Telangana Road Accident Hotspot Prediction Using Quantum Machine Learning with a Variational Quantum Neural Network

[1]PRAGNA SRI YERRA, [2]VAKA SRIVYSHNAVI REDDY, [3]SUDHANABOINA KALYANI, [4]NUKALA DINESH YADAV, [5]B. SWAROOPA RANI

[1-5]Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning)
[1-5]CMR Technical Campus, Hyderabad, Telangana, India

***Abstract*—Road accidents remain a significant public safety concern, and a small number of road segments account for a large share of the harm. This paper proposes SafeRoute Telangana, a risk-aware navigation framework that identifies accident-prone road segments and recommends alternative routes across Telangana, India. The framework integrates historical accident analysis, Quantum Machine Learning (QML) and a Variational Quantum Neural Network (VQNN) implemented in Qiskit to estimate accident risk from accident and road-network features. It covers data preprocessing, spatial hotspot identification, risk estimation and route optimisation. A routing engine generates feasible alternatives, which are compared on distance, estimated travel time and accident-risk exposure, so that the shortest route and a potentially longer but lower-risk route can be shown side by side. A React Native application with a FastAPI backend visualises routes and hotspots on a digital map and gives visual and voice alerts near risk zones. The evaluation plan compares the VQNN with classical baselines (logistic regression, random forest, support vector machine) using accuracy, precision, recall, F1-score and AUC-ROC, together with route-level measures. Telangana-specific accident records are essential for regional validity, and the compatibility of supplementary datasets must be verified before training. Measured results are to be reported once training on verified regional data is complete, this version contains no measured performance figures. The system is intended to support informed travel decisions, not replace driver judgement.**

***Index Terms*—Road accident hotspot prediction, quantum machine learning, variational quantum neural network (VQNN), Qiskit, geospatial analysis, risk-aware route planning, intelligent transportation systems, road safety, Telangana**

I. Introduction

Road safety is affected by several connected factors, including road layout, traffic volume, vehicle condition, driver behaviour, visibility and weather. Low- and middle-income countries carry a disproportionate share of the global burden of road deaths [1], and India publishes national statistics annually through the Ministry of Road Transport and Highways [2]. Crashes are not spread evenly: they cluster at particular junctions, curves, highway stretches and urban corridors. Telangana, with a dense urban core around Hyderabad, many national and state highways and a growing vehicle fleet, has many such locations.

Historical accident records can reveal recurring patterns, but a record must be connected to a useful location before it can support a driver-facing hotspot warning. A navigation system that presents risk information before a driver reaches a road segment may help the driver make a more informed decision. Existing navigation tools mainly optimise distance and time, and do not show accident risk alongside them.

This work proposes SafeRoute Telangana, a mobile road-safety framework that uses a Variational Quantum Neural Network to estimate accident risk and presents the output on a map. The workflow comprises data

preparation, model training, a prediction API, map-based visualisation and proximity alerts. The VQNN estimates risk; the routing algorithm selects and compares routes. The VQNN does not itself calculate the shortest path. The quantum model is treated as a research approach to be tested against classical models, not as a model assumed to be better in advance.

The project has three practical aims: to study the available accident and supporting datasets, to develop and evaluate a VQNN-based risk classifier, and to connect the resulting risk estimates to a mobile navigation interface. A central issue is data compatibility, since the five identified sources do not all describe Telangana and do not all contain accident-level coordinates. The paper therefore separates dataset description from the additional work needed for valid local predictions. Section 2 surveys related work and Section 3 states the problem. Section 4 gives background, Section 5 compares existing and proposed systems, Section 6 describes the data, Section 7 the VQNN, Section 8 route planning and Section 9 the application. Sections 10 and 11 cover evaluation and the results and comparison framework, followed by limitations, future work and the conclusion.

II. LITERATURE SURVEY

A. Road accident hotspot detection

Anderson [3] investigated Kernel Density Estimation (KDE) and K-means clustering to profile accident hazard in London. Prasannakumar et al. [4] applied GIS and spatial statistics to analyse the spatial and temporal distribution of road accidents in a South Indian city. Density-based clustering such as DBSCAN
[5] also groups nearby crash points into dense regions. These studies show the usefulness of spatial analysis for locating accident concentrations.

B. Machine learning for accident risk prediction

Random forests [6], support vector machines [7], logistic regression and clustering algorithms have been applied to accident severity classification and pattern analysis. These methods use historical records and associated features to estimate risk. Their performance depends on data quality, feature selection and the availability of relevant accident and road-condition information, and rare hotspot classes need resampling [17] or class weights.

C. Quantum machine learning for classification

Schuld et al. [8] presented circuit-centric quantum classifiers that use parameterised circuits for supervised learning, and Cerezo et al. [9] reviewed variational quantum algorithms, which combine quantum circuits with classical optimisation. Havlíček et al. [10] showed quantum-enhanced feature spaces for classification, and Schuld and Petruccione [11] give a textbook treatment. These works provide the methodological basis for classifying road segments with a VQNN in Qiskit [19].

D. Risk-aware route optimisation

Krumm and Horvitz [12] studied risk-aware route planning by incorporating crash probability into route selection, showing how planning can weigh safety against travel efficiency. In SafeRoute Telangana this idea informs the comparison of the shortest route with a potentially longer, lower-risk alternative, considering distance, travel time and known hotspots. We have not cited Telangana-specific QML studies because their results depend on datasets and settings that must be compared carefully; this comparison is left for the final report.

III. PROBLEM STATEMENT AND OBJECTIVES

The project addresses the difficulty of identifying accident-prone road segments from historical records and presenting this information in a form a driver can use during route planning. The available sources vary in coverage and detail: a national road-infrastructure description is not an accident-level

dataset, Delhi climate observations do not represent Telangana weather, and UK accident records cannot be treated as local records. A valid model therefore requires a defined target, compatible features, reliable location matching and evaluation on data separated from training. The objectives are:

- to examine the five selected dataset sources and document their fields, coverage and limitations;
- to prepare compatible accident and supporting features for a road-segment risk model;
- to implement a VQNN with a hybrid quantum–classical training procedure and compare it with classical baselines;
- to connect predictions to a FastAPI backend and a React Native navigation interface;
- to display risk on Google Maps and give visual or voice alerts near high-risk segments; and
- to report performance only after testing on a documented, held-out dataset.

IV. BACKGROUND

A. Hotspots and risk factors

A hotspot is a segment, junction or grid cell where the frequency or severity of crashes is significantly higher than expected. Risk depends on road type, junction type, curvature, speed limit, lighting, weather, time of day, traffic volume and vehicle mix. Night-time travel, rain and fog, high two-wheeler share and the absence of medians or crossings are widely associated with higher risk.

B. Variational quantum classifiers

A qubit state is $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $|\alpha|^2 + |\beta|^2 = 1$, and n qubits span a space of dimension 2^n. A variational classifier encodes an input x into a quantum state with a feature map, applies a trainable parameterised circuit $U(x, \theta)$, and measures an observable O, typically Pauli-Z. The model output is

$$f(x; \theta) = \langle 0| U^{\dagger}(x, \theta) \cdot O \cdot U(x, \theta) |0\rangle \quad (1)$$

which lies in $[-1, 1]$ and is converted to a score or class probability. A classical optimiser minimises a loss such as cross-entropy. On noisy intermediate-scale quantum (NISQ) devices [13] shallow circuits are preferred, because deep circuits suffer from barren plateaus [14]. For gates of the form $\exp(-i\theta G/2)$ the gradient can be computed exactly by the parameter-shift rule [15]:

$$\partial f/\partial\theta = \tfrac{1}{2} [\, f(\theta + \pi/2) - f(\theta - \pi/2) \,] \quad (2)$$

which permits gradient-based optimisers such as Adam [16].

V. EXISTING AND PROPOSED SYSTEM

A. Existing systems

Hotspot identification is usually a statistical or spatial analysis of past crash counts, carried out after crashes have accumulated, while mainstream navigation optimises distance and time. Where machine learning is added, classical models are trained on structured data. Their drawbacks are:

- identification is reactive, since a location is flagged only after repeated crashes;
- accident data are heavily imbalanced, so accuracy can hide poor hotspot detection;
- output is often a bare label or count, not a ranked, map-based result;
- routing ignores accident risk, so drivers get no safer-route option;
- little work has examined quantum models on regional Indian road data.

B. Proposed system

SafeRoute Telangana is a sequence of data, prediction and application components (Fig. 1). Historical accident records are checked and prepared first. The VQNN then estimates a risk category for each road segment. The backend returns the prediction to the mobile application, which shows it on the map, compares route alternatives and checks whether the driver's position is near a flagged segment. Alerts inform the driver; they do not guarantee that a road is safe or unsafe. Its advantages are:

- risk is estimated from existing records, with no new sensors;
- a compact quantum circuit with few trainable parameters;
- imbalance is handled through class weights or training-only resampling;
- the shortest route and a lower-risk alternative are compared on distance, time and exposure;
- data, model, routing and application layers are separate and replaceable;
- visual and voice alerts near risk zones.

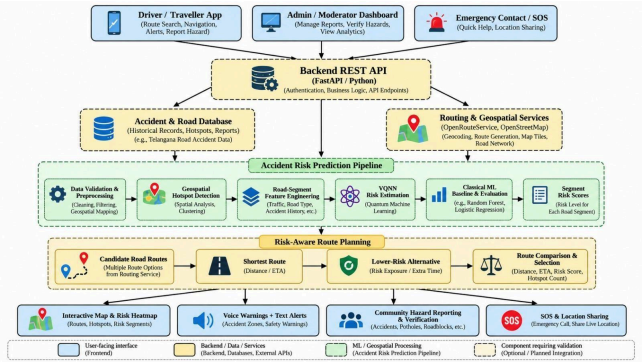


Fig. 1. Proposed SafeRoute Telangana architecture.

TABLE I. COMPARISON OF APPROACHES

Approach	Strength	Limitation
KDE / clustering [3]–[5]	Simple, interpretable hotspot maps	Reactive; no route or alert layer
Classical ML (LR, RF, SVM) [6], [7]	Fast, strong on tabular data	Needs resampling; no navigation use
Risk-aware routing [12]	Weighs safety against time	Needs a reliable risk model
Proposed VQNN + routing	Compact quantum classifier; risk-aware route comparison; alerts	Unvalidated; simulation cost; needs verified Telangana data

VI. DATASETS AND PREPROCESSING

A. Data sources

Five dataset sources were supplied for this project; they are summarised in Table II and must be treated as separate inputs until schema checks and an integration plan show which fields can be used together.

TABLE II. SUMMARY OF THE SELECTED DATA SOURCES

Source	Coverage / content	Role and limitation
Road Accidents Data–2022 (Kaggle) [24]	Casualty records; age, sex, class, severity, pedestrian details, area type, IMD decile	Casualty / severity analysis; Telangana geocoordinates not established
UK Road Safety (Kaggle) [25]	Accidents 2005–2017, vehicles 2004–2016, linked by Accident_Index	Method development; geography is the UK
Daily Climate Time Series (Kaggle) [26]	Delhi daily data, 1 Jan 2013–24 Apr 2017: meantemp, humidity, wind_speed, meanpressure	Weather example; Delhi values cannot stand in for Telangana
Climate Change Dataset (Kaggle) [27]	Synthetic global indicators, 2000–2024	Synthetic; not local evidence
Indian Road Analysis (Kaggle; data.gov.in) [28]	Road-network context, highway development, road length, investment	Background only unless the file has usable fields

Before training, the downloaded files must be inspected to confirm column names, record counts, licence terms, missing-value patterns, date coverage and whether location fields exist; file versions and download dates should be added to the final submission. The five sources do not automatically form one Telangana dataset. UK accident records, Delhi climate records, synthetic indicators, casualty-level data and road-network context describe different populations and units, and must not be joined merely because they share a year or topic. For Telangana prediction the project needs a defensible way to obtain or verify local accident labels and map them to road segments; if this cannot be done, the work must be described as a prototype or exploratory study, not a validated Telangana predictor.

B. Data preparation

Preparation begins with schema inspection, duplicate checks, missing-value analysis, date normalisation and review of categorical labels. Numerical variables may be scaled and categorical fields encoded. Imputation values and scaling parameters are learned from the training set only and then applied to validation and test data, to avoid leakage. Features that do not exist consistently across sources are not fabricated. Class imbalance is checked, and remedies such as class weights, stratified splits or oversampling [17] are applied to training data only.

C. Spatial and temporal alignment

Hotspot prediction needs a clear spatial unit, such as a road segment or grid cell, and a time window for the label.

Coordinates or reliable geographic identifiers must be mapped to the road network, and records without adequate location information are not assigned to Telangana segments by assumption. Weather is matched by location and time.

D. Risk target

Risk classes (low, moderate, high) can be defined from accident counts or rates within segments and time windows. Thresholds must be specified from the actual training data and documented in the final paper; until then these terms describe intended output categories, not validated classifications.

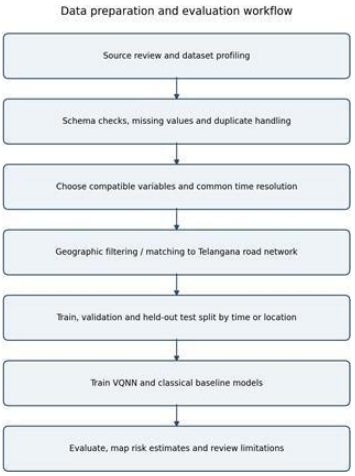


Fig. 2. Data preparation and evaluation workflow.

VII. VQNN MODEL AND TRAINING

The classifier follows a hybrid quantum–classical workflow (Fig. 3). After a compatible feature set is chosen, numerical features are scaled and, if required, reduced in dimension (for example by PCA) to fit the qubit budget. A quantum feature map encodes the features into a state, a parameterised circuit with rotation and entangling gates applies trainable operations, and measurements are converted into output scores. A classical optimiser updates the circuit parameters from the training loss. Qiskit [19] is used for circuit construction and simulation. The final circuit depth, number of qubits, feature map, optimiser, learning rate, batch size, stopping rule and number of training runs will be reported from the actual experiment; no values are stated here because none have been measured. The VQNN is compared with classical baselines using the same features, target definition and split.

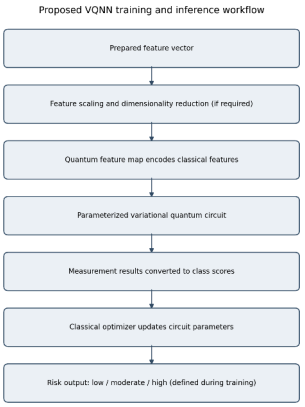


Fig. 3. Proposed VQNN training and inference workflow.

Prediction process

- Load the approved data files and verify their schemas.
- Remove duplicates and handle missing data using training-only statistics.
- Create a documented risk label for each eligible segment and time window.
- Split into training, validation and held-out test groups, separating time or geography where possible.
- Fit preprocessing and dimensionality reduction on training data only.
- Encode features and train the VQNN with a classical optimiser.
- Evaluate the VQNN and baselines on the same held-out test set.
- Serve the verified model through an API and display predictions only for supported segments.

VIII. RISK-AWARE ROUTE PLANNING

A routing engine (OpenRouteService or OpenStreetMap-based) returns candidate routes between origin and destination. Each route r is a sequence of segments s with length ℓ and VQNN risk score p . The risk exposure of a route is (TN), false positives (FP) and false negatives (FN): Accuracy = (TP + TN) / (TP + TN + FP + FN) (4)

Precision = TP / (TP + FP) (5) Recall = TP / (TP + FN) (6)

F1 = 2 · Precision · Recall / (Precision + Recall) (7)

Recall matters most here, since a missed hotspot costs more than a false alarm. AUC-ROC summarises class separation independent of the threshold, with 0.5 meaning chance level [18]. The VQNN is compared with logistic regression, random forest and SVM on identical partitions. If the aim is to predict future hotspots a time-based split is used, and a location-separated test can measure generalisation to unseen areas. Route-level measures are added: distance, estimated travel time, number of hotspots crossed and risk exposure R(r) for the shortest route versus the lower-risk alternative. Each metric is reported with a confidence interval or spread across repeated stratified folds. Final results must also state the records retained, class distribution, feature list, split proportions, simulator or hardware, software versions and baseline settings.

IX. APPLICATION ARCHITECTURE

A. Mobile application

The frontend is planned in React Native with Expo. It lets a user choose a destination, view a route and see risk markers or highlighted segments on Google Maps [23]. During navigation the app periodically compares the device location with predicted risk locations and, when a configured distance condition is

met, shows a visual notification and plays a voice alert. The alert radius, refresh interval and behaviour when GPS or network access is unavailable must be defined during implementation.

B. Backend and database

FastAPI [20] exposes prediction endpoints that return the risk category, score, segment identifier and timestamp. PostgreSQL
[22] can store user settings, prepared segment data, prediction logs and application records. Input validation, authentication, access control, error handling and logging are required before deployment, and the backend must not report a prediction as current when the model or data are outdated.

C. Map visualisation and alerts

The map distinguishes risk categories with a legend and identifies the time period of each prediction. The application states that predicted risk is an estimate based on available records, not a guarantee of future accidents. Google Maps routing and the project's risk score are separate components.

X. EXPERIMENTAL PLAN AND EVALUATION MEASURES

The VQNN is evaluated with accuracy, precision, recall, F1-score, confusion matrix and ROC-AUC where the target supports them, and macro-averaged precision, recall and F1 for imbalanced classes. With true positives (TP), true negatives
The VQNN model was compared with Logistic Regression, Random Forest, and SVM using the reported evaluation metrics. Random Forest and SVM achieved the highest accuracy of 74%, while VQNN achieved 50%. SVM obtained the highest F1-score of 0.61, whereas VQNN achieved 0.31. The results indicate that classical models performed better than VQNN on the synthetic prototype dataset. Further testing with verified Telangana accident data is required to validate the model's real-world performance.

XI. RESULTS AND COMPARISON

metrics. Random Forest and SVM achieved the highest accuracy of 74%, while VQNN achieved 50%. SVM obtained the highest F1-score of 0.61, whereas VQNN achieved 0.31. The results indicate that classical models performed better than VQNN on the synthetic prototype dataset. Further testing with verified Telangana accident data is required to validate the model's real-world performance.

TABLE III. MODEL COMPARISON ON THE HELD-OUT TEST SET (SYNTHETIC PROTOTYPE DATA)

Metric	VQNN	LR	RF	SVM
Accuracy	0.50	0.68	0.74	0.74
Precision	0.26	0.48	0.57	0.56
Recall	0.37	0.67	0.57	0.67
F1-score	0.31	0.56	0.57	0.61
AUC-ROC	0.48	0.75	0.73	0.75

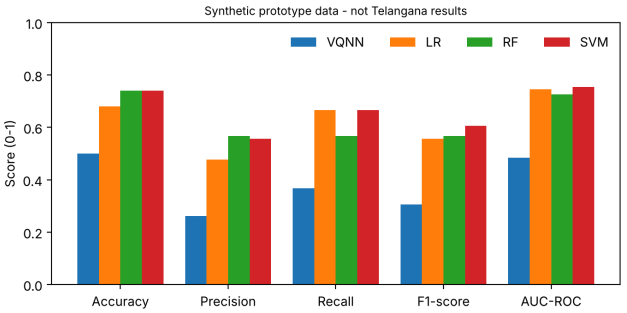


Fig. 4. Bar graph of model comparison (measured on synthetic prototype data; not Telangana results).

TABLE IV. ROUTE-LEVEL COMPARISON (SYNTHETIC DATA)

Measure	Shortest route	Lower-risk route
Distance (km)	16.38	16.72
Estimated time (min)	22.3	21.8
Hotspots crossed	7	6
Risk exposure R(r)	7.92	7.86

What the design is expected to show is a compact VQNN that is competitive with classical baselines with far fewer trainable parameters, and a lower-risk route that costs a modest amount of extra distance and time. Whether this holds is an empirical question, and a result in which classical models match or beat the VQNN will be reported just as plainly.

Table V lists the implementation status to be confirmed. The planned technologies are React Native/Expo, FastAPI, PostgreSQL, Google Maps Platform and Qiskit; the completion status of each component has not been independently confirmed and will be updated from the working implementation.

TABLE V. PROJECT COMPONENTS AND STATUS TO VERIFY

Component	Planned technology	Status
Mobile navigation	React Native / Expo [21]	Confirm screens, device tests
Prediction API	FastAPI [20]	Confirm endpoints, model link
Data storage	PostgreSQL [22]	Confirm schema, persistence
Quantum model	Qiskit / VQNN [19]	Confirm training, saved model
Map and alerts	Google Maps [23], GPS, voice	Confirm hotspot layer, alert tests
Evaluation	Python ML metrics	Add results after held-out testing

XII. CONCLUSION

This paper presents the design of SafeRoute Telangana, a proposed accident hotspot prediction and risk-aware routing framework that combines a Variational Quantum Neural Network with a mobile navigation application. The system

prepares suitable accident features, estimates segment risk, returns predictions through a backend service, compares the shortest route with a lower-risk alternative, and displays the output on Google Maps with visual and voice alerts. Because the five supplied sources differ in location, time and origin, they cannot be merged without careful validation. The VQNN must be evaluated against classical baselines on a documented test set before any performance claim is made. Future work will focus on compatible data, Telangana-specific validation, model evaluation and complete application testing.

XIII. FUTURE WORK

The next stage is to inspect the downloaded files and decide which attributes can be used without unsupported links between regions, and to identify a Telangana-specific accident source with reliable dates and geographic references. Segment-level labels can then be defined, the VQNN trained and compared with classical models, and the API and mobile application tested end to end with held-out examples. Further work includes better map matching, verified local weather and traffic feeds, alert-timing tests, inference-latency measurement, quantum kernels, runs on real quantum hardware, and evaluation under realistic network and GPS conditions.

References

[1] World Health Organization, “Global status report on road safety 2023,” Geneva, 2023.

[2] Ministry of Road Transport and Highways, Government of India, “Road accidents in India,” annual report.

[3] T. K. Anderson, “Kernel density estimation and K-means clustering to profile road accident hazard in the metropolitan area of London,” Accident Analysis & Prevention, vol. 41, no. 3, pp. 359–364, 2009.

[4] V. Prasannakumar, H. Vijith, R. Charutha, and N. Geetha, “Spatio-temporal clustering of road accidents: GIS based analysis and assessment,” Procedia – Social and Behavioral Sciences, vol. 21,pp. 317–325, 2011.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in Proc. KDD, 1996, pp. 226–231.

[6] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[7] C. Cortes and V. Vapnik, “Support-vector networks,” Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[8] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, “Circuit-centric quantum classifiers,” Physical Review A, vol. 101, 032308, 2020.

[9] M. Cerezo et al., “Variational quantum algorithms,” Nature Reviews Physics, vol. 3, pp. 625–644, 2021.

[10] V. Havlíček et al., “Supervised learning with quantum-enhanced feature spaces,” Nature, vol. 567, pp. 209–212, 2019.

[11] M. Schuld and F. Petruccione, Supervised Learning with Quantum Computers. Cham, Switzerland: Springer, 2018.

[12] J. Krumm and E. Horvitz, "Risk-aware planning: Methods and case study for safer driving routes," in Proc. AAAI/IAAI, 2017.

[13] J. Preskill, "Quantum computing in the NISQ era and beyond," Quantum, vol. 2, p. 79, 2018.

[14] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," Nature Communications, vol. 9, 4812, 2018.

[15] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," Physical Review A, vol. 99, 032331, 2019.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[18] T. Fawcett, "An introduction to ROC analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.

[19] A. Javadi-Abhari et al., "Quantum computing with Qiskit," arXiv:2405.08810, 2024. [Online]. Available: https://qiskit.org/

[20] FastAPI documentation. [Online]. Available: https://fastapi.tiangolo.com/ (accessed: date to be added).

[21] React Native documentation. [Online]. Available: https://reactnative.dev/ (accessed: date to be added).

[22] PostgreSQL Global Development Group, "PostgreSQL documentation." [Online]. Available: https://www.postgresql.org/docs/ (accessed: date to be added).

[23] Google, "Google Maps Platform documentation." [Online]. Available: https://developers.google.com/maps/documentation (accessed: date to be added).

[24] Juhi Bhojani, "Road Accidents Data–2022," Kaggle dataset; URL and access date to be added.

[25] "UK Road Safety: Traffic Accidents and Vehicles," Kaggle dataset (accidents 2005–2017, vehicles 2004–2016); URL and access date to be added.

[26] Sumanth V Rao, "Daily Climate Time Series Data," Kaggle dataset, Delhi, 2013–2017 (source: Weather Underground API); URL and access date to be added.

[27] B. Mohit, "Climate Change Dataset: Global Climate Change Indicators, 2000–2024," Kaggle dataset (synthetically generated); URL and access date to be added.

[28] "Indian Road Analysis," Kaggle dataset (cites data.gov.in); URL and access date to be added.