Traffic Flow Prediction Using 1D Convolutional Neural Networks

Final Report: Methodology, Architecture, and Performance Analysis on the METR-LA Dataset.

Project Status: Complete | Model Type: Sequential 1D CNN | Framework: TensorFlow/Keras

NotebookLM

Executive Summary

Core Objective

Forecast multi-sensor traffic flow using historical `METR-LA data` to predict future congestion patterns across the network.

Methodology

Implemented a `1D Convolutional Neural Network (CNN)` treating traffic data as a temporal sequence. Enhanced raw data with cyclical temporal embeddings (`Sine/Cosine hours`) to capture daily rhythms.

Key Results

- **Stability**: Successfully learned macro-level daily periodicity.
- **Performance**: Achieved an R2 Score of `0.538` on the test set.
- **Insight**: Excellent at network-wide averages; **conservative on high-frequency anomalies**.

NotebookLM

Data Ingestion & Cleaning Strategy

Context:

The METR-LA dataset contains significant sensor noise.

The Problem:

Raw data contains numerous '0.0' values. In traffic systems, zero typically indicates sensor failure, not an empty road.

Logic:

1. Replace `0.0` with `NaN`.
2. Linear Interpolation for continuity.
3. Forward/Back fill for edge cases.

Pipeline Step 1: Handling Failures

Raw Vector

`[55.2, 0.0, 0.0, 60.5]`

↓

Mark Missing

`[55.2, NaN, NaN, 60.5]`

↓

Interpolate & Fill

`[55.2, 56.9, 58.7, 60.5]`

NotebookLM

Feature Engineering: Temporal Embeddings

The Challenge:

Neural networks perceive standard time (0-23 hours) linearly.

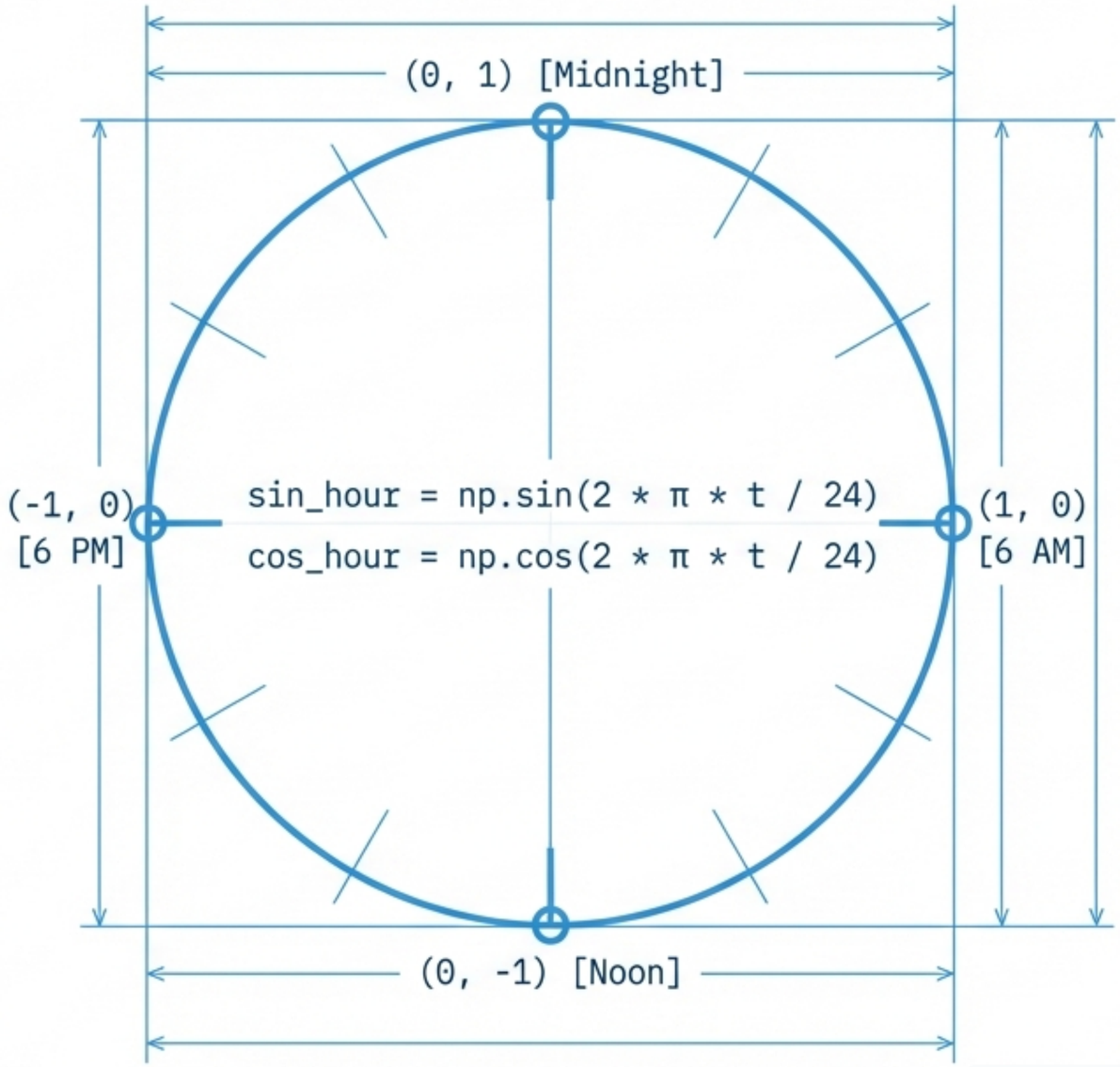
Numerically, **23 (11 PM)** is far from **0 (Midnight)**, despite being adjacent in time.

The Solution:

Cyclical Transformation. We map time onto a unit circle using Sine and Cosine functions.

Impact:

Preserves the periodic nature of the 24-hour day, crucial for learning rush hour patterns.



NotebookLM

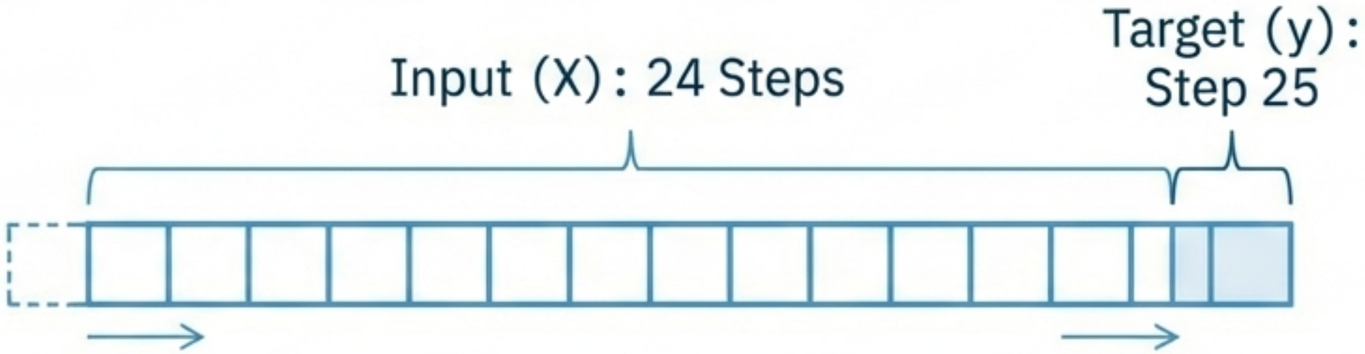
Sequence Generation & Scaling

Scaling Strategy

`MinmaxScaler (0-1)` applied to sensor data only.

Fit only on **Train set** to prevent leakage. Temporal features (`Sin/(Cos)` **excluded** as they are naturally scaled (-1 to 1).

Windowing Logic



Sequence Length: 24 (2 hours).

Dataset Split

Train: ~27,400 samples (80%)
Test: ~6,800 samples (20%)

Train: ~27,400 samples (80%)

Input Shape: (Batch, 24, 209)

NotebookLM

Model Architecture: The 1D CNN Stack

Input Layer
Shape: (24, 209)

↓

Conv1D Block 1
64 Filters, Kernel 3 + BatchNormalization + ReLU

↓

Conv1D Block 2
64 Filters, Kernel 3 + BatchNormalization + ReLU + Dropout(0.2)

↓

Conv1D Block 3
32 Filters, Kernel 3 + BatchNormalization + ReLU

↓

Flatten Layer

↓

Dense Layer
128 Units + Dropout(0.3)

↓

Output Layer
207 Units (Sensors)

1D Convolutions extract local temporal trends efficiently.

NotebookLM

Training Strategy & Dynamics

Configuration

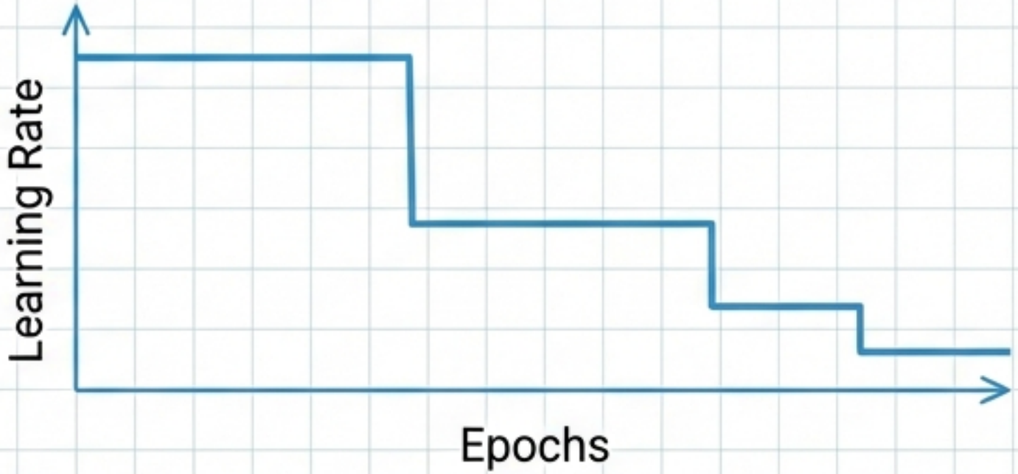
Epochs:	50
Optimizer:	Adam
Loss Function:	Mean Squared Error (MSE)
Batch Size:	32

Adaptive Logic

Callback: ReduceLROnPlateau

Monitor: `val_loss`

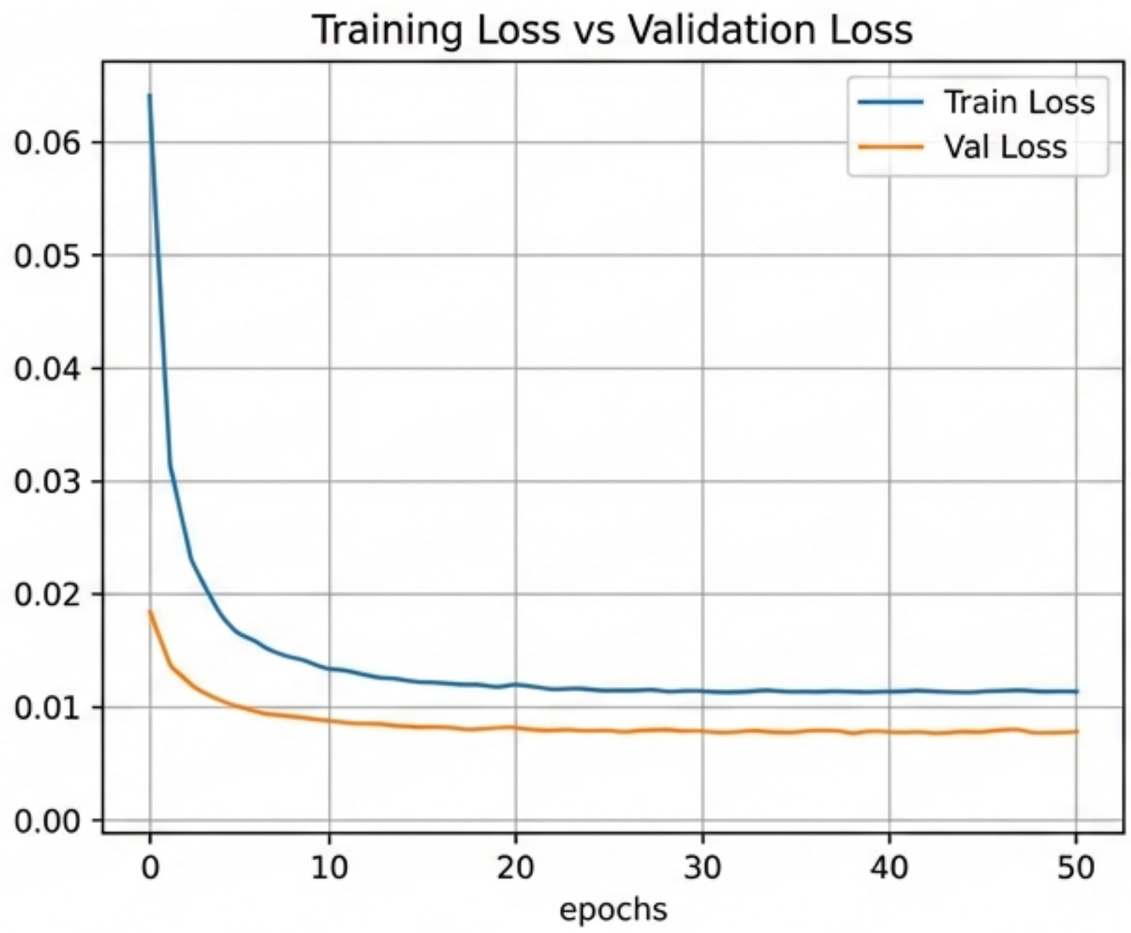
Action: Factor `0.5` if stalled for `3` epochs



Effective fine-tuning: LR stepped down from `0.001` to `0.00001` during training.

NotebookLM

Convergence Analysis

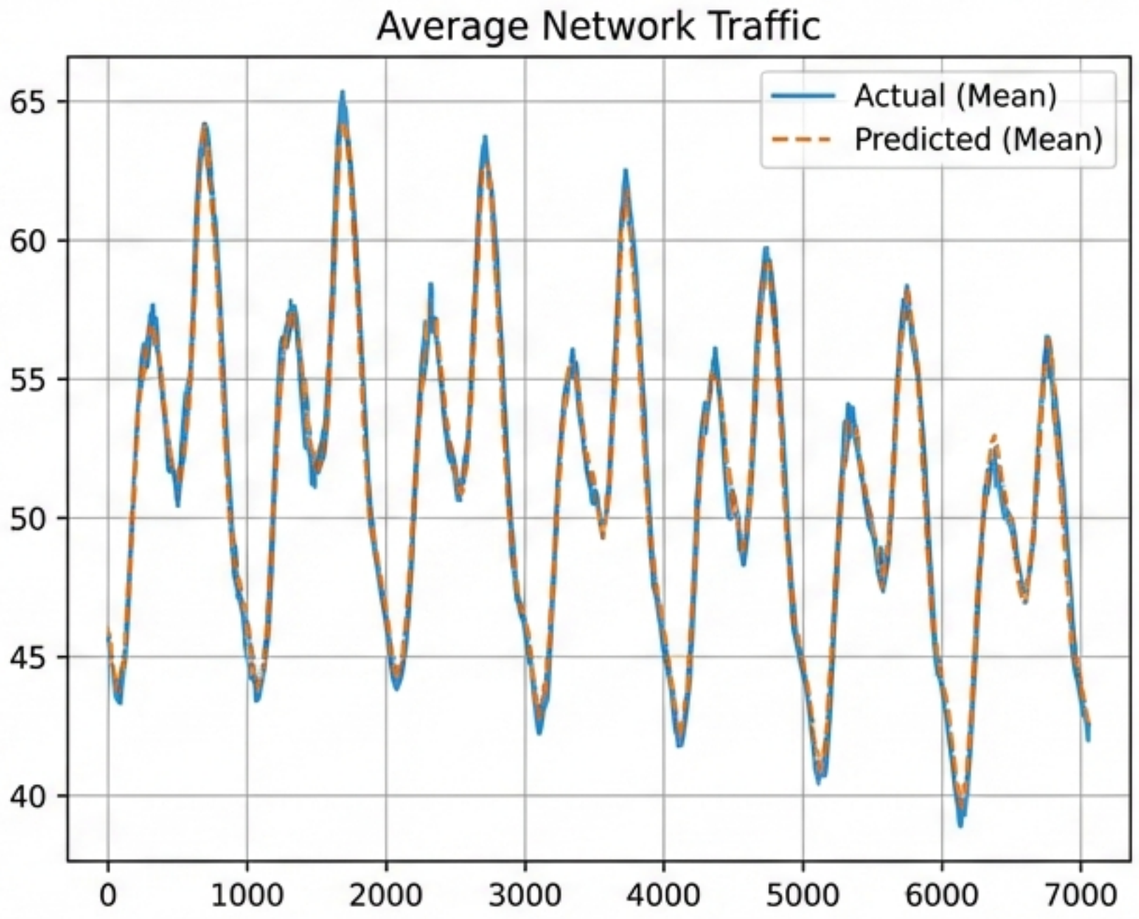


- **Initial Drop:** Rapid descent in first 5 epochs indicates quick learning of dominant daily cycles.
- **Stability:** Both curves stabilize around `0.01` MSE.
- **Generalization:** The consistent gap with Validation Loss lower than Training Loss (likely due to Dropout) confirms no massive overfitting.

NotebookLM

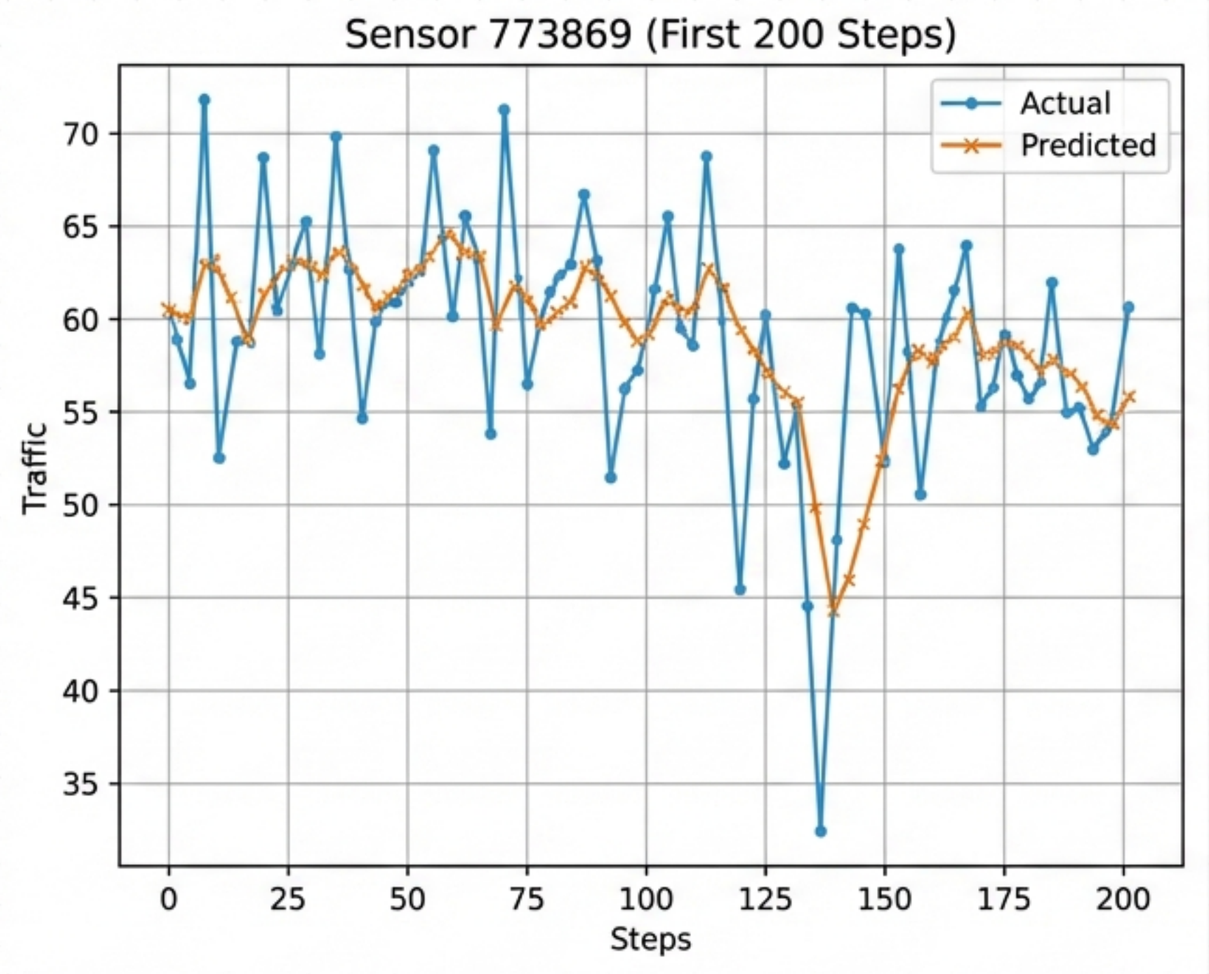
Network-Wide Traffic Trends

- **Macro Performance**: The model excels at predicting the aggregate behavior of the city.
- **Observation**: Nearly perfect overlap on daily peaks (rush hours) and troughs (nighttime).
- **Validation**: Confirms the effectiveness of Cyclical Temporal Embeddings.



NotebookLM

Sensor-Level Granularity



- **Micro Performance:** Zooming into a single sensor reveals limitations.
- **The Lag:** Model follows general shape but lags on sudden changes.
- **The Smoothing Effect:** Typical of MSE loss optimization, the CNN acts as a low-pass filter, averaging out high-frequency noise and extreme anomalies.

NotebookLM

Quantitative Performance Metrics

MAE (Mean Absolute Error)

4.03

Avg deviation in traffic units.

RMSE (Root Mean Squared Error)

6.72

Penalizes outliers heavily.

R2 Score

0.538

Explains ~54% of variance.

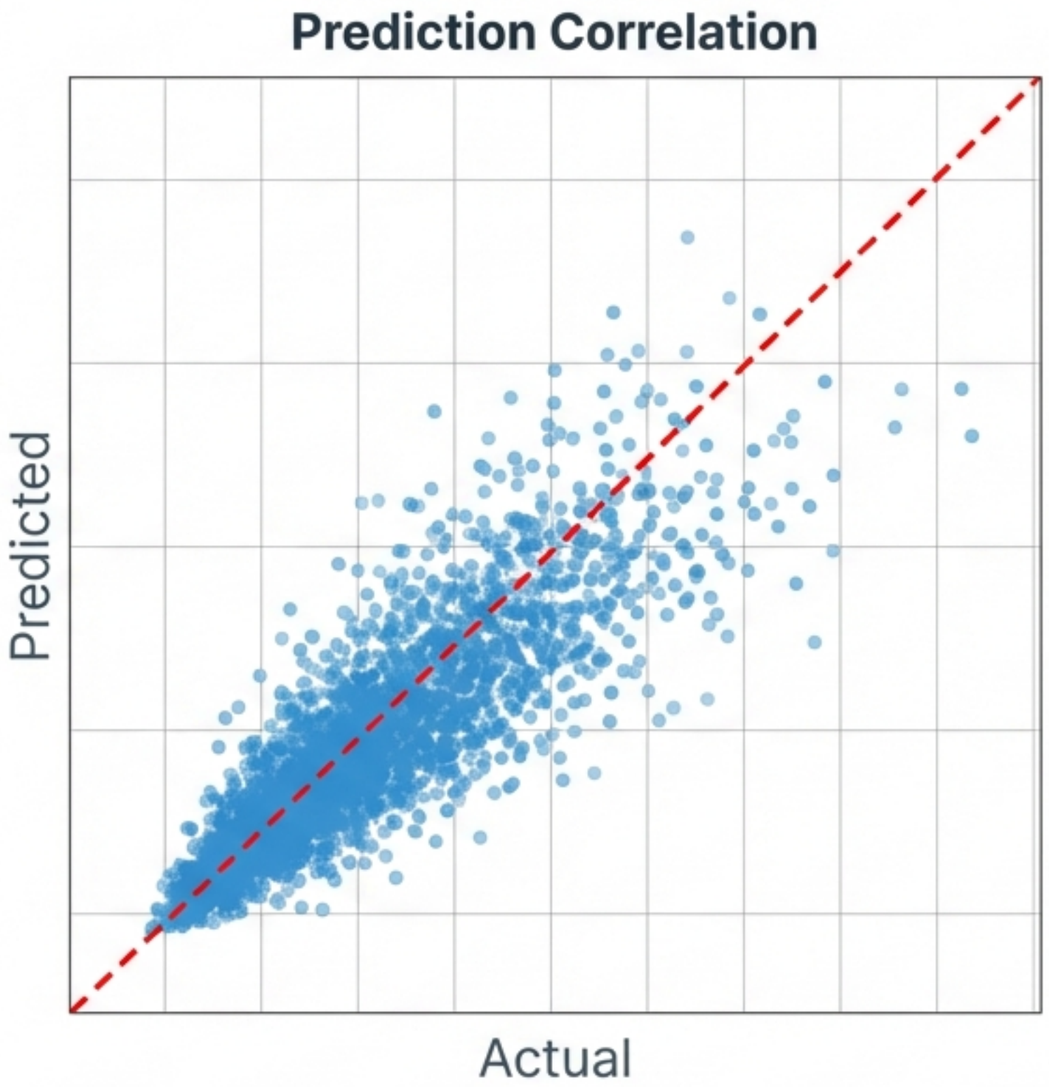
Metrics evaluated on Test Set (6,855 samples). Moderate R2 confirms trend capture but unexplained noise variance.

NotebookLM

Prediction Correlation & Variance

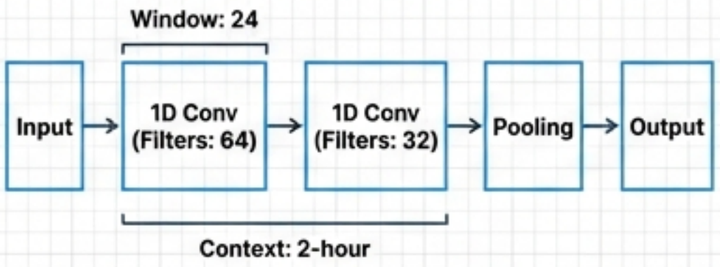
Visualizing Error Distribution:

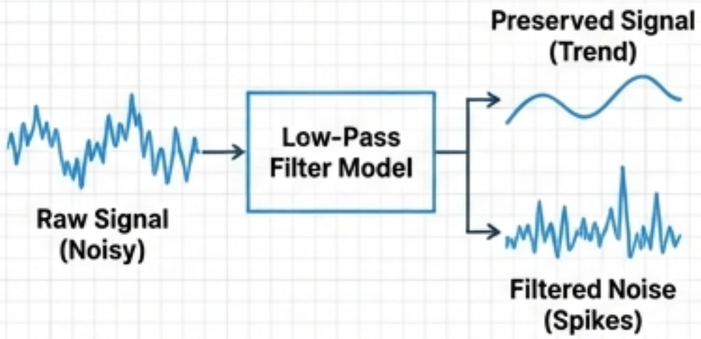
- **Ideal:** Points lie on the red diagonal.
- **Reality:** A visible cloud of dispersion explains the `0.54 R2 score`.
- **Bias:** Tendency to under-predict extreme high traffic values (points below line at top right).
- **Dispersion:** Vertical spread represents unmodelled sensor noise.

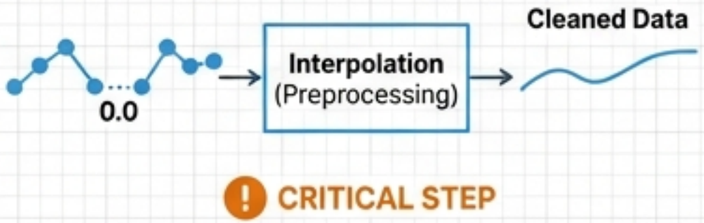


NotebookLM

Key Technical Insights







1. Architecture Validation

The **1D CNN** proved highly effective at feature extraction for periodic trends. The **window size (24)** and **filter depth (64/32)** successfully captured the 2-hour immediate history context.

2. The Smoothing Effect

The model acts as a `low-pass filter`. It successfully preserves the '**signal**' (daily traffic patterns) but filters out the '**noise**' (erratic sensor spikes), which is desirable for trend forecasting but limits anomaly detection.

3. Data Handling Criticality

`Interpolation` of '`0.0`' sensor failures was the single most **critical** preprocessing step. Without it, the model would have learned artificial traffic drops, **degrading performance significantly**.

NotebookLM

Conclusions & Future Directions

Summary: We successfully deployed an end-to-end pipeline—from cleaning sensor failures to training a deep convolutional network—resulting in a robust baseline model (R2 ~0.54) that captures city-wide traffic rhythms.

Recommendations for Improvement

Spatial Awareness

Integrate **Graph Neural Networks (GNNs)** to model physical road connectivity instead of treating sensors in isolation.

Long-Term Dependency

Augment **CNN** with **LSTMs** or **Transformers** to capture dependencies beyond the 2-hour window.

External Features

Incorporate weather and holiday data to explain variance not captured by historical flow.

NotebookLM