Phase-1

Enhancing road safety with ai driven traffic accident analysis and prediction

Name: K Mathesh Kumar

Register Number: 422623104050

Institution: University college of engineering panruti

Department: computer science and engineering

Date of Submission: 28/04/2025

Problem Statement

Road traffic accidents remain a **major global safety challenge**. The World Health Organization estimates over **1.2 million people die in road crashes each year**, with tens of millions more injured.Notably, crashes are the *leading cause of death for children and young adults* worldwide These accidents inflict enormous human and economic losses. In many regions, the situation is worsening: for example, India reported **461,312** road accidents in 2022 ,resulting in **168,491** fatalities. Traditional reactive approaches are not sufficient to prevent tragedies. There is an urgent need for **proactive, data-driven solutions**. Modern AI and machine learning can analyze historical crash data and associated factors to **identify high-risk road segments and predict crash likelihood** before accidents happen.By forecasting when and where accidents are most likely, authorities can intervene in advance, potentially saving lives.

Objectives of the Project

- **Identify Accident Hotspots:** Analyze spatial crash data to find zones with unusually high accident rates .
- **Predict Accident Likelihood/Severity:** Build predictive models that estimate the probability or severity of future accidents given road, traffic and environmental conditions.
- **Analyze Risk Factors:** Determine which factors most strongly influence crash risk or severity.
- **Support Decision-Making:** Provide insights to policymakers and traffic engineers for targeted interventions .
- **Visualization and Reporting:** Create dashboards and visual tools to communicate findings clearly to stakeholders.
- **Future Extensions :**Lay groundwork for real-time alerts or integration with traffic management .

Scope of the Project

Included in Scope:

- **Accident Types:** The project focuses on **road traffic collisions** .It may include analysis of accident *frequency* and *severity*.

- **Data Scope:** Uses *historical crash records* along with related data for analysis.
- **Geographic Focus:** The methodology is general, but an initial case study might focus on a specific region or country . Methods can be applied to any region with available data.
- **Models:** Supervised ML models for classification/regression.Also unsupervised techniques in exploratory analysis.
- **Output:** Generates predictions of crash risk and visualizations to assist planners.

Excluded from Scope:

- **Other Transport Modes:** Air, rail, or maritime accidents are *not* covered. Focus is strictly on **road** traffic.
- **Real-Time Systems:** The initial project uses *static historical data*. Live, real-time streaming.
- **On-Vehicle Technologies:** The project does *not* develop in-vehicle safety systems.
- **Clinical/Emergency Response:** While the project may inform emergency services planning indirectly, medical response protocols or trauma care are not addressed.

Data Sources

- **Kaggle Datasets:** Public repositories like Kaggle host large curated crash datasets. For example, the *"US-Accidents (2016–2023)"* dataset contains millions of traffic incidents across the US, covering location, time, weather, and severity Other examples include global road accident collections and region-specific datasets .
- **Government Open Data:** Official agencies publish crash statistics and raw data. In the US, the Department of Transportation/NHTSA provides the Fatality Analysis Reporting System (FARS) data via APIs. In India, the Ministry of Road Transport & Highways (MoRTH) releases annual accident reports and state-wise data. European countries and others often have open portals for road safety data.
- **City/State Open Portals:** Many cities publish detailed crash records. For example, New York City's OpenData portal offers a *"Motor Vehicle Collisions"* dataset containing every reported crash in the city. Such local datasets include precise geolocations and circumstances.

- **Traffic Authority APIs:** Real-time or historical traffic data can be obtained via APIs. NHTSA's Crash API provides queried access to national crash records. Additionally, some regions have traffic sensor and incident APIs .
- **Supplementary Data:** To enrich analysis, additional sources may be used: road network maps , weather data APIs and demographic/land use data.

Examples of cited sources: The NHTSA Crash API page notes that FARS data from 2010 onward is accessible via their web service. Likewise, NYC's data catalog describes a comprehensive table of police-reported collisions. These show how credible agencies provide raw crash data for analysis.

High-Level Methodology

- **Data Collection:** Gather accident data from the above sources. This may involve downloading CSVs querying open-data APIs ,or scraping government reports. Integrate heterogeneous data into a unified dataset.
- **Data Cleaning:** Address missing or inconsistent entries. Remove duplicates and irrelevant records. Standardize formats. Handle outliers or erroneous entries. Ensuring data quality is crucial before analysis.
- **Exploratory Data Analysis (EDA):** Perform statistical summaries and visualizations to understand patterns. Examples: distribution of accidents by hour of day, day of week, or season; correlation between weather conditions and crash counts; scatter or heatmap of crash locations to spot hotspots. EDA may reveal surprising trends.
- **Feature Engineering:** Create informative input features for modeling. For instance, derive time-related features ,spatial features and encode categorical factors using one-hot encoding. Incorporate external features like traffic density or temperature at crash time. Good feature design can significantly improve predictive accuracy.
- **Model Building:** Use machine learning to predict accident risk or severity. Depending on the goal, this could be a binary classification or multi-class. Candidate algorithms include logistic regression, decision trees/random forests, gradient-boosting, support vector machines, or neural networks. For spatial data, one could even explore convolutional or graph-based models. The process involves splitting data into training/test sets, possibly using cross-validation.
- **Model Evaluation:** Assess performance using appropriate metrics. Because crash data are often imbalanced ,metrics beyond simple accuracy are needed. Precision, recall, F1-score, and area under the ROC curve (AUC)

- **Visualization & Interpretation:** Present the results in an understandable way. Plot predictive risk on maps, time series graphs of accident frequency, and bar charts of factor importance. Interactive dashboards allow users to filter by region or time. Interpreting the model helps explain which factors are driving risk. These visualizations help decision-makers grasp the insights.
- **Deployment (Dashboard/Web App):** Develop a simple interface for stakeholders. For example, build a web dashboard that displays heatmaps of high-risk roads and allows querying predictions for specific areas or conditions. This makes the analysis accessible to non-technical users. The deployed system could also update periodically as new data arrives.
- **Future Extension – Real-Time Data:** In subsequent work, the system can be extended to use live data feeds. For instance, ingesting streaming telemetry from traffic cameras, loop detectors, or connected vehicles would allow **real-time risk prediction**. A continuously updated model could alert traffic managers to emerging danger spots as they happen. This real-time extension would turn a historical analysis into a proactive, live monitoring tool.

Tools and Technologies

- **Programming Language:** Python .
- **Development Environment:** Jupyter Notebook or Google colab.
- **Data Libraries:** Pandas ,NumPy .These handle loading and preprocessing of datasets.
- **Machine Learning:** Scikit-learn , TensorFlow/Keras. These libraries enable model training and evaluation.
- **Data Visualization:** Matplotlib and Seaborn, Plotly Folium or GeoPandas could be used for mapping accident locations.
- **Deployment Framework:** Flask, Streamlit, or FastAPI. These lightweight Python frameworks help create a web app or dashboard to showcase the predictive model and visualizations.
- **Optional Tools:** Git for version control; Google Colab or cloud services for computing if data is large. For GIS data, libraries like Shapely or Geopandas could be used.

These tools form a typical data science stack in Python. Pandas and NumPy handle the raw data, scikit-learn/TensorFlow build models, and Matplotlib/Seaborn/Plotly provide visual insights.

Streamlit or Flask can turn the final analysis into an interactive web interface for decision-makers.

Team Members and Roles :

1. **K Mathesh Kumar** - Project Lead : Oversees the entire project, coordinates tasks, and ensures deadlines are met.

2. **G Harini** : Handles data collection, cleaning, and exploratory data analysis.

3. **V Malini** : Focuses on building and training the AI models for accident prediction.

4. **T Arun Kumar** : Manages data storage, preprocessing, and feature engineering.

5. **F Flora** : Works on deploying the AI model into a web application and ensures it runs smoothly.