Table Of Contents

1.Introduction

Urban congestion poses a growing challenge in modern cities, directly affecting daily commutes, travel planning and transportation efficiency. With the expansion of public transit infrastructure such as the Riyadh Metro, the ability to forecast congestion levels has become crucial for enhancing passenger satisfaction and improving system operations.

In this project, we present **Insiyab**, an AI-powered system designed to predict metro congestion levels based on spatial and temporal features. ***Insiyab*** combines two core artificial intelligence components: a deep learning model trained using **LSTM** architecture for **congestion level prediction** and a **pre-trained natural language processing (NLP)** model capable of interpreting user-provided trip descriptions and converting them into structured data. Through this integration, we aim to develop an intuitive and intelligent system that allows users to interact with AI in a natural way while benefiting from accurate, data-driven insights.

The project contributes to the broader field of smart urban mobility and supports the vision of transforming transportation experiences through artificial intelligence.

2.Project objectives

1. Develop an accurate congestion prediction model using real-world-inspired data and train it using deep learning techniques, specifically Long Short-Term Memory **(LSTM)** networks.

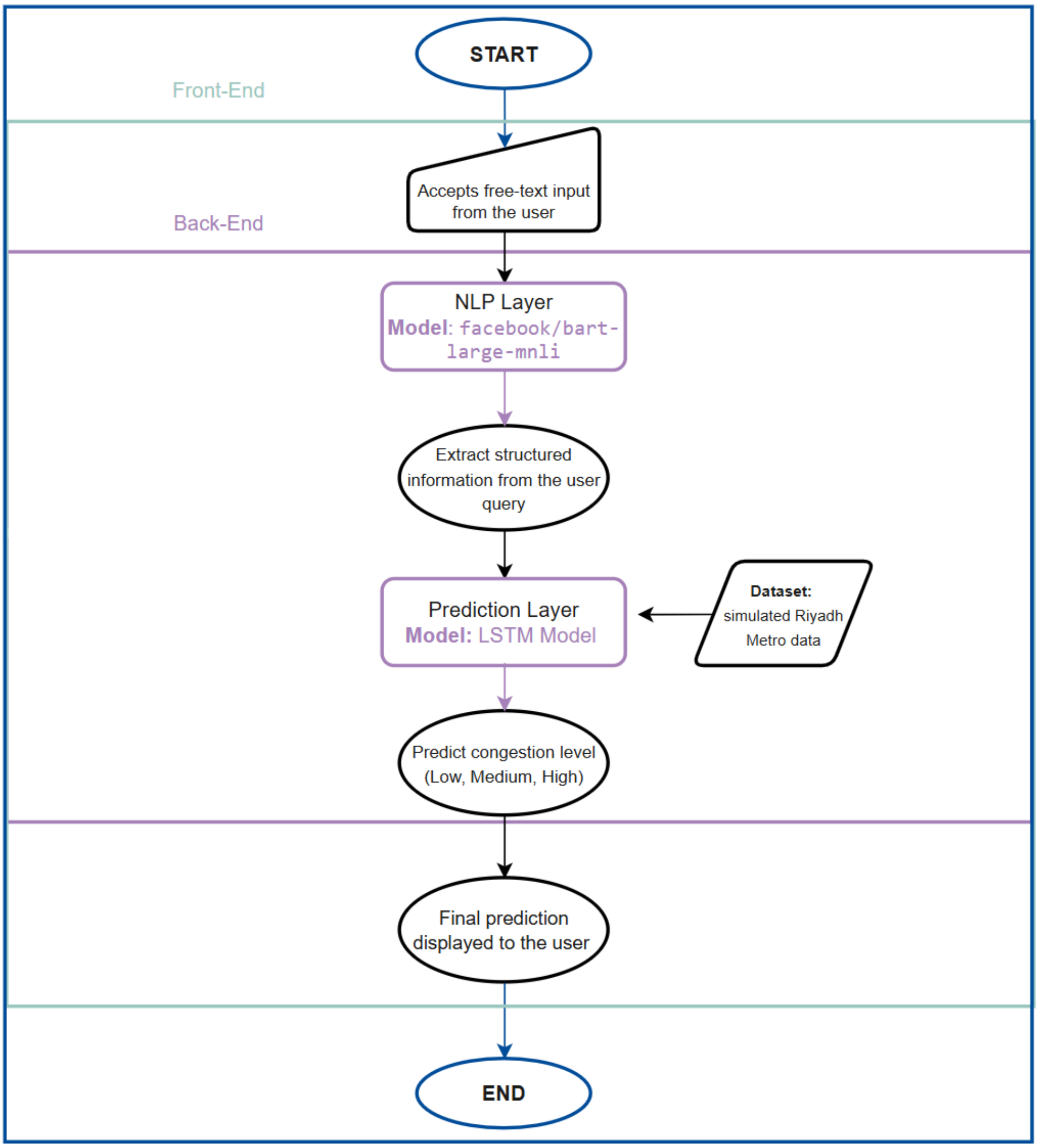
2. Integrate a **pre-trained NLP** model to handle unstructured user inputs and extract meaningful features such as time, station names and day of the week.

3. Create a seamless end-to-end pipeline that combines data preprocessing, training and inference to ensure robust and scalable performance.

4. Provide real-time predictions with interpretability, allowing users to understand not only the predicted congestion level but also the rationale behind the model's output.

5. Design an interactive user interface that enables intuitive interaction with the system, facilitating the application of AI predictions in a real-world context.

3.Diagram


START
Front-End
Accepts free-text input from the user
Back-End
NLP Layer
Model: facebook/bart-large-mnli
Extract structured information from the user query
Prediction Layer
Model: LSTM Model
Dataset: simulated Riyadh Metro data
Predict congestion level (Low, Medium, High)
Final prediction displayed to the user
END

4.Methodology (data processing- model)

Our project is built around the idea of combining two powerful models — one that we trained ourselves, and one that comes pretrained — to create a smart, responsive metro assistant. The goal was to let users interact with the system in the most natural way possible: by simply typing a question in plain English, like "Is Al Batha crowded on Monday at 8 AM?".

To make this happen, we first generated a realistic synthetic structured dataset that mimics Riyadh Metro data, since no public dataset was available. Using this, we trained our own **LSTM model** to predict congestion levels based on the station, day and time.

On the other side, we integrated a **pretrained NLP model** ***facebook/bart-large-mnli*** to understand what the user is asking and extract the needed information without any manual input or rules.

This section explains how these two models were brought together along with data preparation, cleaning and balancing to deliver an interactive experience that feels both intelligent and easy to use.

4.1 Data Collection and Preparation

Due to the lack of publicly available real-world data for the Riyadh Metro, we generated a synthetic dataset (10 thousand instances) that mimics the real characteristics of metro operations. The dataset includes features:

- **Route_ID** – A unique identifier combining the route number and line color (e.g. Red_R1).
- **Start_Station** – The station where the journey begins.
- **End_Station** – The destination station.
- **Hour** – The hour of the day (0 to 23) representing the trip's time.
- **Weekday** – The day of the week (e.g. Monday, Tuesday).
- **Distance_km** – Estimated travel distance between stations (in kilometers).
- **Travel_Time_mins** – Estimated time required for the trip (in minutes).
- **Peak_Hour** – Whether the trip occurs during peak hours (Yes / No).
    - **Passenger_Demand** – The estimated number of passengers on the route.
- **Speed_kmph** – The calculated average speed for the trip (in km/h).
- **Congestion_Level** – The target label indicating crowding level (Low, Medium, or High).

4.2 Data Preprocessing

Before training our models, the dataset was carefully preprocessed to improve performance and reliability. The preprocessing steps included:

- **Data Cleaning**: Missing or incomplete entries were removed to ensure the model trains on clean and consistent data.
- **Encoding**: Categorical variables like station names, weekdays and congestion levels were encoded into numerical format using label encoding to make them suitable for machine learning.
- **Feature Scaling**: Numerical features, especially the hour of the day, were scaled using normalization techniques to standardize their range.
- **Label Balancing (Resampling)**: Initially, the dataset had an imbalanced distribution of congestion levels, with a dominance of the "Medium" class. To ensure fair training and avoid biased predictions, resampling techniques were applied to balance the number of samples for each class.

4.3 Model Design and Integration

We combined two different types of models to handle both natural language understanding and congestion prediction:

- **Natural Language Processing (NLP) Model**: We used a pretrained transformer model, *facebook/bart-large-mnli* [1], to interpret free-form user queries. This model is capable of zero-shot classification and was used to extract structured data (station name, weekday, hour) from the user's input without any additional training.
- **LSTM Model**: A Long Short-Term Memory (LSTM) model was trained on the preprocessed dataset to predict the congestion level based on structured inputs (station, time and day). The model performs multi-class classification with outputs representing congestion levels.

The two models were then integrated, the NLP model processes the user's natural language input and extracts relevant features, which are then passed to the LSTM model. The predicted congestion level is presented back to the user through a simple and intuitive Gradio-based interface.

5. User interface

The project is designed to be **user-friendly interface designed using Gradio**, which is **easy to use and flexible**, allowing users to ask questions in various formats while specifying the **station**, **day** and **time**, without requiring any technical knowledge or the need to select from dropdown menus. Users can interact directly with the model.

Once the user submits the query, the system processes the input, interprets the text and extracts the relevant information such as **station**, **day** and **time**. It then generates an accurate prediction of the **congestion level** (Low, Medium or High) along with the probabilities associated with each category.

Functionality:

Input: The interface expects the user to input a text query in the form of, for example, "What is the congestion at Al Batha station on Monday at 8 AM?".

Output: The output is a **text response** displaying the predicted congestion level (Low, Medium or High) along with the probability for each category.

The model's predictions are displayed in a clear and concise format:

Congestion Level: Predicted category (e.g., High, Medium, Low)

Probabilities: The probabilities for each category, allowing users to understand how confident the model is in its prediction.

Emojis: The UI uses visual emojis to represent the congestion levels:

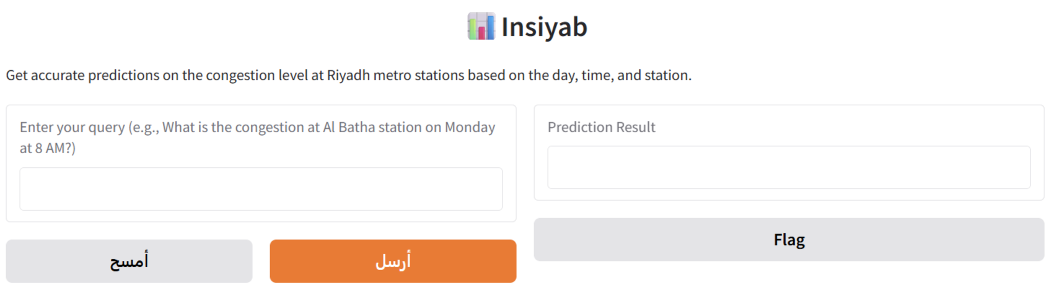
🟢 **Low:** Indicates less crowded stations

🟡 **Medium:** Indicates moderate congestion

🔴 **High:** Indicates highly congested stations

Users can easily input their queries and obtain instant results. This is crucial for providing real-time insights into congestion levels, which could greatly assist commuters in planning their trips.

Screenshot Showing interface:



6.Performance analysis

The performance of the system's performance developed in this project examined two main components: the natural language understanding module, implemented with *facebook/bart-large-mnli* and congestion prediction module with an LSTM neural network.

This section will elaborate and inform on the performance of all models with respect to the experimental results, including interpretation of the metrics and the advantages and limitations of each model.

The *facebook/bart-large-mnli* model was used to extract structured data from user queries.

The model was accurate for the most part with less than 10% of accurate queries due to wrong identification of either station name, date or time. The model also performed well with various phrasings and natural variations from users with an estimated extraction rate of over 90%. The model did have weak performance for vague or very informal queries, which will at least somewhat impact downstream prediction quality. The LSTM neural network was trained to classify congestion from data labelled, as follows: 0 (low), 1 (medium), 2 (high). The classification report for the test set shows that the model performed very well overall with an accuracy of 95% based on 138 samples. The precision, recall and F1 score for each class were:

Class (Congestion)	Precision	Recall	F1-Score	Support
0 (Low)	0.94	1.00	0.97	50
1 (Medium)	0.92	1.00	0.96	44
2 (High)	1.00	0.84	0.91	44

The model has an excellent performance for low and medium congestion, demonstrating an almost perfect recall for those classes from this table. It also achieved perfect precision (1.00) for the high congestion class but at the cost of a somewhat lower recall score (at 0.84), which would indicate there are instances of high congestion that were misclassified, most often as medium congestion.

The confusion matrix gives a better picture of how the model performed:

Actual \ Predicted	0 (Low)	1 (Medium)	2 (High)
0 (Low)	50	0	0
1 (Medium)	0	44	0
2 (High)	3	4	37

The matrix illustrates the primary source of error: out of 44 high congestion samples, 3 were classified as low and 4 were classified as medium. The fact that there were a relatively small number of misclassifications suggests that while the LSTM network learned some strong discriminative learning overall, it was not performing as well at distinguishing high congestion from other classifications, particularly where high congestion classifications overlapped with similar, but, less extreme conditions.

Regarding system behavior, the results demonstrate that BART was very useful as a way to preprocess and LSTM very well as a way to classify, to create a reasonably reliable pipeline. The computed macro and weighted average precision, recall and F1-score all achieved 0.95, confirming that the model produced a reasonably balanced performance across all classes and avoided bias towards a particular class.

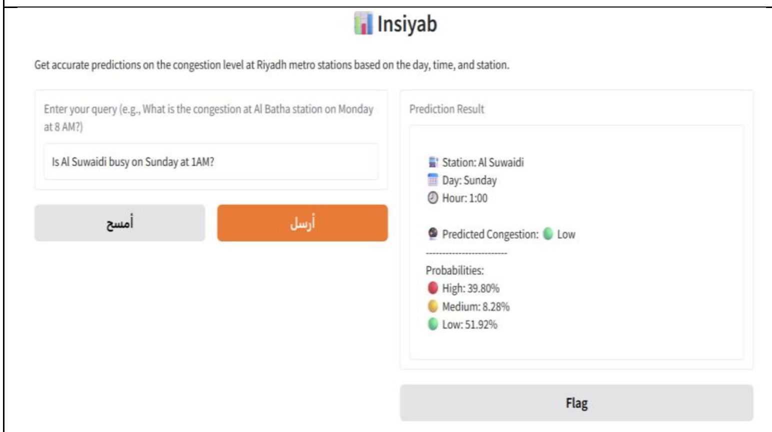
The training behavior of the LSTM also supports the validity of the model. Because the model was trained with early stopping there is no danger that the model will over-fit in the present state. Also, the consistent decrease in loss during training indicates that the model successfully learned. The use of synthetic data and LSTM architecture appears sufficient to adequately capture temporal patterns for predicting congestion. The results also present some limitations. The lower recall for the high congestion class indicates a need for more data or potentially adjustments to the architecture (e.g. trialing attention mechanisms, dropout layers as regularization, etc.). Also, while the BART model handled natural language queries well, local fine-tuning may improve the model's ability to process questions that have a cultural/context relevant to the user's query.

In summary of performance, the system has demonstrated solid and reasonable performance, obtained near perfect classification for the low and medium congestion classes and acceptable performance in the high congestion class. This gives us adequate confidence that we can and will improve the system moving forward; also, as we might gain access to real-world metro data, as well as general architectural improvements in the future.Overall, we perceive the system as "ready" for early or pilot deployment testing, with the potential for refinement and improvements in time.

7.Results and discussion

Present the results of the project by focusing on the system's performance in predicting congestion levels at Riyadh Metro stations based on the **station**, **day** and **time** specified by the user natural language query. The **primary goal** of the system is to provide passengers with real-time information about congestion at metro stations, helping them make informed travel decisions.

The **NLP** model is used to analyze user-entered queries and **LSTM** model predict the congestion level at a specific station at a specific time. The prediction is based on several factors, such as historical data, the day of the week and the time of day.

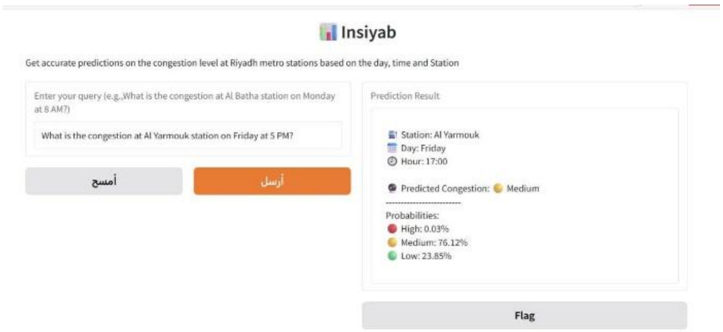
<table>
<tr><th>UI</th><th>Prediction Justification</th></tr>
<tr><td>


</td><td>At Al-Suwaidi Station, the model output forecast for a time forecast of Sunday at 1:00 AM shows a low probability of congestion. The probability of not having congestion was calculated to be a 51.92% probability of having no congestion. In addition there are probabilities of medium congestion with 39.80% and probabilities of high congestion with 8.28%.

This result indicates a low congestion level, primarily because the metro is usually closed or limited to limited operations during the late night hours. For most of the year Riyadh Metro services are unavailable at 1:00 AM. Therefore, the system correctly predicts that Al-Suwaidi Station will experience slight congestion at this time</td></tr>
<tr><td>


</td><td>The output prediction for Imam Muhammad Bin Saud University on Tuesday at 4 PM indicates a high congestion level, with the model predicting 99.91% likelihood for high congestion, while medium and low congestion have negligible probabilities.

This high congestion level can be explained by the fact that 4 PM is generally the time when many people leave work or school, which leads to increased commuter activity. In a densely populated area like a university, this time often coincides with the end of classes or work hours, making it a peak time for congestion. As a result, the system accurately predicts high congestion at this hour.</td></tr>
</table>

<table>
<tr>
<td>
</td>
<td>The output prediction for Al Yarmouk station on Friday at 5 PM indicates a medium congestion level, with the model predicting 76.12% likelihood for medium congestion. The probabilities for high and low congestion are minimal, with high congestion at just 0.03% and low congestion at 23.85%.

This medium congestion level can be attributed to the fact that 5 PM is typically around rush hour in many cities, especially in a place like Riyadh where people may be heading home after a long workday. Additionally, Friday is a weekend day, but it still sees significant commuter activity as many individuals use public transportation to visit shopping centers, cafes, or recreational spots. The proximity of Al Yarmouk station to residential and commercial areas likely contributes to moderate congestion during this time, as people move around for leisure activities. Therefore, the system's prediction of medium congestion aligns with these factors.</td>
</tr>
</table>

These results reflect the **Insiyab** system's ability to provide accurate, real-time predictions of congestion levels at Riyadh Metro stations based on user inputs of the station, day and time. The models demonstrated good predictive ability to predict congestion levels at different times, helping commuters make informed travel decisions.

8. Limitations (challenges)

Although the **Insiyab** system has demonstrated strong performance and promising results in predicting congestion levels at Riyadh Metro stations, there are several challenges and limitations that should be considered:

1. **Reliance on Synthetic Data:**

   Due to the absence of real data for the Riyadh Metro, the model was trained using synthetic data generated to simulate real-world conditions. While the data is relatively accurate, it may not fully capture the complexities and variations of real passenger behavior.

2. **Lack of Advanced Spatial and Temporal Context Integration:**

   Certain important external factors, such as major events, weather conditions, or schedule changes, have not yet been integrated. These factors can significantly impact congestion levels and influence the accuracy of predictions.

3. **Integration with Other Transport Systems:**

   For future expansion, integrating the system with other modes of transportation (such as smart buses) presents a challenge in terms of infrastructure and data format standardization across different systems.

9.Conclusion & Future work

The **Insiyab** project demonstrates the practical application of artificial intelligence in addressing urban mobility challenges specific to Riyadh's metro system. By combining an LSTM-based congestion predictor with a pre-trained natural language model, the system successfully bridges the gap between technical complexity and user accessibility. The results achieved reflect strong model performance, with a test accuracy exceeding 95%, particularly in predicting low and medium congestion levels.

Looking ahead, the system will evolve through several key enhancements. A primary focus will be on developing the user interface to align more closely with real-world deployment standards. Real-time data integration will also be implemented to better reflect live congestion conditions and improve prediction accuracy. Furthermore, the system will be extended to include intelligent responses to user inputs, such as speech recognition and interactive guidance. Additional efforts will be made to improve scalability and support future integration with other transportation systems within the city, such as smart buses—enhancing the system's flexibility and broader impact.

In conclusion, Insiyab offers a scalable, intelligent, and user-centered approach to congestion forecasting in Riyadh Metro, representing a significant step toward building smart transportation infrastructure in alignment with Saudi Arabia’s Vision 2030.

Appendix

https://colab.research.google.com/drive/1xctrnZHOiKoP6hyTJ1g5itwJHFrM2Dc_?usp=sharing

References

[1] Facebook AI, "facebook/bart-large-mnli," Hugging Face, 2020. [Online]. Available: https://huggingface.co/facebook/bart-large-mnli [Accessed: May 2, 2025].

[2] Facebook AI, "facebook/bart-large-mnli," Hugging Face, 2020. [Online]. Available: https://huggingface.co/facebook/bart-large-mnli [Accessed: May 2, 2025].

[3] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997. doi:10.1162/neco.1997.9.8.1735. (General reference on LSTM algorithm)

[4] F. Chollet, "Keras Documentation: Recurrent layers," 2024. [Online]. Available: https://keras.io/api/layers/recurrent_layers/lstm/ [Accessed: May 2, 2025]. (Practical LSTM application in Python)

[5] Kaggle, "Synthetic Data Guide," 2023. [Online]. Available: https://www.kaggle.com/code/synthetic-data [Accessed: May 2, 2025]. (Dataset and synthetic data generation)

[6] Riyadh Metro, "Darb - Official Metro App," Riyadh Development Authority, 2025. [Online]. Available: https://www.riyadhmetro.sa/darb [Accessed: May 2, 2025]. (Metro Riyadh / Darb reference)

[7] Y. Ma, Y. Wang, H. Chen, and Y. Zhang, "Long Short-Term Memory Neural Network for Traffic Speed Prediction Using Remote Microwave Sensor Data," Transportation Research Part C: Emerging Technologies, vol. 54, pp. 187–197, 2015. doi:10.1016/j.trc.2015.03.014. (LSTM applications in traffic prediction)

[8] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," in Advances in Neural Information Processing Systems (NeurIPS), 2015. [Online]. Available: https://papers.nips.cc/paper_files/paper/2015/hash/07563a3fe3bbe7e3ba84431ad9d055af-Abstract.html [Accessed: May 2, 2025]. (LSTM in congestion prediction)