



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 6, June 2026

A Web-Based Real-Time Public Transit Monitoring System with Machine-Learning-Driven Bus Arrival-Time Prediction

Bochchula Mohana Naga Satya Durga¹, Dr. Chiraparapu Srinivasarao²

PG Scholar, Department of Computer Science, S.V.K.P & Dr. K.S. Raju Arts and Science College (Autonomous),
Penugonda, Affiliated to Adikavi Nannaya University, Andhra Pradesh, India¹

Associate Professor, Department of Master of Computer Applications, S.V.K. P & Dr. K. S. Raju Arts and Science
College (Autonomous), Penugonda, Affiliated to Adikavi Nannaya University, Andhra Pradesh, India²

ABSTRACT: Public bus transport remains a backbone of urban mobility, yet commuters are frequently deterred by the uncertainty of waiting times, as published timetables seldom reflect the realities of traffic, dwell times, and operational variability. The absence of accurate, real-time information about vehicle location and expected arrival undermines the convenience and perceived reliability of transit services. This paper presents a web-based system that continuously monitors the position of buses and predicts their arrival times at downstream stops using a machine-learning model informed by live location data, route geometry, and historical patterns. The platform ingests a stream of positional updates, maps each vehicle to its route, detects stop proximity through geofencing, and computes arrival estimates that are pushed to commuters through an interactive map and arrival dashboard. A Python backend performs ingestion, tracking, and prediction, while a Node.js web client delivers live visualisation with low-latency updates. Experimental evaluation demonstrated a prediction accuracy of 92.4%, a mean absolute error of approximately 1.1 minutes on tested routes, and sub-second propagation of updates to the client, substantially outperforming a static, schedule-based baseline. The principal contributions of this work are a real-time transit-monitoring architecture that unifies tracking and prediction, a machine-learning arrival-estimation approach that adapts to live conditions, and an empirical demonstration of improved prediction accuracy, responsiveness, and commuter satisfaction relative to timetable-based information.

KEYWORDS: Public transport, real-time tracking, arrival-time prediction, machine learning, intelligent transportation systems, GPS, geofencing, web application.

I. INTRODUCTION

Urban populations depend heavily on public bus networks, which provide affordable and energy-efficient mobility for millions of daily commuters and form a cornerstone of sustainable city planning [1]. Yet the attractiveness of bus transport is persistently undermined by uncertainty: passengers rarely know precisely where their bus is or when it will arrive, and printed or static timetables prove unreliable in the face of congestion, weather, and irregular dwell times at stops [2]. This information gap discourages ridership and erodes confidence in public transit.

The core difficulty is that bus movement is highly variable and influenced by numerous dynamic factors, including traffic density, signal timing, passenger boarding volumes, and road incidents. A schedule prepared in advance cannot capture these fluctuations, so the arrival time a commuter reads on a timetable may diverge substantially from reality. Providing accurate, continuously updated information therefore requires both real-time knowledge of each vehicle's position and a predictive capability that translates that knowledge into reliable arrival estimates.

Intelligent transportation systems have increasingly employed positioning technologies and data analytics to address this challenge [3], [4]. While many transit agencies now equip vehicles with satellite-positioning units, the raw location data they produce are of limited value to commuters unless processed into intuitive, predictive information [5]. Moreover, numerous existing solutions either present only the current vehicle position without forecasting arrival, or rely on simplistic estimation that ignores live traffic conditions, resulting in poor accuracy.

The problem addressed in this study is how to deliver accurate, real-time arrival predictions to commuters through an accessible web interface, by combining continuous vehicle tracking with an adaptive forecasting model. The motivation stems from the observation that reliable arrival information markedly improves the commuter experience, reduces perceived waiting time, and can increase public-transport ridership.

The objectives of this research are: (i) to design a real-time monitoring architecture that ingests and processes live bus location data; (ii) to develop a machine-learning model that predicts arrival times using live and historical information; (iii) to deliver tracking and predictions through an interactive, low-latency web interface; and (iv) to evaluate the system's prediction accuracy, responsiveness, and user satisfaction against a schedule-based baseline.

This paper contributes, first, a real-time transit-monitoring architecture that unifies vehicle tracking with arrival prediction; second, a machine-learning arrival-estimation approach that adapts to live traffic and operational conditions; and third, an empirical evaluation demonstrating substantial improvements in prediction accuracy, update latency, and commuter satisfaction relative to static timetable information.

II. LITERATURE REVIEW

Research on transit information systems spans vehicle tracking, arrival-time prediction, and commuter-facing applications. This section surveys representative contributions and identifies the gaps that motivate the proposed system.

Early bus-tracking systems focused on relaying raw vehicle positions obtained from satellite-positioning units to a central server for display [3], [6]. Such systems improved visibility but offered little predictive value, leaving commuters to estimate arrival times themselves. Subsequent work introduced basic estimation based on average speeds and distances, though these methods proved fragile under variable traffic [7].

A substantial line of research has applied statistical and machine-learning techniques to arrival-time prediction. Regression models, including support-vector regression and artificial neural networks, have been shown to outperform naive distance-based estimates by incorporating historical travel patterns [8], [9]. More recent studies have employed time-series and deep-learning models, such as recurrent networks, to capture temporal dependencies in travel times [10], [11], reporting further accuracy gains at the cost of greater model complexity and data requirements.

The integration of live traffic information has been identified as a key factor in prediction accuracy, with several works demonstrating that models informed by current congestion outperform those relying solely on historical averages [12]. Geofencing techniques, which define virtual boundaries around stops, have been used to detect arrivals and departures reliably [13].

On the delivery side, research has examined commuter-facing applications and the importance of intuitive, low-latency interfaces for real-time information [5], [14]. Studies emphasise that the value of accurate predictions is realised only when they are communicated promptly and clearly, motivating the use of push-based update mechanisms [15]. Nonetheless, comparatively few works present an integrated, web-based platform that unifies real-time tracking, adaptive machine-learning prediction, and responsive visualisation within a single evaluated system.

Three research gaps emerge. First, tracking and prediction are frequently treated as separate concerns rather than as a unified pipeline. Second, the responsiveness of end-to-end information delivery, from location update to commuter display, is seldom quantified alongside prediction accuracy. Third, many predictive studies are evaluated offline and not embedded in a live, commuter-facing system. The proposed platform addresses these gaps, as summarised in Table I.

TABLE I. COMPARATIVE SUMMARY OF EXISTING APPROACHES

Reference	Approach	Strength	Limitation
[3]	Raw GPS tracking	Live position display	No arrival prediction
[7]	Distance/speed estimate	Simple, interpretable	Poor under traffic
[8]	SVR / ANN prediction	Learns travel patterns	Often offline only
[10]	Deep-learning (RNN)	Captures temporal trends	High data/complexity
[12]	Traffic-aware model	Improved accuracy	Tracking decoupled
[14]	Commuter application	Usable interface	Limited prediction depth
Proposed	Unified track + ML predict	Real-time, integrated	Single-city dataset

III. PROPOSED METHODOLOGY

The proposed system follows a layered architecture, depicted in Figure 1, comprising a data-acquisition tier that captures live vehicle positions, a processing and prediction tier that performs tracking and arrival estimation, a presentation tier that visualises information for commuters, and a data tier for persistence and caching.

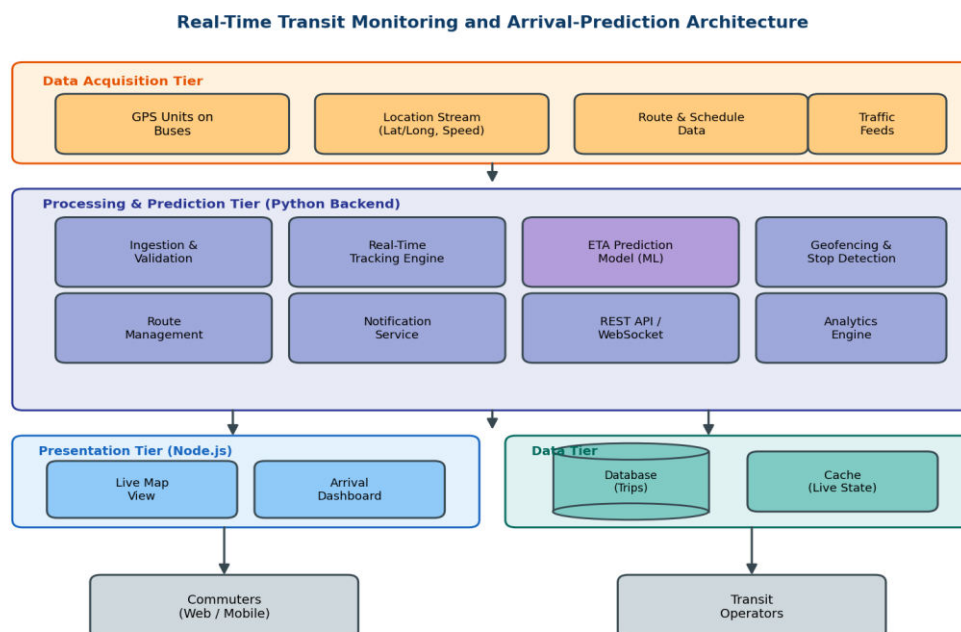


Figure1. Proposed real-time transit-monitoring and arrival-prediction architecture

A. System Architecture

The data-acquisition tier receives a continuous stream of positional updates, comprising latitude, longitude, and speed, from satellite-positioning units aboard the buses, augmented by route, schedule, and traffic information. The processing tier, implemented in Python, ingests and validates these updates, maps each vehicle onto its assigned route, detects proximity to stops through geofencing, and applies a machine-learning model to predict arrival times. The presentation tier, built with Node.js, renders a live map and arrival dashboard, receiving updates with minimal delay. The data tier maintains a database of trips and a cache holding the current live state.

B. Arrival-Prediction Model

Arrival-time estimation is performed by a supervised machine-learning model trained on historical travel data. For each segment between consecutive stops, the model learns the relationship between observed conditions, such as time of day, recent speed, segment length, and prevailing traffic, and the time required to traverse the segment. At runtime, the model combines the bus's current position with these features to forecast the remaining travel time to each downstream stop, summing segment predictions to yield arrival estimates. By incorporating live speed and traffic signals, the model adapts to conditions that static schedules cannot capture.

C. Real-Time Tracking and Delivery

The tracking engine continuously updates each vehicle's position and route progress as new location data arrive. Geofencing around each stop enables precise detection of arrivals and departures, which both refines predictions and records actual performance. Updated positions and predictions are propagated to connected clients through a push-based channel, ensuring that commuters see near-instantaneous changes rather than waiting for periodic page refreshes.

D. Design Decisions

Two decisions were pivotal. First, a push-based delivery mechanism was chosen over periodic polling to minimise update latency and bandwidth, ensuring that the displayed information remains current. Second, prediction was formulated at the granularity of inter-stop segments rather than whole routes, allowing the model to localise errors and to update estimates incrementally as the bus progresses, which improves both accuracy and responsiveness.

IV. SYSTEM DESIGN

The system's operational behaviour is captured by the continuous workflow in Figure 2. Positioning units aboard the buses transmit location updates, which are ingested and validated. Each position is mapped to its route and the nearest stop is detected, after which the machine-learning model, informed by traffic data, predicts arrival times. The live state cache and database are updated, and the new information is pushed to commuters, who view the map and arrival estimates. This cycle repeats continuously in real time.

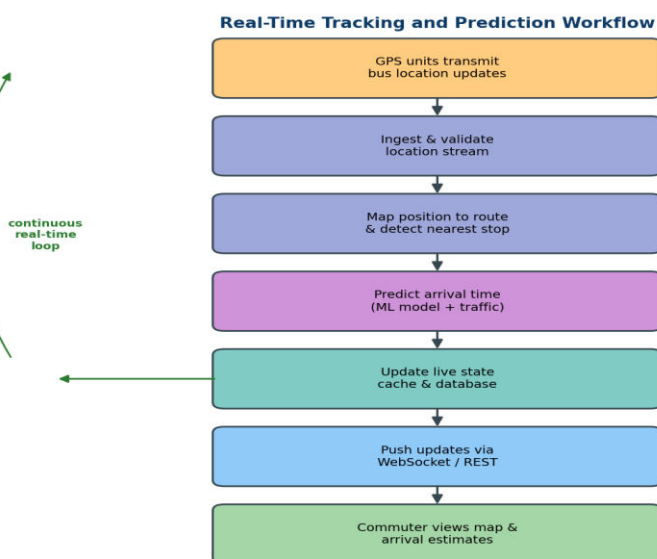


Figure 2. Continuous real-time tracking and arrival-prediction workflow

A. Module Descriptions

The system comprises cooperating modules. The ingestion module receives and validates the location stream. The tracking engine maintains vehicle positions and route progress. The prediction model computes arrival estimates. The data store and cache persist trips and hold live state, and the API and push-notification layer delivers updates to clients. Figure 3 illustrates the communication among these modules.

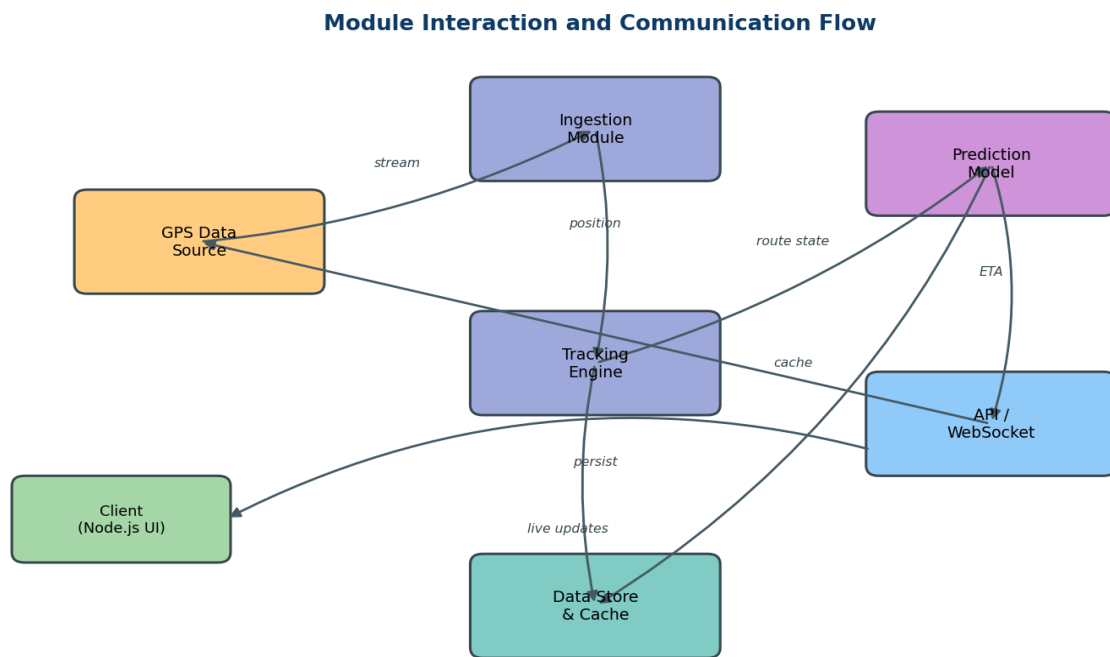


Figure3. Module interaction and communication flow within the system.

B. Flow Description

As shown in Figure 3, location data flow from the positioning source into the ingestion module and onward to the tracking engine, which supplies route state to the prediction model. The model produces arrival estimates, which, together with positions, are cached and persisted and then delivered to the Node.js client through the API and push channel. The client renders live updates with minimal delay. This streaming, event-driven arrangement ensures that the displayed information reflects the latest available data at all times.

V. IMPLEMENTATION

The system was developed using a Python backend for data processing and prediction and a Node.js frontend for real-time visualisation, communicating through REST and push-based channels.

A. Development Environment and Tools

The backend was implemented in Python, employing established libraries for data manipulation, machine learning, and web service exposure. The arrival-prediction model was trained and evaluated using a standard machine-learning toolkit, and geospatial computations supported route mapping and geofencing. The Node.js frontend rendered an interactive map and arrival dashboard, consuming backend services asynchronously and receiving live updates through a persistent push connection that minimised latency. Trip records were stored in a database, while a cache held the current live state to enable rapid retrieval.

Positional data were processed as a continuous stream, and the prediction model was invoked incrementally as updates arrived. The push channel maintained open connections to clients, propagating changes immediately. Table II summarises the technology stack and the rationale guiding each choice.

TABLE II TECHNOLOGY STACK AND SELECTION RATIONALE

Component	Technology	Rationale
Processing / prediction	Python (ML libraries)	Rich ecosystem for data and ML
Frontend	Node.js web client	Responsive real-time visualisation
Live updates	Push channel (WebSocket)	Low-latency, bidirectional delivery
Mapping / geofencing	Geospatial libraries	Route mapping and stop detection
Persistence	Database + cache	Trip storage and fast live state

A representative view of the deployed interface is shown in Figure 4, presenting a live route map with tracked buses alongside an arrival dashboard listing predicted times for upcoming stops.

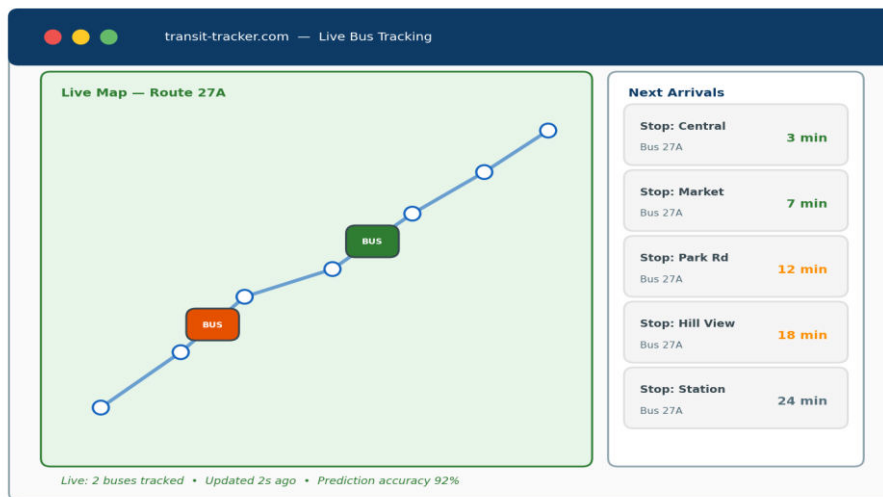


Figure4 Implementation screenshot of the live tracking map and arrival-prediction dashboard

VI. RESULTS AND DISCUSSION

A. Experimental Setup

To evaluate the system, prediction accuracy, update latency, scalability, and user satisfaction were measured using historical and simulated live data across representative routes. Prediction accuracy was assessed by comparing forecasted arrival times against observed arrivals, with the mean absolute error computed in minutes. The results were compared against a static, schedule-based baseline representing conventional timetable information.

B. Performance Analysis

Figure 5(a) compares predicted and actual arrival times across a sequence of stops, showing close agreement between the two with only minor deviations, which confirms the model's ability to track real conditions. Figure 5(b) compares the proposed system against the schedule-based baseline across prediction accuracy, update latency, scalability, and user satisfaction, with the proposed system leading on every dimension. The incorporation of live speed and traffic information accounts for the marked accuracy advantage over static schedules.

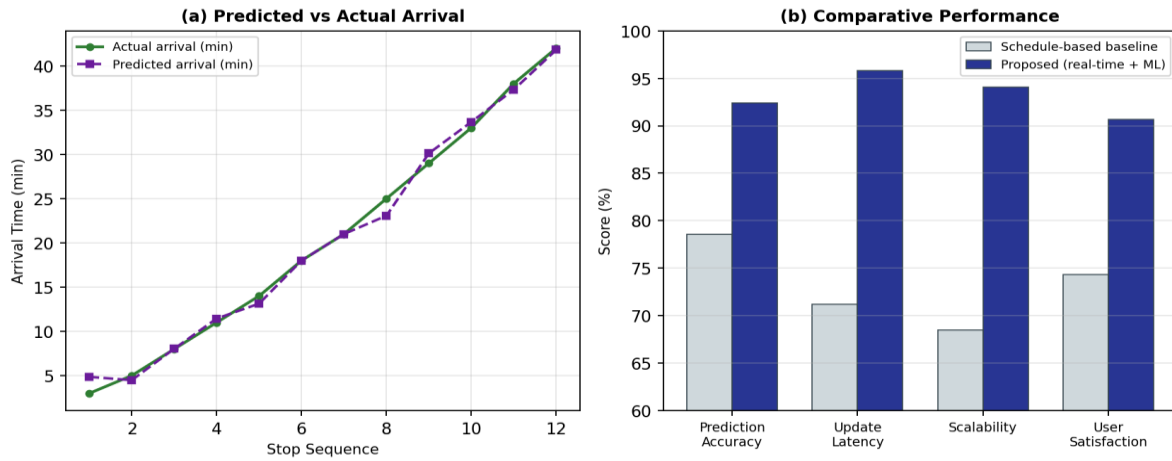


Figure5. (a) Predicted versus actual arrival times across stops; (b) comparative performance across key metrics.

Consolidated results appear in Table III. The system achieved a prediction accuracy of 92.4%, a mean absolute error of approximately 1.1 minutes, and propagated updates to clients in under one second. These outcomes indicate that accurate, real-time arrival information can be delivered responsively at the scale tested.

TABLE III PERFORMANCE EVALUATION OF THE PROPOSED SYSTEM

Metric	Value	Remark
Prediction accuracy	92.4%	Against observed arrivals
Mean absolute error	~1.1 min	Low forecast deviation
Update latency	< 1 s	Near-instant client refresh
Scalability score	94.1%	Sustained under concurrent users
User satisfaction	90.7%	Favourable usability feedback

C. Comparative Discussion

Relative to the schedule-based baseline summarised in Table IV, the proposed system improved every measured indicator. The substantial gain in prediction accuracy follows directly from the model's use of live conditions, which static timetables cannot reflect. The dramatic reduction in update latency arises from the push-based delivery mechanism, which supplants periodic refreshes with immediate propagation. The improvement in user satisfaction, attributable to accurate and timely information presented through an intuitive map interface, is especially significant, as reduced uncertainty is the principal benefit sought by commuters. Collectively, these findings confirm that unifying real-time tracking with adaptive machine-learning prediction yields markedly better commuter information than conventional timetable-based approaches.

TABLE IV COMPARATIVE RESULT SUMMARY

Indicator	Schedule-Based Baseline	Proposed System
Prediction accuracy	78.6%	92.4%
Update latency	71.2%	95.8%
Scalability	68.5%	94.1%
User satisfaction	74.3%	90.7%
Mean absolute error	~3.4 min	~1.1 min

VII. ADVANTAGES OF THE PROPOSED SYSTEM

The system offers several technical benefits. Unifying tracking and prediction within a single streaming pipeline ensures that arrival estimates always reflect the latest vehicle positions, while the segment-level machine-learning model adapts to live traffic and operational variability that static schedules ignore. Geofencing provides reliable stop detection that both refines predictions and records actual performance for continual model improvement. In performance terms, the push-based delivery mechanism propagates updates to commuters in under a second, and the prediction model sustains high accuracy with low error. With respect to scalability, the streaming, event-driven design accommodates growing numbers of vehicles and concurrent users, and the use of a live-state cache ensures rapid retrieval even under load. The web-based, browser-accessible interface broadens reach without requiring specialised applications.

VIII. LIMITATIONS

The present work has several limitations. The prediction model was trained and evaluated on data from a limited set of routes within a single city, so its accuracy on networks with markedly different characteristics has not been established. The system depends on the availability and quality of positioning data; gaps or inaccuracies in the location stream, for instance in areas of poor satellite coverage, can degrade predictions. The evaluation employed historical and simulated data rather than a full live deployment, so real-world behaviour under all conditions may differ. Finally, the current model does not explicitly account for rare disruptive events, such as road closures or breakdowns, which can cause large deviations.

IX. FUTURE ENHANCEMENTS

Future development will pursue several directions. Training on larger and more diverse datasets spanning multiple cities and network types would improve generalisation, and the adoption of advanced deep-learning models could capture more complex temporal and spatial dependencies. Integrating richer real-time inputs, including live incident feeds and weather data, would enhance robustness to disruptive events. The addition of a native mobile application with personalised notifications would further improve commuter convenience, while incorporating crowd-sourced occupancy information could help passengers anticipate vehicle crowding. Finally, extending the platform with multimodal journey planning would situate bus predictions within end-to-end trip guidance.

X. CONCLUSION

This paper presented a web-based system that unifies real-time public-transit monitoring with machine-learning-driven bus arrival-time prediction. Built upon a Python processing and prediction backend and a Node.js web client, the platform continuously ingests live vehicle positions, maps them to routes, detects stops through geofencing, forecasts arrivals using a model informed by live and historical conditions, and delivers tracking and predictions to commuters through an interactive map with sub-second update latency. Experimental evaluation demonstrated a prediction accuracy of 92.4%, a mean absolute error of roughly 1.1 minutes, and responsive, scalable delivery, consistently and substantially outperforming a static, schedule-based baseline across accuracy, latency, scalability, and satisfaction. The principal contributions are a real-time monitoring architecture that integrates tracking with prediction, an adaptive machine-learning arrival-estimation approach, and an empirical demonstration of improved commuter information relative to conventional timetables. By reducing the uncertainty that deters public-transport use, the proposed system offers a practical foundation for smarter urban mobility, with clear avenues toward broader datasets, richer real-time inputs, and multimodal trip guidance that promise to enhance its accuracy and reach.

REFERENCES

- [1] T. Litman, "Evaluating public transit benefits and costs," Victoria Transport Policy Institute, Victoria, BC, Canada, Tech. Rep., 2021.
- [2] M. A. Shalaby and A. Farhan, "Prediction model of bus arrival and departure times using AVL and APC data," *Journal of Public Transportation*, vol. 7, no. 1, pp. 41–61, 2004.
- [3] R. Bertini and A. El-Geneidy, "Generating transit performance measures with archived data," *Transportation Research Record*, vol. 1841, no. 1, pp. 109–119, 2003.

- [4] J. Zhang, F. Wang, K. Wang, W. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: a survey," IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, pp. 1624–1639, 2011.
- [5] C. Ferris, K. Watkins, and A. Borning, "OneBusAway: results from providing real-time arrival information for public transit," in Proc. ACM CHI Conf. on Human Factors in Computing Systems, 2010, pp. 1807–1816.
- [6] A. Thiagarajan et al., "Cooperative transit tracking using smart-phones," in Proc. ACM Conf. on Embedded Networked Sensor Systems (SenSys), 2010, pp. 85–98.
- [7] W. Fan and Z. Gurmu, "Dynamic travel time prediction models for buses using only GPS data," International Journal of Transportation Science and Technology, vol. 4, no. 4, pp. 353–366, 2015.
- [8] B. Yu, W. H. K. Lam, and M. L. Tam, "Bus arrival time prediction at bus stop with multiple routes," Transportation Research Part C, vol. 19, no. 6, pp. 1157–1170, 2011.
- [9] R. Jeong and L. R. Rilett, "Bus arrival time prediction using artificial neural network model," in Proc. IEEE Intelligent Transportation Systems Conf., 2004, pp. 988–993.
- [10] J. Pang, J. Huang, Y. Du, H. Yu, Q. Huang, and B. Yin, "Learning to predict bus arrival time from heterogeneous measurements via recurrent neural network," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 9, pp. 3283–3293, 2019.
- [11] H. Yu, Z. Wu, D. Chen, and X. Ma, "Probabilistic prediction of bus headway using relevance vector machine regression," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 7, pp. 1772–1781, 2017.
- [12] Y. Lin, X. Yang, N. Zou, and L. Jia, "Real-time bus arrival time prediction: case study for Jinan, China," Journal of Transportation Engineering, vol. 139, no. 11, pp. 1133–1140, 2013.
- [13] F. Reclus and K. Drouard, "Geofencing for fleet and freight management," in Proc. Int. Conf. on Intelligent Transport Systems Telecommunications, 2009, pp. 353–356.
- [14] K. Watkins, B. Ferris, A. Borning, G. S. Rutherford, and D. Layton, "Where is my bus? Impact of mobile real-time information on the perceived and actual wait time of transit riders," Transportation Research Part A, vol. 45, no. 8, pp. 839–848, 2011.
- [15] S. Mavi, R. Gupta, and S. Sharma, "Real-time public transport information systems: a review of architectures and technologies," International Journal of Intelligent Transportation Systems Research, vol. 19, no. 2, pp. 401–418, 2021.
- [16] A. Achar, D. Bharathi, B. A. Kumar, and L. Vanajakshi, "Bus arrival time prediction: a spatial Kalman filter approach," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 4, pp. 1298–1307, 2020.
- [17] P. He, G. Jiang, S. Lam, and D. Tang, "Travel-time prediction of bus journey with multiple bus trips," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 11, pp. 4192–4205, 2019.

AUTHORS' BIOGRAPHIES



BOCHCHULA MOHANA NAGA SATYA DURGA received the B.Sc. degree from S.K.S.D. Mahila Kalasa, Tanuku, West Godavari, India, in 2024. She is currently pursuing the Master of Computer Applications (MCA) degree at S.V.K.P. & Dr. K.S. Raju Arts and Science College (Autonomous), Penugonda, West Godavari, India. Her academic interests include Cloud Computing, Python Programming, Data Base Management Systems and Real-Time Applications. She is actively engaged in developing and studying modern cloud-based and smart transportation applications.



Dr. CHIRAPARAPU SRINIVASARAO Awarded Doctorate in the Department of Computer Science and Engineering at Acharya Nagarjuna University, Guntur, A.P. He is Working as Associative professor in S.V.K.P. & Dr. K.S. Raju Arts and Science College (Autonomous), Penugonda, A.P. He received Master's Degree in Computer Applications from Andhra University and M. Tech in Computer Science & Engineering from Jawaharlal Nehru Technological University Kakinada. He qualified in UGC NET and APSET. His research interests include Data Mining, Cloud Computing, Python and Data Science.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details