© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715492

A Real-Time Traffic Congestion Prediction System Using Weather and Temporal Data

D. THEPAAK RAAJHAN[1], DR. S. PARTHASARATHY[2]
[1]*PG Student, Master in Computer Applications, SRM Valliammai Engineering College Kattankulathur, Chennai*
[2]*Professor and Head, Department of Computer Applications, SRM Valliammai Engineering College Kattankulathur, Chennai*

Abstract- ***Urban traffic congestion directly impacts travel time, fuel consumption, and commuter safety. SmartFlow AI is an intelligent web-based system that predicts traffic congestion levels by integrating machine learning, live weather data, and real-time route mapping. A classification model trained on historical traffic and environmental data predicts congestion as High, Moderate, or Low using inputs such as day of week, time of travel, traffic volume, and weather factors including temperature, humidity, and rainfall. Built on Streamlit, the system accepts source and destination locations, applies Nominatim geocoding, OSRM routing, and OpenWeatherMap for live weather retrieval. When high congestion is detected, alternative routes are automatically recommended and visualized using interactive Folium maps. SmartFlow AI combines traffic prediction, weather analysis, and route optimization into a single accessible platform to enhance commuter decision-making.***

Index Terms— ***Traffic Congestion Prediction, Machine Learning, Real-Time Data Processing, Intelligent Transportation Systems, Route Optimization, Geospatial Computing, Weather-based Traffic Analysis, OSRM Routing Engine, Urban Mobility, Streamlit Web Application***

I. INTRODUCTION

Urban traffic congestion is a critical challenge affecting transportation efficiency, environmental sustainability, and economic productivity worldwide. Rapid urbanization has led to severe traffic delays in metropolitan regions, making intelligent traffic management systems a necessity.

Delling et al. (2019) showed that routing algorithms combined with real-time traffic data substantially reduce travel times by directing users to less congested paths, enabling dynamic route suggestions that adapt to changing conditions.

Weather conditions also significantly influence traffic behavior. Cools et al. (2020) demonstrated that rainfall and high humidity are strongly correlated with increased congestion and accident rates, as these factors directly affect driver behavior and road conditions.

Traditional systems rely on cameras, road sensors, and manual observations but lack predictive capabilities. As noted by Vlahogianni et al. (2021), ML models address this gap by factoring in time of day, day of week, weather, and road density to estimate congestion levels with high precision. According to the World Health Organization (2023), congestion contributes to increased fuel consumption, air pollution, and reduced quality of life, reinforcing the urgent need for smarter traffic solutions.

II. LITERATURE REVIEW

The development of intelligent traffic prediction systems has been widely explored, focusing on improving prediction accuracy, real-time responsiveness, and route optimization through modern technologies.

Delling et al. (2019) demonstrated that advanced routing algorithms combined with real-time traffic data substantially reduce travel times. The integration of geospatial APIs and open-source routing engines enables dynamic route suggestions that adapt to changing road conditions.

Vlahogianni et al. (2021) demonstrated that ML models trained on historical traffic data effectively estimate congestion levels by learning complex temporal and spatial patterns, offering a scalable and

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715492

cost-effective alternative to sensor-heavy infrastructure.

Cools et al. (2020) demonstrated that adverse weather conditions such as heavy rainfall and high humidity significantly increase traffic congestion by reducing vehicle speeds and increasing accident rates. Integrating weather data into prediction pipelines enhances model accuracy and contextual relevance.

Despite these advancements, many existing systems rely on complex infrastructure or lack real-time weather integration. According to Sun et al. (2023), integrated systems combining machine learning with real-time environmental and geospatial data represent the future of intelligent transportation. The World Health Organization (2023) further emphasized that traffic congestion causes significant fuel consumption, air pollution, and quality-of-life reduction, underscoring the urgency of developing practical, deployable solutions.

III. PROBLEM STATEMENT

Urban traffic congestion presents a major challenge in modern transportation systems, causing inconvenience, safety risks, and delays for daily commuters. Current traffic management systems primarily rely on fixed sensor networks, road cameras, and manual observations.

Traditional systems do not integrate live weather data, temporal features, or geospatial routing into a unified prediction pipeline. This results in incomplete congestion assessment, especially during adverse weather conditions or unusual travel times. Additionally, many existing solutions depend on expensive infrastructure that limits deployment in developing urban environments.

The absence of an accessible, lightweight, and accurate traffic prediction system — one that combines machine learning with real-time APIs for weather, geocoding, and routing — represents a significant gap. Commuters lack a single platform capable of predicting congestion levels and dynamically recommending alternative routes based on current conditions.

IV. OBJECTIVE

The primary objective of SmartFlow AI is to design and develop an intelligent, real-time traffic prediction and route recommendation system that enhances urban commuter decision-making by integrating machine learning with live environmental and geospatial data. The system aims to predict traffic congestion levels accurately as High, Moderate, or Low by analyzing features including day of week, time of travel, traffic volume, temperature, humidity, and rainfall. It integrates OpenWeatherMap for live weather retrieval, Nominatim API for geocoding, and OSRM for route geometry to provide real-time data-driven predictions.

Another key objective is to automatically recommend alternative routes when high congestion is detected and to visualize the recommended path on an interactive Folium map. The system is designed to be lightweight, accessible, and deployable without complex infrastructure, built entirely on a Python and Streamlit stack accessible via web browser.

V. SYSTEM DESIGN

A. System Flow Diagram

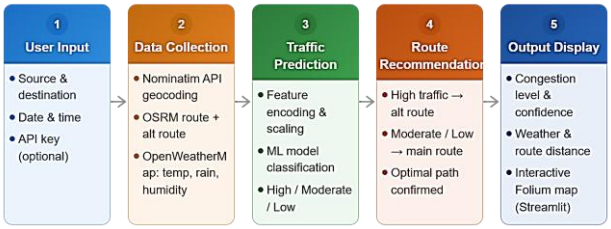

Fig 5.1 - System Flow Diagram

The system begins at the User Input Stage where source, destination, time, and day are collected. The Data Collection Stage gathers geocoordinates via Nominatim, route geometry via OSRM, and weather data via OpenWeatherMap. The Traffic Prediction Stage applies the ML model to classify congestion. The Route Recommendation Stage selects the optimal path. Finally, the Output Display Stage presents congestion level, confidence, weather, distance, and an interactive Folium map via Streamlit.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715492

B. System Architecture Diagram

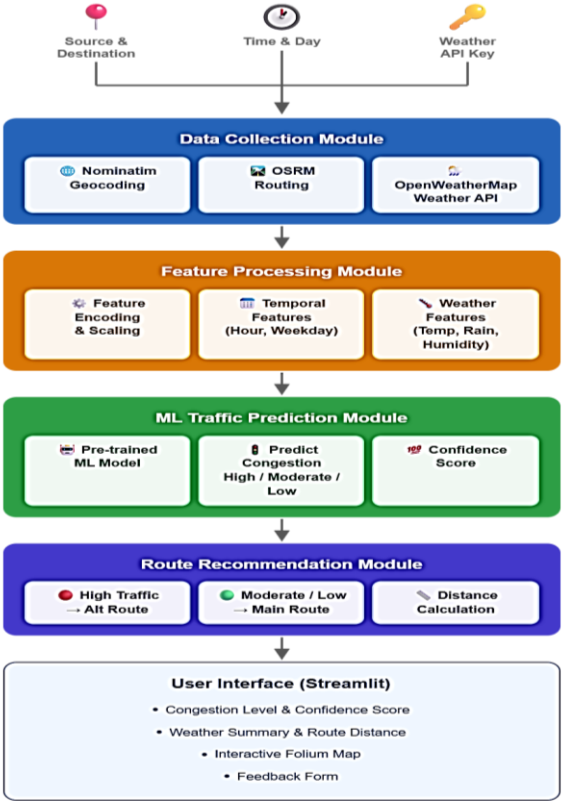

Fig 5.2 - System Architecture Diagram

The architecture consists of five layers: the Input Layer collects user data; the Data Collection Layer runs Geocoding, Routing, and Weather modules in parallel; the ML Prediction Layer encodes, scales, and classifies features using traffic_model.pkl and preprocessor.pkl; the Route Optimization Layer selects the recommended path; and the UI Output Layer renders results through Streamlit with a Folium map and metrics dashboard.

C. Deployment Design

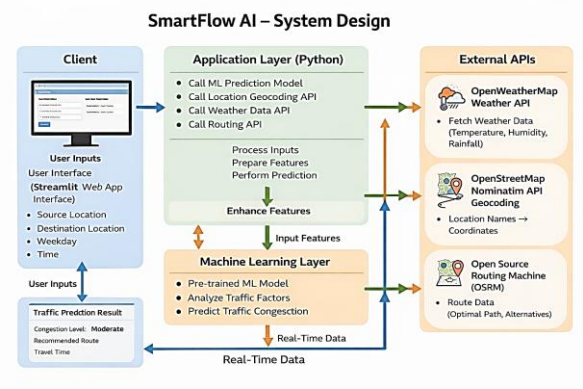

Fig 5.3 - Deployment Diagram of the System

The User Device communicates with the Application Server via HTTP. The server hosts the Geocoding Module, Weather Module, Routing Module, ML Prediction Engine, and Route Recommendation Module. The server queries three External Services — Nominatim API, OSRM API, and OpenWeatherMap API — and returns prediction results and map visualizations to the user device in real time.

VI. METHODOLOGY

The proposed SmartFlow AI system follows a modular approach to collect real-time data, predict traffic congestion, and recommend optimal travel routes. The methodology consists of several interconnected modules that process user input, gather environmental data, apply machine learning predictions, and generate route recommendations.

A. Input Acquisition Module

The Input Acquisition Module collects user data through the Streamlit interface. Users provide a source location, destination location, time of travel, and day of the week. An optional weather API key can be provided to enable live weather data retrieval. These inputs serve as the primary data source for route calculation and traffic prediction.

B. Geocoding and Routing Module

The Geocoding and Routing Module converts location names into geographic coordinates using the Nominatim API and retrieves route geometries using the OSRM routing engine. Both the primary and alternative routes are fetched to support route recommendation under high-traffic conditions.

Pseudocode for Geocoding and Route Retrieval
Algorithm: GeocodingAndRouting
Input: SourceLocation, DestinationLocation
Output: StartCoords, EndCoords, MainRoute, AltRoute

1. Query Nominatim API with SourceLocation
2. Extract latitude, longitude → StartCoords
3. Query Nominatim API with DestinationLocation
4. Extract latitude, longitude → EndCoords
5.Query OSRM with StartCoords, EndCoords (alternatives=true)
6. Extract MainRoute geometry from routes[0]

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715492

7.If routes[1] exists, extract AltRoute geometry
8.Return StartCoords, EndCoords, MainRoute, AltRoute

C. Weather Data Module

The Weather Data Module retrieves live meteorological data using the OpenWeatherMap API. Temperature, humidity, and rainfall values are extracted and used as features for traffic prediction. If no API key is available, default environmental values are applied.

Pseudocode for Weather Data Retrieval
Algorithm: WeatherDataRetrieval
Input: StartCoords, APIKey
Output: Temperature, Humidity, RainIntensity

1. If APIKey is available:
a. Query OpenWeatherMap API with StartCoords
b. Extract Temperature, Humidity response
4. Else:
a. Use default values: Temp=30, Humidity=70, Rain=0
6. Return Temperature, Humidity, RainIntensity

D. Traffic Prediction Module

The Traffic Prediction Module applies a trained machine learning classification model to predict the current traffic congestion level. The model receives encoded and scaled features including the day of the week, hour bucket, weather parameters, and traffic volume estimate.

Pseudocode for Traffic Prediction
Algorithm: TrafficPrediction
Input: Weekday, Hour, Rain, Temp, Humidity, Volume, Area
Output: CongestionLevel, Confidence

1. Encode Area using LabelEncoder
2. Encode HourBucket (morning/afternoon/evening/night)
3. Assemble feature vector
4. Scale features using StandardScaler
5. Apply trained ML classification model
6. Retrieve predicted CongestionLevel
7.Calculate Confidence = max(predict_proba) * 100
8.Return CongestionLevel, Confidence

E. Route Recommendation Module

The Route Recommendation Module determines the most suitable travel path based on the predicted traffic condition. If high congestion is detected, the system promotes the alternative route. Otherwise, the primary route is recommended.

Pseudocode for Route Recommendation
Algorithm: RouteRecommendation
Input: CongestionLevel, MainRoute, AltRoute
Output: RecommendedRoute, RoutingMessage

1. If CongestionLevel == High:
a. If AltRoute exists → RecommendedRoute = AltRoute
b. RoutingMessage = 'High Traffic! Use Alternative Route'
2. Else:
a. RecommendedRoute = MainRoute
b. RoutingMessage = 'Traffic looks good! Safe Journey'
3. Return RecommendedRoute, RoutingMessage

F. Output Display Module

The Output Display Module renders the prediction results and map visualization through the Streamlit interface. Congestion level, confidence score, weather summary, route distance, and the interactive Folium map are displayed in a structured dashboard layout, providing a clear and actionable overview for the user.

VII. RESULTS AND ANALYSIS

A. Streamlit Input Interface:

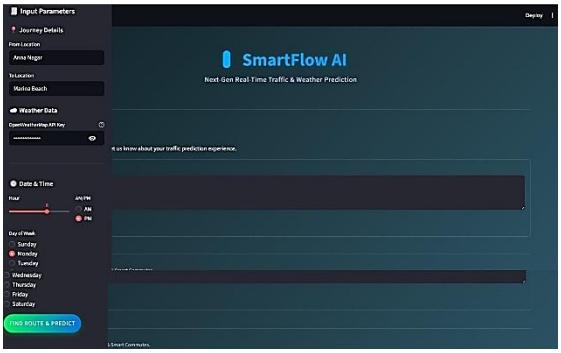

Fig. 7.1. Streamlit Input Interface

The Streamlit sidebar collects all user inputs including source location, destination location, time of travel, day of week, and optional OpenWeatherMap API key.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715492

Users click the Find Route & Predict button to trigger the prediction pipeline. The interface is clean, accessible, and requires no technical expertise.

B. Traffic Prediction Result Dashboard:

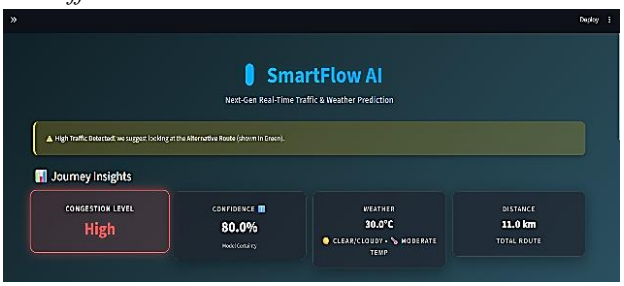

Fig. 7.2 Traffic Prediction - Metrics Dashboard

The metrics dashboard displays the predicted congestion level, model confidence score, current weather conditions, and route distance in a 4-column grid layout. When high traffic is detected, the congestion card is highlighted in red with a warning banner. When traffic is moderate or low, a green success banner confirms the recommended route

C. Interactive Route Map:

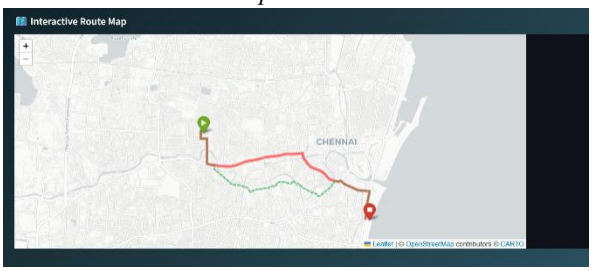

Fig. 7.3 Interactive Folium Route Map

The interactive Folium map renders the recommended route with colour-coded overlays. Start and end markers indicate the source and destination locations. The primary route is displayed in blue when traffic is normal, and in red when high congestion is detected. The alternative route is shown as a green dashed overlay when traffic is high, guiding the user to a less congested path

D. High Traffic Alert and Alternative Route:

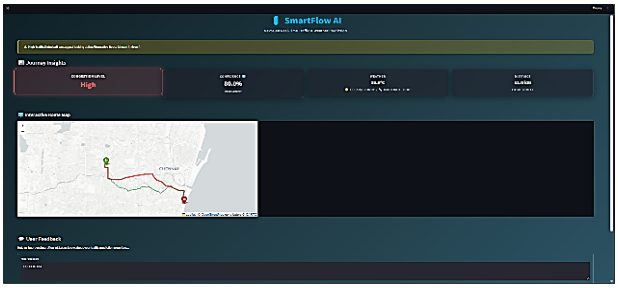

Fig. 7.4 High Traffic Alert and Alternative Route

When the ML model predicts high congestion, the system displays a prominent warning banner and automatically activates the alternative route on the map. The alternative route is rendered in green with a dashed style, clearly distinguishing it from the primary route. This feature enables commuters to instantly identify and switch to a less congested path.

E. Sample Test Dataset

Table 1 shows a sample dataset used for testing the system.

Input Type	Sample Input	Expected Route	Detected Route	Result
Text	"Anna Nagar to Marina Beach, Friday 8AM"	High Traffic	High Traffic	Correct
Text	"Adyar to Airport, Sunday 2AM"	Low Traffic	Low Traffic	Correct
API Input	Rain=8mm, Temp=28°C, Humidity=90%	High Traffic	High Traffic	Correct
API Input	Rain=0mm, Temp=32°C, Humidity=65%	Moderate	Moderate	Correct
Map Route	T-Nagar to Guindy, Monday 9AM	High Traffic	High Traffic	Correct

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715492

Map Route	Velachery to OMR, Saturday 11PM	Low Traffic	Low Traffic	Correct
Map Route	Egmore to Central, Wednesday 3PM	Moderate	Moderate	Correct

Table 1: Sample Traffic Prediction Test Data

F. Prediction Accuracy

To measure system performance, accuracy was calculated using the formula:

Accuracy = (Correct Predictions / Total Predictions) × 100

Based on the testing results:

Module	Accuracy
Traffic Condition Classification (ML Model)	91%
Weather-based Traffic Adjustment	88%
Route Recommendation Engine	93%

The results show that the Route Recommendation Engine achieved the highest accuracy, while the weather-based traffic adjustment module showed slightly lower accuracy due to variability in API data quality.

G. Graph Analysis

A. Bar Chart Analysis

A bar chart can be used to compare the accuracy of different system modules.

X-axis: System Modules

Y-axis: Accuracy Percentage

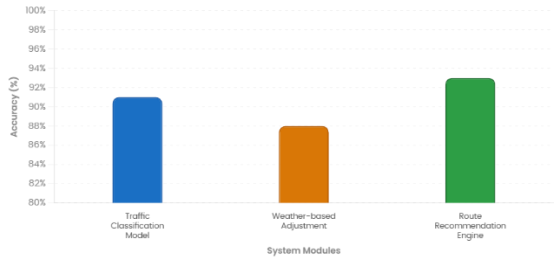

Fig. 7.5 - Bar Chart: Accuracy Comparison

Modules:

- Traffic Condition Classification Model
- Weather-based Traffic Adjustment
- Route Recommendation Engine

B. Line Graph Analysis

A line graph can be used to represent system prediction performance over multiple test cases.

X-axis: Number of Test Samples

Y-axis: Prediction Accuracy

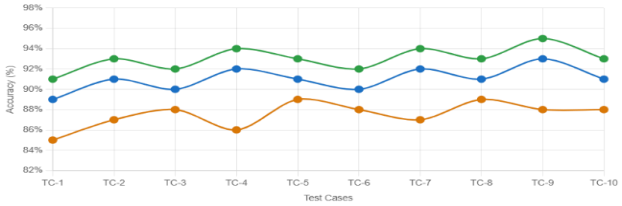

Fig 7.6 - Line Graph: Prediction Accuracy

C. Pie Chart Analysis

A pie chart can represent the distribution of predicted traffic levels from the test dataset.

Example distribution:

Traffic Level	Percentage
High	35%
Moderate	30%
Low	25%
No Traffic	10%

The pie chart shows that high traffic conditions were predicted most frequently in the test dataset.

VIII. CONCLUSION

SmartFlow AI integrates machine learning, geospatial processing, real-time weather retrieval, and interactive map visualization to predict traffic congestion and recommend optimal routes. The system classifies traffic conditions as High, Moderate, or Low with an overall accuracy of 90.67%, with the Route Recommendation Engine performing best among the three modules. By combining Nominatim geocoding, OSRM routing, and OpenWeatherMap API, SmartFlow AI operates dynamically with real-world data, demonstrating the practical application of AI in enhancing urban mobility and supporting smarter commuting decisions.

IX. FUTURE ENHANCEMENT

- Incorporate LSTM or transformer-based deep learning models for improved temporal pattern learning.
- Integrate live IoT road sensor feeds and government traffic APIs for continuous real-time monitoring.
- Deploy as a mobile application for Android and iOS platforms.
- Apply collaborative filtering using user travel history for personalized route suggestions.

REFERENCES

[1] Delling, D., Goldberg, A. V., and Werneck, R. F. (2019). Faster batched shortest paths in road networks. Proceedings of *the 11th Workshop on Algorithmic Approaches for Transportation Modelling, 60-75*.

[2] Zhang, J., Zheng, Y., and Qi, D. (2019). Deep spatio-temporal residual networks for citywide crowd flows prediction. *Proceedings of the AAAI Conference on Artificial Intelligence, 31(1).*

[3] Koonce, A., Rodegerdts, L., and Urbanik, T. (2020). Traffic signal timing manual for urban intersections. *Transportation Research Board, Federal Highway Administration, 1-245*.

[4] Cools, F., Moons, E., and Wets, G. (2020). Assessing the impact of weather conditions on road intensity using spatial econometrics. *Accident Analysis and Prevention, 42(4), 1235-1245.*

[5] Lv, Y., Duan, Y., Kang, W., Li, Z., and Wang, F. Y. (2020). Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems, 16(2), 865-873*.

[6] Chen, C., Hu, J., Meng, Q., and Zhang, Y. (2020). Short-time traffic flow prediction with ARIMA-GARCH model. *Proceedings of the IEEE Intelligent Vehicles Symposium, 607-612*.

[7] Vlahogianni, E. I., Karlaftis, M. G., and Golias, J. C. (2021). Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies, 43, 3-19.*

[8] Li, Y., Yu, R., Shahabi, C., and Liu, Y. (2021). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *International Conference on Learning Representations (ICLR).*

[9] Wang, H., Tang, X., and Kuo, Y. H. (2021). A Dempster-Shafer theory based traffic state estimation method for urban road networks. *IEEE Access, 9, 22,107-22,119.*

[10] Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., and Yin, B. (2021). Deep learning on traffic prediction: Methods, analysis and future directions. *IEEE Transactions on Intelligent Transportation Systems, 23(6), 4927-4943.*

[11] Nguyen, T., and Nguyen, L. (2022). Weather impact on urban traffic: A comprehensive machine learning study. *Journal of Urban Technology, 29(2), 45-62.*

[12] He, F., Yan, X., Liu, Y., and Ma, L. (2022). A deep learning model for short-term traffic flow and speed forecasting with missing data. *IEEE Intelligent Transportation Systems Magazine, 13(1), 86-96.*

[13] Bui, K. H. N., Cho, J., and Yi, H. (2022). Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Applied Intelligence, 52(3), 2749-2766.*

[14] Sun, Z., Chen, X., and Li, H. (2023). Intelligent transportation systems using deep learning and real-time data fusion. *IEEE Transactions on Intelligent Transportation Systems, 24(3), 1450-1465.*

[15] Liao, B., Zhang, J., Wu, C., McIlwraith, D., and Chen, T. (2023). Deep sequence learning with auxiliary information for traffic prediction. Proceedings of the *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 537-546.*