

SafeGo: An Intelligent and Inclusive Safety-Centric Ride-Hailing Platform with Machine Learning-Based Risk Prediction

Laya K Shajan¹, Madhan Senthilkumar², Harshitha Reddy Muramreddy³, Pallavi Jakkula⁴, and Sai Inapakolla⁵

^{1,2,3,4,5}Parul University, Vadodara, Gujarat, India

Under the Guidance of: Ms. Nidhi Shah, Professor, Parul University

Abstract—Ride-hailing apps are built mainly to book a ride, match a driver and reach the destination quickly. Safety and accessibility are usually added afterwards as a separate SOS button or a support-ticket system, so women, elderly riders and persons with disabilities (PWD) are often poorly served. This paper presents SafeGo, a ride-hailing platform that brings passenger-specific ride modes, machine-learning-based safety-risk prediction, accessibility-aware driver matching, PIN-based ride verification, live monitoring, SOS emergency response, transparent fare calculation and an administrative command center into one system. SafeGo provides four ride modes (Normal, Pink, Elderly and PWD), and a Random Forest classifier uses temporal, geographical, trip-level and mode-related features to place each trip in the Stable, Cautious or High Priority category. On a held-out synthetic test set of 2,000 samples the classifier achieved 99.80% accuracy, 99.87% macro precision, 99.78% macro recall and 99.82% macro F1-score, with a mean prediction time of 32.27 ms. Scenario-based tests show that predictions change with the ride mode and trip context. Because the dataset is synthetic, these figures describe model behaviour on the evaluation data and are not a claim about real-world accuracy.

Index Terms—SafeGo, Ride-Hailing, Passenger Safety, Random Forest, Risk Prediction, PWD Accessibility, Driver Matching, PIN Verification, SOS Emergency Response, Intelligent Transportation.

I. INTRODUCTION

App-based ride-hailing has changed how people move around cities. With a few taps a passenger can request a ride, see the driver approach, get a fare estimate and follow the trip on a map [3], [7]. What most platforms have not built well is a safety and accessibility framework around this convenience. They are tuned for speed and availability, not for the different needs different passengers have [9].

The gap matters because the riders who most need a dependable service are often the ones the standard flow serves worst. Women travelling alone after dark, senior citizens, and passengers with reduced mobility all take the same trip as everyone else, but with different risks and different requirements. A platform that treats all of them as one anonymous request cannot respond to those differences, however fast its matching engine is.

A. Structural Shortcomings in Existing Platforms

Three weaknesses recur in current ride-hailing systems:

- 1) **One booking flow for everyone.** A woman travelling alone at night may care most about who is driving and whether someone knows where she is. An elderly passenger may struggle with small buttons and unfamiliar steps and may need a caregiver involved. A wheelchair user needs a vehicle that can actually carry the chair, which a generic “nearest driver” matcher never checks [5], [13].
- 2) **Reactive safety.** Safety is usually engaged only after something has gone wrong, through an SOS button or a

- 3) **Scattered features.** Verification, tracking, emergency handling and administration are often separate tools that share no context, which slows down response and leaves operators with a fragmented view.

B. Motivation

Safety is not a fixed property of a ride. It shifts with the time of day, the route, the distance from well-known safe locations and the character of the area being crossed [4], [14]. A platform that can read these signals before and during the trip can adjust how it treats the ride. SafeGo is built on this idea: safety, accessibility and verification are part of the ride lifecycle itself rather than add-ons.

C. Problem Statement

To design, implement and evaluate an integrated ride-hailing platform that offers specialised transportation for different passenger groups while continuously handling safety-risk assessment, accessibility-aware matching, ride verification, live monitoring and emergency response, and that gives administrators a single command center over all of these.

D. Core Contributions

- A unified ride-hailing platform supporting four specialised ride modes: Normal, Pink, Elderly and PWD.
- A Random Forest safety-risk prediction layer that classifies trips as Stable, Cautious or High Priority.
- Mode-aware driver and vehicle matching, including a vehicle-capability check for PWD rides.
- PIN-based verification before ride initiation and real-time ride monitoring with SOS emergency response.
- Transparent, platform-controlled fare calculation.
- An administrative command center for drivers, passengers, rides, incidents, complaints and analytics.

E. Organisation of the Paper

Section II surveys related work and Section III states the research gap. Section IV describes the architecture, data flow and data model. Section V gives the methodology, including the matching, risk-prediction, verification and fare formulations. Section VI reports the experiments, Section VII compares SafeGo with conventional platforms, Section VIII discusses ethical and privacy considerations, and Sections IX and X cover limitations, conclusions and future work.

II. BACKGROUND STUDY AND LITERATURE SURVEY

This section reviews the work behind each component of SafeGo: ride-hailing and matching, passenger safety and emergency alerting, machine-learning risk prediction, accessibility, and the real-time and pricing infrastructure.

A. Ride-Hailing Systems and Driver Matching

Modern platforms handle booking, navigation, fare estimation and payment reasonably well, mostly by optimising for driver proximity, availability, expected travel time and current demand [3], [9]. Smart ride-booking systems with real-time tracking and basic safety features have been described in [3], and cloud-based mobility platforms in [9]. Matching engines, however, still treat passengers as one broad category. Accessibility, support for elderly riders and gender-based preferences are usually handled outside the core matching logic. Field evidence also shows that ride-hailing outcomes differ across passenger groups: Ge et al. [12] found gender and racial discrimination in transportation network companies, which supports building fairness and inclusion into the platform rather than leaving it to chance.

B. Passenger Safety and Emergency Alerting

Safety research in this area has centred on alert devices and mobile applications. Sharma et al. [1] describe an IoT-based women-safety device with GPS tracking and emergency alerts, and Gupta et al. [2] a mobile emergency alert system built on GPS and GSM. Khan et al. [7] cover real-time location tracking with GPS and IoT, and Iyer et al. [8] look at optimising emergency response through mobile applications. These are effective once an incident has started. Women's perception of safety also depends on the environment they travel through, as discussed by Loukaitou-Sideris [14], which argues for reading the context of a trip and not only reacting to an alarm.

C. Machine Learning for Risk Prediction

Nair et al. [4] discuss machine-learning-based risk prediction in smart transportation systems, and Roy et al. [6] apply AI to driver behaviour analysis for safer transport. Random Forests, introduced by Breiman [10], combine many decision trees trained on random subsets of data and features and take a majority vote. They cope well with mixed numerical and categorical inputs, give class probabilities, and are cheap enough at inference time to sit inside an interactive booking flow, which is why SafeGo uses them. For multi-class problems, accuracy alone can hide weak classes, so macro-averaged precision, recall and F1 are reported as well [11].

D. Accessibility in Transportation

Patel et al. [5] review accessibility solutions for persons with disabilities in transportation systems, and Bezyak et al. [13] document the barriers people with disabilities face when using public transport. A recurring problem is the mismatch between what the passenger needs, such as a wheelchair-accessible vehicle, and what the nearest available vehicle can offer. Closing this gap requires the matcher to check vehicle and driver capability as a hard condition.

E. Real-Time Communication, Security and Pricing

Live location and ride-status updates need a persistent, low-latency channel, for which the WebSocket protocol [16] is a standard choice. Session and role security in web back ends is commonly handled with JSON Web Tokens [17]. On the pricing side, Chen and Sheldon [15] study dynamic pricing on a ride-hailing platform, which motivates keeping price adjustments under platform control and visible to the rider.

III. RESEARCH GAP AND NOVEL CONTRIBUTIONS

A. Gap Analysis

Existing ride-hailing apps are efficient at booking a trip, but safety and accessibility concerns remain scattered across separate features. The literature above exposes four gaps:

- 1) **Isolated safety tools.** Alert devices and tracking apps [1], [2], [7] work on their own and are not connected to matching,

- 3) incident starts. There is little pre-trip and in-trip risk assessment tied to the ride mode [4].
- 4) **Accessibility outside the matcher.** Wheelchair and elderly support is treated as an afterthought instead of a matching constraint [5], [13].
- 5) **No single architecture.** No one design combines ride modes, risk prediction, capability-aware matching, secure verification, live monitoring, emergency response and administrative intervention.

B. Novel Solutions in SafeGo

SafeGo changes the usual flow from *Book* → *Match* → *Ride* to *Passenger Type* → *Specialised Matching* → *Risk Assessment* → *Secure Verification* → *Continuous Monitoring* → *Emergency Intervention*. Safety considerations are therefore woven into several stages of the ride lifecycle at once. Concretely, SafeGo offers:

- 1) **Passenger-type awareness** at the very start of the flow, through four ride modes.
- 2) **Capability-aware matching** that treats vehicle and driver capability as a condition, not a preference.
- 3) **A proactive risk layer** that scores each trip from time, place, distance and mode before it begins.
- 4) **Verification, monitoring and SOS** tied to the same ride record, so that a single incident view exists for administrators.

C. Positioning Against Related Work

Table I places the cited safety and mobility works next to what SafeGo adds. The focus of each work is taken from its title and stated aim; the point is not that these works are weak, but that each addresses one piece of the problem.

TABLE I. RELATED WORK AND WHAT SAFEGO ADDS

Work	Primary focus	Added by SafeGo
[1] Sharma et al.	IoT women-safety device with GPS and alerts	Safety inside the booking flow; Pink Mode
[2] Gupta et al.	Mobile emergency alerts over GPS and GSM	Incident record, admin view, ride context
[3] Verma et al.	Ride booking with tracking and safety features	Risk prediction, capability-aware matching
[4] Nair et al.	ML risk prediction in transportation	Risk tied to ride mode and platform actions
[5] Patel et al.	Accessibility for persons with disabilities	Accessibility used as a matching condition
[7] Khan et al.	Real-time GPS/IoT location tracking	Tracking linked to SOS and verification
[8] Iyer et al.	Emergency response via mobile apps	Central command center for incidents

IV. PROPOSED WORK AND SYSTEM ARCHITECTURE

SafeGo is a single, integrated platform made up of passenger, driver and administrator components that share one backend. Ride management, safety prediction, driver matching, emergency handling, fare processing and administrative monitoring all sit within this platform instead of being spread across separate systems.

A. Layered Architecture

The system is organised in five layers, shown in Fig. 1:

- 1) **Client tier:** web interfaces for passengers, drivers and administrators.
- 2) **Transport and security tier:** the API gateway, token-based authentication and authorisation [17], security middleware and a WebSocket server [16] for live updates.
- 3) **Business services:** ride management, geographic and routing services, driver management and notifications.
- 4) **ML inference tier:** the Random Forest safety predictor, a

unavailable.

- 5) **Persistence and external services:** a document database for users, drivers, rides, ratings and ride-location history, plus telephony, mapping/routing and media-storage services.

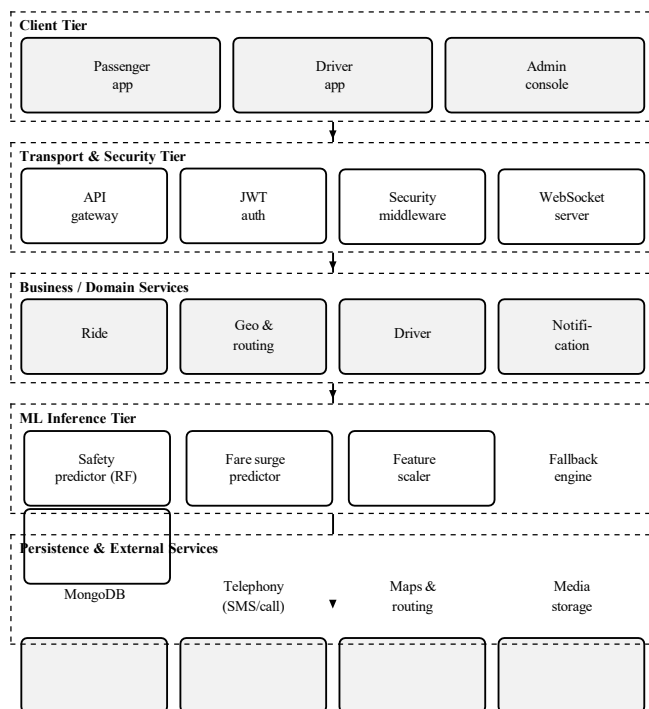


Fig. 1. Layered system architecture of SafeGo.

B. Ride Modes

- **Normal Mode** is the standard flow: the passenger enters pickup and destination, available drivers are identified, a suitable driver is matched and accepts, the passenger verifies the ride with a PIN, the trip begins, location is monitored, and the fare is calculated at the end.
- **Pink Mode** is aimed at women passengers and applies extra driver-selection and monitoring rules defined in SafeGo's configuration.
- **Elderly Mode** offers a simplified booking flow and lets a caregiver take part in the process.
- **PWD Mode** uses accessibility information supplied by the passenger, for example a wheelchair-accessible vehicle requirement, directly in matching.

The mode is chosen from the passenger's own profile and booking choices. How eligibility for Pink, Elderly and PWD modes is confirmed, and how eligible drivers are identified for Pink Mode, is an open design question discussed in Section VIII.

C. Operational Workflow

Fig. 2 shows the end-to-end ride lifecycle. A booking request carries the passenger ID, pickup location, destination, ride mode, any accessibility requirement and the requested time. The backend filters suitable drivers, the safety model scores the trip, the driver accepts, and the PIN check releases the start of the ride. The SOS path can be triggered at any point during the trip. Note that matching runs before scoring in this flow, so the risk class is recorded with the ride and shown to administrators but does not yet influence which driver is selected.

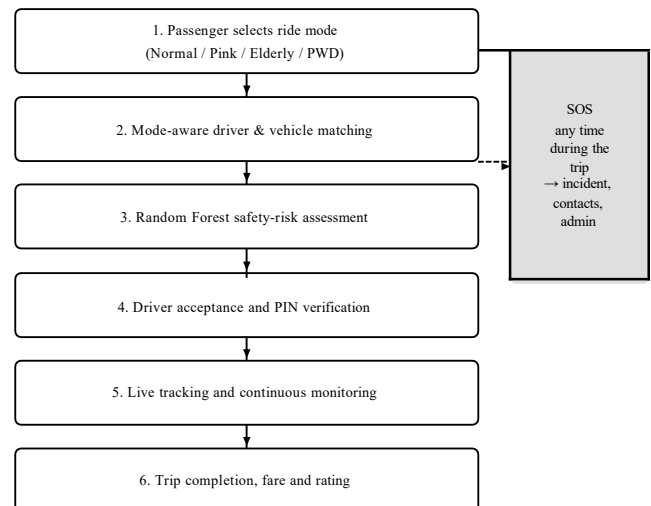


Fig. 2. Ride lifecycle in SafeGo with the SOS branch.

D. Data Flow

The Level-0 data flow diagram in Fig. 3 treats the whole platform as a single process. The passenger sends booking requests, PIN entries and SOS signals into it and receives the risk level, fare and ride status. The driver receives ride requests and PINs and sends back acceptances and live location. The administrator sends monitoring queries and receives incident alerts and analytics. All persistent state, meaning users, rides, incidents and location history, sits in the database that the platform stores to and fetches from.

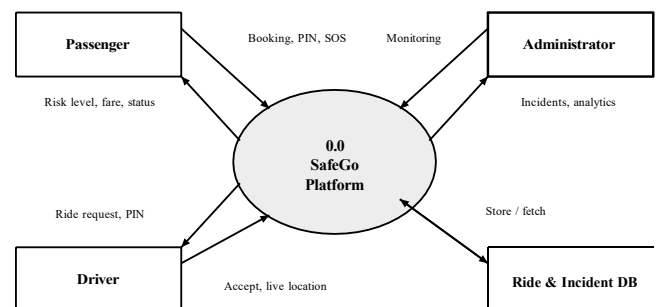
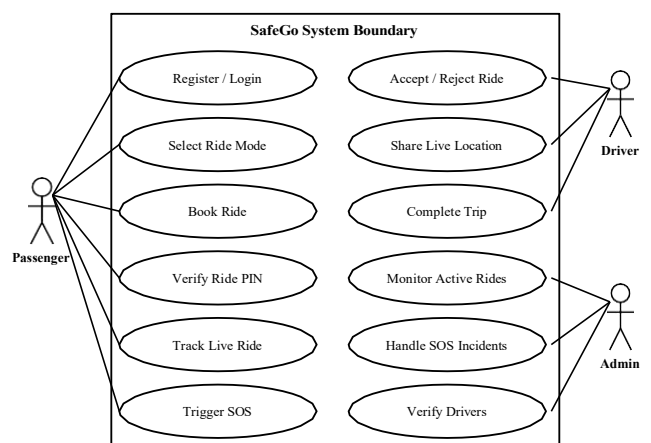


Fig. 3. Level-0 data flow diagram of the SafeGo platform.

E. Roles and Use Cases

Fig. 4 shows what each actor can do. Passengers register, choose a ride mode, book, verify the ride PIN, follow the trip live and trigger SOS. Drivers accept or reject requests, share live location and complete trips. Administrators monitor active rides, handle SOS incidents and verify drivers before they can take rides. Keeping these roles separate limits what each account can see and change.



F. Interaction Sequence

Fig. 5 traces one ride through the system. The backend sends the request to a matched driver, passes the trip features to the ML model and receives a risk class with probabilities, and the driver accepts. The ride starts only after the PIN is validated. During the trip the driver's location is streamed to the backend, an SOS from the passenger raises an incident alert to the administrator, and the trip ends with the fare and a summary sent to the passenger.

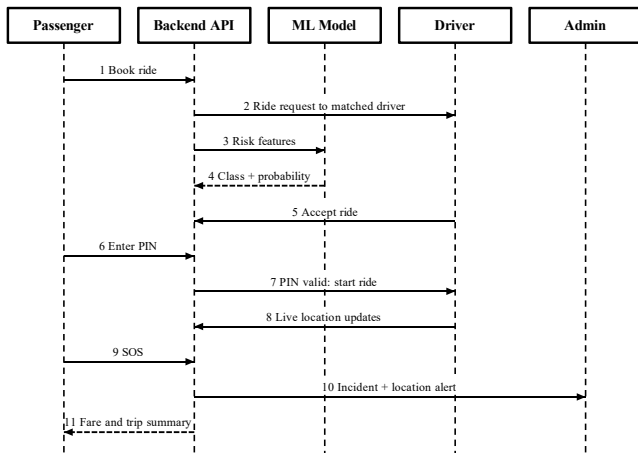


Fig. 5. Sequence diagram of a ride from booking to completion.

G. Ride State Lifecycle

Each ride moves through a small set of states (Fig. 6). A ride is Pending until a driver accepts, becomes Accepted, and moves to In Progress once the PIN is verified. It ends as Completed, or as Cancelled if it is dropped before the trip begins. An SOS during the trip moves the ride into an Emergency state until it is resolved.

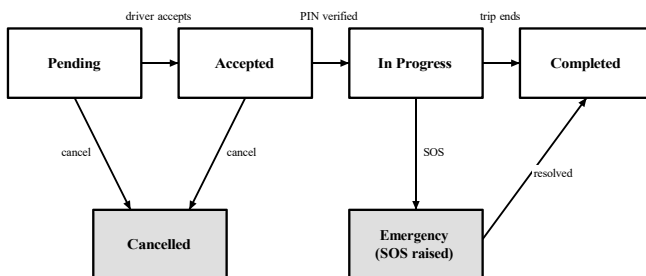


Fig. 6. Ride state diagram.

H. Data Model

The persistent data model links users, drivers, rides and their dependent records (Fig. 7). One user or one driver can have many rides, and each ride can have many location-history points, incident records and, after the trip, a rating. Driver attributes such as verification status, availability, served modes and wheelchair capability are what the matcher reads. The ride record carries the mode, status, risk class, PIN and fare. The figure shows a simplified logical view rather than every field: it does not show the caregiver link that Elderly Mode needs, or any driver attribute needed to establish Pink-Mode eligibility. Both would be added as fields or relations on the user and driver records.

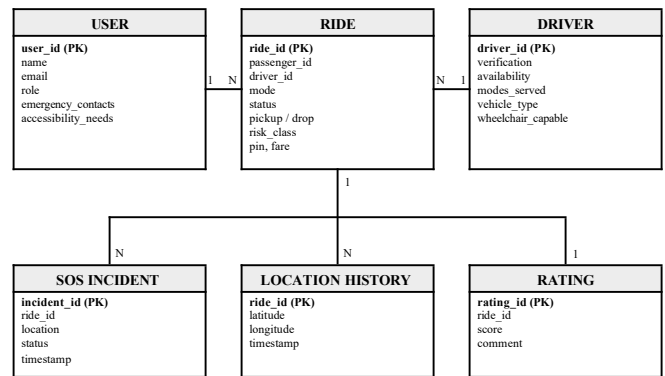


Fig. 7. Simplified entity-relationship view of the SafeGo data model.

I. Resilience and Cross-Cutting Concerns

A safety platform has to keep working when parts of it do not. Table II lists degraded conditions the design has to account for and the intended handling. The fallback engine is the clearest example: if the ML model cannot be reached, a deterministic rule-based engine still supplies a risk level so booking is not blocked. The fallback engine's rules and its agreement with the Random Forest are not evaluated in this paper.

TABLE II. DEGRADED CONDITIONS AND INTENDED HANDLING

Condition	Intended handling
ML model unavailable	Deterministic fallback engine returns a rule-based risk level; booking continues
GPS / location unavailable	Location-dependent features cannot be computed; the request waits for a valid location
Database unavailable	Ride and incident writes fail visibly instead of starting a ride with no record
Network outage	Clients reconnect on the real-time channel and re-synchronise ride status
Telephony service unavailable	Incident is still recorded and shown to admins; contact alerts are retried

Across all layers, the design aims at five system qualities: security (token-based authentication, role-based access and security middleware), scalability (services that can be scaled independently), reliability (the fallbacks above), observability (logging and monitoring of rides and incidents) and compliance (careful handling of personal and location data). These are design aims; load and failure-injection testing have not been carried out in this study.

J. Administrative Command Center

The admin portal gives operators one place to manage or monitor drivers, passengers, active rides, ride history, driver verification, SOS incidents, complaints, safety events and system analytics. Live location and ride-status data reach authorised administrators through the same real-time channel used by the rider and driver apps.

K. Implementation Overview

SafeGo was built as a full-stack application with passenger, driver and administrator components. The build brings together a frontend application, backend APIs, a database, the machine-learning services, mapping and location services, real-time communication and emergency communication services. All of these sit inside one platform, rather than being split into separate apps for each safety-related feature. This matters for maintenance as well as for users: a change to the matching rules or to the risk model is made in one place and applies to every ride mode.

L. Feature Summary by Actor

Table III summarises what each of the three actors can do on the

TABLE III. SAFEGO FEATURES BY ACTOR

Actor	Main capabilities
Passenger	Choose ride mode, book, view driver, verify PIN, track ride, press SOS, see fare, rate
Driver	Register and get verified, go online, accept or reject requests, share location, complete trips
Administrator	Monitor rides, verify drivers, handle SOS incidents and complaints, view analytics

M. Design Requirements

The architecture above follows from a short set of requirements (Table IV). Functional requirements describe what the platform must do, and non-functional requirements describe how well it must do it.

TABLE IV. DESIGN REQUIREMENTS AND WHERE THEY ARE MET

ID	Requirement	Addressed by
FR1	Passenger-specific ride modes	Normal, Pink, Elderly, PWD modes
FR2	Match only suitable drivers and vehicles	Mode filter and capability check
FR3	Assess trip risk before the ride	Random Forest safety classifier
FR4	Prevent unauthorised ride start	Four-digit PIN verification
FR5	Follow the ride live	WebSocket updates, admin view
FR6	Raise and log emergencies	SOS workflow and incident record
FR7	Transparent pricing	Platform-set base and additional fare
NFR1	Fast enough for booking	About 32 ms mean prediction time
NFR2	Secure access	Token authentication, role checks
NFR3	Stay usable if a part fails	Fallback engine, retries
NFR4	Usable by seniors and PWD	Simplified Elderly flow, caregiver role

On security, access is controlled at three points. Users authenticate and receive a token that carries their role [17]; middleware checks the role before each protected route runs; and the PIN check stops a ride from starting without the passenger's participation. Administrator views are separated from passenger and driver views, so operational data such as other users' rides is not exposed to ordinary accounts. The PIN has limits worth stating: a four-digit code has only 10,000 possible values, so it needs attempt limiting, expiry and logout to resist guessing, and it confirms that the passenger took part in starting the ride but does not by itself prove that the arriving driver and vehicle are the ones assigned. Plate or photo confirmation of the driver and vehicle would close that gap.

No formal threat model has been carried out in this study.

V. METHODOLOGY AND MATHEMATICAL FORMULATIONS

A. Mode-Aware Driver Matching

For a ride request r with mode $m(r)$ and accessibility requirement $a(r)$, the matcher first keeps only drivers who are online and eligible for the mode. For PWD rides, vehicle capability is checked as an extra condition before a match is finalised. The eligible set $E(r)$ is

$$E(r) = \{ d \in D_{\text{online}} : m(r) \in M(d) \text{ and } a(r) \subseteq C(d) \} \quad (1)$$

where $M(d)$ is the set of modes the driver serves and $C(d)$ the capabilities of the driver and vehicle. Location evaluation then selects among eligible drivers:

$$d^* = \arg \min_{d \in E(r)} \delta(d, \ell_r) \quad (2)$$

with δ the distance between driver and pickup locations. The pipeline is Ride Request $\rightarrow$ Online Drivers $\rightarrow$ Mode Filtering $\rightarrow$ Capability Check $\rightarrow$ Location Evaluation $\rightarrow$ Suitable Driver, so

SafeGo goes beyond picking the closest driver. Algorithm 1 summarises the steps. Two points limit what this formulation shows. First, δ may be straight-line or road-based distance or travel time, and the choice affects which driver counts as nearest. Second, matching has been checked functionally but not measured: in practice the pools of PWD-capable and Pink-eligible drivers may be thin, so no-match rate and passenger wait time are the quantities that need a simulation or field measurement.

Algorithm 1. Mode-aware driver matching

Input: ride request r (mode, pickup, accessibility need). **Output:** driver d^* or none.
 1. $D \leftarrow$ all drivers currently online.
 2. Keep drivers whose served modes include $m(r)$.
 3. If $m(r) = \text{PWD}$, keep drivers whose vehicle meets $a(r)$.
 4. If $m(r) = \text{Pink}$, apply the Pink-mode selection rules from configuration.
 5. If the remaining set is empty, return none.
 6. Rank remaining drivers by distance to the pickup.
 7. Return the nearest driver d^* .

B. Input Features

The safety model uses a 12-feature representation covering temporal, geographical, trip-level and ride-mode information (Table V). Features pass through a scaler as part of the inference pipeline. Tree ensembles split on thresholds and are unaffected by monotonic rescaling, so the scaler does not change the Random Forest's decisions.

TABLE V. FEATURE GROUPS USED BY THE SAFETY MODEL

Group	Features
Temporal	Pickup hour, day of week
Geographical	Pickup latitude, pickup longitude, hotspot proximity, safe-hub distance
Trip-level	Route distance and other trip variables
Ride mode	Normal, Pink, Elderly or PWD
Contextual	Further contextual variables (12 features in total)

Table V gives the feature groups only. The full list of the 12 features and the source of the reference points behind hotspot proximity and safe-hub distance need to be specified in full for the work to be reproducible.

C. Random Forest Safety-Risk Classifier

Each trip x is assigned to one of three classes, $Y \in \{\text{Stable, Cautious, High Priority}\}$. A Random Forest [10] builds T trees, each splitting nodes to reduce Gini impurity

$$G(t) = 1 - \sum_k p_k^2 \quad (3)$$

where p_k is the fraction of class k samples at node t . The final label is the majority vote of the trees, and the class probability is the share of trees voting for that class:

$$P(Y=k|x) = (1/T) \sum_{t=1}^T 1[h(x) = k] \quad (4)$$

Because each tree sees a different random sample of data and features, the errors of individual trees are partly independent, and the vote is more stable than any single tree [10]. The number of trees, tree depth and any class-weighting used for the reported model are not stated in this paper.

D. Safety Score and Risk Categories

Beyond the class label, the class probabilities can be turned into a safety-oriented score. One simple formulation, shown here for illustration, weights each class by its severity:

$$S(x) = 100 \cdot \sum_k w_k P(Y=k|x) \quad (5)$$

with, for example, $w_{\text{Stable}} = 1$, $w_{\text{Cautious}} = 0.5$ and $w_{\text{High Priority}} = 0$.

These weights are illustrative and have not been validated, and the score is not used in the results below. The three categories are intended to guide how closely a trip is handled: Stable trips follow the normal flow, Cautious trips call for closer attention to location and status, and High Priority trips are the ones administrators

should be able to watch most closely. The exact response for each category is a configuration choice. Two scope notes apply. Each trip is scored once, before it starts; there is no in-trip risk model, so events such as route deviation or an unexpected stop do not change the score. And a High Priority label for a Pink, Elderly or PWD ride should be read as a request for closer handling because of the passenger's needs, not as a claim that the trip itself is more likely to be unsafe.

E. Secure Ride Verification

Before a ride can start, a four-digit PIN must be entered. The backend compares it with the PIN issued for that booking:

$$\text{start}(r) = 1 \text{ if } \text{PIN}_{\text{in}} = \text{PIN}_{\text{stored}}, \text{ else } 0 \quad (6)$$

If the PIN does not match, the ride does not begin. This guards against a trip being started incorrectly or by the wrong person. As noted in Section IV, a four-digit code needs attempt limits and expiry, and it does not verify the driver or vehicle on its own.

F. Emergency and SOS Workflow

When the passenger presses SOS, the system captures the ride and location, creates an incident record, notifies the emergency contacts and the administration, and hands over to emergency assistance (Fig. 8). Logging the event centrally avoids leaving everything to the driver or passenger in the moment.

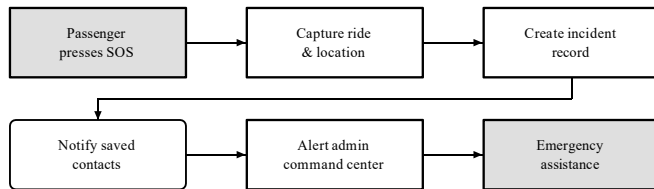


Fig. 8. SOS emergency workflow.

G. Fare Management

Fares combine the standard base fare with any configured dynamic pricing component:

$$F_{\text{final}} = F_{\text{base}} + F_{\text{additional}} \quad (7)$$

Adjustments come from the platform itself and are not set independently by individual drivers, which keeps pricing transparent [15]. The fare-surge predictor shown in the architecture is not described or evaluated further in this paper.

H. End-to-End Ride Procedure

Algorithm 2. Risk-assessed ride lifecycle
Input: booking request r . Output: completed or cancelled ride. 1. Match a driver with Algorithm 1; if none is found, report no match. 2. Build the 12-feature vector x from r and scale it. 3. Obtain the class and probabilities from the Random Forest; if the model is unavailable, use the fallback engine. 4. Store the risk class with the ride record and wait for the driver to accept. 5. On acceptance, issue the PIN; start the ride only if the entered PIN matches (Eq. 6). 6. While the ride is in progress, stream location and status to the passenger and to authorised administrators. 7. If SOS is pressed, run the SOS workflow of Fig. 8. 8. On completion, compute the fare (Eq. 7) and record the rating.

I. Evaluation Metrics

For each class k , precision, recall and F1 are computed from true positives (TP), false positives (FP) and false negatives (FN), and averaged over the $K = 3$ classes [11]:

$$P_k = \text{TP}_k / (\text{TP}_k + \text{FP}_k), R_k = \text{TP}_k / (\text{TP}_k + \text{FN}_k) \quad (8)$$

$$F1_k = 2P_k R_k / (P_k + R_k), \text{ Macro-F1} = (1/K) \sum_k F1_k \quad (9)$$

J. Computational Cost

Matching filters the set of online drivers once and ranks the remainder, so its cost grows linearly with the number of online drivers considered. Random Forest inference costs roughly the number of trees times the depth of each tree, and the trees can be evaluated independently. Neither step involves iterative optimisation at request time, which is consistent with the measured prediction time of about 32 ms. Training is done offline and does not affect the booking flow.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

A. Experimental Setup

The safety-risk model was evaluated on the project's labelled dataset of 10,000 samples, of which 8,000 were used for training and 2,000 were held out for testing. All results below come from the trained Random Forest classifier on that held-out set. The dataset is synthetically generated, so the numbers describe performance on this evaluation set and should not be read as a claim about real-world safety-prediction accuracy. Because the data are synthetic, the labels come from the project's own labelling scheme and not from recorded incidents or expert review; the generation and labelling procedure, and a check for duplicate or near-duplicate samples across the train and test sets, should be documented alongside the results. From the confusion matrix, the test set contains 1,018 Stable (50.9%), 373 Cautious (18.65%) and 609 High Priority (30.45%) trips, so no class is extremely rare.

B. Classification Performance

TABLE VI. SAFETY-RISK CLASSIFICATION PERFORMANCE

Metric	Result
Accuracy	99.80%
Macro Precision	99.87%
Macro Recall	99.78%
Macro F1-score	99.82%
Weighted F1-score	99.80%

C. Confusion Matrix and Per-Class Results

Table VII shows the confusion matrix. Of the 2,000 test samples, 1,996 were classified correctly. The only errors are four High Priority trips predicted as Stable; no Stable or Cautious trip was misclassified.

TABLE VII. CONFUSION MATRIX (ACTUAL VS. PREDICTED)

Actual / Predicted	Stable	Cautious	High Priority
Stable	1018	0	0
Cautious	0	373	0
High Priority	4	0	605

Per-class values follow directly from this matrix (Table VIII). The single weak point is recall on High Priority trips (605 of 609). In a safety setting, missing a high-priority trip is the more costly error, so this is the figure to improve first when real data becomes available. A practical option is to lower the decision threshold for High Priority so that borderline trips are flagged, at the cost of some extra false alarms.

TABLE VIII. PER-CLASS RESULTS DERIVED FROM TABLE VII

Class	Support	Precision	Recall	F1
Stable	1018	99.61%	100.00%	99.80%
Cautious	373	100.00%	100.00%	100.00%
High Priority	609	100.00%	99.34%	99.67%
Macro average	2000	99.87%	99.78%	99.82%

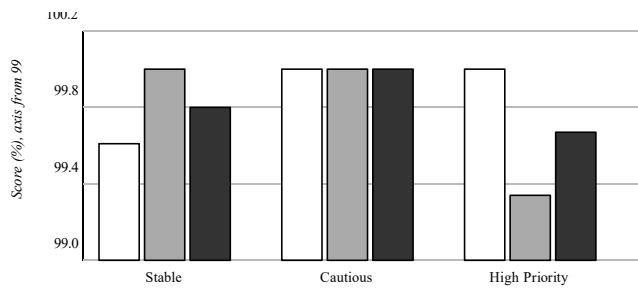


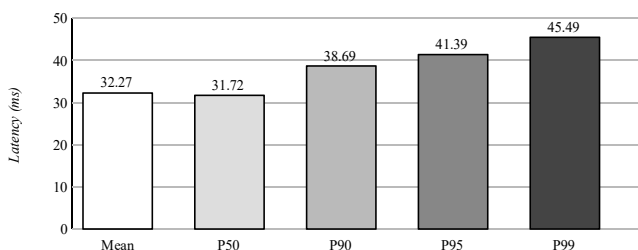
Fig. 9. Per-class precision, recall and F1 (vertical axis starts at 99% to make the small differences visible).

D. Prediction Latency

Prediction time was measured over repeated runs. The mean was 32.27 ms and the 99th percentile 45.49 ms (Fig. 10), which is fast enough for the model to sit inside a real-time booking flow without a noticeable delay. The narrow spread between the median and the 99th percentile, about 14 ms, means that even the slowest predictions stay well below anything a passenger would notice. These figures cover model prediction only; the hardware used and end-to-end booking, WebSocket and SOS-alert latency were not measured in this study.

TABLE IX. PREDICTION LATENCY

Measurement	Mean	P50	P90	P95	P99
Time (ms)	32.27	31.72	38.69	41.39	45.49



confidence (70.19%) among the five, meaning the trees disagree more on it. This is the case where more data, or a review of the labelling rules, would help most. Note that these confidence values are vote shares and not calibrated probabilities.

F. Functional System Testing

Alongside the ML evaluation, the major functional modules were tested directly (Table XI), which gives a fuller picture of how the application behaves end to end. Table XI lists the scope of testing only; case-level results, load testing and usability testing with elderly and PWD users are not reported here.

TABLE XI. FUNCTIONAL TESTING SCOPE

Module	Testing area
Authentication	Registration, login and authorisation
Ride Booking	Pickup, destination and booking
Four Modes	Normal, Pink, Elderly and PWD
Driver Matching	Availability and mode-specific matching
PIN Verification	Ride-start authorisation
Live Tracking	Location and ride status
SOS	Emergency activation and handling
Fare	Fare calculation
Driver Portal	Requests and ride management
Admin Portal	Drivers, rides and incidents
Accessibility	PWD-specific ride requirements

G. Illustrative Walkthroughs

PWD ride. A passenger who needs a wheelchair-accessible vehicle books in PWD Mode. The matcher removes every online driver whose vehicle does not meet the requirement, ranks the rest by distance and offers the request to the nearest. The trip features are scored and the ride is labelled High Priority (96.59% in Table X), the driver accepts, and the ride starts only after the PIN is entered.

Evening Pink ride. A woman books in Pink Mode in the evening. Pink-mode driver-selection rules are applied before ranking, the trip is scored High Priority (70.19%), and live tracking and the SOS button are available throughout. The lower confidence tells

To check that the model responds to context, five scenarios were run (Table X). The output changes with the combination of trip context and ride mode instead of defaulting to one answer: a daytime downtown Normal ride is Stable, a late-night suburban bypass is Cautious, and the PWD, Pink and Elderly examples are High Priority.

TABLE X. SCENARIO-BASED ROUTE-RISK PREDICTIONS

Scenario	Mode	Prediction	Confidence
Normal daytime downtown	Normal	Stable	95.95%
Late-night suburban bypass	Normal	Cautious	86.58%
Accessibility ride	PWD	High Priority	96.59%
Evening women ride	Pink	High Priority	70.19%
Senior citizen ride	Elderly	High Priority	98.31%

Three observations follow. First, all three special modes land in High Priority in these examples and no Normal ride does. Because each scenario changes the mode and the trip context together, Table X cannot separate their effects, and the classifier may be leaning heavily on ride mode. A controlled test that scores the same trip in all four modes, together with a feature-importance analysis, is needed before drawing that conclusion. Second, what High Priority means for these modes needs care: it is intended to prompt closer handling of a passenger with particular needs, not to say the trip is riskier. Keeping passenger vulnerability and trip risk as separate signals is a design question for future work, since merging them can flood administrators with alerts and label whole passenger groups as risky. Third, the Pink-mode scenario has the lowest operators that this is a borderline case worth reviewing as real data accumulates.

H. Discussion and Threats to Validity

The classifier separates Stable, Cautious and High Priority trips very well on the synthetic test data, and its low inference time suits an interactive booking flow. These numbers need careful reading, for six reasons.

- 1) **Easy class boundaries.** Scores this close to 100% on synthetic data usually mean the labels can be recovered from the features, for example because they were generated by rules over the same inputs.
- 2) **Distribution shift.** Real trips are noisier and their patterns change across cities, seasons and events, none of which the synthetic data captures.
- 3) **Single split.** One 80/20 split gives one estimate, and the model settings are not reported; repeated or cross-validated runs would show the variance.
- 4) **No comparison models.** Other classifiers were not benchmarked, so the evidence supports that a Random Forest works here, not that it is the best choice.
- 5) **Mode and context confounded.** The scenario test changes mode and context together, so it does not show how much the prediction depends on ride mode alone.
- 6) **Parts not evaluated.** Matching quality, the fare-surge predictor and the fallback engine were not measured, so their performance is unknown.

The results therefore show that the pipeline works end to end and that the model reacts to mode and context, not that it will match this

accuracy in deployment.

VII. COMPARATIVE ANALYSIS AND ADVANTAGES

Table XII compares SafeGo with a typical conventional ride-hailing app. SafeGo treats safety and passenger-specific needs as part of the core ride lifecycle rather than as extras layered on top.

TABLE XII. CONVENTIONAL RIDE-HAILING VS. SAFEGO

Elderly Mode	Not necessarily	Yes
PWD Mode	Limited / varies	Yes
Accessibility-aware matching	Limited	Yes
ML safety prediction	Limited	Yes
PIN ride verification	Platform dependent	Yes
Live ride monitoring	Yes	Yes
SOS	Platform dependent	Yes
Centralised SOS management	Limited / varies	Yes
Dynamic fare	Yes	Yes
Driver verification	Yes	Yes
Unified admin command center	Platform dependent	Yes

This comparison is at feature level against conventional apps in general, and the Conventional column varies by platform. Some existing platforms already offer PIN-based ride start, SOS, trip sharing, women-preference options and accessible-vehicle options. SafeGo's contribution is therefore best read as bringing these together with a pre-trip machine-learning risk score and capability-aware matching in one ride lifecycle and one administrative view, not as introducing each feature for the first time. A comparison against named applications is left for future work.

Core advantages:

- **Inclusive by design:** ride modes and capability-aware matching address women, elderly riders and PWD directly.
- **Proactive safety:** the risk model flags context before the trip starts, and SOS covers what happens during it.
- **Integrated operations:** one platform and one command center for routine ride management and safety incidents.
- **Real-time ready:** a mean prediction time of about 32 ms fits inside the booking flow.
- **Resilient:** a deterministic fallback keeps risk assessment available if the ML service is not.

VIII. ETHICAL, PRIVACY AND DEPLOYMENT CONSIDERATIONS

A platform that classifies trips by risk and stores location and accessibility details carries responsibilities beyond model accuracy.

- 1) **Sensitive data.** Location traces, gender-based ride choices and disability information are personal data. Only what the matcher and safety layer need should be collected, access should be restricted by role, and users should be told clearly how the data is used. A deployment would also need explicit consent for location and disability data, defined retention limits, encryption in transit and at rest, and alignment with applicable law such as India's Digital Personal Data Protection Act, 2023.
- 2) **Fairness.** Risk labels tied to places can end up penalising neighbourhoods or the drivers who work there. Evidence of discrimination on ride-hailing platforms [12] is a reminder to audit outcomes by area and passenger group before deployment.
- 3) **Human oversight.** A High Priority label should prompt attention, not automatic punishment of a driver or refusal of a

passenger. Administrators keep the final decision.

- 4) **Emergency integration.** Passing SOS events to real emergency authorities needs legal and operational agreements that go beyond the software.
- 5) **Inclusive design.** Elderly and PWD users should be involved in testing the interfaces, so that the modes work in practice and not only on paper [13].
- 6) **Mode eligibility.** Pink, Elderly and PWD modes depend on who the passenger is, and Pink Mode also depends on identifying eligible drivers. Confirming eligibility without excluding people or exposing sensitive details, and preventing misuse of a mode by ineligible riders, is unresolved here and needs a policy as well as a technical mechanism.

IX. LIMITATIONS

- 1) Synthetic dataset: the reported metrics are not a measure of real-world safety-prediction performance.
- 2) Real-world generalisation: real transportation environments are far less predictable than the current dataset.
- 3) GPS dependence: safety and route features need accurate location data and a working network connection.
- 4) Emergency services: connecting to emergency authorities in a live deployment needs legal, operational and organisational groundwork.
- 5) Model validation: testing on larger, real-world datasets is needed to judge how well the model generalises.
- 6) Specification and evaluation gaps: the full 12-feature list, model settings and dataset generation are not fully specified, and matching, the fare-surge predictor, the fallback engine, end-to-end latency and usability with elderly and PWD users were not evaluated.
- 7) Scope of risk assessment: each trip is scored once before it starts, risk does not yet influence driver selection, and there is no in-trip anomaly detection. Mode eligibility checks and PIN hardening are also open.

X. CONCLUSION AND FUTURE WORK

This paper presented SafeGo, a ride-hailing platform that brings transportation, passenger safety, accessibility and emergency response into one connected system. Beyond basic A-to-B booking, it adds four ride modes, mode-aware driver matching, PIN-based verification, live monitoring, SOS response, dynamic fare calculation and centralised administration. On a held-out synthetic set of 2,000 samples, the Random Forest safety-risk classifier reached 99.80% accuracy and 99.82% macro F1 with a mean prediction time of 32.27 ms, and scenario tests showed that its output shifts with ride mode and context. The next step is testing against real-world data and field conditions.

A. Future Directions

- 1) Train the safety model on larger real-world transportation datasets and expand real-world safety-hub and hotspot data.
- 2) Improve route-risk prediction with richer geographical information.
- 3) Improve accessibility-oriented vehicle matching and evaluate the platform with real passenger and driver feedback.
- 4) Improve the explainability of safety predictions, for example with SHAP-based feature attributions [18].
- 5) Integrate IoT-based in-vehicle safety hardware [1] and explore audio-based anomaly detection.
- 6) Develop offline emergency communication mechanisms and evaluate dynamic fare prediction under real demand conditions.

- 8) Run a controlled ablation that scores the same trip in all four ride modes, with feature-importance analysis, and separate passenger vulnerability from trip risk.
- 9) Add in-trip risk monitoring (route deviation, unexpected stops), and let the risk class inform driver selection.
- 10) Simulate driver supply for PWD and Pink rides to measure no-match rate and wait time, and compare SafeGo with named ride-hailing applications.
- 11) Harden PIN verification (attempt limits, expiry, driver and vehicle confirmation), define mode-eligibility verification, and complete a formal threat model and privacy review.

ACKNOWLEDGEMENT

The authors thank Ms. Nidhi Shah, Professor, Parul University, for guidance and supervision throughout the design, implementation and evaluation of this work.

REFERENCES

- [1] R. Sharma *et al.*, “IoT-Based Women Safety Device with GPS Tracking and Emergency Alerts,” *IEEE Access*, 2021.
- [2] S. Gupta *et al.*, “Mobile-Based Emergency Alert System Using GPS and GSM,” *International Journal of Engineering Research*, 2020.
- [3] A. Verma *et al.*, “Smart Ride Booking System with Real-Time Tracking and Safety Features,” *Springer Conference Proceedings*, 2022.
- [4] P. Nair *et al.*, “Machine Learning-Based Risk Prediction in Smart Transportation Systems,” *Elsevier Journal*, 2022.
- [5] K. Patel *et al.*, “Accessibility Solutions for Persons with Disabilities in Transportation Systems,” *Journal of Transport & Health*, 2021.
- [6] D. Roy *et al.*, “AI-Based Driver Behaviour Analysis for Safe Transportation,” *IEEE Smart Cities Conference*, 2023.
- [7] M. Khan *et al.*, “Real-Time Location Tracking System Using GPS and IoT,” *International Journal of Computer Applications*, 2020.
- [8] S. Iyer *et al.*, “Emergency Response Optimization Using Mobile Applications,” *IEEE Conference Publications*, 2021.
- [9] J. Singh *et al.*, “Cloud-Based Smart Mobility Platforms for Urban Transport,” *IEEE Access*, 2022.
- [10] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
- [11] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, Jul. 2009.
- [12] Y. Ge, C. R. Knittel, D. MacKenzie, and S. Zoepf, “Racial and gender discrimination in transportation network companies,” NBER Working Paper No. 22776, Oct. 2016.
- [13] J. Bezyak, B. Sabella, and R. Gattis, “Public transportation: An investigation of barriers for people with disabilities,” *Journal of Disability Policy Studies*, vol. 28, no. 1, pp. 52–60, 2017.
- [14] A. Loukaitou-Sideris, “Fear and safety in transit environments from the women’s perspective,” *Security Journal*, vol. 27, no. 2, pp. 242–256, 2014.
- [15] M. K. Chen and M. Sheldon, “Dynamic pricing in a labor market: Surge pricing and flexible work on the Uber platform,” in *Proc. ACM Conf. Economics and Computation (EC)*, 2016, p. 455.
- [16] I. Fette and A. Melnikov, “The WebSocket Protocol,” IETF RFC 6455, Dec. 2011.
- [17] M. Jones, J. Bradley, and N. Sakimura, “JSON Web Token (JWT),” IETF RFC 7519, May 2015.
- [18] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 4765–4774.