., and Quak, H. (2025). CO_2 performance assessment of urban last-mile logistics using GPS data. Transportation Research Part D, 131, 104093. https://doi.org/10.1016/j.trd.2024.104093

[20] Mangiaracina, R., Perego, A., Seghezzi, A., and Tumino, A. (2019). Innovative solutions to increase last-mile delivery efficiency in B2C e-commerce. International Journal of Physical Distribution and Logistics Management, 49(9), 901–920. https://doi.org/10.1108/IJPDLM-02-2019-0048

[21] McCormack, E., Zhao, W., Dailey, D. J., and Sosa, T. (2010). Using GPS truck data to develop a freight performance measurement system. Transportation Research Record, 2168, 83–91. https://doi.org/10.3141/2168-09

[22] Patel, N., Jafri, S., and Figliozzi, M. (2022). Clustering GPS commercial vehicle stops into freight activity patterns. Transportation Research Record, 2676(2), 490–500. https://doi.org/10.1177/03611981211036024

[23] Peng, B., Song, X., and Krogh, B. (2025). GPS data-driven courier stop activity analysis in high-density urban environments. Journal of Transport Geography, 116, 103853. https://doi.org/10.1016/j.jtrangeo.2024.103853

[24] Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81–106.

https://doi.org/10.1007/BF00116251

[25] Ratner, A. J., De Sa, C., Wu, S., Selsam, D., and Re, C. (2017). Data programming: Creating large training sets quickly. Proceedings of the VLDB Endowment, 11(3), 269–282. https://doi.org/10.14778/3137628.3137680

[26] Savelsbergh, M., and Van Wegen, T. (2016). City logistics: Challenges and opportunities. Transportation Science, 50(2), 579–590. https://doi.org/10.1287/trsc.2016.0675

[27] Shen, L., and Stopher, P. R. (2014). Review of GPS travel survey and GPS data-processing methods. Transport Reviews, 34(3), 316–334. https://doi.org/10.1080/01441647.2013.856882

[28] Siripirote, T., Sumalee, A., and Ho, H. W. (2020). Statistical estimation of freight activity analytics from GPS data of trucks. Transportation Research Part E, 140, 101986. https://doi.org/10.1016/j.tre.2019.101886

[29] Song, L., Shangguan, Q., Zhao, P., Guo, X., and Fu, T. (2023). Deep learning-based vehicle activity identification from GPS data in urban logistics. Computers in Industry, 145, 103847. https://doi.org/10.1016/j.compind.2023.103847

[30] Suel, E., Bhatt, P., Wasilewski, J., Bhattacharya, S., and Bhattacharya, A. (2021). Measuring and predicting urban stop demand for last-mile logistics with deep learning. Transportation Research Part C, 131, 103257. https://doi.org/10.1016/j.trc.2021.103257

[31] Vapnik, V. N. (1995). The nature of statistical learning theory. Springer. https://doi.org/10.1007/978-1-4757-2440-0

[32] Wu, J., Peng, B., and Yao, E. (2024). Integrating GPS data and deep learning for dynamic courier routing optimisation. Transportation Research Part C, 159, 104431. https://doi.org/10.1016/j.trc.2023.104431

[33] Yang, X., Huan, Y., Li, Q., and Li, L. (2022a). GPS-based identification of urban freight journey ends of heavy trucks. Transportation Research Part C, 136, 103380. https://doi.org/10.1016/j.trc.2021.103380

[34] Yang, X., Huan, Y., Wang, K., and Li, Q. (2022b). Intercity freight travel end identification from GPS data of heavy trucks. Transportation Research Part C, 143, 103561. https://doi.org/10.1016/j.trc.2022.103561

[35] Yang, X., Lu, Y., and Zhao, X. (2016). A trajectory-based clustering method for identifying freight stops from GPS data. Transportation Research Record, 2548, 90–98. https://doi.org/10.3141/2548-11

[36] Zhou, Z.-H. (2018). A brief introduction to weakly supervised learning. National Science Review, 5(1), 44–53. https://doi.org/10.1093/nsr/nwx106