iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VII **Month of publication:** July 2025

DOI: https://doi.org/10.22214/ijraset.2025.73273

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

UrbanFlow360: A Cloud-Native Architecture for Real-Time Urban Traffic Congestion Prediction

Akash Tiwari[1], Aashish Dewangan[2], Kishan Kumar Bouri[3], Yash Mathur[4]
Indian Institute of Technology Jodhpur Postgraduate Diploma in Data Engineering

Abstract: ***UrbanFlow360 is a cloud-native Software-as-a- Service (SaaS) platform for real-time traffic congestion prediction and urban traffic analytics. Developed using a modular mi- croservice architecture and containerized with Docker, the system integrates predictive machine learning pipelines and offers live analytics through an interactive Streamlit dashboard. This paper details the end-to-end development lifecycle of UrbanFlow360, including dataset preprocessing, model training, application de- velopment, containerization, and deployment on AWS Elastic Container Registry (ECR) and Amazon EC2. The architecture demonstrates best practices in reproducibility, portability, and scalability of intelligent transportation systems.***

Index Terms: ***Traffic prediction, cloud-native, AWS, machine learning, Streamlit, ECR, EC2, Docker, containerization, urban analytics, smart cities***

I. INTRODUCTION

Urban areas face mobility challenges due to rising vehicle volumes, infrastructure limitations, and unpredictable conges- tion. Traditional systems fall short in providing predictive insights.

UrbanFlow360 addresses this gap using cloud-native tech- nologies. It enables real-time traffic congestion prediction via a browser-based dashboard powered by Streamlit and deployed through AWS infrastructure.

II. TEAM AND CONTRIBUTIONS

1) Akash Tiwari: Cloud and DevOps lead — EC2, Docker, ECR, IAM roles.
2) Aashish Dewangan: Data preprocessing and ingestion — handled wide CSVs, JSONs, and schema design.
3) Kishan Kumar Bouri: ML pipeline development and training.
4) Yash Mathur: Streamlit frontend integration and UI/UX refinement.

III. DATASET DESCRIPTION

Data was collected from traffic logs across Delhi and Bangalore. A sample of the uploaded dataset is shown in Figure 1.

IV. PREPROCESSING PIPELINE

We designed a consistent preprocessing flow to normalize column names, fill missing values, and encode categorical features.

1) Missing value imputation

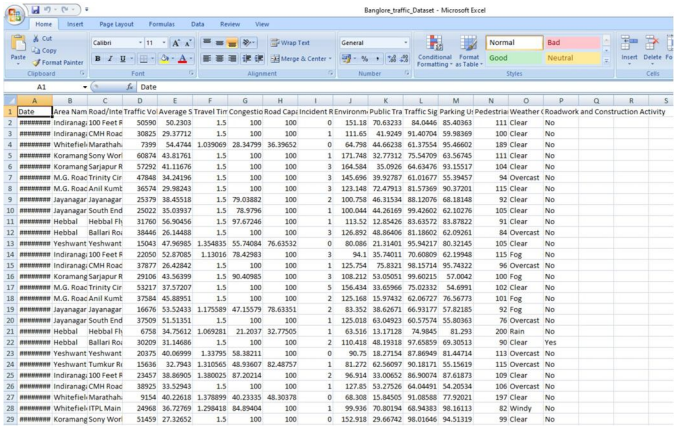


Fig. 1. Uploaded CSV preview showing traffic metrics

©IJRASET: All Rights are Reserved

2) Time and weekday feature extraction
3) Encoding and scaling
Figure 2 visualizes this pipeline.

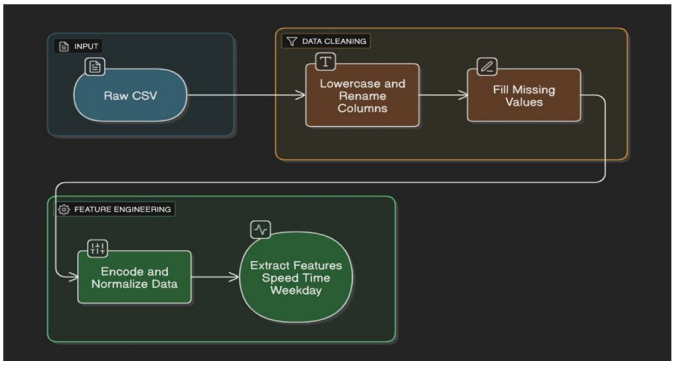


Fig. 2. Preprocessing stages for input data

V. ARCHITECTURE OVERVIEW

UrbanFlow360's modular architecture includes preprocess- ing, model inference, UI, and cloud deployment.

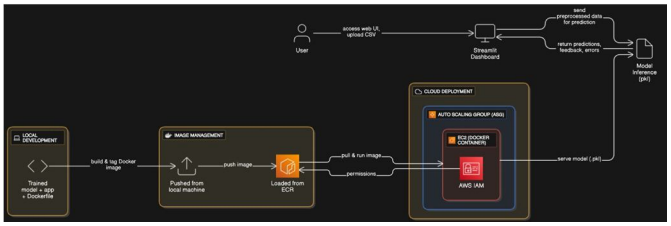


Fig. 3. UrbanFlow360 system architecture

VI. MODEL DEVELOPMENT

A. *Algorithm*

We used Random Forest Regressor for its interpretability and robustness.

B. *Training*

```
model = RandomForestRegressor()
model.fit(X_train, y_train)
joblib.dump(model, "model.pkl")
```

C. *Evaluation*

We measured accuracy using:

- Mean Absolute Error (MAE)
- Root Mean Square Error (RMSE)
- R2 score

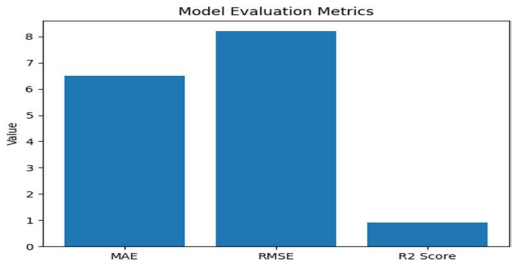


Fig. 4. Model performance metrics

VII. FRONTEND AND STREAMLIT INTEGRATION

Users interact via a Streamlit-based dashboard.

```
if predict_button:
    result_df = predict_csv_congestion(df, city)
    st.dataframe(result_df)
    st.bar_chart(result_df['Congestion'].
        value_counts())
```

Figure 5 shows file upload and city selection.
Figure 6 shows the prediction output with congestion chart.

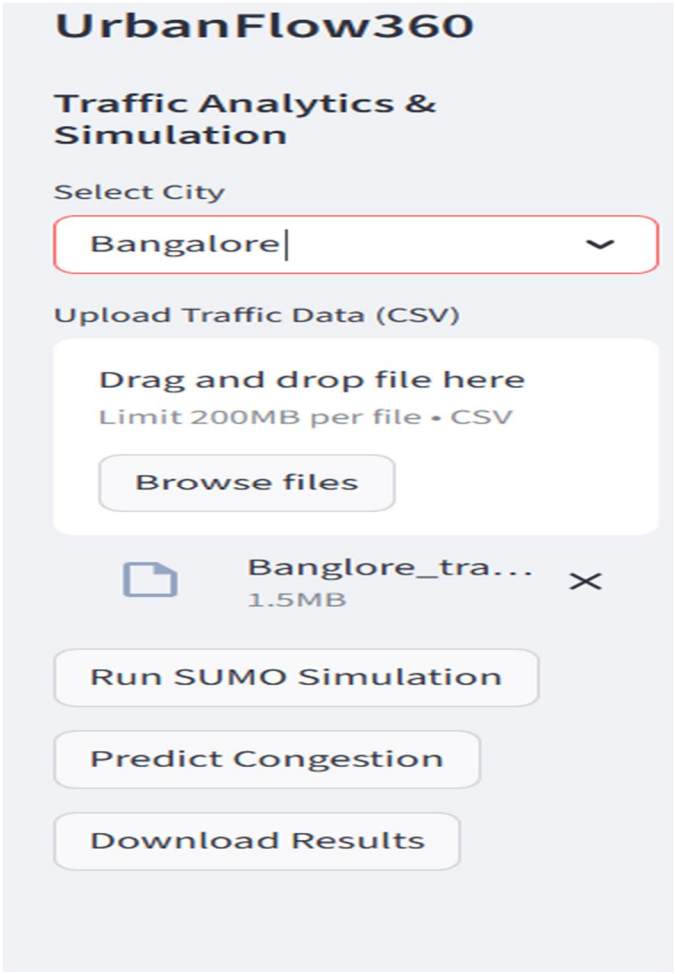


Fig. 5. Streamlit sidebar for file upload and city selection

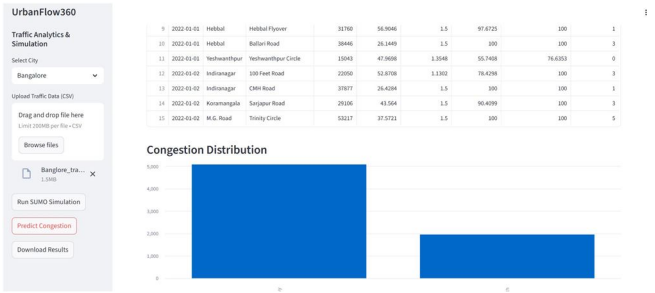


Fig. 6. Predicted congestion and bar chart output

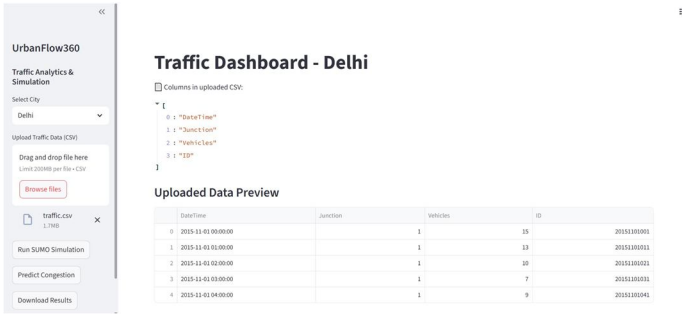


Fig. 7. Complete UI view after running predictions

VIII. DOCKER CONTAINERIZATION

UrbanFlow360 is packaged in Docker for reproducibility.

A. *Dockerfile*

```
FROM python:3.10-slim
WORKDIR /app
COPY . .
RUN pip install -r requirements.txt
CMD ["streamlit", "run", "frontend/app.py"]
```

B. *Push to ECR*

```
docker build -t urbanflow360 .
docker tag urbanflow360:latest \
  631158448576.dkr.ecr.ap-south-1.amazonaws.com/
      urbanflow360-backend:latest
docker push 631158448576.dkr.ecr.ap-south-1.
    amazonaws.com/urbanflow360-backend:latest
```

IX. AWS DEPLOYMENT (EC2 + ECR)

We deployed the model as a Docker container on EC2 after pushing it to ECR.

1) Create IAM role and ECR repo
2) Push Docker image to ECR
3) Launch EC2, SSH, pull image
4) Run app: docker run -p 8501:8501 urbanflow360

X. RESULTS

The app performed well on city-level congestion predictions.

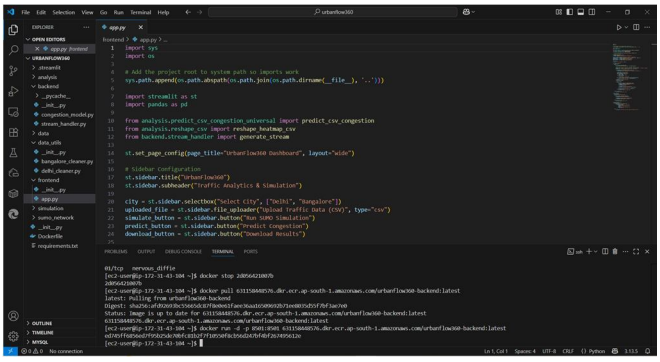


Fig. 8. Docker image pushed to AWS ECR

©IJRASET: All Rights are Reserved

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

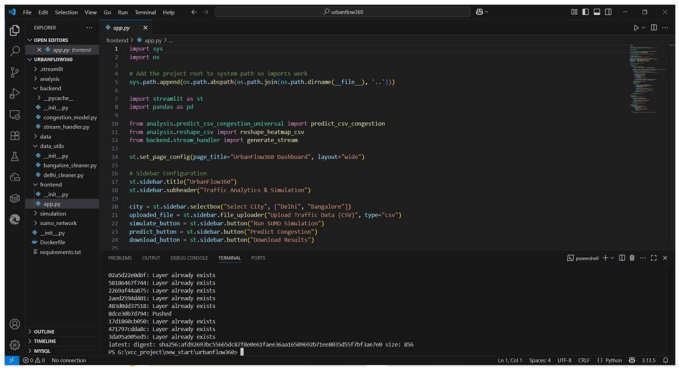


Fig. 9. SSH terminal on EC2 running Docker container

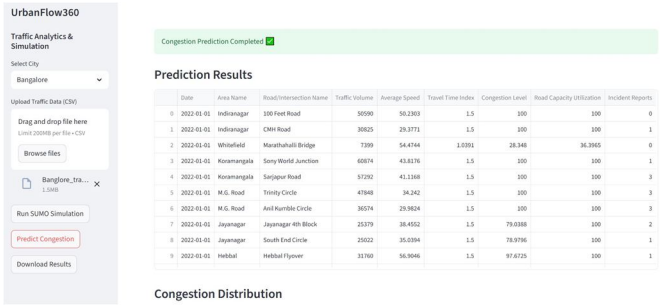


Fig. 10. Deployed dashboard accessed via EC2 public IP

Area	Avg Speed (km/h)	Predicted Congestion (%)
Koramangala	43.8	92.1
M.G. Road	29.9	85.5
Whitefield	54.5	21.2

TABLE I
SAMPLE CONGESTION PREDICTIONS FOR BANGALORE



Fig. 11. Bar chart visualization of congestion levels

XI. COMPARISON WITH EXISTING TOOLS

1) Dashboards like TomTom offer historic data, not predic- tions.
2) Our tool allows CSV uploads + real-time ML predictions.
3) Dockerized, portable, and open-source.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

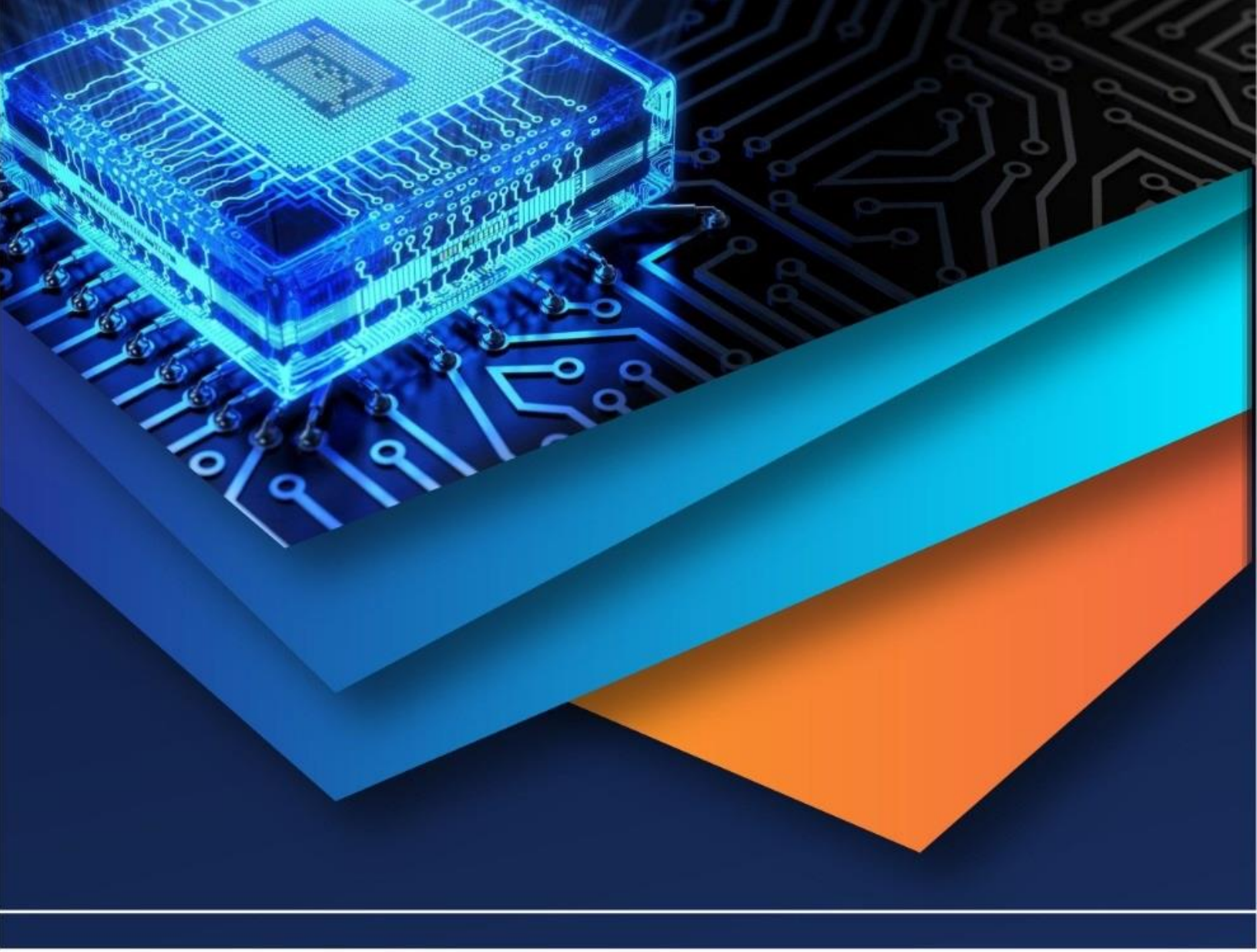
XII. CONCLUSION

UrbanFlow360 bridges cloud computing and urban mobility analytics. It is portable, scalable, and accessible to both cities and researchers. Future work includes multi-city support, live APIs, and SUMO integration for simulation.

REFERENCES

[1] DLR SUMO - https://sumo.dlr.de/docs/
[2] AWS Docs - https://docs.aws.amazon.com/
[3] Streamlit - https://docs.streamlit.io/
[4] Scikit-learn - https://scikit-learn.org
[5] Docker Docs - https://docs.docker.com/
[6] GitHub Repo - https://github.com/KishanBouri/urbanflow360

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)