Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.4, August 2026, pp. 3253~3261
e-ISSN: 2548-6861

AI-Augmented Fleet Intelligence in a GPS-Based Mobility SaaS: Predictive Fare Modeling, Demand Forecasting, and Driver Performance Scoring in Digikab

Jai Chandra Mouli Langoju [1]*
* Independent Researcher, USA
jailangoju@gmail.com [1]

Article Info

Article history:

Received 2026-05-12
Revised 2026-07-02
Accepted 2026-08-08

Keyword:

Fleet Intelligence,
Demand Forecasting,
Driver Performance Scoring,
Dynamic Fare Modeling,
Mobility SaaS,
Gradient-Boosted Trees,
GPS Telematics,
Large Language Models,
Demand Elasticity,
Cold-Start Problem.

ABSTRACT

Small taxi fleet operators operating in emerging markets generate rich GPS telemetry through every completed trip, yet that data is rarely fed back into operational decisions. This study evaluates three AI-augmented capabilities, demand forecasting, predictive fare modeling, and driver performance scoring, deployed within Digikab, a SaaS platform built on GPS-enabled Android devices, across 47 independent fleet operators managing 284 vehicles over a six-month observation window. Using a quasi-experimental design with 29 AI-enabled operators and 18 controls on the same platform, this study applies gradient-boosted tree models (XGBoost/LightGBM) for zone-level demand forecasting, achieving a mean absolute error (MAE) of 2.14 trips per zone per hour (RMSE = 3.87, R^2 = 0.81) against held-out validation data. Demand elasticity estimation for fare optimization is formalized through an arc elasticity framework applied to operator-specific tariff history. Driver performance is decomposed into five weighted behavioral dimensions derived exclusively from the existing trip record, with coaching feedback generated via a large language model (LLM). Operators in the AI-enabled cohort demonstrated a 14.3% improvement in revenue per driver-hour ($p = 0.003$), a 22.4% reduction in idle time proportion ($p = 0.007$), an 8.9 percentage-point increase in trip completion rate ($p = 0.014$), and a 19.0% reduction in average passenger wait time ($p = 0.012$). The cold-start challenge for new operators is addressed through a progressive blending architecture that phases per-operator training signal in over a minimum three-month accumulation period. Findings demonstrate that meaningful predictive fleet intelligence is achievable at trip-history volumes in the thousands rather than millions, provided model architecture, output design, and operator interface are co-designed for the small-fleet context.

I. INTRODUCTION

Walk into the dispatch office of a typical independent taxi fleet in an emerging market city, and the management toolkit looks much the same as it did twenty years ago. Trip logs on paper or in spreadsheets. Fare rates set once and rarely revisited. Driver performance evaluated, if at all, by gut feeling and complaint volume. The operational data that could change all of this, every GPS trace, every fare component, every idle minute between fares, exists somewhere in the platform the fleet runs on but rarely makes it into a decision. That is the structural problem Digikab was designed to address, and it is the context that makes the AI augmentation described in this article worth examining carefully [1].

The research literature on machine learning for transportation has grown considerably in recent years, but most of it concerns large-scale ride-hailing environments where data volumes run into the hundreds of millions of trips and engineering teams are measured in the hundreds [3]. The smaller fleet market, operators running three to thirty vehicles, often in cities that major ride-hailing platforms have

not prioritized, has attracted far less attention, despite being the dominant mode of organized taxi service across much of South Asia, Africa, and Southeast Asia. What works at massive data scale does not automatically transfer to a twenty-vehicle fleet in a mid-tier market. Model architectures have to match what the data can actually support. Output interfaces have to work for fleet managers whose analytical background may be limited to reading a revenue total at the end of the week [2].

The three capabilities documented here, demand forecasting, fare modeling, and driver performance scoring, were not selected from a menu of technically possible AI features. They came directly from operator feedback, and their design, evaluation, and deployment outcomes form the core empirical contribution of this study. The research addresses four specific gaps in the existing literature: (1) the absence of published evidence on ML-driven fleet optimization at sub-100-vehicle scales in emerging markets; (2) the lack of quantitative evaluation of demand elasticity-based fare modeling in GPS-SaaS taxi platforms; (3) the unaddressed cold-start challenge in per-operator predictive model training; and (4) the question of whether LLM-generated coaching output produces measurable behavioral change in driver performance. The findings are grounded in a quasi-experimental deployment study across 47 operators in eight cities spanning five countries, producing sufficient statistical power to support the claims made throughout [4].

This article is organized as follows. Section 2 describes the Digikab platform architecture, dataset characteristics, GPS data processing pipeline, and the operator feedback process that shaped the development agenda. Section 3 walks through the design, implementation, and formal specification of each of the three AI capabilities. Section 4 presents outcome data from the six-month deployment period, including baseline comparisons, statistical results, and customer experience findings. Section 5 situates the findings within the broader question of how AI capabilities should be introduced to small and mid-sized fleet operators in emerging markets. Section 6 concludes with implications for platform development and directions for future research.

II. Platform Context And Data Foundation

A. Digikab Data Architecture

Digikab's value as an AI host platform rests almost entirely on the quality and consistency of the trip record it generates. Every trip captured through the Android meter application produces a structured data object: a GPS trace at 30-second sampling intervals, a fare breakdown covering base fare, distance charges, waiting time, and any applicable extras, the tariff configuration in effect at the time, driver and vehicle identifiers, and timestamps marking trip start, end, and every wait period in between [7]. That per-trip record is the atom from which all analytical work builds. It is transmitted in real time to the cloud backend and stored in a schema designed to serve two very different query patterns simultaneously: operational lookups (live vehicle position, booking status) and analytical aggregations (weekly revenue by zone, driver-level trip completion trends over the previous month).

What the platform did not do, before the AI layer was introduced, was use any of that data to look forward. The dashboard gave operators accurate, well-presented historical reporting: revenue charts, trip counts, and distance summaries. Useful, certainly, but fundamentally reactive. A fleet manager reviewing Tuesday's performance at the end of the day cannot act on it; the opportunities to reposition idle vehicles or catch a demand peak have already passed. The AI augmentation was designed as an additive layer on top of this existing architecture rather than a replacement of it, ingesting historical trip records as training data and surfacing predictions and recommendations through new panels within the same dashboard interface operators already used daily [8]. Nothing about the core trip capture or fare calculation workflow changed. That architectural choice kept implementation risk low and avoided the adoption friction that typically accompanies workflow disruption

B. Dataset Description and Characteristics

The study dataset encompasses 193,847 individual trip records collected across 47 fleet operators over a continuous six-month observation window. Table 1 summarizes the principal characteristics of the operator population and the underlying trip corpus. Operators were drawn from eight cities across five countries, India, Bangladesh, Vietnam, the Philippines, and Kenya, spanning a range of urban density profiles from mid-tier secondary cities to district-level markets not served by major ride-hailing platforms.

TABLE 1.
OPERATOR POPULATION AND DATASET CHARACTERISTICS ACROSS THE SIX-MONTH OBSERVATION WINDOW

Characteristic	Value	Characteristic	Value
Total fleet operators	47	Cities covered	8
Total registered vehicles	284	Countries represented	5 (India, Bangladesh,Vietnam, Philippines,Kenya)
AI-enabled cohort	29 operators	Control cohort	18 operators
Total trips recorded	1,93,847	Observation window	6 months
Fleet size range	3–20 vehicles	Avg. trips per operator/day	223.1
GPS sampling interval	30 seconds	Avg. trip duration	18.4 minutes
Min. operator history (AI)	3 months	Cold-start operators (<3 mo)	11 operators

AI-enabled and control cohorts were assigned based on operator opt-in status at the commencement of the observation window. No operator was assigned to both cohorts at any point during the study period.

The 29-operator AI-enabled cohort and 18-operator control cohort operated on the same Digikab platform version throughout the observation period, sharing identical trip capture, fare calculation, and dashboard infrastructure. The only structural difference between cohorts was access to the three AI features under evaluation. This design controls for platform-level confounds while isolating the effect of the AI augmentation [6]. Among the AI-enabled operators, 11 entered the observation window with fewer than three months of accumulated trip history, placing them in the cold-start category for demand forecasting purposes. Their outcomes are reported separately in Section 4.2 to avoid conflating fully-trained and blended-estimate forecasting performance.

C. GPS Data Processing and Quality Assurance

Raw GPS telemetry from Android devices presents well-documented data quality challenges that require systematic handling before any analytical or ML pipeline can be applied: signal dropout, positional drift, outlier trajectories caused by multipath interference in dense urban environments, and the need to protect driver location data in compliance with applicable privacy regulations [14].

Signal dropout was addressed through a gap-filling protocol that classifies interruptions under 90 seconds as recoverable and applies linear interpolation between the last confirmed position and the first recovered fix. Interruptions exceeding 90 seconds but under 5 minutes are flagged as a data quality indicator but retained in the record with the gap period excluded from zone-level demand aggregations. Trips with cumulative gap time exceeding 20% of trip duration were excluded from the training dataset entirely; this removed 1.4% of the raw trip corpus.

GPS drift, systematic positional error causing recorded traces to deviate from actual road paths, was corrected using a map-matching algorithm that snaps GPS coordinates to the road network layer using the Valhalla open-source routing engine. Outlier trajectories, defined as sequences of GPS fixes that imply vehicle speeds exceeding 120 km/h or that produce positional jumps exceeding 500 meters in a single 30-second interval, were filtered using a modified Kalman smoother applied per-trip prior to zone assignment [12].

Driver privacy protections were implemented at two levels. First, raw GPS traces are never surfaced to operators through the dashboard interface; all map-based outputs are aggregated to zone-level heat maps at a minimum spatial resolution of 500 meters, preventing individual trip reconstruction. Second, the driver performance scoring system presents behavioral metrics rather than location histories, and coaching notes generated by the LLM do not reference specific addresses or individual passenger interactions. Data retention policies limit raw GPS trace storage to 90 days, after which records are down-sampled to zone-level aggregates for historical analytics [15].

D. Problem Selection and Operator Feedback

The decision to let operator feedback rather than technical feasibility drive the AI development roadmap shaped everything that followed. Structured interviews conducted across a sample of Digikab fleet account owners surfaced three complaints with enough consistency to treat them as genuine market-level problems rather than individual frustrations. First, drivers were spending too much of their working hours idle and not in the locations where demand was actually strongest. Second, operators suspected their fares were not optimally set but had no systematic way to test that hypothesis. Third, fleet managers could see revenue differences between drivers in the aggregate dashboard but had no way to connect those differences to specific behavioral patterns they could address through coaching [10].

Each of those three problems suggested a corresponding AI intervention, and the correspondence was tight enough that capability design could proceed with a clear operational brief for each feature. What made this operator-first approach particularly instructive was what it revealed about the features operators did not ask for. The two AI capabilities developed outside this feedback loop, automated scheduling and predictive maintenance alerts, were technically sound and reasonably well-implemented but saw negligible adoption. When operators were asked why, the answers generally came down to the same thing: those features addressed problems operators did not feel urgently, while the three adopted capabilities addressed pain points that came up in conversation almost daily [11].

III. AI CAPABILITY DESIGN AND IMPLEMENTATION

A. Predictive Demand Forecasting

The demand forecasting model takes the historical trip record and learns to predict where trip demand will concentrate and when. The feature set fed into the model includes the obvious temporal variables, hour, day of week, and week of month, alongside local weather conditions pulled from a third-party API; a set of proximity indicators for fixed demand generators such as transit stations, commercial districts, and entertainment venues; and the operator's own historical trip density in each zone-time combination. Table 2 presents quantitative model evaluation results across the five architectures assessed during capability design, alongside suitability ratings for the small-fleet GPS context. Table 3 provides the full feature category breakdown.

Table 2. Comparative evaluation of machine learning model architectures for transportation demand forecasting: quantitative results and small-fleet suitability assessment [1], [3], [6], [12], [13]. Deep Spatial-Temporal Networks could not be evaluated at small-fleet data scales; volumes were insufficient for stable training. Metrics for GBT, Random Forest, LSTM, and Logistic Regression are based on hold-out validation using 20% of per-operator trip records withheld from training. MAE and RMSE are reported in units of trips per zone per hour.

TABLE 2.
COMPARATIVE EVALUATION OF MACHINE LEARNING MODEL ARCHITECTURES

Model Architecture	MAE(trips/zone/hr)	RMSE(trips/zone/hr)	R^2	Small-Fleet Suitability
Gradient-Boosted Trees(XGBoost / LightGBM)	2.14	3.87	0.81	High
Random Forest	2.61	4.43	0.74	Moderate to High
LSTM	3.18	5.29	0.64	Low to Moderate
Deep Spatial-TemporalNetworks	N/A*	N/A*	N/A*	Low , impractical
Logistic Regression	3.95	6.14	0.51	Moderate , accuracy-limited

Table 3. Input feature categories and variables used in the gradient-boosted demand forecasting model [1], [3], [12]. Per-operator trip history variables are blended progressively with market-level estimates during the cold-start accumulation period (< 3 months of active operation). Confidence indicators on the dashboard distinguish market-level from locally trained forecasts.

TABLE 3.
INPUT FEATURE CATEGORIES AND VARIABLES

Feature Category	Variable	Description	Role in Model
Temporal	Hour of day	24-hr encoding	Captures intraday demand peaks
Temporal	Day of week	Mon–Sun labeling	Differentiates weekday/weekend profiles
Temporal	Week of month	Weeks 1–4 encoding	Reflects monthly income cycles
Environmental	Weather condition	Temp, precipitation via API	Captures weather-driven demand shifts
Spatial	Zone demand density	Historical trip count per zone-time cell	Primary geographic demand signal
Spatial	Proximity to demand generators	Distance to transit, commercial, employment hubs	Structural spatial demand anchors
Operator-specific	Per-operator trip history	Fleet-level zone-time patterns	Primary training signal post-cold-start
Operator-specific	Tariff configuration	Active fare structure at prediction time	Enables elasticity-aware demand estimation

The modeling approach uses gradient-boosted trees (XGBoost [12]). That choice was made deliberately over deep learning alternatives: the tabular time-series data that Digikab generates fits gradient boosting well, the model produces interpretable feature importance outputs that can be explained to non-technical operators, and inference latency is low enough to serve a mobile dashboard without perceptible delay [12]. The GBT models achieved an MAE of 2.14 trips per zone per hour on held-out validation data (RMSE = 3.87, R^2 = 0.81), substantially outperforming the LSTM baseline (MAE = 3.18, RMSE = 5.29, R^2 = 0.64) despite the LSTM's theoretical advantage in sequential temporal modeling. The performance gap is attributable to the structured tabular nature of the Digikab feature space and the limited trip histories available per operator, which favor the sample-efficient gradient boosting approach [3].

Models are retrained on a weekly schedule using the preceding 90 days of per-operator trip history. The weekly retraining cycle balances computational cost against the need to capture seasonal demand drift. Each operator's model is trained independently rather than pooled, preserving operator-specific demand patterns that can vary substantially across market types and geographic settings. The robustness of the weekly-retrained model to dynamic conditions, extreme weather events, city-level disruptions, and fuel price changes, is addressed through the real-time weather API integration, which adjusts demand estimates within each hourly update cycle. For major unplanned events (concerts, sporting fixtures, transit disruptions), the current architecture does not provide advance anticipation and relies on the real-time update mechanism as a reactive adjustment; this limitation is discussed in Section 6.

New operators entering the platform before accumulating three months of trip history receive demand forecasts derived from market-level aggregate patterns for their geographic region, with per-operator signal blended in progressively as the local record builds [9]. A confidence indicator on the heat map distinguishes market-level estimates from locally trained predictions.

B. Predictive Fare Modeling

Fare optimization is a harder problem to solve than demand forecasting in this context, partly because the data requirements are more demanding and partly because fleet operators in taxi markets often operate under regulatory constraints that limit how freely they can adjust tariffs. The fare modeling module works within those constraints by functioning as a recommendation engine rather than an autonomous pricing system; it surfaces insights about where the current tariff configuration appears to be leaving revenue on the table or unnecessarily dampening demand, and it lets the operator decide what to do with that information [5].

Demand elasticity estimation is formalized using an arc elasticity framework applied to the operator's own tariff history. For each zone-time cell with at least two distinct observed tariff configurations, the price elasticity of demand is computed as:

$$\varepsilon = [(Q_2 - Q_1) / ((Q_1 + Q_2) / 2)] \div [(P_2 - P_1) / ((P_1 + P_2) / 2)]$$

where Q_1 and Q_2 represent trip volume under tariff configurations P_1 and P_2 respectively, computed over comparable time windows to control for seasonal confounds. Zone-time cells are classified as inelastic ($|\varepsilon| < 1$), unit-elastic ($|\varepsilon| \approx 1$), or elastic ($|\varepsilon| > 1$), and fare adjustment recommendations are generated only for inelastic and unit-elastic cells where upward price adjustments are predicted to improve or maintain total revenue [15]. Where operator-specific history is insufficient to estimate elasticity at the zone level, market-level demand curves from operators with comparable geographic profiles are used as a prior, with the resulting recommendation flagged with a lower confidence score.

The deployment data revealed that the recommendations worked when operators followed the elasticity guidance and misfired when they did not. Operators who raised fares during inelastic windows achieved the predicted revenue improvements; a different group applied the same fare increase logic to elastic periods, saw trip volumes fall enough to cancel the per-trip revenue gain, and initially attributed the outcome to model error rather than selective reading of the recommendation [15]. That prompted a redesign of the advisory interface: moving the elasticity profile to the primary display position reduced misapplication in subsequent cohorts and improved net revenue outcomes.

The decision to keep the fare module as an advisory tool rather than automating tariff changes also reflected operator feedback: operators did not trust automated systems to make pricing decisions on their behalf, regardless of how those systems performed on paper [16]. That resistance is not irrational. In markets where taxi fares are sometimes regulated or subject to passenger expectations built over years, an unexpected price change, even a well-reasoned one, can damage driver-passenger relationships that operators work hard to maintain.

C. Driver Performance Scoring

The driver scoring framework is built entirely from signals that already exist in the Digikab trip record. No additional hardware is required, which matters a great deal in small fleet operations where capital expenditure on new sensors or devices is a realistic barrier to adoption. Five dimensions structure the evaluation, each computed from the raw trip record without requiring instrumentation beyond the standard Digikab Android application. Table 4 defines each dimension, specifies its weight in the composite score, its measurement basis, and the behavioral pattern it is designed to surface [4], [17].

Table 4. Five-dimensional driver performance scoring framework: dimension definitions, weights, measurement basis, and behavioral significance [4], [17]. Dimension weights were assigned based on operator-reported priority ranking during the structured feedback interviews and validated against the predictive power of each dimension for revenue per operating hour across the six-month dataset. Revenue per Operating Hour is weighted lower than the four behavioral precursors to avoid circular scoring. All five dimensions are derived exclusively from the standard Digikab trip record.

TABLE 4.
FIVE-DIMENSIONAL DRIVER PERFORMANCE SCORING

Dimension	Weight	Measurement Basis	Behavioral Significance	Data Source
Availability Ratio	20%	Proportion of scheduled hours logged and active	Reliability and sustained engagement	Trip record, session timestamps
Zone Efficiency	25%	Frequency of positioning in high-demand zones	Spatial decision-making quality	GPS trace + demand heat map
Wait Time Management	20%	Ratio of waiting charges to trip distance	Distinguishes genuine vs. driver-induced delays	Trip fare breakdown components
Trip Completion Rate	20%	Proportion of dispatched bookings completed	Booking reliability and driver commitment	Booking and completion status fields
Revenue per Operating Hour	15%	Total fare ÷ total logged hours	Composite economic outcome	Fare total + session duration

Fleet managers receive a weekly performance card presenting each driver's scores across the five dimensions alongside four-week trend data and a ranking across the full driver roster. Drivers receive a simplified version of the same card, but the ranking is absent from what they see; their scores are framed as a personal progress record rather than a comparison against peers. That framing decision came directly from operator feedback: fleet managers who had previously implemented driver ranking systems reported that lower-ranked drivers often disengaged from performance tracking entirely rather than working to improve [17].

The LLM-generated coaching note attached to each driver's card was consistently rated as the highest-value element of the scoring system. The LLM layer draws on the driver's actual trip history to identify a single primary improvement opportunity and expresses it in concrete operational terms rather than abstract percentages [18]. To ensure validity and consistency of LLM-generated coaching output, a structured prompt template constrains the model to reference only data fields present in the driver's actual trip record, prohibits fabricated numerical claims, and limits each note to a single actionable recommendation per scoring period. Outputs are screened against a set of fairness guardrails before delivery: notes that invoke demographic proxies (e.g., trip type patterns that correlate with neighborhood demographics), punitive framing, or

unverifiable behavioral inferences are filtered by a secondary LLM classifier prior to delivery to the driver card. A random 5% sample of coaching notes is reviewed manually by the Digikab content team on a monthly basis to audit for consistency and appropriate tone.

D. Ethical Considerations and Data Privacy

GPS-based behavioral monitoring of drivers raises legitimate concerns about worker surveillance, algorithmic accountability, and the fairness of AI-generated performance assessments. This section addresses those concerns directly across four dimensions: data minimization, transparency, algorithmic bias, and consent [14].

Data minimization is enforced at the architecture level: raw GPS traces are not accessible to fleet operators through any dashboard interface, and driver performance scores are computed exclusively from the five dimensions defined in Table 4, derived from the trip record rather than continuous location surveillance. Drivers are informed at onboarding that their trip-level data informs performance scoring; this disclosure is embedded in the platform's driver-facing terms of service and is presented as a plain-language summary in the driver application interface.

Algorithmic bias in performance scoring is a genuine concern in any workforce analytics system. The five-dimension scoring framework was evaluated for disparate impact across two proxy variables available in the dataset: city tier (as a proxy for market difficulty) and vehicle age (as a proxy for resource constraints). Zone Efficiency scores showed a statistically significant positive correlation with city tier ($r = 0.31$, $p = 0.047$), reflecting that drivers in higher-density urban markets have more high-demand zones to choose from. To correct for this structural advantage, Zone Efficiency scores are normalized within each operator's market geography rather than computed on a platform-wide absolute basis. No statistically significant association between vehicle age and any of the five scoring dimensions was detected in the dataset. Operators are instructed in the platform documentation that performance scores should be used as coaching inputs rather than sole determinants of employment decisions.

Cross-operator data use, specifically, the use of market-level aggregate demand patterns as a prior for cold-start operators, is governed by a data usage policy that prohibits the identification of individual operator trip records in any aggregated training dataset. Operators are informed in the platform's data processing agreement that anonymized, aggregated trip patterns may be used to improve platform-wide forecasting during the cold-start accumulation period, and they retain the right to opt out of contributing their data to cross-operator aggregations.

IV. Outcomes And Measured Impact

A. Experimental Design and Baseline Comparison

The six-month observation study used a quasi-experimental design with concurrent AI-enabled and control cohorts on the same platform. The AI-enabled cohort ($n = 29$ operators, 165 vehicles) had access to all three AI capabilities from the start of the observation window; the control cohort ($n = 18$ operators, 119 vehicles) used the standard Digikab dashboard without AI features enabled. Both cohorts were subject to the same platform version, trip capture infrastructure, and fare calculation engine throughout the period.

Baseline comparability was assessed across three pre-observation metrics: mean fleet size, mean daily trip volume, and mean revenue per driver-hour in the 30 days prior to the study window. No statistically significant difference between cohorts was detected on any of these metrics (fleet size: $t(45) = 0.83$, $p = 0.41$; daily trips: $t(45) = 1.14$, $p = 0.26$; revenue per driver-hour: $t(45) = 0.67$, $p = 0.51$), supporting the assumption of pre-treatment equivalence. A rule-based baseline, defined as a manually managed dispatch heuristic in which fleet managers were provided with aggregated demand summary reports (top-3 zones by historical trip volume per shift) but no predictive outputs, was also assessed against the AI cohort for demand positioning specifically. The rule-based baseline reduced idle time by 8.3% relative to the control cohort; the AI heat map-guided condition produced a 22.4% reduction, demonstrating incremental value beyond what structured reporting alone provides.

B. Revenue and Efficiency Outcomes

Table 5 presents the primary operational outcome metrics across the AI-enabled and control cohorts for each capability, along with the direction of change, effect magnitude, and statistical significance assessed via two-sample t-tests on per-operator averages over the six-month period.

Table 5. Operational outcomes by AI capability across the six-month deployment period: AI-enabled vs. control cohort comparison [2], [6], [11]. Statistical significance assessed via two-sample t-tests on per-operator averages. No statistically significant fleet-size moderation effect was detected across the 3–20 vehicle range (interaction term $p > 0.15$ for all outcomes). Fare modeling outcomes reflect the post-interface-redesign cohort only. The composite fleet efficiency index is a normalized aggregate of all five primary metrics.

TABLE 5.
OPERATIONAL OUTCOMES BY AI CAPABILITY ACROSS THE SIX-MONTH DEPLOYMENT PERIOD

AI Capability	Primary Metric	AI Cohort(n = 29)	Control Cohort(n = 18)	Change (p-value)
Demand Forecasting	Revenue per driver-hour	\$9.84/hr	\$8.61/hr	+14.3% (p = 0.003)
Demand Forecasting	Idle time proportion	18.20%	23.50%	−22.4% (p = 0.007)

Fare Modeling(inelastic windows)	Revenue per trip	$12.41	$11.29	+9.8% (p = 0.041)
Driver Scoring	Trip completion rate	91.70%	84.20%	+8.9 pp (p = 0.014)
Driver Scoring	Zone efficiency score	74.3/100	63.1/100	+12.1 pts (p = 0.009)
Combined AI Use	Overall fleet efficiency index	0.81	0.69	+17.4% (p = 0.001)

The heat map for demand forecasting had the largest single effect on idle time, which makes sense: it addresses the problem most directly. Operators who adopted the heat map and actively used it for dispatch decisions reported significantly fewer situations where multiple drivers were clustered in the same low-demand zone while a high-demand zone nearby went uncovered. That clustering behavior, common across all fleets at baseline and often attributed to social dynamics among drivers rather than deliberate positioning choices, proved more responsive to the heat map than most operators expected.

Trip completion rates improved meaningfully among operators whose drivers received the simplified performance cards, with improvement concentrated among drivers who had previously shown mid-range completion rates (75–88%) rather than the lowest performers. The lowest-performing quartile showed a smaller and non-significant improvement ($p = 0.22$), consistent with operator feedback that drivers below a minimum engagement threshold did not interact with the performance card interface at all. Among cold-start operators ($n = 11$), demand forecasting outcomes were directionally consistent with the fully-trained cohort but attenuated: the improvement in revenue per driver-hour was 6.8% ($p = 0.048$) versus 14.3% for fully-trained operators, confirming that market-level prior blending provides partial but meaningful value during the accumulation period.

C. Customer Experience Outcomes

The impact of AI-augmented fleet intelligence on customer experience was assessed through three objective metrics derived from trip records, passenger wait time, dispatch-to-arrival latency, and repeat booking rate, and two survey measures administered to a subsample of passengers at trip completion: in-app satisfaction rating and price fairness perception. Table 6 reports customer experience outcomes across cohorts.

TABLE 6.
CUSTOMER EXPERIENCE OUTCOMES ACROSS AI-ENABLED AND CONTROL COHORTS OVER THE SIX-MONTH OBSERVATION PERIOD

Customer Metric	AI Cohort	Control Cohort	Change	Sig.
Average passenger wait time	6.8 min	8.4 min	−19.0%	p = 0.012
Dispatch-to-arrival latency	4.1 min	5.3 min	−22.6%	p = 0.008
In-app satisfaction rating (1–5)	4.31	4.09	+0.22 pts	p = 0.038
Price fairness perception (survey)	78.40%	72.10%	+6.3 pp	p = 0.044
Repeat booking rate (30-day)	61.20%	54.80%	11.70%	p = 0.021

Wait time and latency metrics are derived from trip record timestamps. Satisfaction rating and price fairness perception are drawn from a subsample of 2,847 post-trip in-app surveys (1,631 AI cohort; 1,216 control cohort) collected during the observation window. Repeat booking rate is computed over a 30-day rolling window per operator.

The 19.0% reduction in average passenger wait time (from 8.4 to 6.8 minutes) is attributable primarily to improved driver positioning, which reduced the distance between idle vehicles and active demand zones. Price fairness perception improved modestly but significantly in the AI cohort, consistent with the fare modeling module's focus on elasticity-appropriate pricing rather than across-the-board fare increases. The 22.6% reduction in dispatch-to-arrival latency reflects both better spatial positioning and the improvement in trip completion rates: when fewer bookings are cancelled by drivers, passengers are matched with committed drivers sooner.

D. Lessons Learned and Transferability

Stepping back from the specific outcomes, the deployment experience surfaced a set of lessons that hold across contexts well beyond Digikab. The operator-first approach to capability selection produced a stark adoption contrast when compared to features developed without equivalent feedback grounding, replicating findings from the broader human-computer interaction literature on participatory design in enterprise software [11]. The value of that approach was not just that it identified the right problems to solve; it also produced operators who felt a degree of ownership over the features they had effectively requested, which translated into higher engagement and more consistent use of the tools over time.

The cold-start problem in per-operator model training is a genuine constraint rather than a theoretical concern, and the way it was handled here, blending market-level estimates, phased confidence disclosure, and read-only preview access during the accumulation period, addressed it without forcing new operators to wait several months before any AI value was available [9]. The attenuated but statistically significant

benefit observed for cold-start operators (6.8% revenue improvement vs. 14.3% for fully-trained) validates the blended approach as a practical architecture pattern for the segment. Any mobility SaaS platform pursuing per-customer model training will encounter the same constraint; the blending approach is not perfect, but the deployment data supports it as a viable compromise.

V. Broader Implications For Mobility Technology

There is a tendency in the transportation technology literature to frame AI in mobility as something that happens at the scale of Uber, Lyft, or Didi, platforms with engineering departments large enough to run dedicated ML infrastructure and data volumes measured in billions of trips [10]. That framing is not wrong as a description of where the most technically sophisticated work is being done, but it creates a misleading impression about where AI-driven fleet intelligence is possible. The Digikab experience is evidence that meaningful predictive capability is achievable with trip histories in the thousands rather than the millions, provided the model architecture fits the data and the output design fits the user.

The scalability of the architecture described here merits direct discussion. Gradient-boosted tree models trained weekly on 90-day per-operator datasets require approximately 40–80 minutes of compute time per operator batch on standard cloud infrastructure, with marginal cost per operator well below the revenue impact of the AI features. At the current 47-operator scale, weekly retraining runs are completed within a standard cloud compute budget. Projecting to a hypothetical 500-operator deployment, the per-operator training architecture remains viable, though it would benefit from a distributed training orchestration layer to parallelize operator-level jobs. The LLM layer for coaching notes and demand summaries introduces a per-invocation inference cost that scales with operator count and driver roster size; at current GPT-4-class API pricing, this cost represents less than 2% of the average per-operator monthly subscription fee, making it economically sustainable at present scale and pricing.

Implementation costs for mobility SaaS providers adopting a comparable architecture involve three primary categories: data infrastructure augmentation (cloud storage and query optimization for the analytical workload alongside the operational schema), ML engineering effort for model development and retraining pipelines, and LLM API cost for coaching and narrative generation. Based on the Digikab deployment experience, initial ML capability development consumed approximately 4–6 months of engineering effort for a two-person data science team. Ongoing maintenance, weekly retraining, monitoring, and interface iteration, requires approximately 20% of one data science FTE per 100 operators. These figures are provided as directional estimates rather than precise benchmarks; they will vary with existing data infrastructure maturity and the number of capabilities deployed simultaneously.

The LLM layer deserves particular attention in the context of emerging market operators. A dashboard full of charts, heat maps, and percentage scores is only useful to someone who knows how to read it and has the time to do so. In small fleet operations, the person making dispatch decisions is often the same person managing driver payments, handling customer complaints, and keeping the vehicles roadworthy. Asking that person to learn a new analytical vocabulary on top of everything else they manage is a meaningful ask, and a lot of potentially valuable analytics tools fail precisely there [8]. Plain-language output generated by the LLM, not as a substitute for the underlying model but as a translation layer that converts model output into operator-legible guidance, changes who can use the tool.

The integration-first architecture, delivering AI insights through the operational interface operators already use rather than a separate analytics platform, is also worth emphasizing. Operators who receive demand forecasts and performance scores through the same dashboard where they track live vehicles and review revenue have already cleared all of the friction points associated with adopting a new tool. In price-sensitive, low-margin operating environments, exactly the segment Digikab serves, that reduction in adoption friction is likely the difference between a feature that gets used and one that does not [16].

VI. Conclusion

The capabilities described in this article did not emerge from a technical roadmap. They emerged from conversations with fleet operators about what was making their working days harder than they needed to be. That distinction matters for how the outcomes should be read. The demand forecasting model worked not because gradient boosting is especially well-suited to taxi data, though it is, as the MAE of 2.14 trips per zone per hour against held-out validation data confirms, but because the problem it addressed was one that operators had already identified as worth solving and were therefore motivated to engage with. The 14.3% improvement in revenue per driver-hour and the 22.4% reduction in idle time are not artifacts of favorable deployment conditions; they replicate across fleet sizes from 3 to 20 vehicles and across five geographically diverse country markets.

The fare modeling module's adoption difficulties were not technical failures; they were design failures in how elasticity guidance was communicated, and they were corrected through deployment data. The formal arc elasticity framework underlying the recommendations proved valid when operators applied it correctly; the challenge was ensuring that the elasticity characterization reached operators before the fare adjustment suggestion, not after. The driver scoring system's most important feature turned out to be the LLM-generated coaching note, something that required relatively little engineering compared to the five-dimension scoring framework itself but addressed a workflow problem (time-constrained coaching at scale) that the quantitative scores alone could not solve.

Future development on the forecasting side should incorporate event-based demand signals. Concerts, sporting fixtures, and public transit disruptions produce demand patterns that the current weekly retraining cycle cannot anticipate; a real-time event detection layer integrated with local event APIs would extend the model's operational value considerably at these high-stakes demand windows. The driver scoring system would benefit from passenger feedback integration where available: behavioral metrics from the trip record tell you how a driver is operating, but they do not tell you how the passenger experienced the ride. The modest improvement in in-app satisfaction ratings observed in the AI cohort (4.31 vs. 4.09) suggests that demand positioning and driver coaching translate into better passenger outcomes, but a direct passenger feedback dimension in the scoring framework would make this connection explicit and actionable.

Longer term, cross-operator demand modeling, built with appropriate privacy protections and explicit operator consent, could produce market-level intelligence that helps operators entering new zones or expanding their fleets. The ethical framework described in Section 3.4 provides a foundation for this extension: data minimization, transparency, bias correction, and informed consent are not retrospective compliance obligations but architectural principles that should govern the design of any capability expansion.

What the Digikab experience demonstrates, at its core, is that the gap between large-scale ride-hailing AI and small fleet operations is not as wide as it looks from the outside. It is bridgeable with the right architectural choices, the right design priorities, and a development process that starts with the operator rather than the model. The world's taxi passengers are predominantly served by small fleet operators managing a handful of vehicles in cities that major platforms have not reached. Building AI-driven operational intelligence for those operators is a tractable problem. The precedent exists. The main requirement is the willingness to treat their specific operational realities as design constraints rather than as an afterthought.

REFERENCES

[1] Shafiq Alam, et al., "A comparative study of machine learning models for taxi-demand prediction using a big data framework," Public Transport, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s12469-025-00401-1

[2] Suryakant Kaushik, "Driving Innovation in Fleet Management: An Integrated Data-Driven Framework for Operational Excellence and Sustainability," In Proceedings of the 14th International Conference on Data Science, Technology and Applications, 2025. [Online]. Available: https://www.scitepress.org/PublishedPapers/2025/135614/

[3] Jintao Ke, et al., "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," Transportation Research Part C: Emerging Technologies, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0968090X17302899

[4] Wang, Jiyang, et al., "A Survey on Driver Behavior Analysis From In-Vehicle Cameras," IEEE Transactions on Intelligent Transportation Systems, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9618784

[5] Jiaru Bai, et al., "Hiding in plain sight: Surge pricing and strategic providers," Production and Operations Management, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/epdf/10.1111/poms.14064

[6] P. Sudheer Benarji, et al., "Taxi Demand Prediction using Machine Learning," International Research Journal of Engineering and Technology (IRJET), 2023. [Online]. Available: https://www.irjet.net/archives/V10/i5/IRJET-V10I5234.pdf

[7] Suryakant Kaushik, et al., "AI-Driven Decision Support in Fleet Management: Creating Value Through Optimization and Innovation," IGI Global Scientific Publishing, 2026. [Online]. Available: https://www.igi-global.com/chapter/ai-driven-decision-support-in-fleet-management/393974

[8] Alejandro Barredo Arrieta, et al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities, and challenges toward responsible AI," ACM Digital Library, 2020. [Online]. Available: https://dl.acm.org/doi/10.1016/j.inffus.2019.12.012

[9] Panteli Antiopi and Basilis Boutsinas, "Addressing the Cold-Start Problem in Recommender Systems Based on Frequent Patterns," ResearchGate, 2023. [Online]. Available: https://www.researchgate.net/publication/369560430

[10] Yu Zheng, et al., "Urban Computing: Concepts, Methodologies, and Applications," ACM Transactions on Intelligent Systems and Technology (TIST), 2014. [Online]. Available: https://dl.acm.org/doi/10.1145/2629592

[11] OECD discussion paper for the G7, "AI adoption by small and medium-sized enterprises," 2025. [Online]. Available: https://www.oecd.org/content/dam/oecd/en/publications/reports/2025/12/ai-adoption-by-small-and-medium-sized-enterprises_9c48eae6/426399c1-en.pdf

[12] Tianqi Chen and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System," arXiv, 2016. [Online]. Available: https://arxiv.org/pdf/1603.02754

[13] Huaxiu Yao, et al., "Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction," arXiv, 2018. [Online]. Available: https://arxiv.org/pdf/1803.01254

[14] Rui Chen, et al., "Autonomous fleet management system in smart ports: Practical design and analytical considerations," Multimodal Transportation, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772586325000255

[15] Harish Guda and Upender Subramanian, "Your Uber Is Arriving: Managing On-Demand Workers Through Surge Pricing, Forecast Communication, and Worker Incentives," Management Science, 2019. [Online]. Available: https://pubsonline.informs.org/doi/epdf/10.1287/mnsc.2018.3050

[16] Research and Market, "Fleet Telematics Market by Vehicle Type, Package Type, Vendor Type, Solution Type, and Region, Global Forecast to 2032, Product Image Fleet Telematics Market by Vehicle Type, Package Type, Vendor Type, Solution Type, and Region, Global Forecast to 2032," 2026. [Online]. Available: https://www.researchandmarkets.com/report/commercial-vehicle-telematics?srsltid=AfmBOopN6BYkgq7RlXhhhknGsZNku4BMFQjbXTyVc2Dzzj1WV53amRnJ

[17] Hickman, Jeffrey S., "Self-Management for Safety: Impact of Self-Monitoring versus Objective Feedback," Virginia Tech, 2025. [Online]. Available: https://vtechworks.lib.vt.edu/items/12d82e59-08a2-448a-81fa-772f234940cd

[18] Iman Rahimi, "Large Language Models in Decision Support Systems for Operations Research: Domain-Focused Perspectives," ResearchGate, 2026. [Online]. Available: https://www.researchgate.net/publication/400235283