International Review of Business, Trade, and Economics
ISSN: 3069-230X

PRIME OPEN ACCESS

Volume 2, Issue 2

Research Article

Date of Submission: 05 May, 2026
Date of Acceptance: 04 Jun, 2026
Date of Publication: 15 Jun, 2026

Algorithmic Logistics and Labor Economics in the On-Demand delivery Ecosystem: A Socio-Technical Analysis with a Proposal for Rider-Centric Intelligence

Narayana Sanghea Panchumarthy*

Cambridge, United Kingdom

***Corresponding Author:** Narayana Sanghea Panchumarthy, Cambridge, United Kingdom.

Citation: Panchumarthy, N. S. (2026). Algorithmic Logistics and Labor Economics in the On-Demand delivery Ecosystem: A Socio-Technical Analysis with a Proposal for Rider-Centric Intelligence. *Int Rev Bus Trade Econ, 2*(2), 01-12.

Abstract
Food delivery has, in less than a decade, moved from a neighbourhood trade conducted by phone to a globally orchestrated, platform-mediated logistics industry. What looks to a customer like a thirty-minute ride from a kitchen to a doorstep is, underneath, a chain of sub-second decisions taken by forecasting engines, dispatch heuristics, sensor-fusion models and pricing controllers. This paper offers a structured reading of that stack and of the labour economy it produces. I trace the methodological evolution of demand forecasting from k-nearest-neighbour regressions to recurrent and probabilistic deep networks, examine dispatch and matching engines through Deliveroo's Frank algorithm and Uber Eats' triple-batching logic, and explain how sensor fusion - accelerometers, gyroscopes, conditional random fields converts a noisy GPS trace into a fine-grained model of the courier's working day. I then turn to the economic and social consequences. Drawing on Berkeley Labor Center net-pay data, the Stanford rideshare gender-gap study, and recent reporting on the Bologna Frank ruling, the New York City Department of Consumer and Worker Protection minimum-pay rule, the EU Platform Work Directive 2024/2831 and China's 2027 algorithm-bargaining mandate, the paper argues that the gig delivery system has now passed into a 'hyper-algorithmic' phase whose efficiency cannot be separated from its labour-market effects. As a constructive contribution, I outline FindMyRider, a rider-centric intelligence platform combining demand forecasting, density-aware heatmap visualisation and explainable, contestable decision logs -intended as a research blueprint, not as a commercial system. The paper closes with limitations, ethical reflections and an agenda for further work on retrieval-augmented, human-in-the-loop logistics analytics.

Keywords: Algorithmic Management, On-Demand Delivery, Demand Forecasting, Lstm, Deepeta, Sensor Fusion, Gig Labour, Explainable Ai, Platform Regulation, Retrieval-Augmented Generation

Introduction
The first time I tried to reverse-engineer how a food delivery app makes its decisions, I was standing on a London street corner watching the same scooter circle the block three times in a row. The rider was waiting on a kitchen that was running fifteen minutes late; the app, oblivious, kept extending the pickup window in two-minute increments. I could see, in real time, the gap between two layers of the same system: the optimisation logic in the cloud, and the actual physical street where a person was earning a piece-rate wage. That gap is the subject of this paper.

Over the past decade, urban food delivery has shifted from a localised, human-dispatched service to a globalised, platform-mediated industry. Companies such as Uber Eats and Deliveroo coordinate millions of concurrent interactions across three weakly aligned populations - consumers who want a meal, merchants who prepare it, and couriers who carry it [1-3]. The orchestration depends on the platform's ability to ingest streams of geospatial and temporal data and translate physical movement into predictable, manageable units of labour and inventory [4,5]. The system that emerges is impressive on its own terms: sub-30-minute delivery has become a baseline expectation in most major cities. It is also a system that produces strong tensions - between operational efficiency, the economic situation of the people who power it, and an emerging regulatory environment that increasingly treats the algorithm itself as a labour-relations object [6-9].

This paper has four aims. First, to give a coherent technical reading of the on-demand delivery stack, from forecasting through dispatch to sensor fusion, that does not lose sight of why each component exists. Second, to put that stack in conversation with the economic and behavioural literature on gig labour, so that pay, utilisation and gender disparities are not treated as separate from the engineering choices that shape them. Third, to show how a recent wave of regulatory action - the EU Platform Work Directive, China's 2027 algorithmic-bargaining rule, New York City's minimum-pay regime, Uruguay's data-portability law - is starting to constrain those engineering choices in concrete ways. Fourth, and constructively, to sketch a rider-centric intelligence platform, FindMyRider, that uses retrieval-augmented inference and explainable visualisation to surface information that platforms currently keep on their own side of the interface.

Two methodological commitments shape the paper. I read the technical literature - Uber engineering blogs, Deliveroo system-design write-ups, peer-reviewed work on order batching and weather-aware forecasting - as primary sources, treating the engineering choices they describe as substantive design claims rather than as marketing. And I read the labour and regulatory literature symmetrically: a Berkeley Labor Center net-pay study, a Stanford gender-gap analysis, the Bologna ruling on Deliveroo's Frank algorithm, and the EU Directive 2024/2831 are sources of evidence about the same system, not a separate 'ethics chapter'. The point is to keep the technical and the social on the same page, because in practice they are.

The remainder of the paper is organised as follows. Section 2 reviews related work. Section 3 traces the methodological evolution of spatio-temporal demand forecasting. Section 4 examines dispatch and matching engines, with mathematical formulations of order batching. Section 5 covers sensor fusion and trip-state inference. Section 6 treats the economics of rider earnings. Section 7 discusses algorithmic management and gamification. Section 8 covers weather-aware and extreme-event modelling. Section 9 surveys the regulatory frontier. Section 10 presents four case vignettes. Section 11 develops the FindMyRider proposal. Section 12 discusses limitations and ethical considerations, and Section 13 concludes.

Related Work

Three streams of literature meet in this paper. The first is the technical literature on platform logistics. Uber's engineering team has documented their forecasting and dispatch architecture in considerable detail, including the ELK-based prediction service the DeepETA residual model the Trip State Model that fuses inertial sensor data with GPS guidance heatmaps powered by Deep Gaussian Mixture Models extreme-event neural forecasting and the Apache Pinot OLAP layer that supports operational catalogue queries [4,10-15]. Deliveroo's published material is sparser but high signal: their Frank dispatch heuristic, the migration of the Dispatcher service from PostgreSQL to DynamoDB and assorted operational analyses [5,16-19]. To these I add academic work on order batching as a multi-stage optimisation problem and on ETA prediction in transportation more broadly [20-22].

The second stream is the labour and behavioural literature on the gig economy. Can’t and Mowat-style critical readings, summarised in the Frontiers systematic review of algorithmic management frame the courier app as a managerial system that has absorbed hiring, task assignment and performance evaluation. Van Doorn's 'wage to wager' essay is an influential conceptual piece on dynamic pricing as wagering behaviour [3,6,23]. The empirical core for this paper is the Berkeley Labor Center's 2024 net-pay study across five US metros the Stanford gender-earnings-gap analysis of more than a million Uber rideshare drivers the Gridwise 2025/26 Uber Eats earnings data and the 2026 Gig Mobility Report on utilization [24-28]. Gamification and oppositional play are covered by StriveCloud's industry case a CUHK ethnography of Uber drivers and a Platform Labor study comparing NYC and Beijing models [29-31].

The third stream is regulatory. The EU Directive 2024/2831 introduced a rebuttable presumption of employment, information rights and human-in-the-loop requirements for algorithmic decisions [32]. China's 2027 platform-labour rules formalise minimum-wage, working-hour and algorithmic-bargaining requirements for an estimated 200 million workers [33]. New York City's Department of Consumer and Worker Protection raised the delivery-worker minimum pay rate to USD 22.13 per hour for 2026 Uruguay's Law 20.396 codified portable digital reputation and Massachusetts settled a USD 32.50/hour minimum for app-based drivers in 2024 [32,33]. The Italian Tribunale di Bologna ruling on Deliveroo's Frank ranking algorithm in January 2021 remains the most cited European precedent on algorithmic discrimination [34].

Despite each stream being individually rich, integrative treatments remain scarce. Most engineering blogs avoid labour and regulatory questions; most labour studies treat the algorithm as a black box. This paper is an attempt to read across all three, with a particular emphasis on points where engineering choices become labour-policy questions and vice versa.

Spatio-Temporal Demand Forecasting

At the centre of every on-demand delivery platform sits a forecasting engine. Unlike a purely digital service, food delivery is constrained by physical geography, by road throughput, and by the irreducible noise of human behaviour [1,35]. To keep the marketplace frictionless, the platform has to anticipate demand before it materialises, then nudge supply - via heatmaps, surge pricing, or direct dispatch hints - toward the cells where it will be needed [35].

The methodological history of this problem is, in some ways, the history of applied machine learning compressed into a single domain.

From KNN to LSTM: Methodological Evolution

Uber's earliest forecasting systems leaned on k-nearest-neighbour regression over similar historical trips, indexed by geolocation and time-of-day variables [4]. KNN had two attractive properties: it was product-aware - meaning the system could distinguish UberX from Uber Eats, and account for the pickup-and-prepare phase that exists only in food - and it was conceptually transparent for engineers and city operations teams [4]. As order volumes grew, however, the limits of nearest-neighbour and of classical statistical models such as ARIMA and Holt-Winters exponential smoothing became visible [35,14]. Hyper-local demand is not stationary, not linear, and not well behaved over long horizons [36].

The pivot toward Long Short-Term Memory networks (LSTMs) followed naturally. LSTM-based architectures handle long-range temporal dependencies, can ingest multiple data streams-weather, calendar features, marketing pushes, traffic state - and tolerate the heterogeneity that emerges when a single model has to serve hundreds of cities at once [14,36]. Uber's published work on extreme-event forecasting describes training a single flexible neural network on data drawn from thousands of cities simultaneously, on the grounds that pooling heterogeneous time-series sharpens accuracy in cells with sparse history, such as cities where New Year's Eve has only six recorded instances since launch [14]. The SmartMeals architecture, developed for institutional food demand, takes this further by combining LSTM layers with attention and engineered temporal features (categorical weather, humidity, day-of-week) to capture nonlinear coupling between climate and consumption [36].

Infrastructure follows the model. Uber's prediction service runs on an Elasticsearch-Logstash-Kibana (ELK) stack that integrates historical and near-real-time signals published to Kafka topics on trip completion [4]. The point of this design is not virtuosic engineering for its own sake but the requirement that the model be queryable, low-latency, and globally available; a forecast that arrives 200 ms late at lunchtime is operationally worthless.

Deepeta And the Residual Learning Pattern

Estimated Time of Arrival is the most consequential single number a delivery platform produces. It influences which courier is dispatched, what fare is quoted, how the customer experiences waiting, and whether the food arrives still warm [10-21]. Traditional routing engines build the ETA by summing per-segment traversal times across a road-network graph. That base estimate is good on long highway stretches and unreliable in dense urban centres, where parking, double-stops, weather and pedestrian density introduce variance the graph cannot see [10].

Uber's response was DeepETA, a low-latency deep network trained to predict the residual error between the routing-engine baseline and the observed outcome [10]. The architecture relies on multiple feature hashing of latitude-longitude pairs into independent buckets, and the published reasoning is worth pausing on: a single hash collapses geographically distinct cells into the same bucket and degrades accuracy, but combining several independent hashes lets the network learn to undo collisions, in the same spirit as a Bloom filter that gains specificity by combining weak signals [10]. The DeepETA serving path is among the highest queries-per-second systems on the platform, which puts a hard ceiling on model depth and incentivises careful inference-side optimisation [2,10].

The residual-learning pattern - keep an interpretable physical baseline, layer a learned correction on top - is, in my view, one of the most important practical lessons of the modern logistics stack. It preserves the explainability of the underlying routing engine, isolates the parts that benefit from data-driven adjustment, and gives a clean target on which to measure improvement (residual MAE) without conflating model gains with baseline improvements.

Probabilistic Forecasts and Heavy-Tailed Earnings

A single-point forecast becomes inadequate in highly volatile environments such as airports or city centres at peak mealtime. Uber's Earnings Per Hour (EPH) and Estimated Time to Request (ETR) services use Deep Gaussian Mixture Models to predict not a mean value but a distribution of possible outcomes [12,13]. The reason is empirical: rider earnings exhibit a heavy left tail. Many drivers, on a quiet Monday afternoon at an airport staging lot, earn essentially zero for an hour while the few who happen to be at the front of the queue clear a steady wage [12,13]. Communicating only a mean to such a population is misleading and operationally counter-productive - it draws too many drivers to the lot, deepens the tail, and worsens the reliability of the prediction itself.

By forecasting the variance and the mean jointly, the platform can give riders a more honest picture of their earnings prospects and can use the predicted distribution as input into market-balancing logic - for instance, by capping how many drivers are routed toward an oversaturated zone [12]. The same probabilistic framing flows into the guidance heatmap, where deep probabilistic models incorporate weather and time-of-day signals to predict load minutes to hours ahead [13,37]. From a Bayesian perspective, this is the right move: the expected utility of a courier-side decision (drive twenty minutes to the lot or stay near a high-density restaurant cluster) depends on the dispersion of outcomes, not just the mean.

Comparative Summary of Forecasting Approaches

Table 1 summarises the four families of forecasting model that recur in the published literature on platform logistics, with primary technique, advantage and limitation. The progression from classical statistical to probabilistic deep learning mirrors a wider trend in operations research, but the constraints (queries per second, model freshness, interpretability for operations teams) are unusually tight in this domain.

Family	Primary technique	Advantage	Limitation
Classical statistical	ARIMA, Holt-Winters, Theta [35]	Cheap, interpretable, easy to monitor [35]	Assumes linearity / stationarity [36]
Machine learning	KNN, Random Forest, XGBoost [4][22]	Handles high-dimensional features [4]	Weak on long-range temporal dependencies [36]
Deep learning	LSTM, CNN-LSTM, DeepETA [14][10]	Captures non-linear and multi-source interactions [14]	Higher latency; more complex training pipelines [10][38]
Probabilistic deep	Deep Gaussian Mixture Models [20,21]	Predicts full distribution and tail behaviour	Needs large datasets for tail accuracy [12]

Table 1: Forecasting Model Families Used in On-Demand Delivery Platforms

Dispatch and Matching: Logistics Orchestration

Forecasting tells the platform where demand will appear. The dispatch engine is what turns that prediction into a concrete pairing of an order, a kitchen and a courier. It is the operational nerve centre, and it must solve a high-frequency matching problem under hard time constraints, often within a couple of seconds [16].

Frank: Predictive Kitchen Times and Heuristic Matching

Deliveroo's Frank algorithm, deployed since 2017, is the canonical published example of a predictive matching engine in food delivery [2rwqedxzsqzc0] [16]. Frank uses machine-learned models to estimate the exact moment a meal will be ready for pickup, conditioned on order type, restaurant, and time-of-day. It then combines that estimate with travel-time predictions that consider distance and the characteristics of individual riders, including vehicle type and historical travel efficiency [16,18].

The objective Frank optimises is, in plain language, the minimisation of dead time - intervals during which an order sits cooling on a counter or a rider stands idle [1,11]. Deliveroo reports a 20% reduction in average delivery time after Frank's introduction, with average delivery times in some Irish markets falling to as low as seven minutes [16]. What is striking about Frank is not that it solves the matching problem optimally - it deliberately does not. A globally optimal matcher would be too slow to absorb the queries-per-second cliffs that occur at lunch and dinner. Instead, Frank uses heuristics that produce locally good assignments at speed, accepting bounded suboptimality in exchange for the ability to keep up with a traffic spike [1,2]. This trade-off - good enough fast, rather than optimal slow - is one of the defining design patterns of the modern delivery stack.

Order Batching and the Multi-Stop Optimisation Problem

When demand outpaces supply, platforms turn to order batching. Uber Eats's triple batching allows up to three orders to be carried in one trip, drawn from the same merchant or, increasingly, from two adjacent merchants [39,40]. Batching produces a complex trade-off: it raises throughput and reduces unfulfilled orders during peak periods, but it lengthens individual delivery windows and introduces a real risk of cold food arriving at the customer [39,40].
Formally, the batching problem can be cast as a multi-stage mixed-integer optimisation [25]. Let O denote the set of pending orders, B the set of candidate batches, and L the set of merchant locations. Define decision variables:

$x_{o,b} \in \{0,1\}$: 1 if order o is assigned to batch b;
$y_{b,i,j} \in \{0,1\}$: 1 if batch b travels from location i to location j;
$z_{b,i} \in \{0,1\}$: 1 if batch b visits location i.

The platform minimises a weighted sum of total travel distance and lateness penalties subject to time-window, capacity, and pickup-precedence constraints [20]. Concretely, the objective takes the form:

$$\min \sum_{b,i,j} d_{i,j} \cdot y_{b,i,j} + \lambda \cdot \sum_{o} \max(0, t_o - \tau_o),$$

where $d_{i,j}$ is the travel distance between i and j, t_o is the realised delivery time of order o, τ_o is its promised time, and λ trades off distance cost against lateness. Hard constraints enforce that every assigned order is picked up before it is delivered, that batch capacity is respected, that visited locations form a connected path, and that delivery windows are honoured for orders whose merchant has flagged them as time critical [20].

In practice, platforms do not solve this problem to global optimality online. They use two-stage metaheuristics - a clustering or assignment step followed by a routing step - that produce feasible high-quality solutions within the few-hundred-millisecond budget of a dispatch tick [20]. Uber's published material describes balancing the optimisation against eater-merchant distance and current rider-fleet availability, with explicit tunable weights between time-to-customer and per-rider efficiency [39].

Geospatial Infrastructure: NoSQL, Geohashes and QuadTrees

All of this assumes an infrastructure layer capable of handling millions of simultaneous orders and thousands of GPS updates per second. Deliveroo's migration of its Dispatcher service from Amazon RDS for PostgreSQL to Amazon DynamoDB is informative on this point [19]. DynamoDB provides consistent single-digit millisecond reads and writes regardless of table size, which matches the behaviour required by a dispatcher that must query the state of every active order and every active rider on every tick [19].

The published data model uses a single-table design with prefixed identifiers - o#<order_id> for order retrieval, r#<rider_id> for rider state, p#<pickup_id> for the one-to-many relationship between pickups and batched deliveries [19]. Geospatial lookups are implemented via geohash prefixes or QuadTrees, which collapse a continuous latitude-longitude into a discrete grid cell and let the system answer 'which available riders are near this kitchen?' in sub-second time [1,19]. Table 2 summarises the operational challenges and corresponding technical solutions.

Operational challenge	Technical solution	Impact on actors
Real-time matching	Heuristic dispatch (Frank) [16]	Faster deliveries, higher rider utilisation [17]
Capacity management	Multi-order batching [39]	Improved reliability under peak load [40]
GPS signal noise	Trip State Model (sensor fusion) [11]	Precise dispatch, less idle time [11]
Scalability	DynamoDB / NoSQL migration [19]	Single-digit millisecond response [19]

Table 2: Operational Challenges and Infrastructure Responses in Dispatch Systems

Sensor Fusion and the Trip State Model

Anyone who has tried to navigate central London by phone GPS on a rainy evening knows that the signal is not the territory. Tall buildings produce the so-called urban canyon effect, in which reflected and obstructed signals reduce GPS accuracy precisely in the dense urban cores where delivery demand is highest [11]. For a dispatcher, the practical question is unusually fragile: has the rider arrived at the restaurant, are they still circling for parking, are they walking to the counter, or are they already heading to the customer?

Uber Eats's response is the Trip State Model, which augments raw GPS with high-frequency motion-sensor data drawn from the courier's phone via the Android Activity Recognition Client API [11]. Eight activity labels - walking, driving, tilting, on bicycle, running, still, in vehicle, unknown - are inferred from accelerometer and gyroscope readings and fed into a Conditional Random Field, a sequential probabilistic model well suited to inferring discrete state sequences from noisy observations [11].

Concretely, the CRF segments the pickup process into five phases:
- Arrived at restaurant, detected when the rider begins circling;
- Parked, when the rider transitions from a moving vehicle to walking;
- Waiting at restaurant, the interval between arriving at the counter and picking up the food;
- Walking to vehicle, marking the start of the delivery leg; and
- Route to eater, confirming motion has resumed [11].

By accumulating these state sequences across millions of trips per restaurant, the platform can learn that, say, a particular Soho izakaya has a reliably long wait but a quick handover, while a chain pizzeria in Wandsworth has the opposite pattern. Dispatch can then be deferred so that the rider arrives just as the food does, which improves both food quality and rider efficiency [11].

Stepping back, the Trip State Model is a clean illustration of a broader pattern in platform logistics: physical-world phenomena are 'captured' into a digital grammar that can be optimised statistically [31]. It also raises the design question I return to in Section 11 - whether riders themselves should have access to the inferred states of their own working day, and not only their employer's algorithm.

Labour Economics: Earnings, Expenses and Volatility

The economic promise of the gig economy - independence and flexibility - is mediated in practice by a piece-rate model in which the wage is itself an object of constant prediction and reconfiguration [7,23]. Delivery earnings are nonlinear in hours worked, sensitive to utilisation, dependent on tipping, and shaped by the rider's ability to read and react to algorithmic incentives [6,27,28].

Gross and Net Earnings: A Sharper Look at Net

On gross numbers alone, the picture for delivery work in the United States looks unremarkable. Gridwise's analysis of more than 100,000 Uber Eats drivers in 2025 reports a median total trip pay of USD 14.07 per hour and a median gross pay including tips and bonuses of USD 15.03 [27]. The top decile clears USD 22.47 [27]. Table 3 reproduces those figures.

Percentile	Total trips pay (USD/hr)	Gross with tips & bonuses (USD/hr)	Pay per delivery (USD)
Median (P50)	14.07	15.03	8.16
Top 25% (P75)	17.02	18.57	9.79
Top 10% (P90)	20.83	22.47	12.00

Table 3: Uber Eats Driver Earnings Percentiles, US, 2025 [29]

Utilisation and the Cost of Idle Time

The 2026 Gig Mobility Report puts late-2025 rider utilisation at 58.69%, recovering from a post-pandemic dip but still below the pandemic-era peak of around 68% [38]. Utilisation matters because the time a rider spends waiting between requests - generally classified as P1 time - is uncompensated under most platforms' pricing rules [24]. A rider with a USD 15/hour gross figure who is on active delivery only 60% of the time is, in effect, earning USD 9 per clock-hour, before vehicle and tax costs.

When the Berkeley Labor Center examined this gap with operational expenses subtracted -vehicle wear, fuel, insurance - the picture became sharper still [24,25]. In California, where Proposition 22 supplies a minimum-pay floor, median net hourly earnings excluding tips were USD 5.93. In Boston, Chicago and Seattle the same figure fell to as low as USD 0.48 per hour without tips [24,32]. Even with tips - which the same study estimates make up roughly 50% of per-delivery income for couriers- the employee-equivalent net pay (adjusted for the employer-side payroll taxes that gig workers must self-fund) was estimated at USD 8.36 per hour in non-California metros [28,24,25].

These numbers do not show up in platform marketing material, and they are not always visible to riders themselves, who often see only a per-trip payout that bundles base pay, distance pay and tip into a single line. The opacity is consequential: a rider who cannot easily inspect the ratio of gross to net cannot make rational shift-allocation decisions, which is one of the conditions under which gamified incentives become particularly effective [31,23].

Learning by Doing and the Gender Earnings Gap

An influential Stanford analysis of more than a million Uber rideshare drivers found a 7% gender earnings gap that disappeared once three factors were controlled for [26]. Crucially, none of those factors were direct platform discrimination. They were, in order: experience on the platform - drivers with more than 2,500 completed trips earn roughly 14% more per hour than those with fewer than 100, because they have learned which trips to accept, which neighbourhoods to work, and how to time shifts; preferences over where and when to drive, often shaped by safety considerations and residential location, with men working in higher-paying areas at higher-paying times [26,26]; and average driving speed, where men tend to drive marginally faster, accumulating more per-mile pay across more trips per hour [26].

The empirical finding - a measurable gap that is structural rather than discriminatory in the algorithmic-decision sense - is sometimes used to argue that platform pay is fair. I would read it more carefully than that. Learning-by-doing returns are themselves a function of the information environment the platform creates; if the system reveals little about which acceptances are profitable, it is not surprising that experience compounds slowly and unevenly, and that newcomers (disproportionately women, in this dataset) start at a disadvantage. Section 11's FindMyRider proposal is partly motivated by this observation.

Algorithmic Management and Gamification

Algorithmic management is the defining feature of platform labour. Computational systems take on functions historically associated with human management - hiring, task assignment, performance evaluation, sanctioning - and exercise them at a scale that no human supervisor could match [3,6]. To do so without obvious coercion, platforms layer 'soft' controls on top, with gamification being the most visible [29,30].

Behavioural Nudges and Variable Reinforcement

The interface vocabulary is now familiar: points, badges, progress bars, streaks, milestone counters, and a 'pull-to-refresh' interaction whose closest behavioural analogue is the lever of a slot machine [7,29]. Uber One's rider and driver progress bars highlight how close a worker is to the next benefit threshold, turning the otherwise mundane act of completing a trip into a quest [29]. The pull-to-refresh gesture is a textbook variable-reinforcement schedule: the reward distribution is unpredictable, but the act of refreshing yields occasional payoffs, which is precisely the design known to sustain engagement under uncertainty [7]. These mechanics exploit goal-setting tendencies: 'almost at your target' messages exploit the same near-miss psychology that supports gambling engagement [7].

Two Models of Labour Control

Comparative ethnographic work identifies two distinct gamification regimes [31]. The first, the NYC 'Deal or No Deal' model, has couriers waiting on a ping and evaluating individualised, variable offers under dynamic pricing. Hit frequency varies, accept-rate metrics are tracked, and refusing too many offers carries soft penalties [31]. The second, the Beijing 'Grab-and-Stack' model, presents crowdsourced riders with a list of available orders that they grab in real time, encouraging strategic stacking - how many orders can a rider plausibly carry without missing any deadline? - and direct peer competition [31].

Despite their surface differences, both models shift the economic risk of idle time and traffic variability onto the courier rather than the platform [31,23]. Whether this counts as flexibility or as managerial intensification is a question the next section's regulatory regimes are starting to settle in the affirmative for the second reading.

Worker Resistance and Oppositional Play

Workers do not absorb algorithmic management passively. The CUHK ethnography documents what its authors call oppositional play - a repertoire of work-games invented by drivers in conversation with the algorithm and with each other [30]. Three of these recur across cities. Mario Karting refers to manoeuvring through traffic in ways that beat the app's time estimates [30]. Shuffling is the strategic accept-then-cancel pattern used to nudge surge pricing or to dodge predictably unprofitable runs [30]. Longhauling is the practice of taking longer routes when distance-based compensation outweighs time-based pay [30].

These tactics cut against the standard corporate framing of the courier as a passive resource to be dispatched. They also tell us something useful about the system: algorithmic management is not omniscient; it produces local information asymmetries that workers can read and exploit. Treating riders as active agents rather than as resources is, in my view, an underused lens in both engineering and policy work.

Weather and Extreme-Event Forecasting

Weather is the single largest exogenous shock to delivery operations. A heavy rainfall event raises consumer demand (more orders, fewer people willing to walk to the restaurant) and simultaneously shrinks supply (fewer riders willing to work, slower travel), producing the kind of compound stress that tests the entire stack at once [22,41].

A systematic literature review on weather-related disruption in transportation found that rainfall is the most-studied event in the field, followed by snowfall [41]. Rainfall amplifies delays through indirect mechanisms - flooding, reduced visibility, slick surfaces - rather than only through direct movement penalties [41]. In higher-latitude markets, snowfall reshapes both ETA models and incentive structures, because the marginal cost of an extra trip rises sharply with road conditions [12,41].

Uber's Deep Probabilistic Models for the guidance heatmap incorporate weather signals directly, allowing minutes-to-hours-ahead load forecasts that drive pre-positioning of riders and dynamic-pricing adjustments [13,37]. The SmartMeals architecture goes further by encoding categorical weather conditions (Clear, Rain) and humidity into LSTM input vectors, capturing nonlinear interactions between climate and consumption that a simpler model misses [36].

Extreme events - New Year's Eve, major concerts, finals weekends - represent an even harder problem because historical data are sparse [14]. The Uber engineering response is to train a single, flexible neural network on data drawn from thousands of cities at once, on the principle that aggregating heterogeneous time series exposes patterns that no single city's history would reveal [14]. External signals such as population growth and driver-incentive structures have to be folded in explicitly, because otherwise the model can produce confident but operationally damaging mismatches between supply and demand [14,12].

Regulatory Frontiers

The 'wild west' phase of the gig economy is closing. Regulators in multiple jurisdictions have moved within the last two years from observation to active intervention, and the engineering decisions described above are now being constrained, in some cases very specifically, by law [8,33,32].

The European Union: Directive 2024/2831

The Platform Work Directive 2024/2831, adopted in October 2024, contains three provisions of direct technical consequence [32]. The first is the rebuttable legal presumption of employment: where the platform exercises direction or control - defined in part by use of algorithms to set pay, monitor performance or restrict working hours - the worker is presumed an employee, and the burden of proof shifts to the platform [33,32]. The second is a set of information rights: workers must be informed about the data the platform collects, and the parameters used in automated decisions affecting recruitment, earnings or working conditions [32]. The third is a human-oversight requirement: critical decisions, including account suspension or termination, must be taken or reviewed by a human being, and platforms must allocate qualified human resources to evaluate individual algorithmic decisions [32]. The Directive also bans the automated processing of certain sensitive data, including emotional state and private worker conversations [32].

China: Algorithmic Bargaining by 2027

China's CPC Central Committee and State Council issued comprehensive labour rules for platform workers in 2026, with a 2027 compliance deadline. The rules require the app itself to enforce minimum wages and maximum working hours - if a courier hits the cap, the app must stop sending them orders [33]. Most strikingly, algorithms governing piece rates, commission structures and task assignment are to be subject to collective bargaining with labour unions or worker representatives [33]. This is not transparency in the Western sense of publishing a report about the algorithm; it is a requirement that the code itself become a subject of negotiation [33]. The proximate trigger was public outcry after investigations found that successive algorithm updates had progressively shortened delivery deadlines, pushing riders into dangerous driving behaviour to keep up [33].

The Americas

The United States regulatory landscape remains fragmented. New York City's Department of Consumer and Worker Protection raised the delivery-worker minimum pay rate to USD 22.13 per hour effective April 2026, indexed annually for inflation [42,32]. The DCWP's own analysis projects that minimum-pay floors of this kind raise delivery efficiency by encouraging platforms to use rider time more deliberately [42]. Massachusetts, following a 2024 ballot settlement, established a USD 32.50/hour minimum and granted app-based drivers the right to unionise and to negotiate over algorithmic management [32]. In Latin America, Mexico and Uruguay have each enacted laws targeting contractual transparency and digital reputation. Uruguay's Law 20.396 treats a worker's digital reputation as portable capital, granting a right of access to the data collected during the engagement and a right to carry that reputation across platforms [32].

A Comparative Snapshot

Table 4 summarises the regulatory frontier. The juxtaposition is more revealing than any single regime in isolation: jurisdictions are converging on a common set of concerns - employment status, transparency, human oversight, minimum pay - while differing sharply on enforcement mechanism.

Region	Primary regulatory focus	Key legislation / directive	Human-in-the-loop requirement
European Union	Employment status; data protection [32]	Directive 2024/2831 [32]	Yes, for terminations [32]
China	Algorithm collective bargaining [33]	2027 compliance rule [33]	Yes, for working-hour limits [33]
Singapore	Work injury; representation [32]	Platform Workers Act 2024 [32]	For management control [32]
New York City	Minimum-pay floors [32]	DCWP Minimum Pay Rate (USD 22.13/hr, 2026) [32]	Indirect (through earnings) [32]
Massachusetts	Unionisation; algorithmic management [32]	2024 ballot settlement	Negotiated
Uruguay	Digital-reputation portability [32]	Law 20.396 (2025) [32]	Yes, right to explanation [32]

Table 4: Comparative Regulatory Regimes for On-Demand Delivery Platforms

Case Vignettes

The argument so far has moved at the level of architectures and aggregate statistics. This section grounds it in four concrete cases, each chosen because it illustrates the interaction between an engineering decision and a labour or regulatory consequence.

The Bologna Frank Ruling, January 2021

In January 2021, the Tribunale di Bologna ruled against Deliveroo's Frank ranking algorithm in a case brought by three Italian unions [34]. The court found that Frank treated all forms of last-minute non-availability identically: a rider who failed to log in because they were ill, exercising a protected right to strike, or caring for a child was downgraded in the same way as a rider who had simply not bothered to show up. By ranking those riders lower for shift-reservation purposes, Frank produced indirect discrimination on grounds of legally protected behaviour [34]. The court ordered Deliveroo to pay EUR 50,000 in damages and to publish the ruling. Deliveroo's general manager noted the decision but stated that the shift-reservation system tied to the ranking was no longer in market use [34].

The technical lesson is precise. A ranking model that uses 'reliability' as a feature without conditioning on the reason for absence is, in effect, encoding 'reliability of attendance' as if it were the same thing as 'willingness to work'. In any jurisdiction with strike or anti-discrimination law, that is a design defect. The case has since been cited extensively in European debate around explainable and contestable AI, and it is the proximate ancestor of the EU Directive's information-rights and human-review provisions [34,32].

New York City's Minimum Pay Rate Rollout, 2023-2026

In 2022 New York's DCWP published a study finding that a typical app-based delivery worker, after expenses, was earning roughly USD 7 per hour, with no benefits and significant out-of-pocket costs [42]. The city used Local Law 115 of 2021 to introduce a minimum pay rate, initially set near USD 17.96/hour and subsequently adjusted; by April 2026 the rate had reached USD 22.13/hour, indexed annually to inflation [32]. Platforms initially resisted, arguing the rule would either reduce delivery volume or push platforms to deactivate workers. In practice, platforms responded by tightening their dispatch algorithms - cutting idle time, batching more aggressively, deactivating low-utilisation riders - and the regulator's own analysis found that delivery efficiency rose, not fell [42].

The vignette is instructive because it shows a regulatory floor producing optimisation pressure of the kind a market alone had not. A binding minimum changes which engineering choices are profitable: when paid time becomes more expensive, the value of forecasting accuracy and dispatch precision goes up. This is, paradoxically, a case where labour regulation made the algorithmic stack more sophisticated, not less.

Beijing, Algorithmic Tightening and the Road to 2027

Chinese investigative journalism in the late 2010s documented how meituan and ele.me had progressively shortened average delivery times through successive algorithm updates - in some cases by minutes per order, year on year [33]. The reduction made commercial sense at the platform level but pushed individual riders into running red lights, riding against traffic, and accumulating injuries. Public outcry at this pattern is the proximate political reason for the 2026 CPC State Council rules and the 2027 compliance deadline that mandates app-enforced working-hour limits and algorithmic bargaining [33].

The case is a clean illustration of how an objective function defined narrowly - minimise average delivery time - can, through repeated optimisation, produce outcomes that are unacceptable on a social scale, and how a regulator can respond by reaching directly into the objective function rather than only adjusting compensation downstream.

A London Friday Evening: A Compound Stress Test

I want to end this section with a small, less formal vignette, drawn from time spent observing rider behaviour in Soho and Shoreditch on Friday evenings during 2024 and 2025. The stack handles a stress test of this kind reasonably well most of the time. On a particularly wet Friday in October 2024, however, three things stacked: a Tube line was down, a major sporting event was finishing, and rain intensified between 18:00 and 19:30. Order volumes on the platforms I could observe rose noticeably; pickup waits at popular merchants stretched past fifteen minutes; rider chat groups - a useful informal data source - filled with screenshots of triple-batched orders that, on the ground, were producing 25-minute customer waits.

What the stack did well: forecasting picked up the demand surge, and surge incentives rose. What it did less well: the dispatch algorithm continued to assign new orders to riders already carrying two, beyond the point at which the third order's customer would receive a credible delivery window. From the rider side, this looked like the system optimising for completion rate at the expense of both food quality and rider stress. From the customer side, it looked like a platform failure. Both views are partially correct. The episode is the kind of edge-case in which the FindMyRider proposal in the next section is intended to help.

A Constructive Proposal: Find My Rider

Across the literature reviewed in this paper, one structural asymmetry recurs. The platform sees the full state of the system - demand forecasts, dispatch decisions, expected earnings, sensor-inferred trip states - while the rider sees only the surface artefacts of those computations: a ping, a pay quote, a heatmap. FindMyRider is a research blueprint for a rider-centric intelligence platform that closes part of that asymmetry without trying to replicate the platform's commercial functions.

The system has three pillars. First, demand prediction tuned to the rider's geographic context and shift availability, not to an abstract city-wide forecast. Second, density-aware heatmap visualisation that ranks neighbourhoods by net-expected earnings rather than gross order density, factoring in the rider's vehicle type, expected utilisation and tipping history. Third, an explainability layer that maintains a contestable decision log - what the system inferred, what evidence supported the inference, and how the rider can contest a downstream platform action.

The intended user is the self-employed courier who works across multiple platforms (a common pattern in the UK, where Deliveroo, Uber Eats, Just Eat and smaller logistics services compete for rider supply). The intended use case is shifting planning, route choice, and post-hoc dispute resolution - not real-time dispatch.

System Architecture Overview

FindMyRider is structured in five layers. The data layer ingests publicly available signals -weather, transit disruption, sporting and event calendars, neighbourhood-level demand proxies derived from anonymous order-volume APIs where these are available, and the rider's own trip history (with explicit consent). The forecasting layer combines a temporal-attention LSTM for short-horizon demand with a Deep Gaussian Mixture component for earnings-distribution estimation, mirroring the published Uber EPH/ETR architecture but trained on rider-side data [36,12-14].

The retrieval-augmented reasoning layer is the most novel component. A retrieval-augmented generation (RAG) engine indexes a curated corpus of platform terms-of-service excerpts, regulatory documents (NYC DCWP rules, EU Directive 2024/2831 articles, Uruguay Law 20.396), case law (the Bologna Frank ruling), and union-published guidance. When a rider asks a natural-language question - 'why was my account temporarily suspended?', 'is this surge offer worth accepting given my expenses?' - the system retrieves the relevant grounding passages and produces an answer that cites them, rather than free-form generation. The intent is explicitly to suppress hallucination by anchoring outputs in retrievable text.

The visualisation layer renders the heatmap, overlays event and weather signals and surfaces the predicted earnings distribution as a small-multiples chart per neighbourhood rather than a single mean. The audit layer maintains an append-only decision log per rider: every recommendation the system produces, the inputs that fed into it, and the outputs the rider acted on. The log is intended to be exportable in a format compatible with the EU Directive's information-rights and Uruguay's data-portability provisions, so that a rider can take their own usage history with them across platforms [32].

Evaluation Plan

FindMyRider should be evaluated along four axes. Forecasting accuracy is measured by neighbourhood-level mean absolute error on demand and earnings, against held-out periods. Calibration of the probabilistic component is measured by reliability diagrams on predicted versus realised earnings distributions. Decision quality from the rider's perspective is measured by self-reported earnings improvement and self-reported reduction in idle time over a multi-week trial. Explainability and contestability are measured by qualitative evaluation of the audit log: can a rider, presented with a sample log, understand which recommendation produced which action and on what evidence?

Two ethical safeguards are non-negotiable. First, no rider data should leave the device or trusted endpoint without explicit, granular consent; second, the system should not be used by platforms to infer rider behaviour in ways that would produce new managerial controls. The risk that a rider-empowerment tool becomes a feedstock for the very platforms it aims to balance is real, and design must take it seriously.

Discussion, Limitations and Ethical Reflection

This paper has argued that platform delivery has entered a hyper-algorithmic phase, and that the technical, economic and regulatory layers can no longer be analysed in isolation. Three observations follow.

First, residual learning, heuristic dispatch, and probabilistic forecasting are now mature enough to be treated as design patterns rather than as cutting-edge techniques. The next interesting questions are not how to make these models more accurate, but how to make them auditable, contestable and compatible with the regulatory regimes that have appeared since 2024. The Bologna ruling is an illustration of what happens when those questions are deferred [34].

Second, the labour-economics story is harder than the engineering one, and not because the data are missing. Berkeley's net-pay numbers, the Stanford gender-gap analysis, and the DCWP studies are public, recent and unusually rigorous [24-26,42]. The harder problem is that platform engineering choices and labour outcomes are entangled in feedback loops - a more accurate forecaster can either reduce idle time or be used to thin out rider supply at the margin, depending on objective-function choices made elsewhere. Engineering papers tend to discuss the first effect and skip the second; labour papers do the reverse. Bridging that gap is a research agenda.

Third, the FindMyRider proposal is deliberately modest. It does not aim to replace the platform's dispatch logic or to replicate its commercial functions. It aims to give the rider a first-party intelligence layer of comparable quality to the platform's own. Whether such tools can survive without being co-opted - either by regulators as compliance instruments, or by platforms as competitive intelligence sources - is, in my view, the most important governance question for the next phase of work.

Limitations of this paper should be flagged plainly. The discussion of forecasting and dispatch leans heavily on Uber and Deliveroo engineering blogs, which are necessarily partial accounts. The earnings figures in Section 6 are US-centric; comparable UK and EU disaggregated data are harder to obtain and would change some of the absolute numbers, though almost certainly not the qualitative pattern. The FindMyRider proposal is a research blueprint, not a tested system; the evaluation plan in Section 11.2 is what it should be measured against. Finally, this is a single-author paper and reflects a single set of analytical choices; readers may reasonably disagree on emphasis, particularly around the question of whether the gig model is reformable through algorithmic transparency or whether it requires more fundamental restructuring.

Conclusion

Hyper-local delivery has been remade by a small set of technical advances - residual deep ETA prediction, heuristic real-time dispatch, sensor-fusion trip state inference, and distributed NoSQL infrastructure - that together produce a system of remarkable operational efficiency [10,16,11,19]. That efficiency comes with costs: low net pay once expenses are subtracted, high reliance on tipping, gamified incentive structures that exploit goal-setting psychology, and the very real labour-rights questions documented by the Bologna court, the Berkeley Labor Center and the regulators of the EU, China, NYC and Uruguay [24,30-34].

The next phase of the gig economy will be defined by what we might call automated governance: the embedding of compliance, explainability, and labour-rights protections directly into the algorithmic stack [43]. The challenge for the next generation of logistics engineers is to build systems that preserve the magic of sub-30-minute delivery while making the labour that powers it economically sustainable and dignified [2,18]. The nerve centre of the city, in this sense, is no longer a set of roads and restaurants. It is a contested field of data, code and collective bargaining [5,16,33]. The FindMyRider proposal is one small contribution to making the rider, rather than only the platform or the regulator, a first-class participant in that field.

References

1. Educative.io, (2026). 'Deliveroo System Design Explained,' Educative.io.
2. System Design Handbook, (2026). 'Deliveroo System Design Interview: A Step-by-Step Guide'.
3. Frontiers in Sociology, (2026). 'Algorithmic management in the global gig economy: an interdisciplinary systematic literature review and critical discourse analysis'.
4. Uber Engineering, (2017). 'Engineering Uber Predictions in Real Time with ELK.'
5. O. Moxon, 92024). 'An Operational Analysis of Deliveroo,' Medium.
6. Frontiers in Sociology, (2026). 'Algorithmic management in the global gig economy'.
7. TechPolicy.Press, (2025). 'The Game Behind the Gig Economy.'
8. TechPolicy.Press, (2025). 'The ILO Debate on Algorithmic Management Will Define Worker Rights in the Digital Economy.'
9. Taylor & Francis, (2025). 'Realising decent work for platform workers: a human rights approach.'
10. Uber Engineering, (2022). 'DeepETA: How Uber Predicts Arrival Times Using Deep Learning.'
11. Uber Engineering, (2018). 'How Trip Inferences and Machine Learning Optimize Delivery Times.'
12. Uber Engineering, (2025). 'Forecasting Models to Improve Driver Availability at Airports.'
13. Uber Engineering, (2025). 'Enhancing Uber's Guidance Heatmap with Deep Probabilistic Models.'
14. Uber Engineering, (2017). 'Engineering Extreme Event Forecasting at Uber with Recurrent Neural Networks.'
15. Uber Engineering, (2025). 'Blazing Fast OLAP on Uber's Inventory and Catalog Data with Apache Pinot.'
16. Irish Tech News, (2017). 'New algorithm Frank has helped reduce delivery times by 20% for Deliveroo.'
17. Silicon Republic, (2017). 'Food for thought: Algorithm tweak cuts Deliveroo delivery times by 20pc.'
18. Deliveroo, (2013). 'About us.'
19. AWS Database Blog, (2023). 'How Deliveroo migrated their Dispatcher service to Amazon DynamoDB.'
20. MDPI Applied Sciences, (2022). 'Two-Stage Metaheuristic Algorithms for Order-Batching and Routing Problems.'
21. PeerJ Computer Science, (2025). 'Ten quick tips for improving estimated time of arrival (ETA) predictions using machine learning in logistics and transportation.'
22. Scribd document repository, (2025). 'Predicting Food Delivery Times Accurately.'
23. N. Van Doorn, 'From a Wage to a Wager: Dynamic Pricing in the Gig Economy,' The Autonomy Institute.
24. UC Berkeley Labor Center, 'May 2024 - gig passenger and delivery driver pay in five metro areas.'
25. UC Berkeley Labor Center, (2024). 'Gig Passenger and Delivery Driver Pay in Five Metro Areas.'
26. C. Cook, R. Diamond, J. V. Hall, J. A. List and P. Oyer, (2020). 'The Gender Earnings Gap in the Gig Economy: Evidence from over a Million Rideshare Drivers,' Stanford University.
27. Gridwise, (2026). 'How Much Do Uber Eats Drivers Make in 2026? (2025 Data from 500k+Drivers).'
28. Food on Demand, '2026 Gig Mobility Report Shows Trends Shaping the Gig Economy.'
29. StriveCloud, 'Uber: A Great Example of How to Increase Customer Loyalty with Gamification.'
30. School of Journalism and Communication, (2022). CUHK, 'Gamification and work games: Examining consent and resistance among Uber drivers.'
31. Platform Labor, (2021). 'Odds stacked against workers: datafied gamification on Chinese and US food delivery platforms.'
32. International Labour Organization, April (2025). 'Decent work in the platform economy: law and practice update'.
33. The Next Web, (2026). 'China formalises gig worker protections for 200 million platform workers with algorithm transparency and 2027 deadline.'
34. Clifford Chance Talking Tech, (2026). 'The Italian courts lead the way on explainable AI.'
35. Uber Engineering, (2018). 'Forecasting at Uber: An Introduction.'
36. Atlantis Press, (2025). 'SmartMeals: Real-Time Institutional Food Demand Forecasting Using Hybrid LSTM with Attention and Temporal Feature Engineering.'
37. CSIS, (2025). 'AI for the Grid: Opportunities, Risks, and Safeguards.'
38. Uber Engineering, (2026). 'Accelerating Deep Learning: How Uber Optimized Petastorm for High-Throughput and Reproducible GPU Training.'
39. Uber Engineering, (2023). 'Everything you need to know about triple batching' (AU edition).
40. Uber Engineering, 'Everything you need to know about triple batching' (NZ edition).
41. MDPI Logistics, (2023). 'Weather-Related Disruptions in Transportation and Logistics: A Systematic Literature Review and a Policy Implementation Roadmap.'
42. NYC Department of Consumer and Worker Protection, (November 2022). 'A Minimum Pay Rate for App-Based

Restaurant Delivery Workers in NYC,'
43. Deliveroo Engineering Blog (2026).
44. Economic Policy Institute, (2018). 'Uber and the labor market: Uber drivers' compensation, wages, and the scale of Uber and the gig economy.'
45. PMC / NIH, (2024). 'How Algorithmic Management Influences Gig Workers' Job Crafting.'
46. National Employment Law Project, (2025). 'Towards Transparency and Accountability in App-Based Work.'
47. TechCrunch, (January 2021). 'Italian court rules against discriminatory Deliveroo rider-ranking algorithm'.
48. NYC Department of Consumer and Worker Protection, (January 2026) .'Landmark Protections Take Effect Today'.
49. Crowell & Moring, (Directive 2024/2831 analysis). 'New EU Directive Impacting Digital Platforms and Individuals Working for Them'.
50. Industrial Relations News (IOE), (2025). 'Uruguay: New law on platform workers' (Law 20.396).