**Phase-3**

**Student Name: V.Sujithra**

**Register Number: 611423106106**

**Institution:** Mahendra Engineering College For Women

**Department:** BE.Electronics And Communication Engineering

**Date of Submission:** 10-05-2025

**Github Repository Link: git@github.com:sujithra-project/EBPL-Sujithra.git**

### **1. Problem Statement**

Develop an AI-powered system that analyzes historical traffic data and predicts

high-risk accident locations, enabling proactive measures to enhance road safety.

Goals

*1.* ***Reduce Traffic Accident*s:** Enable proactive measures to prevent accidents.

2. *I****mprove Road Safety*:** Enhance safety for drivers, pedestrians, and cyclists.

3. ***Inform Policy Decisions*:** Provide insights for policymakers to develop targeted safety initiatives.

***Impact Potential***

1.***Lives Saved*:** Reduce traffic-related fatalities and injuries.

2. ***Economic Benefits*:** Minimize economic losses due to accidents.

3.***Improved Quality of Lif*e:** Enhance overall quality of life for road users*.*

### **Abstract**

This project utilizes AI-driven analysis and prediction to identify high-risk accident

locations, enabling proactive measures to enhance road safety, reduce traffic

related fatalities and injuries, and inform data-driven policy decisions for safer

transportation infrastructure.

1**. Real-time Analysis:** Enables real-time traffic monitoring and accident

prediction.

2*.* **Data-Driven Insights**: Provides actionable insights for policymakers and

stakeholders.

3. I**mproved Emergency Response:** Enhances emergency response times and

effectiveness.

4. **Reduced Economic Burden:** Minimizes economic losses due to accidents.

5. **Scalability:** Can be applied to various regions and cities.

### **3. System Requirements**

1. **Servers:** High-performance servers for data processing and model

training.

2. **Storage:** Ample storage for historical traffic data and model outputs.

3. **Networking:** Reliable network infrastructure for data transmission.

**Software Requirements**

1. **Programming Languages:** Python, R, or other languages suitable for

machine learning.

2. **Machine Learning Frameworks:** TensorFlow, PyTorch, or Scikit-learn.

3. **Data Analysis Tools:** Pandas, NumPy, Matplotlib, Seaborn.

4. **Database Management:** Relational databases (e.g., MySQL) or NoSQL

databases (e.g., MongoDB).

**Data Requirements**

1. **Historical Secondary traffic Data:** Accidents, traffic volume, speed, road

conditions.

1. **Real-time Data:** Traffic updates, weather, road closures.

**Functional Requirements**

1. **Data Ingestion**: Collect and preprocess traffic data.

2. **Model Training:** Train machine learning models for accident prediction.

3. **Prediction:** Generate predictions and risk scores.

4. **Visualization:** Display results and insights.

**Non-Functional Requirements**

1. **Scalability:** Handle large datasets and high traffic volumes.

2. **Accuracy:** Ensure accurate predictions and insights.

3. **Security:** Protect sensitive data and ensure system integrity

### 

### **4. Objectives**

**Primary Objectives**

1. **Improve Road Safety:** Reduce traffic-related fatalities and injuries.

2. **Predict High-Risk Locations:** Identify areas prone to accidents. 3. **Inform Policy Decisions:** Provide data-driven insights for policymakers.

**Objectives**

1. **Optimize Emergency Response:** Enhance response times and effectiveness.

2. **Minimize Economic Losses:** Reduce costs associated with accidents.

3. **Improve Traffic Management:** Optimize traffic flow and reduce congestion.

**Technical Objectives**

1. **Develop Accurate Models:** Create reliable machine learning models for

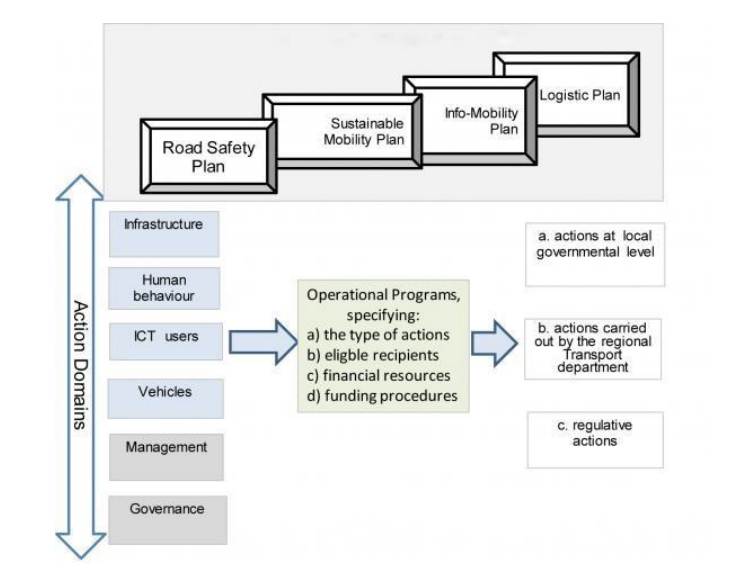
accident prediction.

2. **Integrate Real-Time Data:** Incorporate real-time traffic data for improved

predictions.

3. **Visualize Insights:** Display results and insights effectiveness.

**5. Flowchart of Project Workflow**



### **6. Dataset Description**

**Key Features**

1. **Accident Location:** Geographic coordinates (latitude, longitude).

2. **Time and Date:** Timestamp of accident occurrence.

3. **Road Conditions:** Weather, lighting, and road surface conditions.

4. **Traffic Volume:** Traffic density and speed data.

5. **Accident Severity:** Severity of accident (e.g., fatal, injury, property

damage).

**Data Sources**

1. **Government Records**: Official accident reports.2. **Traffic Sensors:** Real-time traffic data.

3. **Weather Services:** Historical weather data.

### 

### **Data Preprocessing**

**Preprocessing Steps**

1. **Data Cleaning:** Handle missing values, outliers, and inconsistencies.

2. **Data Transformation**: Convert data types, normalize/scale features.

3. **Feature Engineering:** Extract relevant features (e.g., time of day, day of

week).

4. **Data Integration:** Combine data from multiple sources.

5. **Data Splitting:** Split data into training, validation, and testing sets.

**Techniques**

1. **Handling Missing Values**: Imputation, interpolation, or removal.

2. **Feature Scaling:** Normalization, standardization, or log transformation.

3. **Encoding Categorical Variables:** One-hot encoding, label encoding.

**Goals**

1. **Improve Model Performance:** Enhance accuracy and reliability.

2. **Reduce Noise:** Minimize impact of irrelevant or erroneous data.

### 

### **Exploratory Data Analysis (EDA)**

1. **Understand Data Distribution:** Examine variable distributions and

relationships.

2. **Identify Patterns:** Discover trends, correlations, and insights.3. **Inform Modeling:** Guide feature selection, engineering, and model

choice.

**EDA Steps**

1. **Descriptive Statistics:** Calculate means, medians, modes, and standard

deviations.

2. **Data Visualization:** Use plots (e.g., histograms, scatter plots, heatmaps)

to illustrate relationships.

3. **Correlation Analysis:** Examine relationships between variables.

4. **Geospatial Analysis:** Visualize accident locations and patterns.

**Potential Insights**

**1. High-Risk Areas:** Identify locations with increased accident frequency.

**2. Temporal Patterns:** Discover time-of-day, day-of-week, or seasonal

trends.

**3**. **Correlations:** Identify relationships between variables (e.g., weather,

traffic volume).

### **9. Feature Engineering**

**Feature Engineering Objectives**

**Extract Relevant Features:**

Create new features that capture meaningful information.

**Improve Model Performance:**

Enhance accuracy and reliability.

**Time-Based Features:**

Time of day, Day of week, Month, Season

**Location-Based Features:**

Latitude and longitude

Road type (e.g., highway, urban)

Proximity to intersections or landmarks

**Weather Features:**

Temperature

Precipitation

Visibility

**Traffic Features:**

Traffic volume

Speed

Congestion level 5.

**Derived Features:**

Accident density (accidents per km²)

Road curvature or gradient

**Techniques**

1. **Feature Extraction:** Extract relevant information from existing features.

2. **Feature Transformation**: Apply transformations (e.g., normalization,

encoding).

3. **Feature Creation:** Generate new features through calculations or

combinations.

### 

### **10.Model Building**

**Model Objectives**

1. **Predict Accident Risk:** Identify high-risk locations and times.

2. **Improve Accuracy:** Develop reliable models for accident prediction.

**Potential Models**

**Machine Learning Models**:

Random Forest

Gradient Boosting

Support Vector Machines (SVM)

**Deep Learning Models:**

Convolutional Neural Networks (CNN) for spatial analysis

Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM)

networks for temporal analysis

**Statistical Models:**

Logistic Regression

Poisson

**Model Evaluation Metrics**

1. **Accuracy:** Measure model accuracy.

2. **Precision:** Evaluate precision of positive predictions.

3. **Recall:** Assess recall of actual accidents.

4. **F1-Score:** Balance precision and recall.

5. **Area Under the ROC Curve (AUC-ROC):** Evaluate model

performance.

**Model Selection**

**Cross-Validation:** Use techniques like k-fold cross-validation to evaluate

model performance.

**Hyperparameter Tuning:** Optimize model hyperparameters for improved

performance.

### 

### **11. Model Evaluation**

**Model Evaluation Objectives**

1. **Assess Performance:** Evaluate model accuracy and reliability.

2. **Compare Models:** Determine the best-performing model.

**Evaluation Metrics**

1. **Accuracy:** Measure model accuracy.

2. **Precision:** Evaluate precision of positive predictions.

3.**F1-Score:** Balance precision and recall.

4**.Area Under the ROC Curve (AUC-ROC):** Evaluate model performance.

5**.Mean Absolute Error (MAE):** Measure average error.

**Evaluation Techniques**

1. Cross-Validation: Use k-fold cross-validation.

2. Confusion Matrix: Analyze true positives, false positives, true negatives,

and false negatives.

1. ROC Curve: Plot true positive rate vs. false positive rate.

**Model Comparison**

1. Compare Metrics: Evaluate models based on chosen metrics.

2. Statistical Tests: Use tests like paired t-tests or Wilcoxon signed-rank tests.

### **12. Deployment**

*Model Accuracy: 0.50 Accident Likelihood:*

**13. Source code**

*Import necessary libraries*

*import pandas as pd*

*from sklearn.model\_selection import train\_test\_split*

*from sklearn.ensemble import RandomForestClassifier*

*from sklearn.metrics import accuracy\_score*

*Sample dataset (replace with actual data)*

*data = {*

*'Speed': [60, 70, 80, 90, 60, 70, 80, 90],*

*'Weather': [0, 0, 1, 1, 0, 0, 1, 1], # 0: Clear, 1: Rainy*

*'Road\_Type': [0, 1, 0, 1, 0, 1, 0, 1], # 0: Highway, 1: Urban*

*'Accident': [0, 1, 1, 1, 0, 0, 1, 1] # 0: No accident, 1: Accident*

*}*

*df = pd.DataFrame(data)*

*Define features (X) and target variable (y)*

*X = df[['Speed', 'Weather', 'Road\_Type']]*

*y = df['Accident']*

*Split data into training and testing sets*

*X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)*

*Train a random forest classifier*

*model = RandomForestClassifier()*

*model.fit(X\_train, y\_train)*

*Make predictions*

*y\_pred = model.predict(X\_test)*

*Evaluate model performance*

*accuracy = accuracy\_score(y\_test, y\_pred)*

*print(f"Model Accuracy: {accuracy:.2f}")*

*Use the model to predict accident likelihood for new data*

*new\_data = pd.DataFrame({'Speed': [75], 'Weather': [0], 'Road\_Type': [1]})*

*prediction = model.predict(new\_data)*

*print(f"Accident Likelihood: {prediction[0]}")*

**14. Future scope**

**Future Directions**

1. Integration with Smart City Infrastructure: Incorporate with smart city systems

for real-time traffic management.

2. Expansion to Other Modes of Transportation: Apply the model to predict

accidents for other modes, such as cycling or pedestrian traffic.

3. Incorporating Additional Data Sources: Utilize additional data sources, such as

social media or IoT devices.

4. Development of Real-Time Alert Systems: Create systems to alert authorities and drivers of potential accidents.

1. Global Applicability: Adapt the model for use in different cities or countries.

**Potential Impact**

1. Reduced Accidents: Decrease traffic-related fatalities and injuries.
2. Improved Traffic Management: Optimize traffic flow and reduce congestion.

3. Enhanced Public Safety: Inform policy decisions and infrastructure development.

1. **Team Members and Roles**

**V . Sujithra –** Data Cleaning and EDA

**S . Priyadharshini –** Feature Engineering and models development

**N . Rani –** Documentation and reporting