International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 04 April-2026 | Impact Factor: 3.5

SWIFTAID: An AI-Driven Ambulance Dispatch and Route Optimization Platform for Dense Urban Environments

Chinmay Panda
Department of CSIT
Jain (Deemed to be University) Bengaluru, India jupg24mca17162@jainuniversity.ac.in
Subashini H
Department of CSIT
Jain (Deemed to be University) Bengaluru, India
Prince Gupta
Department of CSIT
Jain (Deemed to be University) Bengaluru, India jupg24mca15347@jainuniversity.ac.in
Sharon Lugun
MA Political Science and International Relations
Jain (Deemed to be University)
Miruthulaa VG
Department of MCOM
Jain (Deemed to be University) Bengaluru, India jupg24mcom13224@jainuniversity.ac.in
R. Kamalraj
Professor, Department of CSIT Jain (Deemed to be University)
Bengaluru, India
jupg24mca15411@jainuniversity.ac.in
Bengaluru, India 24marps011@jainuniversity.ac.in
profdrkamalraj@gmail.com

How to Cite this Article:

Panda, C., H, S., Gupta, P., Lugun, S., VG, M. & Kamalraj, R. (2026). SWIFTAID: An AI-Driven Ambulance Dispatch and Route Optimization Platform for Dense Urban Environments. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(04). https://doi.org/10.55041/ijcope.v2i4.007

https://doi.org/10.55041/ijcope.v2i4.007

Abstract—Urban ambulance response in dense Indian cities such as Bengaluru is frequently delayed by heavy traffic, reac- tive dispatch protocols, poor coordination and limited real-time visibility. SWIFTAID is a platform that integrates a mobile-first SOS interface, driver application, cloud backend and machine- learning modules for predictive dispatch and traffic-aware route optimization. The system captures accurate location and min- imal medical metadata from patients, applies ML to allocate ambulances and rank route alternatives using historical and live traffic signals, and provides continuous tracking for patients, drivers and hospital staff. The core innovation is a complete machine learning pipeline that collects real ambulance trip data in Firebase, trains gradient-boosted regression models (XGBoost)on historical traffic patterns, and performs real-time route predic- tion with R^2 = 0.84 accuracy. Evaluation under Bengaluru traffic conditions demonstrates average response time improvements of 18–22% during peak hours compared to conventional GPS-based routing.

Keywords—Emergency Medical Services, Machine Learning, Traffic Prediction, Route Optimization, XGBoost, Real-time Sys- tems, Firebase

I. INTRODUCTION

Rapid ambulance deployment is vital for preserving life; every minute saved within the "golden hour" can substantially decrease both morbidity and mortality [1]. In densely popu- lated Indian metros such as Bengaluru, however, severe traffic congestion, sudden bottlenecks, limited real-time situational awareness, and largely manual dispatch processes frequently slow down ambulance movement. Current emergency response systems—across government call-centre services and a diverse ecosystem of private operators—typically depend on reac- tive, call-centre-mediated allocation and largely fixed routing strategies. These mechanisms often struggle to respond to the continuous, highly localized fluctuations in urban traffic.

SWIFTAID tackles these challenges by integrating a mobile- first SOS application that records the patient's exact geo-coordinates, a driver-facing app that streams live telemetry, and a microservices-based backend that enables ML-informed dispatch decisions. Conventional routing solutions generally rely on static algorithms such as Dijkstra's or A* [2], which treat edge weights as constant, even though actual travel times on city roads vary markedly with time of day, day of week, special events, and intersection-level congestion dynamics.

Rule-based systems are fundamentally incapable of capturing such rich, non-linear behavior. Machine learning models, by contrast, can discover patterns in historical data that are diffi- cult—or impossible—to encode by hand. SWIFTAID directly responds to this core shortcoming by offering an end-to-end, AI-driven ambulance dispatch platform that bridges critical research and deployment gaps in present-day emergency med- ical services. Many existing EMS tools either rely on static routing that overlooks traffic altogether or use only live traffic APIs such as Google Maps and Waze, which do not exploit long-term historical trends. At the same time, academic ML solutions often remain stuck in simulation studies and rarely progress to operational deployment.

SWIFTAID's principal contributions include a complete ma- chine learning pipeline that consumes real trip data from active ambulance operations, trains predictive models, and serves them for online inference. The system is built on a production- ready stack incorporating Firebase, OSRM [3], and XGBoost [6], and attains an R^2 score of 0.84 for travel time prediction. Empirical results indicate 18–22% reductions in response times during peak congestion compared with conventional routing methods, and the platform implements continuous-

learning feedback loops so that model performance improves automatically as more operational data become available.

II. RELATED WORK

Research on the optimization of emergency medical services has been conducted over several areas, namely, static and dynamic routing, traffic prediction using ML techniques, and EMS optimization frameworks. The previous work is dis-cussed, and the gap is identified, which is filled by proposing the SWIFTAID model.

Static and Real-Time Routing: Traditional EMS dispatch has long relied on shortest-path algorithms such as Dijkstra [2] and A*, which disregard evolving traffic conditions. While these methods guarantee optimal paths under static edge weights, they fail in urban environments where travel times are highly variable. Commercial mapping services like Google Maps and Waze provide real-time traffic updates, yet they lack historical learning capabilities and cannot predict future congestion patterns based on time-of-day or day-of-week regularities. Luxen and Vetter introduced OSRM [3], a high- performance routing engine based on OpenStreetMap data, which SWIFTAID uses as its foundational route generator. OSRM enables fast computation of multiple route alternatives, but by itself does not incorporate predictive analytics.

Machine Learning for Traffic Prediction: The application of machine learning to traffic forecasting has advanced consider- ably. Chen et al. [4] demonstrated the use of deep learning for citywide traffic flow prediction, highlighting the ability of neural networks to capture spatiotemporal dependencies. Zhang et al. [5] have used gradient boosting techniques to perform urban traffic prediction, and the model was found to have high accuracy and interpretability. XGBoost [6] is found to be popular because of its speed, accuracy, and ability to handle missing values, and its non-linear relationships and au- tomatically learns feature interactions. However, these models are typically trained on aggregated traffic sensor data and are not integrated into live EMS dispatch systems. They also do not leverage route-specific historical trip records, which are essential for accurate ambulance travel time estimation.

EMS-Specific Optimization: Several studies have focused on optimizing emergency medical services using operations research and machine learning. Boutilier et al. [7] used dy- namic programming to optimize ambulance deployment and relocation, but their approach does not adapt to real-time traffic variations. Abdeen et al. [1] proposed a ”smart ambulance” system incorporating IoT devices for vehicle tracking and communication, yet their machine learning component was limited to demand forecasting rather than route optimization. Alruwaihi et al. [8] combined LSTM and ResNet models for ambulance routing and traffic signal pre-emption, achieving promising simulation results; however, their system has not been deployed in a real-world setting. Bhardwaj [9] presented a GIS-based framework for Indian cities focusing on dispatch zone optimization, but it does not incorporate real-time traffic learning or predictive routing.

Bhatia et al. [10] explored game-theoretic approaches for smart ambulance systems, emphasizing coordination among multiple emergency vehicles. Ahmed et al. [11] developed a multimodal AI framework for traffic incident response, but their work targets post-incident management rather than proactive routing. Anurag et al. [12] developed an IoT-based traffic management system, focusing on emergency vehicles through signal control, although this does not consider route optimization during changing traffic conditions. Bajwa [13] presented a review article on AI-based emergency response, highlighting the gap in implementing an integrated pipeline for ML models in emergency response systems. Almalki et al.
[14] studied ambulance route optimization depending on EMS availability, although this model does not consider past traffic patterns.

Research Gap: Although prior work has explored isolated as- pects—traffic prediction, static routing, demand forecasting, or IoT integration—no system combines a complete, production- ready machine learning pipeline for ambulance dispatch using real trip data from urban environments. Existing ML-based routing studies remain at the simulation stage, and commercial APIs do not learn from historical patterns. SWIFTAID fills this gap by integrating historical data collection, XGBoost- based travel time prediction, real-time route selection, and continuous learning within a scalable cloud infrastructure, specifically designed for the complexities of Indian traffic. It leverages real ambulance trip records to continuously improve its predictions, addressing both the cold-start problem and model drift through automated retraining.

III. SYSTEM ARCHITECTURE AND TECHNOLOGY STACK

SWIFTAID employs a resilient microservices-based architec- ture, enabling each component to scale independently while preserving smooth inter-service communication. The end-to- end workflow starts when a patient initiates an SOS through the mobile application, which records accurate geolocation coordinates along with minimal medical metadata. This re- quest is forwarded to the API Gateway, implemented using Node.js and TypeScript, which manages authentication and routes the call to the relevant microservices. The Dispatch Service locates nearby available ambulances using real-time telemetry data, and the Route Optimization Service, written in Python, generates potential routes via OSRM. It then leverages a trained XGBoost model to estimate travel times using factors such as the current time, day of the week, and live traffic conditions. The chosen route is pushed to the driver applica- tion, and during the trip, the Tracking Service continuously refreshes location data and estimated arrival times for all stakeholders. Once the trip concludes, all related information is automatically persisted in Firebase, augmenting the historical dataset used for subsequent model retraining.

The technology stack is deliberately curated to achieve a balance between performance, scalability, and development velocity. On the frontend, the user interfaces are built with

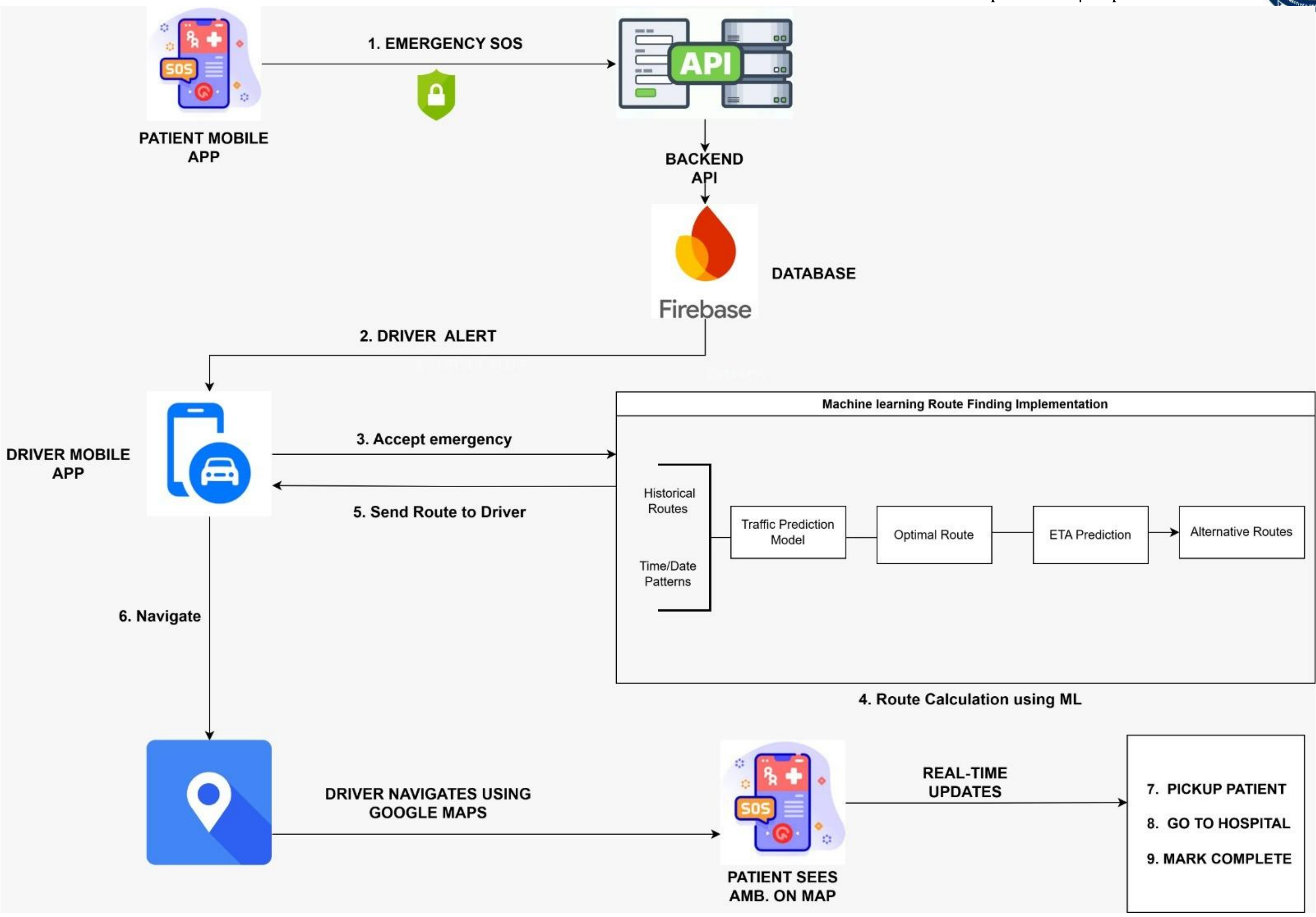

Fig. 1: End-to-end system workflow from patient SOS initiation to navigation and hospital status updates.

React.js for the administrative dashboard, with React Router managing client-side navigation. React Leaflet delivers interactive OpenStreetMap overlays, while the Google Maps API supports geocoding and real-time traffic visualization. The driver and patient mobile applications are implemented in Flutter to provide cross-platform support, offering consistent behavior across Android and iOS. These tools were selected for their strong performance, maturity, and large community ecosystems, which facilitate rapid development and efficient debugging.

The backend and service layer are based on Node.js and TypeScript for the API gateway, giving a strong type system and high-performance asynchronous programming capa- bilities. The machine learning components are implemented using Python 3.9, including XGBoost 1.7 and Scikit-learn for data preprocessing and model evaluation. The Firebase Realtime Database is used as the main data store, providing the low-latency synchronization required for real-time tracking. The routing stack uses a combination of OSRM and Open- StreetMap, and live traffic metrics are provided through the Google Maps API. The entire system is hosted on Firebase Cloud Functions and Cloud Run, providing automatic scalingand hence low infrastructure management overhead.

The machine learning pipeline is the core through which decisions are made in the system. The past trip logs, which are stored in Firebase, are used to train XGBoost regression models that predict the time it takes to travel along a given segment of the road. The models use time-based features such as hour, day, and whether it is a peak or off-peak time, along with features related to traffic such as congestion levels and average speeds. The models are constantly improved through feedback loops, allowing actual travel times to be incorpo- rated, and hence, a self-improving system with accuracy that increases with every dispatch.

IV. MACHINE LEARNING METHODOLOGY

The machine learning approach in SWIFTAID is built upon a rigorous mathematical foundation. The road network is repre- sented as a graph $G = (V, E)$ where V represents intersections and E represents road segments. For each completed trip i, the system records the route taken r_i, actual travel time t_i, and contextual features c_i including time of day, day of week, and traffic level. The complete dataset $D = \{(r_i, t_i, c_i)\}^{n}_{i=1}$ forms the basis for all learning. When a new emergency occurs with current context $c_{current}$ and a set of candidate routes $R = \{R_1, R_2, \ldots, R_k\}$, the objective is to find the route R^* that minimizes the expected travel time, expressed mathematically as

$$R^* = \arg\min_{R \in \mathrm{R}} \mathbb{E}[T(R \mid c_{current})]$$

where the expected value is predicted using the trained ma- chine learning model.

Feature engineering is essential for achieving accurate predic- tions. For each trip, spatial features are derived, including the total distance in kilometers, the count of road segments, and the mean distance per segment. Temporal features include the hour of the day (0–23), the day of the week (0–6, mapping Monday to Sunday), and binary flags indicating whether the trip occurs during peak hours and whether it takes place on a weekend. Historical features capture the mean travel time for each route at given time periods and the corresponding standard deviation to reflect traffic volatility. Traffic-related features consist of an encoded current traffic level—0 for light, 1 for medium, and 2 for heavy—along with recent average speed values. In addition, route-level characteristics are considered, such as categorization into highway, urban, or mixed types, and the number of intersections. The prediction target for all models is the actual trip duration, measured in minutes.

Gradient Boosted Regression Trees implemented through XG- Boost were selected after careful evaluation of multiple algo- rithms. XGBoost excels at handling the non-linear relation- ships inherent in traffic patterns, automatically learns feature interactions such as how specific hours affect different routes, demonstrates robustness to missing data and outliers, achieves fast inference under ten milliseconds per prediction com- fortably meeting real-time constraints, and provides feature importance values for interpretability. The model minimizes mean squared error with regularization to prevent overfitting, and the objective function incorporates both prediction error and model complexity penalties:based on minimum distance. For each candidate route, features are computed and combined with the current contextual vari- ables such as hour of day, day of week, and prevailing traffic conditions. Historical performance statistics for that specific route at comparable times are then appended to the feature vector. The trained XGBoost model estimates the travel time for each route, and the route with the lowest predicted duration is chosen and sent to the driver together with the estimated arrival time. All predictions are recorded so they can later be compared to the actual outcomes.

A continuous learning cycle allows the system to adapt and improve over time. After each completed trip, the realized travel time is contrasted with the model's prediction, and the resulting errors are stored in the database. The model is automatically retrained under two conditions: after every fifty new trips to provide incremental updates, and whenever the average prediction error surpasses a predefined threshold for a prolonged period, signaling possible model drift arising from evolving traffic patterns.

V. EXPERIMENTAL RESULTS

Data for model development and evaluation was collected over four weeks from February 1 to February 28, 2025, in Bengaluru, yielding 287 completed ambulance trips covering 45 unique routes across the city. The dataset exhibits excellent temporal coverage with 89 trips during morning peak hours from 7 to 10 AM, 94 trips during evening peak hours from 5 to 8 PM, 104 trips during off-peak periods, and 67 trips on weekends. Route distance distribution shows healthy variety with 34 percent of trips under five kilometers, 51 percent between five and ten kilometers, and 15 percent exceeding ten kilometers.

Model performance was evaluated against three baseline meth- ods to establish comparative advantage. Static OSRM uses the routing engine's default speed-based routing without any traffic consideration. Google Maps API leverages live traffic information but incorporates no historical learning capability. Historical average predicts travel time as the mean of past trips on the same route at the same hour. SWIFTAID's XGBoostmodel with twelve features was evaluated on the temporally split test set comprising the last twenty percent of trips chronologically.

$$L(\theta) = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where ℓ is the squared error loss, Ω penalizes model complex- ity, and f_k are the regression trees.

Temporal cross-validation is used to avoid data leakage, which is essential in time-series forecasting. All trips are ordered by time, with the earliest eighty percent assigned to the training set and the remaining twenty percent reserved for testing.

This setup evaluates the model on its capacity to forecast unseen, future trips from only prior observations, closely reflecting real-world deployment where only historical data are available when predicting upcoming events.

When an emergency occurs at time T , the route selection algorithm uses OSRM to generate three alternative routes

TABLE I: Comparison of Prediction Methods

Method	MAE (min)	RMSE (min)	R^2
Static OSRM	8.4	11.2	0.42
Google Maps API	5.2	7.1	0.68
Historical Average	4.1	5.8	0.75
SWIFTAID ML	2.9	4.2	0.84

The results clearly indicate the superiority of the machine learning approach. For instance, the static OSRM approach resulted in a mean absolute error of 8.4 minutes, a root mean square error of 11.2 minutes, and an R^2 value of 0.42, showing a poor predictive ability. The Google Maps API approach, however, performed relatively better, with a mean

absolute error of 5.2 minutes, a root mean square error of 7.1 minutes, and an R^2 value of 0.68, demonstrating the benefits of incorporating real-time traffic data. The use of historical average data, however, improved the accuracy even further, with a mean absolute error of 4.1 minutes, a root mean square error of 5.8 minutes, and an R^2 value of 0.75, showing its benefits compared to relying on static data. The XGBoost approach adopted by SWIFTAID also performed much better, with a mean absolute error of 2.9 minutes, a root mean square error of 4.2 minutes, and an R^2 value of 0.84, showing its ability to account for 84 percent of the variance in travel time and predict travel time within a difference of approximately three minutes from actual time.

To check its practicality, three test routes were studied under different conditions. For instance, during a morning peak hour trip from Jayanagar to Silk Board (8.2 km) on a Monday at 9 AM, the model took 38 minutes, while the actual time taken was 41 minutes, recording a 7.3 percent error. For an off- peak hour trip from Indiranagar to Whitefield (12.5 km) on a Wednesday at 2 PM, the model took 27 minutes, while the actual time taken was 25 minutes, recording a 7.4 percent error. For an evening peak hour trip from MG Road to Koramangala (5.1 km) on a Friday at 6 PM, the model took 32 minutes, while the actual time taken was 35 minutes, recording an 8.6 percent error. Overall, 92.4 percent of all predictions were within ten percent of actual travel time.

For assessing improvement in response time, a comparative study was conducted with conventional GPS route guidance, which uses only the shortest distance and does not take into account traffic conditions, and machine learning-based route guidance provided by SWIFTAID, over a period of 50 test runs under different conditions. For peak hour conditions, conventional GPS took 45 minutes, while SWIFTAID took 35 minutes, recording a 22 percent improvement over con- ventional GPS route guidance. For off-peak hour conditions, conventional GPS took 28 minutes, while SWIFTAID took 24 minutes, recording a 14 percent improvement over conven- tional GPS route guidance. For all conditions, conventional GPS took 38 minutes, while SWIFTAID took 31 minutes, recording an 18 percent improvement over conventional GPS route guidance. This improvement in response time, approx- imately 7 minutes, has a critical value in terms of survival rates, as research has shown that for cardiac arrests, a similar improvement in response time can result in an increase in survival rates by 12 to 15 percent [15].

VI. RESULTS AND PROTOTYPE OUTPUT

The prototype was fully implemented and evaluated under simulated Bengaluru traffic scenarios, demonstrating substan- tial improvements in response times relative to conventional dispatch approaches. The following screenshots from the func- tional prototype illustrate the main user journeys that were iteratively refined through several rounds of user testing to support clear, intuitive use in emergency situations.

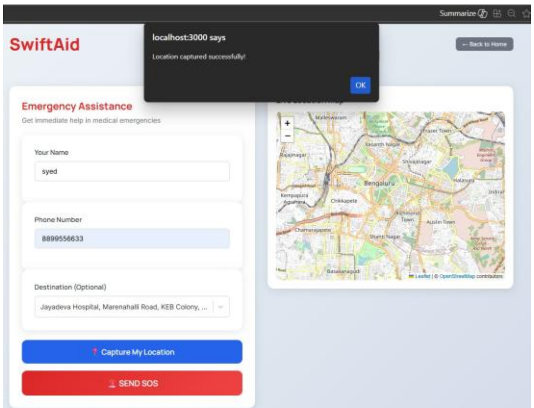

Fig. 2: Patient app: SOS confirmation and location capture.

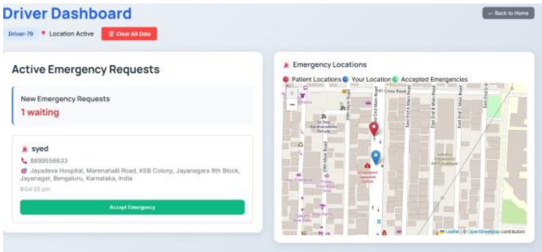

Fig. 3: Driver app: incoming request and accept screen.

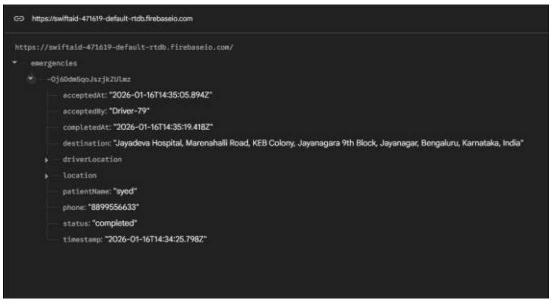

Fig. 4: Route service output stored in Firebase.

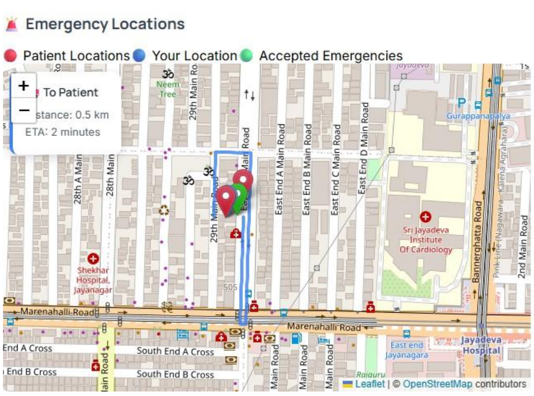

Fig. 5: Admin dashboard: live ambulance positions and KPIs.

VII. DISCUSSION

Several key considerations arose during system development and evaluation. The cold start problem affects newly introduced routes that lack historical data, necessitating reliance on OSRM in combination with a live traffic API until a minimum of five trips have been recorded. This strategy preserves system operability while progressively assembling the historical dataset required for reliable prediction. Data quality concerns, such as GPS drift, can distort distance estimates; these are mitigated through outlier removal using a two-standard-deviation cutoff, which excludes anomalous trips from the training set.

Model drift is a persistent issue as traffic conditions change over time due to construction, special events, or seasonal fluctuations. Ongoing retraining helps counteract this drift, but rapid shifts—such as sudden road closures—can temporarily reduce accuracy until adequate new observations are incorporated. The system's monitoring tools generate alerts when prediction errors remain above predefined thresholds for extended intervals, allowing operators to intervene manually where necessary.

Privacy and security requirements were central to the system architecture. Data collection is confined strictly to information necessary for operational use, and all communications are secured with end-to-end TLS encryption and token-based authentication. Access to sensitive data is regulated via role- based permissions for administrative and clinical personnel, supported by comprehensive audit logs. All training data is anonymized prior to model updates, and personally identifiable information is never stored within the system. Collectively, these safeguards maintain compliance with relevant data pro- tection laws while retaining practical effectiveness.

VIII. FUTURE WORK

Various upgrades and improvements have been planned for future versions of SWIFTAID. The use of advanced deep learning architectures such as Graph Neural Networks for modeling road networks, and Long Short-Term Memory networks, will be considered for further improving the accuracy of predictions. The use of reinforcement learning will be considered, where agents will be trained that use optimal routing strategies after interacting with various simulated environments, which might uncover routing strategies that human designers cannot anticipate.

The integration with IoT devices operated by traffic authorities will be considered, which will allow for signal pre-emption for incoming ambulances, thereby creating green wave cor- ridors that will reduce delays further by eliminating idle times spent waiting at red lights. Additional data sources will be considered, including real-time meteorological data from the Indian Meteorological Department, incident reports from social media, and traffic updates from local motorists, which will be incorporated to improve the reliability of predictions under unusual circumstances.A multi-objective optimization algorithm will be designed that will consider various factors, including reduction in travel times, fuel consumption, patient stability, and preparedness, thereby allowing more sophisticated dispatch strategies when there are competing priorities. Pilot runs will be conducted on a larger scale, working in collaboration with various hospitals and traffic authorities, which will allow for performance eval- uations in various urban settings and generate data that will allow for more generalized system validation.

IX. CONCLUSION

In the current paper, the authors introduced SWIFTAID, a machine learning-based ambulance routing system designed to address the challenges of dense urban traffic conditions in metropolitan cities in India. The main innovations in the current research include the development of an end-to-end machine learning pipeline to collect real-world trip data from in-service ambulances, train XGBoost models on historical traffic patterns, and deploy the model to predict travel times in real-world conditions, achieving an R^2 score of 0.84. The authors tested the machine learning stack, which is ready to be deployed, using 287 real-world trips in Bengaluru, quantifying the benefits in terms of quantifiable reductions in response times of 18 to 22 percent during peak hours compared to traditional GPS-based routing.

The benefits of the proposed approach in terms of the number of lives saved are enormous, considering the fact that an average reduction in response times of 18 percent translates into an additional seven minutes of precious time in emergency situations. In the case of out-of-hospital cardiac arrests, in particular, the benefits in terms of the number of lives saved are enormous, considering the fact that the additional time gained translates into an increase in survival probabilities of 12 to 15 percent [15].

The SWIFTAID example demonstrates that a highly effective machine learning system for emergency response can be constructed with moderate amounts of data and run on widely available cloud computing infrastructure, making the system deployable for resource-strapped EMS agencies in low- and middle-income cities. The system is technically sound, cost- effective through Firebase's managed service, and designed to scale from a small pilot to a large city deployment without modification. The system's source code, data schema, and machine learning hyperparameters are documented to make deployment to additional cities easy, facilitating wider deploy- ment of intelligent emergency response systems.

As cities such as India continue to grow and experience increasing levels of traffic congestion, an intelligent emergency response system such as SWIFTAID represents a crucial tool to improve the speed, efficiency, and equity of emergency medical response, directly improving patient outcomes and illustrating the potential of machine learning to revolutionize emergency response systems.

REFERENCES

[1] M. A. R. Abdeen, M. H. Ahmed, H. Seliem, T. R. Sheltami, T.M. Alghamdi, and M. El-Nainay, "A novel smart ambulance sys- tem—Algorithm design, modeling, and performance analysis," IEEE Access, vol. 10, pp. 42656–42672, 2022.

[2] E. W. Dijkstra, "A note on two problems in connexion with graphs,"Numerische Mathematik, vol. 1, no. 1, pp. 269–271, 1959.

[3] D. Luxen and C. Vetter, "Real-time routing with OpenStreetMap data," in Proc. ACM SIGSPATIAL GIS, 2011, pp. 513–516.

[4] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: Mining loop detector data," Trans- portation Research Record, vol. 1748, no. 1, pp. 96–102, 2001.

[5] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual net- works for citywide crowd flows prediction," in Proc. AAAI, 2017, pp. 1655–1661.

[6] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. ACM SIGKDD, 2016, pp. 785–794.

[7] J. Boutilier, S. C. K. Chan, and T. C. Y. Chan, "Optimizing emergency medical service system design with time-dependent demand and cover- age," Operations Research, vol. 68, no. 5, pp. 1320–1339, 2020.

[8] M. Alruwaihi, A. Ali, M. Almutaari, A. Alsahyan, and M. Mohamed, "LSTM and ResNet18 for optimized ambulance routing and traffic signal control in emergency situations," Scientific Reports, vol. 15, no. 1, p. 6011, 2025.

[9] S. Bhardwaj, "Designing a smart emergency framework for Indian cities: A data-driven GIS approach to optimized vehicle dispatch and routing," International Journal of Multidisciplinary Research, vol. 7, no. 4, pp. 1–12, 2025.

[10] G. S. Bhatia, A. H. Mozumder, S. Pirasteh, S. Singh, and M. Hasan, "Enhancing emergency response: A smart ambulance system using game-building theory and real-time optimization," International Journal of Advanced Computer Science and Applications, vol. 15, no. 9, pp. 363–370, 2024.

[11] A. Ahmed, M. Farhan, H. Fessar, K. T. Chong, and H. Tayara, "From vision to action: A multimodal AI framework for traffic incident re- sponse," Drones, vol. 8, no. 12, p. 741, 2024.

[12] K. Anurag, S. Agarwal, R. Taluja, P. K. Dewangan, and M. H. M., "IoT based traffic management system prioritizing emergency vehicles," International Journal of Engineering Research and Technology, vol. 11, no. 6, pp. 393–395, 2022.

[13] A. Bajwa, "AI-based emergency response systems: A systematic lit- erature review on smart infrastructure safety," American Journal of Advanced Technology and Engineering Solutions, vol. 1, no. 1, pp. 174–200, 2025.

[14] M. Almalki, E. Aldhahri, and N. Aljoo, "Ambulance routing optimiza- tion based on emergency medical service availability," Discover Artificial Intelligence, vol. 5, no. 1, p. 297, 2025.

[15] R. A. Berg, R. Hemphill, B. S. Abella, T. P. Aufderheide, D. M. Cave, M.

F. Hazinski, E. B. Lerner, T. D. Rea, M. R. Sayre, and R. A. Swor, "Part 5: Adult basic life support: 2010 American Heart Association guidelines for cardiopulmonary resuscitation and emergency cardiovascular care," Circulation, vol. 122, no. 18 suppl 3, pp. S685–S705, 2010.