

Project Proposal
<b>Project Name: AI-Based Traffic Congestion Prediction and Visualization</b>
<p>PROJECT DESCRIPTION (&lt; 150 words)</p> <p>This project aims to predict and visualize real-time traffic congestion by combining historical traffic data with social media (Twitter) reports. A graph-based model will represent intersections as nodes and roads as edges, allowing dynamic updates of traffic weights. Deep learning methods (e.g., LSTM or CNN) will be applied to analyze both numerical traffic data and textual traffic incident reports. Results will be presented through an interactive heatmap, enabling users and city planners to quickly identify congestion hotspots and optimize routes. By merging data-driven insights with user-friendly visualizations, this system can help reduce commute times and carbon emissions.</p>
<p>SOLUTION (Deliverables). Write a bullet point list of what you expect your software will achieve. I do not hold you to this list for your end-product.</p> <ul style="list-style-type: none"><li>• Real-time traffic congestion forecasting using deep learning models</li><li>• Integration of Twitter data (NLP analysis) for incident detection</li><li>• Graph-based road network representation with dynamic weights</li><li>• Interactive heatmap and route optimization recommendations</li><li>• Dashboard for historical trend analysis and performance tracking</li></ul>
<p>DATASETS (if any used).</p> <ul style="list-style-type: none"><li>• Historical and real-time traffic data from public or private APIs</li><li>• Twitter feeds with location-based traffic/incident reports</li></ul>
<p>Expected Tools (Cloud DBs, Hardware, &amp; Python Libraries to be used).</p> <ul style="list-style-type: none"><li>• Cloud database services (e.g., AWS RDS, Google Cloud SQL)</li><li>• Python libraries:<ul style="list-style-type: none"><li>• <b>NetworkX</b> for graph modeling</li><li>• <b>TensorFlow/Keras</b> or <b>PyTorch</b> for deep learning</li><li>• <b>Pandas</b>, <b>NumPy</b>, <b>scikit-learn</b> for data processing</li><li>• <b>spaCy</b> or <b>NLTK</b> for NLP on tweets</li><li>• <b>Bokeh</b> or <b>Plotly</b> for interactive visualizations</li></ul></li></ul> <p>GPU-enabled hardware for faster training (if available)</p>

Rough Timeline (Fill in the columns):

**Weeks Project Task Timeline**

Week	Tasks/Goals	Deliverables
1	Project Setup & Data Acquisition	<ul style="list-style-type: none"><li>- Define project objectives &amp; success metrics</li><li>- Acquire/setup APIs for traffic and Twitter data</li><li>- Establish development environment</li></ul>
2	Data Exploration, Preprocessing & Baseline ML	<ul style="list-style-type: none"><li>- Perform EDA on collected data (traffic &amp; tweets)</li><li>- Clean and preprocess data</li><li>- Implement baseline ML models for congestion forecasting</li></ul>
3	Deep Learning Model Development & NLP Integration	<ul style="list-style-type: none"><li>- Develop LSTM/CNN models for time-series traffic data</li><li>- Implement NLP pipeline (spaCy/NLTK) for Twitter data</li><li>- Combine textual features with ML predictions</li></ul>
4	Graph Modeling & Route Optimization	<ul style="list-style-type: none"><li>- Construct road network graph (NetworkX) with dynamic traffic weights</li><li>- Implement or refine route optimization algorithms</li><li>- Validate integrated model performance</li></ul>
5	Visualization, Front-End & Testing	<ul style="list-style-type: none"><li>- Create interactive dashboards/heatmaps (Bokeh/Plotly)</li><li>- Integrate real-time data refresh</li><li>- Conduct end-to-end testing and refine model/UI</li></ul>
6	Documentation, Deployment & Final Presentation	<ul style="list-style-type: none"><li>- Document all components (data pipeline, models, APIs, usage)</li><li>- Deploy system (cloud or local server)</li><li>- Prepare &amp; deliver final project presentation</li></ul>