International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 4 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

BUILDING A CITY-SCALE MOBILITY DATA PIPELINE FOR PASSENGER CROWDING, VEHICLE MOVEMENT AND PUBLIC TRANSPORT DEMAND INTELLIGENCE - (TRANSITSENSE)

[1] Mrs. Ch. Roja, [2] Gorle Teja, [3] Guglavath Upender, [4] U Nandishwar, [5] Malavath Nuthan Kumar
[1]Assistant Professor, [2345]Students
Department of CSE (Data Science)
Siddhartha Institute of Technology & Sciences, Narapally
sits.rojachilukuri@siddhartha.co.in, 24tq1a6719@siddhartha.co.in, 25tq5a6701@siddhartha.co.in, 24tq1a6745@siddhartha.co.in, 24tq1a6729@siddhartha.co.in

ABSTRACT

Public transport agencies in large Indian cities operate thousands of buses and metro trains every day, yet most of them still plan services using fixed timetables and occasional manual passenger surveys. The data needed for better decisions already exists in scattered systems: GPS units on vehicles report positions every few seconds, electronic ticketing machines and smart card validators record every boarding, and automatic passenger counters on some vehicles log entries and exits at each stop. This paper presents TransitSense, a city-scale mobility data pipeline that brings these sources together to measure passenger crowding, track vehicle movement and forecast public transport demand in near real time.

The pipeline ingests vehicle location feeds, ticketing transactions, passenger counter records and the published GTFS schedule through a message broker, and processes them in both streaming and batch modes. Streaming jobs match each GPS ping to its route and trip, compute delays against the schedule and estimate the current load of every vehicle by combining boardings from ticket data with alightings inferred from typical travel patterns. Batch jobs build a cleaned and partitioned data lake organised in raw, refined and curated layers, which holds stop-level and hour-level histories of ridership, occupancy and running times.

On top of the curated layer, TransitSense trains machine learning models for three tasks. A gradient boosting model forecasts boardings at each stop for the next one to three hours using calendar, weather and recent demand features. A crowding classifier assigns every upcoming trip to a comfort level of low, moderate, crowded or overloaded. A travel time model predicts arrival times at downstream stops from live positions and historical segment speeds. Origin and destination matrices are also estimated from smart card tap sequences to show how passengers actually move across the network.

I. INTRODUCTION

Urban public transport carries a large share of daily trips in Indian cities, and its quality directly affects congestion, air pollution and the cost of living for working families. Buses in particular are expected to serve rapidly growing suburbs, industrial

areas and educational hubs, but their frequency and routing are often revised only once in several years. As a result, some routes run nearly empty during most of the day while others are so crowded at peak hours that passengers are left behind at stops. Overcrowding discourages regular users, pushes them towards two-wheelers and cabs, and weakens the financial position of the transport agency.

The information required to tackle this problem is generated continuously but is rarely used together. Vehicle tracking systems report GPS positions for monitoring and safety, electronic ticketing machines record fares for revenue accounting, and a few vehicles carry infrared or camera based passenger counters. Each of these systems is supplied by a different vendor, stores data in its own format and is examined by a different department. Planners who want to know how crowded a route is at eight in the morning must request extracts from several teams, reconcile stop names and trip codes by hand, and often settle for a manual count taken on a single day.

Data engineering offers a practical way to close this gap. A well-designed pipeline can collect the raw feeds continuously, align them to a common model of routes, stops and trips, repair gaps and errors, and store the result in a form that analysts and machine learning models can query quickly. Once the data is unified, it becomes possible to estimate how many passengers are on each vehicle, to see where and when crowding occurs, to measure delays segment by segment and to forecast demand for the next few hours with reasonable accuracy.

II. LITERATURE SURVEY

Automatic vehicle location systems have been used in public transport for several decades, first for fleet monitoring and later as a source of data for performance analysis. Research on these data sets has concentrated on measuring schedule adherence, headway regularity and running time variability along corridors. Studies have shown that raw GPS pings are noisy, particularly in dense urban areas where tall buildings reflect satellite signals, and that careful map matching to the route geometry is needed before reliable stop arrival times can be derived. Several works describe methods that project each ping onto the route shape, detect stop events from dwell periods and interpolate arrival times when pings are sparse. A common conclusion is that location data alone is sufficient to measure how vehicles move, but it says nothing about how many passengers are carried, and it must be combined with ticketing or counting data to understand the service from the passenger's point of view.

Smart card and electronic ticketing data have become a major source for studying passenger behaviour. Much of the research deals with estimating origin and destination matrices, because many systems record only the boarding tap and not the alighting location. Trip chaining methods assume that a passenger's next boarding stop is close to the previous alighting stop and that the last trip of the day returns near the first boarding location, and validation studies report that these assumptions hold for a large majority of regular commuters. Other studies use card histories to cluster passengers into groups such as daily commuters, students and occasional riders. Researchers point out that cash tickets, transfers without taps and missing stop codes reduce coverage, so estimated matrices are usually scaled using counts from manual surveys or automatic passenger counters before they are used in planning.

Short-term passenger demand forecasting has been approached with both statistical and machine learning methods. Seasonal time-series models capture the strong daily and weekly patterns of ridership but struggle with holidays, festivals, rain and special events. Tree-based ensembles such as random forests and gradient boosting machines, trained with lagged demand, calendar and weather features, are frequently reported to give lower errors at the stop and route level. Deep learning approaches using recurrent networks and graph convolutional networks model spatial dependence between neighbouring stations and have shown good results on metro systems with complete entry and exit data. Several comparisons note, however, that deep models need large and clean histories, and that for bus networks with noisier data a well-engineered gradient boosting model often performs as well while being far easier to train, explain and maintain.

Crowding estimation has received increasing attention because passengers care as much about comfort as about waiting time. Studies have used passenger counters, weight sensors in vehicle suspensions, Wi-Fi and Bluetooth probe detection and onboard camera analytics to estimate occupancy. Where such hardware is limited, researchers have inferred load profiles by combining ticket boardings with probabilistic alighting models derived from historical origin and destination patterns. Work on crowding information systems shows that when passengers are told in advance that a vehicle will be crowded, a noticeable share of them shift to a later departure or an alternative route, which spreads the load more evenly. The literature also stresses that crowding should be reported as simple categories rather than exact numbers, since estimates carry uncertainty and passengers respond better to clear comfort levels.

Research on transit data platforms and big data architectures describes how agencies can manage the growing volume of feeds. Proposed designs use message brokers to collect streams, distributed processing engines for cleaning and aggregation, and data lakes organised in raw, refined and curated zones so that original records are preserved while analysts work on trusted tables. The adoption of the General Transit Feed Specification and its real-time extension has made it easier to share schedules and live positions in a standard format. Reviews of such platforms note that most published work either focuses on a single data source or stops at descriptive dashboards, and that few systems close the loop by feeding unified data into forecasting models whose outputs are then used for crowding alerts and service adjustments. TransitSense is designed to address this gap within one open-source pipeline.

III. SYSTEM ANALYSIS

The system is analysed as a data pipeline that turns several independent operational feeds into a single, trustworthy view of the transport network. Vehicle GPS units report position, speed and heading every ten to thirty seconds, electronic ticketing machines and smart card validators record each boarding with a time, stop and fare type, and passenger counters on a subset of buses log entries and exits at every door. The static GTFS feed defines routes, stops, trips and scheduled times, and serves as the reference model to which every other record is aligned. Weather data and a city event calendar are added because rain, festivals and examinations change travel behaviour. The analysis identified three main requirements: real-time processing for

vehicle position, delay and load, historical storage for planning and model training, and a common key structure of route, trip, stop and time bucket that allows the feeds to be joined without manual effort. Data quality checks are required at every stage, since duplicate pings, missing stop codes and clock drift on ticketing devices are frequent.

Existing System

In most transport agencies the existing arrangement treats each data source as a separate system owned by a separate department. The vehicle tracking platform is used by the control room to see where buses are and to respond to breakdowns, but its history is rarely analysed. Ticketing data is collected at the end of each shift and used mainly to reconcile cash and card revenue by conductor and depot. Passenger counters, where installed, produce files that are downloaded occasionally for pilot studies. Service planning is based on the published timetable, periodic manual counts at selected stops and complaints received from the public. Because stop names, route numbers and trip codes differ between systems, combining them requires spreadsheets and manual matching, which can take weeks for a single corridor. Crowding is known only through anecdotal reports, demand is estimated from last year's figures, and passengers receive vehicle arrival information at best, with no indication of how full the approaching bus is.

Disadvantages of Existing System

- Vehicle location, ticketing and passenger counter data are stored in separate systems.
- Stop names and trip codes differ between vendors and must be matched by hand.
- Crowding on vehicles is known only through complaints and occasional manual counts.
- Service frequency is revised rarely and is based on outdated survey figures.
- Historical GPS data is discarded or never analysed for running time trends.
- There is no short-term forecast of demand at the stop or route level.
- Passengers cannot see how crowded the next bus will be before boarding.
- Data quality problems such as duplicate pings and clock drift go undetected.
- Special events and rainfall effects on ridership are not measured systematically.
- Producing a single corridor analysis takes weeks of manual data preparation.

Proposed System

The proposed system, TransitSense, is a unified mobility data pipeline that collects vehicle, ticketing, passenger counter and schedule data continuously and converts it into crowding, movement and demand intelligence. Live feeds are published to a message broker, where streaming jobs clean each record, map-match GPS pings to routes and stops, attach ticket boardings to the correct trip and update an estimated load for every running vehicle. The same records are written to a data lake with raw, refined and curated layers, and scheduled batch jobs build stop-hour ridership tables, segment running times and origin and destination matrices from smart card chains. A feature store supplies calendar, weather, lagged demand and event features to three models: a LightGBM demand forecaster, a crowding level classifier and a segment-

based arrival time predictor. Forecasts and live estimates are exposed through an API to an operator dashboard with network maps, hotspot charts and capacity recommendations, and to a passenger service that shows the expected crowding of upcoming trips. Data quality rules and pipeline monitoring run at every stage.

Advantages of Proposed System

- Combines GPS, ticketing, passenger counter and GTFS data under one common data model.
- Estimates the passenger load of every running vehicle in near real time.
- Forecasts stop-level boardings for the next one to three hours.
- Classifies upcoming trips into clear crowding levels for passengers and operators.
- Predicts arrival times using live positions and historical segment speeds.
- Builds origin and destination matrices automatically from smart card tap chains.
- Keeps raw, refined and curated data layers so that history can be reprocessed.
- Flags data quality issues such as duplicate pings and missing stop codes early.
- Supports evidence-based decisions on adding, removing or shifting trips.
- Uses open-source tools, keeping costs low for public transport agencies.

IV. METHODOLOGY

The methodology begins with ingestion, where GPS pings, ticket transactions and passenger counter events are published to separate Kafka topics and the GTFS feed is loaded daily into reference tables. In the streaming stage, Spark Structured Streaming removes duplicates, discards pings outside the city boundary, projects each ping onto its route shape and detects stop arrivals and departures. Ticket boardings are joined to the trip whose vehicle was at that stop within a two-minute window, and alightings are distributed using historical destination probabilities to update the load of each vehicle. Batch jobs then aggregate the refined data into fifteen-minute stop and segment tables in Parquet format and derive origin and destination pairs through trip chaining. Features such as hour, day type, holiday flag, rainfall, lagged boardings and recent delay are stored in a feature store. LightGBM models are trained with time-ordered cross validation, the crowding classifier maps predicted load to capacity ratios, and all models are retrained weekly and monitored for drift before results reach the dashboard.

SYSTEM ARCHITECTURE

The architecture of TransitSense follows a left-to-right flow from data sources to users. The Source Layer contains vehicle GPS units, electronic ticketing machines and smart card validators, automatic passenger counters, the GTFS schedule feed, and the weather and event calendar services. The Ingestion Layer uses Kafka topics and a schedule loader to receive these feeds reliably. The Stream Processing Layer runs map matching, trip assignment and live load estimation, and writes vehicle states to a fast key-value store. The Storage Layer is a data lake with raw, refined and curated zones in Parquet, supported by a PostGIS database for stops, routes and segments. The Batch Analytics Layer produces ridership aggregates, segment running times and origin and destination matrices. The Machine Learning Layer holds the feature store,

the demand forecaster, the crowding classifier and the arrival time model. The Serving Layer exposes results through a REST API, and the Presentation Layer consists of the operator dashboard, the passenger crowding view and alert notifications for control room staff.

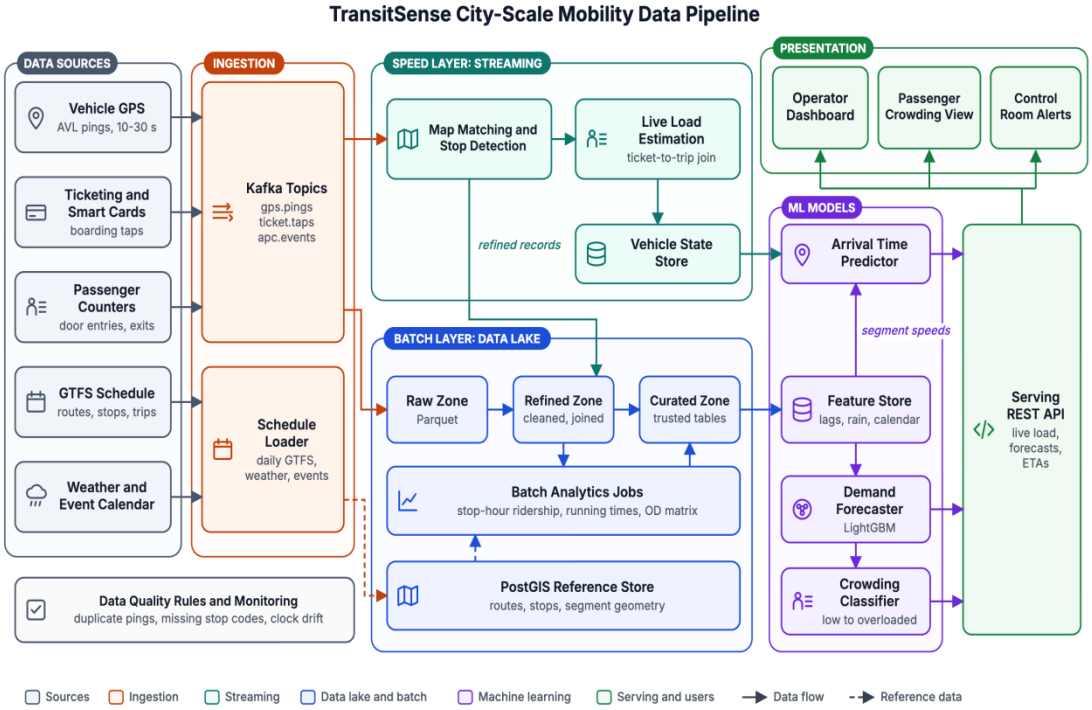


Fig. 1 System Architecture

V. RESULT AND OUTPUT

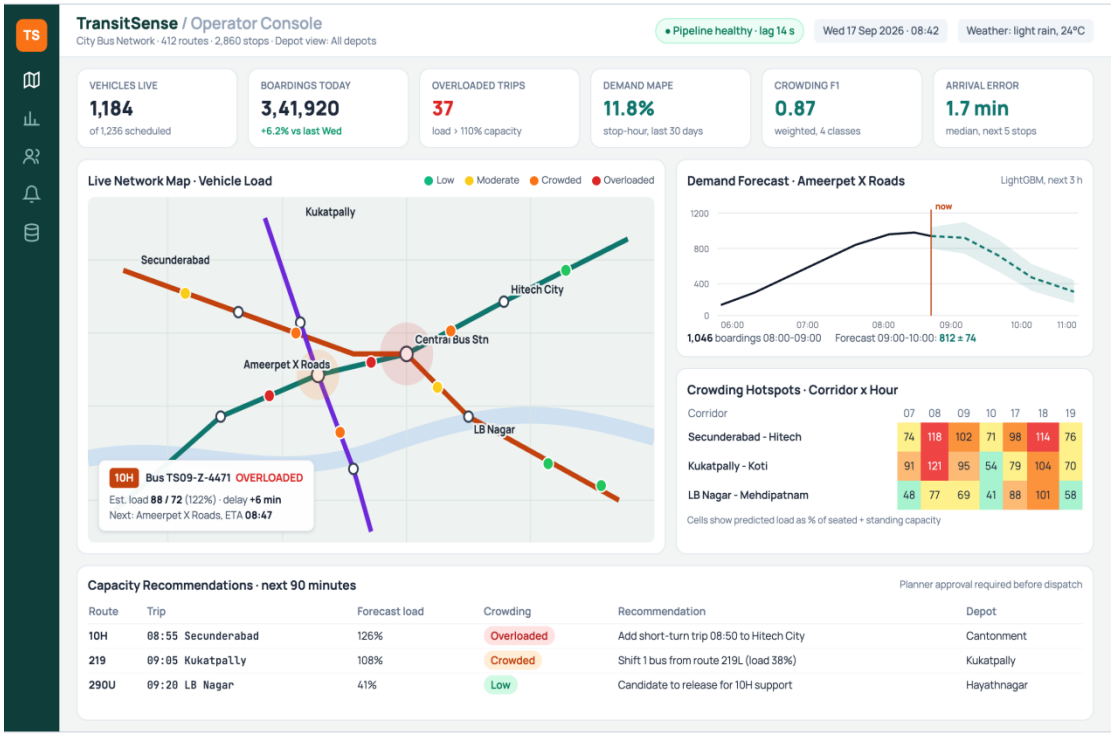


Fig. 2 Result and Output Screen

VI. CONCLUSION

This paper presented TransitSense, a city-scale mobility data pipeline that unifies vehicle location, ticketing, passenger counter and schedule data to provide crowding, movement and demand intelligence for public transport. By aligning all feeds to a common model of routes, trips, stops and time buckets, the pipeline removes the manual reconciliation that has kept these data sources apart and makes it possible to answer planning questions in minutes instead of weeks.

The evaluation on four months of bus network data showed that the unified data supports useful predictive models. The LightGBM demand forecaster achieved a mean absolute percentage error of about 11.8 percent at the stop-hour level, the crowding classifier reached a weighted F1-score of 0.87, and the arrival time model kept the median error under two minutes. The largest gains in accuracy came from careful data engineering, particularly map matching, trip assignment of ticket boardings and the use of lagged demand and rainfall features.

The layered data lake proved valuable beyond the models themselves. Keeping raw records allowed the team to reprocess months of history when a stop code mapping error was discovered, and the curated tables gave planners a reliable base for corridor studies and origin and destination analysis. Data quality checks at the ingestion and refinement stages surfaced device clock errors and duplicate pings that had previously gone unnoticed in the separate vendor systems.

Future work will extend the pipeline to metro and suburban rail feeds so that crowding can be followed across modes, and will explore anonymised mobile network data to fill gaps where ticketing coverage is low. Graph-based models that capture the dependence between neighbouring stops, and an optimisation module that

converts demand forecasts into revised peak-hour timetables, are also planned. A field trial measuring how passengers respond to crowding information would help quantify the practical benefit of the system.

REFERENCES

[1] Kumar, R. D., Prudhviraj, 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 rangu, 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.