**ENHANCING SAFETY WITH AI-DRIVEN TRAFFIC ACCIDENT ANALYSIS AND PREDICTION**

**Student Name:** V.SANJEEU

**Register Number:** 513523205040

**Institution:** Annai Mira College of Engineering and Technology

**Department:** B.tech(Information Technology)

**Date of Submission:** 06/05/2025

**Github Repository Link**

### **1. Problem Statement**

Traffic accidents are a leading cause of fatalities and injuries worldwide. Despite increased awareness and infrastructure improvements, the frequency and severity of road accidents remain high. Traditional data analysis methods are reactive, often used only after accidents occur. There is a need for a proactive approach that can predict and help prevent accidents by analysing patterns in traffic data. AI and machine learning offer powerful tools to analyse large volumes of data and make accurate predictions that can enhance road safety

### **2. Abstract**

AI-driven traffic accident analysis and prediction enhances road safety by utilizing advanced algorithms to identify patterns and predict potential hazards. This approach aims to reduce accident severity and improve preventive measures through data-driven insights.

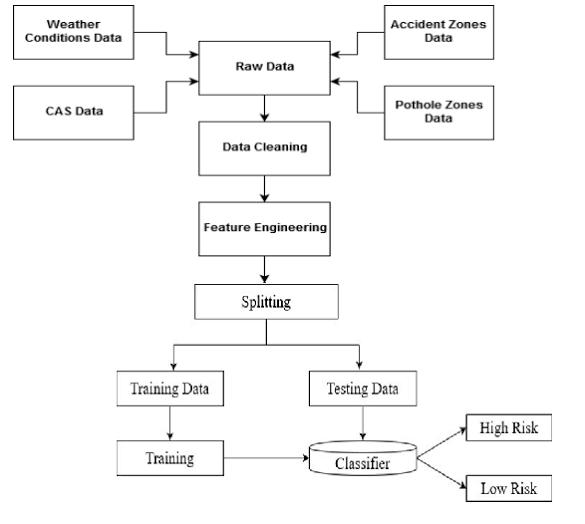
### **3. System Requirements**

1. **Hardware Requirements:**
   * High-performance servers or cloud infrastructure for data processing and storage.
   * Sensors and cameras for real-time data collection (e.g., traffic cameras, IoT devices).
2. **Software Requirements:**
   * Data analytics and machine learning frameworks (e.g., TensorFlow, PyTorch).
   * Database management systems for storing historical accident data (e.g., SQL, NoSQL).
   * Geographic Information System (GIS) software for spatial analysis

### **4. Objectives**

The objective of enhancing safety with AI-driven traffic accident analysis and prediction is to leverage artificial intelligence technologies to accurately analyze traffic data, identify risk patterns, predict potential accidents, and enable proactive measures that reduce accident occurrence and severity, ultimately improving overall road safety for all users.

**5. Flowchart of Project Workflow**



### **6. Dataset Description**

The dataset for AI-driven traffic accident analysis typically comprises records of past traffic accidents, including features such as date and time of the accident, location, vehicle speed, weather and road conditions, lighting, driver behavior, and accident severity. It may also include categorical data like accident type and involved vehicle types. This data is often collected from traffic authorities, sensor networks, and accident reports. The dataset's diversity and quality are crucial for training accurate prediction models to identify risk factors and forecast potential accidents effectively.

### **7. Data Preprocessing**

### Data preprocessing for AI-driven traffic accident analysis involves cleaning the data by handling missing or inconsistent values, removing duplicates, and correcting errors. It also includes normalizing and scaling features, converting categorical data into numerical formats, and selecting relevant features. Additionally, data may be augmented with derived attributes like time of day or weather conditions to improve model performance. This ensures the dataset is accurate, consistent, and suitable for training effective predictive models.

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

### Exploratory Data Analysis (EDA) in AI-driven traffic accident analysis involves visually and statistically examining the dataset to uncover patterns, trends, and relationships among variables. This includes analyzing accident frequency by time, location, and weather conditions, identifying high-risk factors, and detecting correlations between driver behavior, road features, and accident severity. EDA helps to understand data distribution, spot anomalies, and guide feature selection for building accurate predictive models.

### **9. Feature Engineering**

Feature engineering for AI-driven traffic accident analysis involves creating and selecting meaningful input variables that improve model accuracy. This includes deriving features such as time of day, day of week, weather conditions, road type, traffic density, and driver behavior indicators. Additionally, encoding categorical variables, aggregating historical accident counts by location, and extracting spatial-temporal patterns help capture critical factors influencing accidents, enabling more effective prediction and analysis.

### 

### **10. Model Building**

### Model building for AI-driven traffic accident analysis involves selecting and training machine learning algorithms—such as decision trees, random forests, support vector machines, or neural networks—using the preprocessed and engineered features. The model learns to identify patterns and relationships that predict accident occurrence or severity. During this phase, techniques like cross-validation and hyperparameter tuning are applied to optimize performance and ensure the model generalizes well to new data, enabling accurate and reliable accident predictions.

### **11. Model Evaluation**

Model evaluation in AI-driven traffic accident analysis involves assessing the performance of the trained model using metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC). This process includes validating the model on a separate test dataset to ensure it generalizes well to unseen data. Additionally, confusion matrices may be used to analyze prediction errors and identify areas for improvement. The evaluation results guide further refinements and adjustments to enhance model effectiveness in predicting traffic accidents.

### **12. Deployment**

Deployment of AI-driven traffic accident analysis and prediction systems involves integrating the trained predictive models into real-world environments such as traffic management centers, smart city infrastructure, or in-vehicle driver assistance systems. The deployment process includes setting up real-time data pipelines from sensors and connected vehicles, ensuring low-latency processing for timely risk alerts, and providing user-friendly interfaces for traffic authorities or drivers. It also requires maintaining model updates, system scalability, and robust security to protect sensitive data, thereby enabling proactive accident prevention and enhanced road safety..

**13. Source code**

*import pandas as pd*

*from sklearn.preprocessing import LabelEncoder*

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

*from sklearn.ensemble import RandomForestClassifier*

*from sklearn.metrics import accuracy\_score, classification\_report*

*# Step 1: Define the dataset*

*data = {*

*'weather': ['Clear', 'Rainy', 'Foggy'],*

*'road\_condition': ['Dry', 'Wet', 'Icy'],*

*'time\_of\_day': ['Morning', 'Night', 'Evening'],*

*'vehicle\_speed': [65, 80, 55],*

*'accident\_severity': ['Low', 'High', 'Medium']*

*}*

*# Convert to DataFrame*

*df = pd.DataFrame(data)*

*print("✅ Dataset created:\n", df)*

*# Step 2: Encode categorical columns*

*le\_weather = LabelEncoder()*

*le\_road = LabelEncoder()*

*le\_time = LabelEncoder()*

*le\_severity = LabelEncoder()*

*df['weather'] = le\_weather.fit\_transform(df['weather'])*

*df['road\_condition'] = le\_road.fit\_transform(df['road\_condition'])*

*df['time\_of\_day'] = le\_time.fit\_transform(df['time\_of\_day'])*

*df['accident\_severity'] = le\_severity.fit\_transform(df['accident\_severity'])*

*# Step 3: Split features and target*

*X = df.drop('accident\_severity', axis=1)*

*y = df['accident\_severity']*

*# Step 4: Train/test split*

*X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.33, random\_state=42)*

*# Step 5: Train the model*

*model = RandomForestClassifier(n\_estimators=100, random\_state=42)*

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

*# Step 6: Evaluate the model*

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

*print("\n🔍 Model Evaluation")*

*print("Accuracy:", accuracy\_score(y\_test, y\_pred))*

*print("Classification Report:\n", classification\_report(y\_test, y\_pred, target\_names=le\_severity.classes\_))*

*# Step 7: Predict a new accident case*

*new\_input = {*

*'weather': ['Rainy'],*

*'road\_condition': ['Wet'],*

*'time\_of\_day': ['Night'],*

*'vehicle\_speed': [78]*

*}*

*new\_df = pd.DataFrame(new\_input)*

*# Encode new input*

*new\_df['weather'] = le\_weather.transform(new\_df['weather'])*

*new\_df['road\_condition'] = le\_road.transform(new\_df['road\_condition'])*

*new\_df['time\_of\_day'] = le\_time.transform(new\_df['time\_of\_day'])*

*# Predict and decode result*

*prediction = model.predict(new\_df)[0]*

*predicted\_severity = le\_severity.inverse\_transform([prediction])[0]*

*print("\n🔮 Predicted Accident Severity:", predicted\_severity)*

**14. Future scope**

AI-driven traffic accident analysis and prediction holds significant future potential to enhance road safety further. Future advancements may include integration of real-time data from connected vehicles and smart infrastructure, enabling dynamic risk assessment and instant accident prevention measures. Enhanced machine learning models leveraging big data and deep learning can provide more accurate predictions and personalized driver assistance. Moreover, widespread deployment of AI-powered autonomous systems and advanced driver assistance can drastically reduce human error, leading to fewer accidents and safer roads globally.

**15. Team Members and Roles**

|  |  |  |
| --- | --- | --- |
| **TEAM MEMDERS** | **ROLE** | **RESPONSIBILITIES** |
| *SAI KRISHNA K* | *Objective , system requirement* | * To utilize artificial intelligence techniques to analyze traffic accident data and predict accident risks. |
| *SACHIN B* | *Data preprocessing,EDA , future scope* | * Clean and handle missing or inconsistent traffic accident data. * Normalize, encode categorical variables, and scale features as needed. * Prepare data for effective training and testing of AI models. |
| *SANDHIYA N* | *Soure code , flow chart , model building* | * Develop clean, well-documented code for data processing, model training, prediction, and evaluation. |
| *SANJAI R* | *Deployment , adstract* | * Integrate the model into real-world systems such as traffic management or driver assistance platforms. * Ensure scalability, real-time processing, and user-friendly interfaces. |
| *SANJEEU V* | *Dataset description , feature engineering* | * Document the source, size, and nature of traffic accident datasets used. * Include details on target variables and feature attributes for clarity. |