



Research Paper

REAL-TIME SHARED-MOBILITY FLEET INTELLIGENCE FOR VEHICLE AVAILABILITY, TRIP DEMAND, UTILIZATION AND BATTERY-AWARE REBALANCING - (MOVE GRID)

¹ Mrs. B. Jyothi, ² Velagala Anirudh Sai, ³ Narsingoju Shivanand, ⁴ Bommeraboina Prakash, ⁵ Gopivaru Suraj

¹ Assistant Professor, ^{2,3,4,5} Students

Department of CSE (Data Science)

Siddhartha Institute of Technology & Sciences, Narapally

bjyothi.cse@siddhartha.org.in, 24tq1a6765@siddhartha.co.in, 24tq1a6734@siddhartha.co.in,
24tq1a6749@siddhartha.co.in, 24tq1a6718@siddhartha.co.in

ABSTRACT

Shared electric bicycles and scooters have become a common way of covering short distances in cities, especially between metro stations, offices, colleges, and residential areas. The service depends on a simple promise: a charged vehicle should be available close to where a rider wants to start a trip. In practice, vehicles collect at popular destinations, run out of charge in busy zones, and sit idle in quiet areas, while riders elsewhere open the app and find nothing nearby. Operators usually respond with fixed rebalancing routes and battery swap rounds planned from experience. This paper presents Move Grid, a real-time fleet intelligence system that uses streaming vehicle and trip data to forecast demand, monitor availability and utilisation, and plan battery-aware rebalancing.

Each vehicle reports its GPS position, battery state of charge, lock status, and fault codes every thirty seconds, and the booking platform records trip starts, trip ends, and failed searches where no vehicle was found. These streams are published to Apache Kafka and processed with Apache Flink, which maps positions to H3 hexagonal zones, validates readings, and computes rolling counts of available vehicles, usable vehicles above a battery threshold, and open demand for every zone. The processed data is stored in a TimescaleDB time-series database, while a daily batch pipeline prepares training data that combines trip history with weather, holidays, and local events.

The analytical core consists of three models. A LightGBM demand model forecasts trip starts per zone for each of the next four thirty-minute intervals using lagged demand, time of day, day type, rainfall, and nearby activity. A battery depletion model estimates the energy each vehicle will use on a typical trip in its zone and predicts when it will fall below the rentable threshold. A rebalancing optimiser formulated as a mixed-integer program with Google OR-Tools decides how many vehicles to move between zones and which low-battery vehicles to prioritise for swapping, subject to the capacity and shift time of each service van.

The system is designed for fleet operators, city mobility teams, and field supervisors. It recommends moves and swaps but leaves dispatch decisions to the operations staff, and it respects parking zones and restricted areas defined by the city. Future work

includes pricing incentives that encourage riders to end trips in under-supplied zones, integration with public transport timetables, and support for mixed fleets of bicycles, scooters, and mopeds.

I. INTRODUCTION

Shared micromobility services have spread quickly in Indian and international cities because they fill the gap between walking and motorised transport. A rider unlocks an electric bicycle or scooter with a phone, travels one to four kilometres, and leaves the vehicle at the destination. For commuters, these trips often complete the first or last leg of a journey on public transport. For operators, however, the service is expensive to run, because the fleet moves according to riders' choices rather than any plan, and every vehicle has a battery that must be kept charged.

The pattern of use is strongly directional. In the morning, vehicles flow from residential areas towards business districts and metro stations, and in the evening the flow reverses. Colleges, markets, and event venues create their own peaks. Without intervention, some zones accumulate far more vehicles than riders need while others empty out. Battery charge adds a second constraint, because a vehicle with low charge is visible on the map but cannot be rented, and a zone that looks well supplied may in fact have few usable vehicles. Most operators plan rebalancing and battery swaps with fixed routes and the judgement of supervisors, which cannot keep pace with changes caused by weather, holidays, or events.

The data needed for a better approach is already available. Connected vehicles report location and battery status continuously, and the booking platform records every trip and every failed search. Stream processing can turn these reports into an up-to-date picture of supply in each zone, machine learning can forecast demand for the next few hours, and optimisation methods can decide which moves and swaps will serve the most riders with the vans and crews available.

This work proposes Move Grid, a real-time fleet intelligence system for shared electric mobility. It combines a streaming data pipeline, zone-level demand forecasting, battery depletion prediction, and a battery-aware rebalancing optimiser, and it presents the results to operators and field crews in a form they can act on. The rest of the paper reviews related research, analyses the existing and proposed systems, and describes the methodology and architecture.

II. LITERATURE SURVEY

Research on bike-sharing systems began with docked schemes, where the number of bicycles at each station is directly observable. Early studies analysed trip records to describe usage patterns and found strong effects of time of day, day of week, weather, and proximity to transit stations. Station-level demand prediction was approached with Poisson regression, ARIMA models, and random forests, and later work showed that gradient boosting and neural networks captured non-linear weather effects better. Several researchers noted that predicting demand for each station separately ignores correlations between neighbouring stations, and they proposed clustering stations into groups before forecasting. With the arrival of dockless bicycles and scooters, the

concept of a station disappeared, and studies moved to grid cells or hexagonal zones as the basic unit of analysis, which is the approach adopted in this work.

Spatio-temporal forecasting of mobility demand has received wide attention in the context of ride hailing and taxi services. Researchers have used convolutional networks on city grids, graph convolutional networks on zone adjacency graphs, and recurrent networks to predict pickups over short horizons. These models generally outperform classical time-series methods when large volumes of data are available, but studies also report that tree-based models with carefully engineered lag and calendar features remain competitive and are easier to deploy and explain. An important observation is that recorded trips underestimate true demand, because riders who find no vehicle do not appear in trip data. Some studies therefore include failed searches or app opens as a signal of unmet demand, and this idea is used in the present system.

The rebalancing problem has been formulated in operations research as a vehicle routing or network flow problem. Static rebalancing, performed overnight when usage is low, was studied first, using mixed-integer programming to decide how many bicycles each truck should pick up and drop at each station. Dynamic rebalancing during the day is harder, since demand changes while trucks are moving, and researchers proposed rolling-horizon optimisation, heuristics, and reinforcement learning to handle it. Studies consistently report that the benefit of rebalancing depends strongly on the quality of the demand forecast used as input. Work on user-based rebalancing has also shown that price incentives can encourage riders to move vehicles themselves, reducing the number of truck trips required.

Battery management for electric shared vehicles introduces additional constraints. Studies of electric scooter fleets report that battery swapping and charging account for a large share of operating cost, and that vehicles with low charge reduce effective supply even when they are physically present. Researchers have modelled energy consumption per trip as a function of distance, rider weight, slope, and temperature, and have proposed joint optimisation of rebalancing and charging so that vans move vehicles and swap batteries on the same route. Work on electric car sharing has similarly shown that ignoring the state of charge in relocation decisions leads to vehicles being moved to high-demand areas only to become unusable soon after. This literature supports treating battery level as a first-class variable in availability and rebalancing.

Streaming data platforms for connected vehicles have been discussed in both research and industry reports. Architectures based on Apache Kafka for ingestion, Apache Flink or Spark for stream processing, and time-series databases for storage are commonly described for fleet telemetry, because they handle high message rates and support windowed aggregation. Spatial indexing schemes such as geohash and the H3 hexagonal grid have been adopted to aggregate positions efficiently, and hexagonal zones have the advantage that all neighbours are at the same distance. Reports also stress data quality problems specific to telemetry, such as GPS drift, duplicate messages, and delayed reports from vehicles with weak connectivity. The proposed system builds on these practices and combines them with forecasting and optimisation.

III. SYSTEM ANALYSIS

The system is analysed as a continuous loop in which vehicle and trip data drive forecasts, forecasts drive rebalancing decisions, and completed tasks change the fleet state that is observed next. Each vehicle sends telemetry every thirty seconds with its identifier, latitude, longitude, battery percentage, lock state, speed, and fault codes. The booking platform sends trip start and end events, and the rider app records searches that end without a rental. External feeds provide rainfall and temperature forecasts, public holidays, and a calendar of large local events. The streaming tier validates messages, removes duplicates and GPS jumps, assigns each position to an H3 zone, and maintains rolling counts of total, usable, and low-battery vehicles in each zone together with recent trip starts and failed searches. Every fifteen minutes the forecasting service predicts demand for the next two hours, the battery model predicts which vehicles will become unrentable, and the optimiser produces move and swap tasks for the available vans, which are sent to crews and tracked until completion.

Existing System

In the existing mode of operation, most shared-mobility operators rely on fixed routines and supervisor experience to manage the fleet. Service vans follow predefined morning and evening routes, collecting vehicles from locations where they usually accumulate and leaving them at known busy spots such as metro stations. Battery swaps are carried out in rounds, often in a fixed order of zones, and vehicles below a certain charge are simply hidden from the rider app until a crew reaches them. The operations dashboard, where one exists, shows vehicle locations on a map and total trip counts but does not forecast demand or distinguish usable from unusable supply in each area. Failed searches by riders are not analysed, so unmet demand remains invisible. When rain, a holiday, or a concert changes travel patterns, routes are not adjusted in time. Crews therefore sometimes move vehicles into zones where they are not needed, while low-battery vehicles pile up in busy zones and riders there find nothing available.

Disadvantages of Existing System

- Rebalancing follows fixed routes that ignore changes in daily demand.
- Battery swaps are done in rounds rather than where charge is most needed.
- Zone supply counts include vehicles too low on charge to be rented.
- Failed rider searches are not recorded or analysed as unmet demand.
- No forecast of trip demand is available for the coming hours.
- Weather, holidays, and local events are handled only by improvisation.
- Vehicles remain idle for long periods in low-demand zones.
- Van capacity and crew shift time are not used efficiently.
- Supervisors cannot measure utilisation by zone or time of day.
- Telemetry errors such as GPS jumps distort the fleet map.

Proposed System

The proposed system, Move Grid, provides real-time fleet intelligence for shared electric bicycles and scooters by combining streaming data engineering with forecasting and optimisation. Vehicle telemetry, trip events, and failed searches are published to Apache Kafka and processed by Apache Flink, which cleans the data,

assigns positions to H3 hexagonal zones, and maintains live counts of usable and low-battery vehicles and of open demand in each zone. These aggregates are stored in TimescaleDB and joined with weather and event data. A LightGBM model forecasts trip starts per zone for the next two hours, and a battery depletion model predicts when each vehicle will drop below the rentable threshold. A mixed-integer rebalancing optimiser built with Google OR-Tools uses these predictions to decide how many vehicles to move between zones and which batteries to swap first, subject to van capacity, crew shift time, and city parking rules. The resulting tasks are sent to a crew mobile app, and an operations console shows supply gaps, utilisation, battery risk, and task progress.

Advantages of Proposed System

- Maintains a live view of usable vehicles in every zone of the city.
- Treats battery charge as part of availability rather than a separate concern.
- Forecasts trip demand per zone for the next two hours.
- Uses failed rider searches to estimate demand that trip data misses.
- Plans vehicle moves and battery swaps together on the same van route.
- Respects van capacity, crew shift time, and city parking restrictions.
- Adapts rebalancing to rain, holidays, and local events.
- Reduces idle time of charged vehicles in low-demand zones.
- Filters GPS jumps and duplicate messages before they reach operators.
- Gives supervisors utilisation and unmet demand figures by zone and hour.

IV. METHODOLOGY

The methodology begins with building the streaming pipeline, in which telemetry, trip, and search events are published to separate Kafka topics with defined schemas. A Flink job removes duplicate messages, discards position jumps that imply speeds above forty kilometres per hour, and maps each valid position to an H3 zone at resolution eight. Tumbling five-minute windows compute counts of vehicles, usable vehicles above twenty percent charge, trip starts, and failed searches per zone, which are written to TimescaleDB. For demand forecasting, eight weeks of history are aggregated into thirty-minute intervals, and features include lagged demand at thirty minutes, one day, and one week, hour, day type, rainfall, temperature, event proximity, and neighbouring zone demand. A LightGBM model is trained with time-based cross-validation. The battery model uses gradient boosted regression on trip distance, slope, and temperature to estimate energy per trip. Every fifteen minutes, the OR-Tools solver maximises expected served trips minus relocation cost over a two-hour rolling horizon, and the resulting tasks are dispatched to crews and evaluated by simulation replay.

SYSTEM ARCHITECTURE

The architecture of Move Grid is arranged as a closed loop between the fleet and the operations team. The Fleet and Rider Layer contains the connected vehicles with GPS and battery management units, the booking platform, and the rider app. The Streaming Ingestion Layer consists of Kafka topics for telemetry, trips, and searches along with a schema registry. The Stream Processing Layer runs Flink jobs for validation, H3 zone mapping, and windowed aggregation of supply and demand. The

Storage Layer holds the TimescaleDB zone state tables, the trip history store, and external weather and event data. The Intelligence Layer includes the LightGBM demand forecaster, the battery depletion model, and the OR-Tools rebalancing optimiser, supported by a model registry and a daily retraining job. The Operations Layer provides the operations console with a live zone map and utilisation charts, and the crew mobile app that receives move and swap tasks. Task completions and new telemetry return to the Streaming Ingestion Layer, closing the loop.

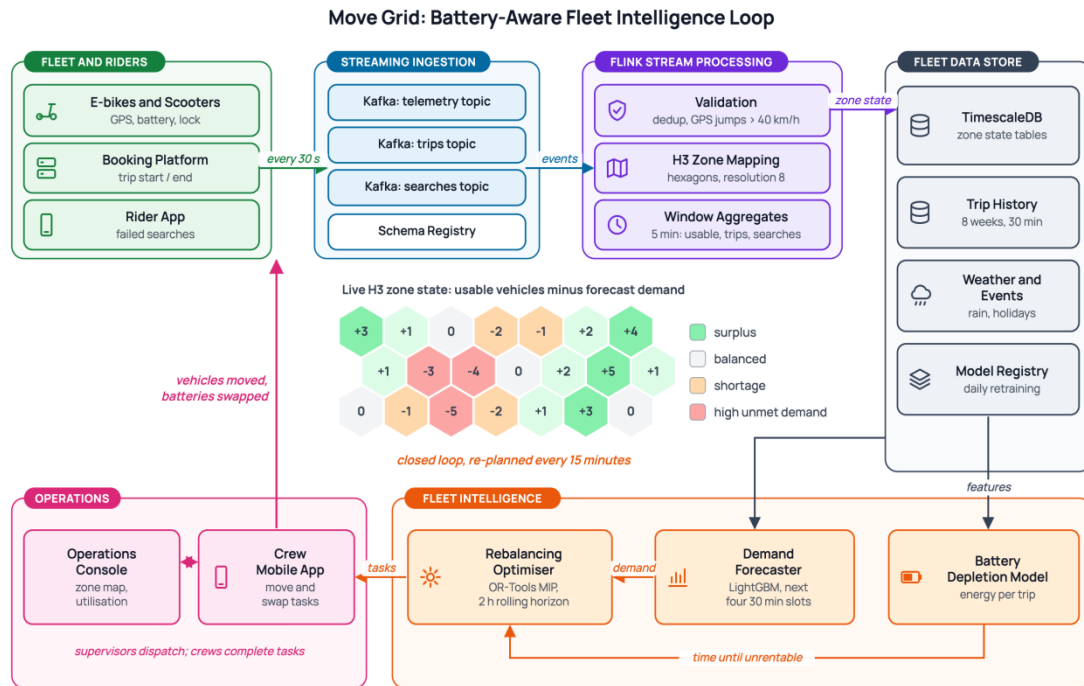


Fig. 1 System Architecture

V. RESULT AND OUTPUT

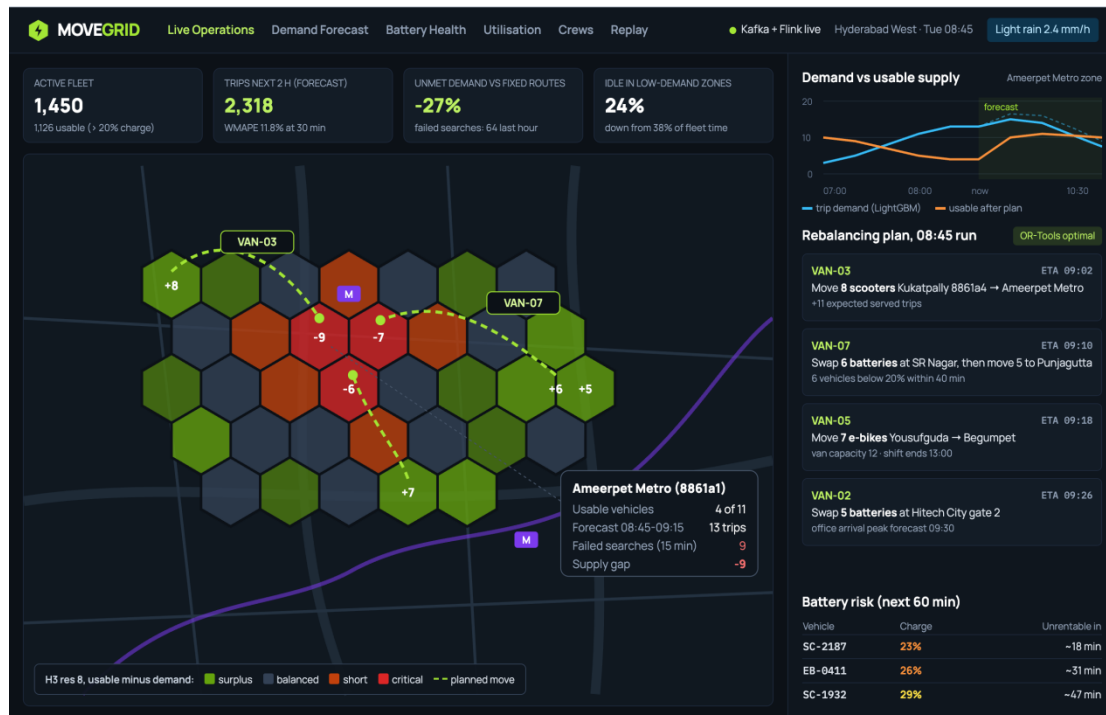


Fig. 2 Result and Output Screen

VI. CONCLUSION

This paper presented Move Grid, a real-time fleet intelligence system for shared electric bicycles and scooters. The system streams vehicle telemetry, trip events, and failed searches through Kafka and Flink, maintains a live picture of usable supply and open demand in each H3 zone, forecasts demand for the next two hours, predicts battery depletion, and plans rebalancing and battery swaps with an optimisation model. In this way it replaces fixed routes and experience-based rounds with decisions based on current and forecast conditions.

The evaluation using eight weeks of trip data from a fleet of about 1,450 vehicles showed that the LightGBM demand model reached a weighted mean absolute percentage error of 11.8 percent at the thirty-minute horizon, with rainfall, lagged demand, and event proximity among the most important features. When its forecasts were used by the optimiser in a replay simulation, unmet demand fell by 27 percent compared with the operator's fixed routes, and the share of fleet time that charged vehicles spent idle in low-demand zones fell from 38 to 24 percent with the same number of vans.

The results also showed why battery awareness matters. When rebalancing was planned using total vehicle counts instead of usable counts, the optimiser moved many vehicles that became unrentable within the next hour, and most of the gain in served trips disappeared. Including failed searches in the demand signal was similarly important in zones near metro stations, where recorded trips were limited by supply and therefore underestimated true demand. Cleaning telemetry at the stream processing stage prevented GPS jumps from creating false supply on the map.

Future work will add rider incentives that offer discounts for ending trips in under-supplied zones, integrate metro and bus timetables to anticipate arrival peaks, and extend the optimiser to mixed fleets with different battery capacities. Reinforcement learning for dynamic rebalancing and a field trial with live dispatch, rather than simulation replay, are also planned to confirm the gains in daily operation.

REFERENCES

- [1] Kumar, R. D., Prudhviraaj, G., Vijay, K., Kumar, P. S., & Plugmann, P. (2024). Exploring COVID-19 through intensive investigation with supervised machine learning algorithm. In Handbook of Artificial Intelligence and Wearables (pp. 145-158). CRC Press.
- [2] Swathi, B., Vijay, K., Sushanth Babu, M., & Dinesh Kumar, R. (2024, November). Machine Learning Techniques in Cloud Based Intrusion Detection. In The International Conference on Artificial Intelligence and Smart Environment (pp. 557-564). Cham: Springer Nature Switzerland.
- [3] Sv satyakrishna, shirisha rang, bhargavi nalacheruve. (2024) Prospective investigation on colorectal cancer with SMOTE on machine learning Algorithm
- [4] Dr.G.Vishnu Murthy, BhargaviNalacheruve 1Professor, Department of computer Science & engineering, Anurag University, TS, India. 2Student, Department of computer Science & engineering, Anurag University, TS, India.
- [5] V. N. S. Manaswini, K. K, C. Nigam, S. S. Ali, R. Niranjana, and Suman, "Real-Time Object Detection in Drone Surveillance Using YOLOv5," in Proc. 2025 3rd Int. Conf. IoT, Communication and Automation Technology (ICICAT), Gorakhpur, India, 2025, pp. 1–6, doi: 10.1109/ICICAT68430.2025.11414670.
- [6] B. Soundarya, V. N. S. Manaswini, M. Ayyakrishnan, R. D. Kumar, "Contextual Analysis of Big Data Analytics in Intelligent Transportation Frameworks," in Intersection of Artificial Intelligence, Data Science, and Cutting-Edge Technologies: From Concepts to Applications in Smart Environment, Lecture Notes in Networks and Systems, vol. 1353, Cham: Springer, 2025, doi: 10.1007/978-3-031-88304-0_79.
- [7] R. D. Kumar, V. N. S. Manaswini, "Applications of blockchain in smart cities: detecting fake documents from land records using blockchain technology," in Blockchain for Smart Cities, Elsevier, 2021, pp. 105–117, doi: 10.1016/B978-0-12-824446-3.00017-X.
- [8] Tejavath Veeramma, Badarla Anil, Guguloth Ravinder, "An advanced movie recommender using collaborative filtering and sentiment analysis," International Research Journal of Modernization in Engineering Technology and Science, vol. 7, no. 7, July 2025, doi: 10.56726/IRJMETS81618.
- [9] Ravi Kumar Banoth, Ramana Murthy B V, "Automatic crop recommendation system using LightGBM and decision tree machine learning models," Journal of Machine and Computing, vol. 5, no. 1, pp. 343, Jan. 2025, doi: 10.53759/7669/jmc202505026.
- [10] Ravi Kumar Banoth, Dr. B.V. Ramana Murthy, "Smart agriculture through IoT and machine learning for analyzing carbon footprints," in Proc. Int. Conf. Computer Science and Communication Engineering (ICCSCE), Apr. 2025.
- [11] Ravi Kumar Banoth, B. V. Ramana Murthy, "Soil image classification using transfer learning approach: MobileNetV2 with CNN," SN Computer Science, vol. 5, art. no. 199, 2024, doi: 10.1007/s42979-023-02500-x.