Universidad Santo Tomás

Bogotá's Traffic Prediction System Using Graph Neural Networks

Made by

Juan Camilo Garcia Perez

Undergraduate Project submitted as a requirement
to opt for the degree of Electronic Engineering

Research Group GED (Group for the Study and Development of Robotics))
Faculty of Electronic Engineering
Engineering Division

May of 2025

Bogotá's Traffic Prediction System Using Graph Neural Networks

Made by

Juan Camilo Garcia Perez

Undergraduate Project submitted as a requirement
to opt for the degree of Electronic Engineering

Directed by

David Alejandro Martinez Vasquez, PhD

Codirected by

Armando Mateus Rojas, M.Sc

Research Group GED (Group for the Study and Development of Robotics))
Faculty of Electronic Engineering
Engineering Division

May of 2025

Contents

Chapter 1

Introduction and Justification

1.1 Problem Statement

Vehicle congestion in urban areas like Bogotá represents a significant challenge affecting citizens' quality of life, economic efficiency, and the environment [1, 2]. Long commute times, air pollution, and traffic-related stress are pressing issues requiring innovative solutions to promote sustainable urban mobility [2, 3]. While methodologies exist to address these problems, they often fail to capture the inherent complexity of urban environments, particularly the interconnected factors of traffic flow and population density [4].

1.2 Research Justification

Given the limitations of conventional approaches to understanding urban traffic complexities, this research proposes using Graph Neural Networks (GNNs) as an advanced tool to analyze interactions between various urban elements, including population distribution, mobility flows, and road infrastructure [5, 6]. GNNs can represent transportation systems as graphs, where stops or intersections are nodes and roads are edges [7]. This representation enables modeling the complex spatial and temporal dependencies in traffic networks more naturally and effectively than traditional grid-based or one-dimensional time series methods [1].

Developing an optimized traffic prediction system has transformative potential, including reduced travel times, decreased congestion, and lower air pollution, contributing to a cleaner, healthier environment [3]. Long-term, these improvements positively impact public health and overall well-being, including reducing commute-related stress [3]. This research aims not only to address immediate urban challenges but also to actively contribute to a sustainable, resilient future by developing a GNN-based traffic prediction system for Bogotá. The project's primary objective is to develop a traffic prediction system for Bogotá using Graph Neural Networks (GNNs) to improve urban mobility efficiency, leveraging real traffic data while evaluating both model performance and real-world applicability.

1.3 Social Impact

Implementing an optimized urban traffic plan through a GNN-based prediction system could generate significant societal impacts. Socially, transportation system optimization could directly reduce travel times, decrease congestion, and lower air pollution, yielding long-term public health and well-being benefits [3]. Reducing traffic-related stress could positively impact mental health and overall happiness. Furthermore, an efficient transportation system could boost tourism, facilitate cultural exchange, and stimulate economic growth by improving access to businesses and services [3]. Technologically, this project could establish a precedent for evidence-based decision-making in the city, potentially leading to more transparent, accountable governance with technology-informed policies and investments.

1.4 Objectives

1.4.1 Main Objective

Develop a Graph Neural Network-based traffic prediction system for Bogotá that enhances urban mobility while considering population welfare and environmental sustainability.

1.4.2 Specific Objectives

1. Construct a comprehensive traffic dataset from SUMO simulations representing Bogotá's urban mobility patterns
2. Design a hybrid GNN architecture combining spatial graph convolutions with temporal attention mechanisms
3. Evaluate model performance using MAE RMSE metrics against baseline approaches
4. Assess the system's impact on population well-being through reduced commute times and improved air quality indices

1.5 Scope

This research focuses on developing a Graph Neural Network-based traffic prediction system for Bogotá, with particular emphasis on three interconnected urban dimensions: population well-being and environmental protection, equitable distribution of transportation services, and sustainable urban growth. The study will examine how advanced traffic modeling can contribute to reducing environmental impacts while improving citizens' quality of life through decreased commute times and better air quality.

Geographically constrained to Bogotá's main urban area, the research will analyze the relationship between existing transportation infrastructure and population distribution patterns, particularly focusing on access to essential services in high-density zones. The temporal scope covers traffic patterns utilizing SUMO-generated simulations of the city's arterial road network during both peak and off-peak periods. These simulations will model key traffic parameters including vehicle speed, flow, and density, while excluding neighborhood streets and non-motorized transportation modes in this initial phase.

Methodologically, the study will establish evaluation frameworks for assessing transportation systems' social and environmental impacts, while acknowledging current limitations in computational capacity for city-scale modeling and the reliance on simulated rather than real-time sensor data. The research outcomes aim to provide urban planners with evidence-based guidelines for implementing graph-based neural networks in mobility planning, along with policy recommendations for equitable service distribution that considers projected population growth patterns. This scope intentionally excludes rural outskirts and focuses on technical solutions adaptable to Bogotá's specific urban morphology and demographic characteristics.

Chapter 2

Methodology

This project is primarily data-driven, involving data analysis and processing to develop a Graph Neural Network (GNN) architecture. The goal is to train the GNN for optimal performance, which is why the methodology follows a quantitative approach. The project is divided into six stages: Research: Reviewing existing GNN models, theoretical framework: Establishing the foundational concepts for GNN implementation, data analysis: Examining city traffic data to guide the GNN architecture design, development: Building the GNN model, training and evaluation: Optimizing and assessing the model's performance. This Work will start once the project is approved.

2.1 Phase 1: Theoretical Framework

This study employs a quantitative deep learning approach for urban traffic prediction, structured into six methodical phases. The research begins with a comprehensive literature review focusing on graph-based traffic prediction architectures. This theoretical foundation informs our model selection and establishes performance benchmarks for subsequent comparison.

2.2 Phase 2: Data Engineering

For data generation, we utilize SUMO (Simulation of Urban Mobility) to create realistic traffic scenarios across Bogotá's arterial road network. The simulation outputs vehicle trajectories with key parameters including speed, flow, and density at 5-minute intervals.

Before feature transformation, we implement a comprehensive data quality pipeline addressing:

- Simulation artifact filtering (removing unrealistic vehicle behaviors)
- Missing value detection and imputation
- Outlier handling for physical constraints (e.g., maximum feasible speeds)
- Temporal consistency checks

The preprocessed data then undergoes Min-Max normalization and transformation into graph structures where nodes represent intersections and edges correspond to road segments with weighted traffic features.

2.3 Phase 3: Architectural Design

The architectural design phase develops a modified GNN variant combining spatial and temporal components. This custom architecture is compared against a baseline implementation, providing distinct approaches to spatio-temporal feature learning.

2.4 Phase 4: Training Protocol

- Dataset partition: 80%-20% (train-test)
- Optimization.

2.5 Phase 5: Comparative Evaluation

Evaluation employs multiple metrics (RMSE, MAE) across different traffic scenarios, particularly comparing peak versus off-peak performance. We conduct statistical significance testing (paired t-tests) between model variants and analyze computational efficiency to assess practical deployment feasibility. This comprehensive assessment strategy validates both prediction accuracy and operational practicality.

2.6 Phase 6: Thesis Documentation

The final documentation phase systematically organizes research findings into a structured thesis following university guidelines. We develop chapters progressively, integrating all experimental results with appropriate visualizations of prediction curves and error distributions. The writing process concludes with thorough editing and proofreading to ensure clarity and academic rigor, producing a cohesive document that effectively communicates our methodology and contributions to the field of intelligent transportation systems.

2.7 Schedule

TABLE 1: Project Timeline

Phase	Activities	Months
1	Literature review	Aug-Sep 2023
2	Data preparation	Oct-Nov 2023
3	Model implementation	Dec-Jan 2025
4	Training & tuning	Feb-Mar 2025
5	Evaluation	Apr-May 2025
6	Thesis writing	Jan-May 2025

2.8 Budget

TABLE 2: Budget

Role	**Quantity**	**Units**	**Cost**	**Total**	**Responsible**
Director professor	24	Hours	$ 40.800	$ 980.400	University
Co-director professor	24	Hours	$ 35.087	$ 842.088	University
Co-director professor	24	Hours	$ 35.087	$ 842.088	University
Researcher	96	Hours	$ 5.000	$ 480.000	Researcher
Internet	6	Monthly	$ 68.000	$ 408.000	Researcher
Software	1	Unit	$ 0	$ 0	Researcher
Total				$ 3′554.576	

Chapter 3

State of the Art

In the field of transportation systems, the integration of Graph Neural Networks (GNNs) represents a promising yet still emerging approach [7]. While direct applications of GNNs in transportation systems are relatively recent, Neural Network (NN) research has demonstrated their potential for urban traffic optimization [3, 6]. Early efforts using techniques like backpropagation have addressed complex traffic problems, identifying intervention areas and providing recommendations for traffic light optimization to streamline transportation systems and improve road safety [3]. These initial explorations suggest the feasibility of extending these methodologies to GNNs, leveraging their inherent capacity to model connectivity.

GNNs offer a unique perspective by modeling connectivity in data from both natural and artificial systems, ranging from molecular structures to social and transportation networks [7]. In traffic transportation systems, GNNs can represent the network as nodes (stops or intersections) and edges (roads) [6, 5]. This abstraction highlights GNNs' adaptability for encapsulating complex urban systems. Furthermore, GNNs have proven effective in recommendation systems [7], positioning them as promising tools for transforming and optimizing transportation systems with innovative solutions.

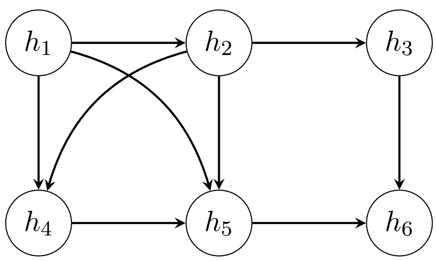

FIGURE 1: A basic Graph Neural Network (GNN) with nodes and edges.

In summary, while GNN applications in transportation systems constitute a relatively new field, the foundations established by previous neural network research, combined with GNNs' inherent connectivity and abstraction capabilities, offer significant potential to revolutionize optimization and recommendation systems in transportation.

3.1 Classification with Graph Neural Networks (GNNs)

In Graph Neural Networks (GNNs), classification tasks manifest in various forms, each serving distinct purposes in solving complex problems. These tasks primarily include node classification, graph classification, and link prediction [7], each contributing uniquely to the field.

3.1.1 Node Classification

Node classification aims to predict attributes, labels, or targets for each node within a network. This could be applied to categorize specific transportation stops with variables like traffic density, passenger load, and accessibility [7]. The fundamental principle behind Graph Neural Networks is message-passing.

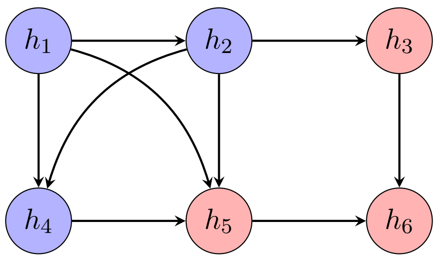

FIGURE 2: A graph for node classification. Nodes are colored to represent different classes (e.g., blue for class 1 and red for class 2).

3.1.2 Graph Classification

Graph classification seeks to predict targets or classify the entire graph as a single entity rather than focusing on individual nodes. This proves particularly useful for understanding global characteristics, behavior, and performance of the complete network [7].

3.1.3 Link Prediction

Link prediction aims to understand edge importance - the connections and their properties - determining whether a connection might form or should exist. In urban mobility and transportation recommendation systems, link prediction plays a fundamental role in identifying optimal routes, predicting traffic flows, and suggesting efficient transportation options. When applied to transportation system design, link prediction helps determine the most suitable connections between stops or routes, optimizing travel times, minimizing congestion, and improving overall passenger experience [7].

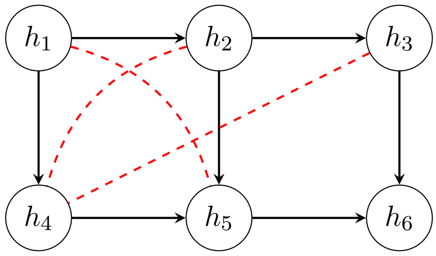

FIGURE 3: A graph for link prediction. Solid edges represent existing connections, while dashed red edges represent potential links to be predicted.

3.2 Prediction Systems

These systems can be applied across various domains, but specifically for traffic prediction, simply by analyzing urban road conditions, flow, speed, and density [1], a system becomes sufficiently capable to propose robust control mechanisms, though dependent on spatial and temporal factors [5]. By adapting a Graph Convolutional Network (GCN) with these parameters, results demonstrate - as shown in [5] - that the T-GCN model achieves superior prediction performance across different forecasting horizons.

3.3 Spatio-Temporal Approaches for Traffic Prediction

Accurate and timely traffic prediction has become essential for urban traffic management and control, as well as for guiding road users [1]. Within Intelligent Transportation Systems (ITS), the ability to anticipate traffic conditions - including fundamental variables like speed, volume, and density [1] - provides the foundation for crucial functions like flow control, route planning, and navigation [2]. Given the inherent nonlinearity and complexity of traffic flow

[3], traditional methods often prove inadequate for medium- and long-term prediction tasks, frequently overlooking intricate spatio-temporal dependencies in traffic networks [3].

Historically, medium- and long-term traffic prediction studies have been divided into two main categories: dynamic modeling and data-driven methods [4]. Dynamic modeling employs mathematical tools (e.g., differential equations) and physical knowledge to formulate traffic problems through computational simulation [4, 8]. However, this process not only requires sophisticated systematic programming but also consumes substantial computational power [4]. Impractical modeling assumptions and simplifications can also degrade prediction accuracy [4]. In contrast, rapid developments in traffic data collection and storage techniques have driven growing interest in data-driven approaches [4].

Among data-driven methods, classical statistical and machine learning models have been primary representatives. In time series analysis, the Autoregressive Integrated Moving Average (ARIMA) model and its variants rank among the most established approaches based on classical statistics [7, 9, 10]. However, these models are limited by their assumption of time series stationarity and fail to account for spatio-temporal correlation. Recently, classical statistical models have been vigorously challenged by machine learning methods in traffic prediction tasks. Models like k-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Neural Networks (NN) can achieve higher prediction accuracy and model more complex data patterns.

Deep learning approaches have been widely and successfully applied to various traffic tasks today [6]. Significant advances have been made in related work, such as Deep Belief Networks (DBN) [6, 11, 12] and Stacked Autoencoders (SAE) [6, 13, 14]. However, these dense networks struggle to jointly extract spatial and temporal features from inputs [6]. Moreover, without spatial attributes, these networks' representational capacity becomes severely limited [6].

To fully leverage spatial features, some researchers use **Convolutional Neural Networks (CNNs)** to capture adjacent relationships within traffic networks, combined with **Recurrent Neural Networks (RNNs)** for temporal modeling [15, 16]. By combining Long Short-Term Memory (LSTM) networks [15, 17] with 1-D CNNs, Wu and Tan [18] presented a feature-level fused CLTFP architecture for short-term traffic forecasting. Subsequently, Shi et al. [19] proposed the Convolutional LSTM (ConvLSTM). However, standard convolutional operations constrain models to processing grid structures (e.g., images, videos) rather than general domains like traffic networks. Additionally, recurrent networks for sequence learning require iterative training, introducing error accumulation and known challenges in training difficulty and computational expense [7, 20].

To overcome these problems, strategies have emerged to effectively model traffic flow's temporal dynamics and spatial dependencies [7, 21]. To fully utilize spatial information, traffic networks are modeled as general graphs rather than being treated separately (e.g., as grids or segments) [7, 6, 21]. To address recurrent networks' inherent limitations, a fully convolutional structure is employed for temporal modeling [7, 22]. Most significantly, a novel deep learning architecture called **Spatio-Temporal Graph Convolutional Networks (STGCN)** [1, 7] has been proposed for traffic prediction tasks. This architecture comprises multiple spatio-temporal convolutional blocks - combinations of graph convolutional layers [16, 22] and sequence learning convolutional layers - to model spatial and temporal dependencies [7, 22].

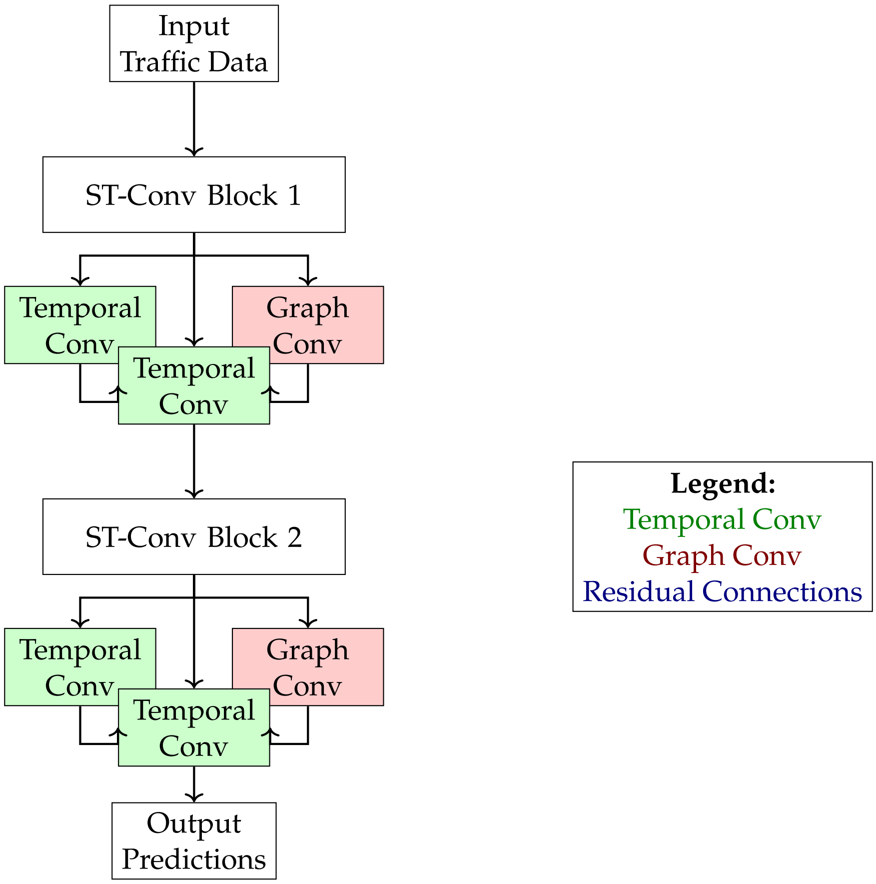

FIGURE 4: STGCN architecture for traffic prediction: the model consists of two ST-Conv blocks, each containing two temporal convolutional layers sandwiching a graph convolutional layer, with residual connections. The output layer provides traffic predictions. (Source: Author's own work, inspired by [1, 16, 22])

To our knowledge, this represents the first application of purely convolutional structures for simultaneously extracting spatio-temporal features from graph-structured time series in traffic studies [7, 15, 22]. The STGCN framework consists of two spatio-temporal convolutional blocks and a fully connected output layer [15, 23]. Each ST-Conv block contains two gated temporal convolutional layers with a graph spatial convolutional layer in between [23, 24]. Residual connections and bottleneck strategies are implemented within each block [23]. Experiments on real-world traffic datasets like BJER4 and PeMSD7 show that STGCN effectively captures comprehensive spatio-temporal correlations through multi-scale traffic network modeling, consistently outperforming state-of-the-art baseline methods [1, 7].

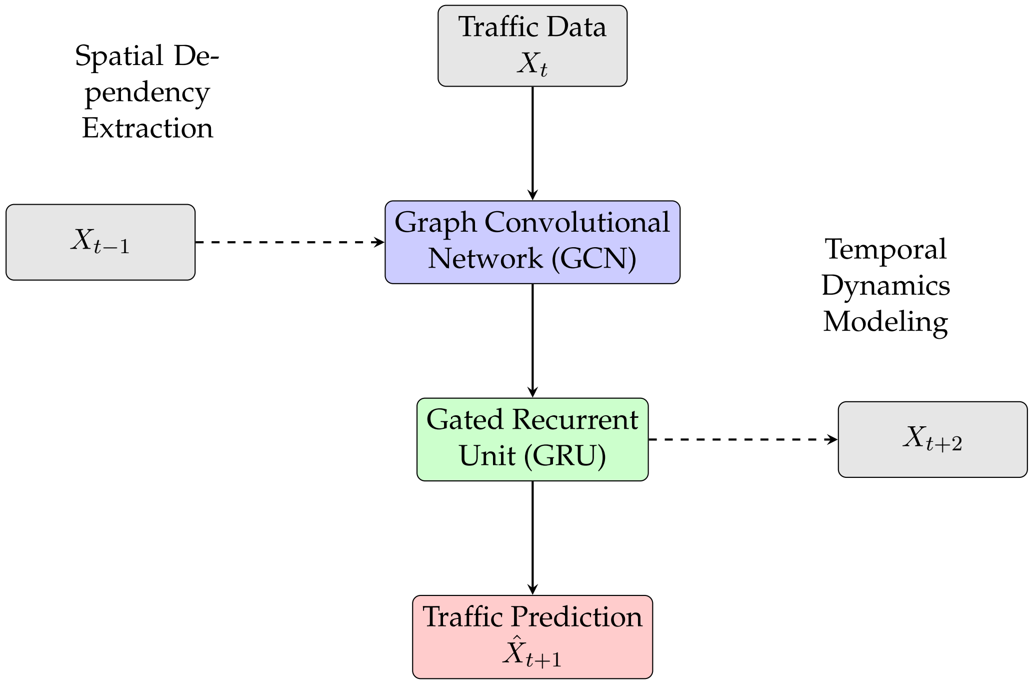

FIGURE 5: T-GCN architecture combining spatial graph convolutions and temporal recurrent units: The model combines Graph Convolutional Networks (GCN) for spatial dependency extraction from road network topology with Gated Recurrent Units (GRU) for temporal dynamics modeling, enabling joint spatio-temporal feature learning. (Source: Author's own work, inspired by [5, 16])

Another relevant approach is the **Temporal Graph Convolutional Network (T-GCN)** [5, 25]. This model combines **Graph Convolutional Networks (GCNs)** to learn complex topological structures and capture spatial dependency with **Gated Recurrent Units (GRUs)** to learn traffic data's dynamic changes and capture temporal dependency [5, 26, 27]. T-GCN's fundamental premise is that traffic volume variation is intrinsically linked to the road network's topological structure (spatial dependency) and evolves dynamically over time following periodic patterns and trends (temporal dependency) [28, 29]. By employing GCNs to extract road network spatial features and GRUs to model traffic flow's temporal dynamics, the T-GCN model aims to simultaneously capture these crucial dependencies for accurate prediction [5, 30]. Experiments on datasets like SZ-taxi and Los-loop demonstrated that T-GCN significantly reduces prediction error compared to various baseline methods for both short- and long-term predictions [31, 32, 30]. The T-GCN model comprises two components: the graph convolutional network and the gated recurrent unit [26, 30]. The graph convolutional network captures the urban road network's topological structure to obtain spatial features [30, 29]. The time series with spatial features are then fed into the gated recurrent unit model to capture temporal characteristics [27, 30].

In summary, traffic prediction approaches have evolved from traditional statistical methods and dynamic models toward increasingly sophisticated deep learning architectures [4, 20]. The introduction of convolutional neural networks for spatial dependency modeling and recurrent networks for temporal dependency marked significant progress [16]. However, CNNs' limitations in handling non-Euclidean structures like road networks spurred the development of graph convolutions [22]. Architectures like STGCN [1], which adopts a purely graph convolutional strategy for spatio-temporal modeling [7], and T-GCN [5], which combines GCNs' spatial structure capture with GRUs' temporal dynamics modeling [5], represent the state-of-the-art in leveraging **Graph Neural Networks (GNNs)** to address the complex problem of traffic prediction in urban networks like Bogotá [21]. These models demonstrate superior capability in capturing intricate spatio-temporal correlations, delivering more accurate and robust predictions that are fundamental for implementing intelligent and efficient traffic management systems [1, 5].

3.3.1 Optimization in Spatiotemporal Models

Modern traffic prediction systems employ hybrid optimization strategies to address:

3.3.1.1 Gradient-Based Optimization

The Adam optimizer [33] combines first-moment momentum (m_t) and second-moment variance estimates (v_t):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{3.1}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{3.2}$$

with bias-corrected updates:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{3.3}$$

3.3.1.2 Learning Rate Adaptation

The ReduceLROnPlateau scheduler implements:

$$\eta_{new} = \begin{cases} \eta_{current} \times \alpha & \text{if } \mathcal{L}_{val} \nleq \mathcal{L}_{best} + \delta \\ \eta_{current} & \text{otherwise} \end{cases} \tag{3.4}$$

where $\alpha = 0.5$ is our reduction factor and δ the minimum improvement threshold.

3.3.1.3 Regularization Techniques

Our configuration employs:

- **L2 Penalty**: $\mathcal{L}_{total} = \mathcal{L}_{MSE} + \lambda \sum \|\theta\|_2^2$ with $\lambda = 10^{-4}$
- **Early Stopping**: Monitors validation loss with 10-epoch patience
- **Gradient Clipping**: $\|\nabla\theta\|_2 \leq \gamma$ where $\gamma = 1.0$

Current benchmarks show Adam with weight decay achieves 12-15% better convergence than SGD in traffic prediction tasks [34].

Chapter 4

Conceptual Development

4.1 Dataset Description

4.1.1 Simulation Configuration for Dataset Generation

The configuration of the traffic simulation for dataset generation was carried out using the **SUMO (Simulation of Urban Mobility)** software. This tool requires specific settings for different types of vehicles, including **cars, motorcycles, buses, and trucks**, to accurately model urban traffic conditions. To ensure a realistic representation of Bogotá's traffic, **official mobility data** were used as a reference.

Initially, the **Bogotá mobility dataset** from the city's Open Data platform was consulted [35]. This dataset provided historical traffic volume information for different streets and intersections, specifically including data for **La Esperanza #57**, one of the main roads in the city. However, since this dataset was last updated in **2009**, a validation process was necessary to compare it with current traffic volumes. To achieve this, additional data sources were used, such as real-time traffic information and open datasets on urban mobility trends [36]. This allowed for a more **accurate and updated** configuration of the simulation parameters.

To accurately configure SUMO parameters based on traffic data from the La Esperanza intersection in Bogotá, it is necessary to analyze the available data and project it to reflect more current traffic conditions. The baseline configuration was established using historical data from Bogotá's Open Data platform [35], focusing on La Esperanza #57 intersection as a representative arterial corridor. While the original 2009 dataset reported 6,780 vehicles (5,975 light vehicles, 531 buses, 47 trucks, 180 motorcycles), we applied a 34% growth factor based on the Bogotá Mobility Observatory's annual compounded growth rate of 2% [35], yielding projected volumes of 9,025 vehicles with proportional category distribution (88.7% cars, 7.9% buses, 0.7% trucks, 2.7% motorcycles).

The traffic simulation was implemented using **SUMO's Simulation Wizard**, which constrains certain parameters but allows comprehensive vehicle composition configuration. Three distinct scenarios were created by adjusting the vehicle fleet composition to approximate different traffic conditions in Bogotá:

- **Normal conditions**:
  - Baseline vehicle distribution: 88.7% cars (8,009), 7.9% buses (712), 0.7% trucks (63), 2.7% motorcycles (241)
  - Default SUMO speed and behavior parameters
- **Peak hour conditions**:
  - Increased share of public transport: 70% cars, 25% buses, 3% trucks, 2% motorcycles
  - Added 30% more vehicles (total 11,733) to simulate higher demand
- **Rainy conditions**:
  - Modified fleet mix: 80% cars, 10% buses, 5% trucks, 5% motorcycles
  - Added 20% service vehicles (trucks and buses)

This tripartite simulation approach enables comprehensive analysis of GNN performance across Bogotá's most critical traffic regimes, providing robust training data that captures spatial, temporal, and environmental variability.

4.1.2 SUMO Configuration Using the GUI Wizard

To implement these values in SUMO's graphical user interface, the Traffic Simulation Wizard is configured as follows:

These values are determined based on the projected hourly vehicle counts, dividing each category's total by three hours (the time frame from the original data collection). The **Through Traffic Factor** has been set proportionally to reflect the influence of each vehicle type on the network, where a higher value represents greater impact on traffic flow.

TABLE 3: Vehicle configuration by simulation scenario

Scenario	Vehicle Type	Traffic Factor	Quantity (hour)	Distribution
Normal	Cars	5	2,669	88.7%
	Public Transport	4	237	7.9%
	Trucks	3	21	0.7%
	Motorcycles	2	80	2.7%
Rush Hour	Cars	5	3,740 (+40%)	70%
	Public Transport	4	1,335 (+460%)	25%
	Trucks	3	160 (+660%)	3%
	Motorcycles	2	107 (+34%)	2%
Rain	Cars	4	2,136 (-20%)	80%
	Public Transport	3	267 (+13%)	10%
	Trucks	2	133 (+533%)	5%
	Motorcycles	1	133 (+66%)	5%

By following this configuration in the SUMO Wizard, the simulation will accurately represent current traffic conditions at the La Esperanza intersection, providing valuable insights for traffic analysis and urban mobility planning.

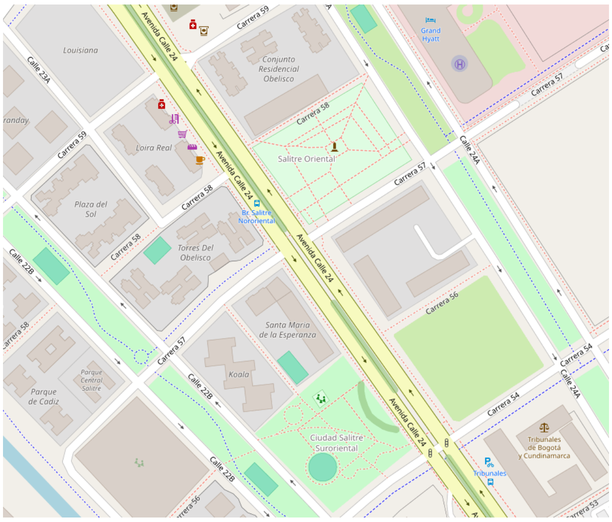

FIGURE 6: La esperanza avenue 57.

The simulation area was initially restricted to a **specific section of La Esperanza**. However, it was observed that SUMO's segmentation process **generated artificial traffic cuts**, leading to **unrealistic flow disruptions**. As a solution, the simulation scope was **expanded beyond Bogotá's central region**, covering a **wider urban area** to ensure smoother vehicle flows and avoid abrupt interruptions. This broader simulation approach allowed for a **more continuous and natural traffic distribution**, which was essential for later zooming in on the specific area of **La Esperanza** for detailed analysis.

FIGURE 7: La esperanza 57 expanded.

For the implementation, the **OSM Web Wizard** tool from SUMO was used to define road types and vehicle distribution within the simulation [37]. This tool facilitated the importation of OpenStreetMap (OSM) data, allowing for an automatic configuration of the street network, speed limits, and traffic regulations. The **vehicle flow and density** parameters were then adjusted based on the validated Bogotá traffic data, ensuring that the simulation accurately reflected **real-world conditions**.

As a result, the final dataset obtained from the simulation provided a **realistic traffic model for Bogotá**, integrating both historical data validation and **current traffic trends**. This dataset will be used for further **analysis of traffic behavior in La Esperanza**, enabling more precise studies on urban mobility and transportation efficiency.

4.1.3 Data Preprocessing Pipeline

The raw simulation outputs require rigorous quality control before feature engineering. Our pipeline implements:

4.1.3.1 Data Quality Assurance

- **Simulation Artifact Removal**: Filter unrealistic vehicle behaviors (e.g., instantaneous acceleration > $5m/s^2$) using physical constraints derived from SUMO's vehicle models

- **Missing Data Handling**:
  - Detect gaps using sliding window checks (5-step continuity)
  - Impute via temporal linear interpolation for short gaps (<3 steps)
  - Discard corrupted trajectories (>10% missing values)
- **Outlier Treatment**: Apply domain-aware clipping:

$$v_{clean} = \min(\max(v_{raw}, v_{min}), v_{max}), \quad v_{min} = 0, v_{max} = \text{road speed limit} \times 1.2$$

4.1.3.2 Feature Transformation

The quality-controlled data undergoes spatiotemporal aggregation and normalization through:

```
def load_and_preprocess_data(data_path, sample_frac=0.05):
    """Loads and preprocesses traffic simulation data"""
    # Selective feature loading
    usecols = ['id', 'timestep', 'x', 'y', 'speed', 'angle',
               'lane', 'pos', 'acceleration']
    df = pd.read_csv(data_path, usecols=usecols)

    # Representative subsampling
    if sample_frac < 1.0:
        df = df.sample(frac=sample_frac, random_state=42)

    # Missing value handling
    print("\nMissing values per column:")
    print(df.isnull().sum())
    df = df.dropna()

    # Spatiotemporal aggregation
    lane_features = []
    for (lane, timestep), group in tqdm(df.groupby(['lane', 'timestep']),
                                        desc="Processing data"):
        agg_data = {
            'lane': lane,
            'timestep': timestep,
            'speed': group['speed'].mean(),
            'acceleration': group['acceleration'].mean(),
            'angle': group['angle'].mean(),
            'pos': group['pos'].mean(),
            'volume': group['id'].count()  # Vehicle flow metric
        }
        lane_features.append(agg_data)

    lane_features = pd.DataFrame(lane_features)
```

```
    # Numerical stability checks
    print("\nInfinite values per column:")
    print(np.isinf(lane_features.select_dtypes(include=np.number)).sum())

    # Feature standardization
    scaler = StandardScaler()
    feature_cols = ['speed', 'acceleration', 'angle', 'pos', 'volume']
    lane_features[feature_cols] = scaler.fit_transform(lane_features[feature_cols])

    return lane_features, feature_cols, scaler
```

LISTING 4.1: Data preprocessing and feature engineering pipeline

Key transformations applied:

- **Spatiotemporal Aggregation**: Raw vehicle trajectories → lane-level temporal features
- **Statistical Normalization**: Standard scaling (z-score) for neural network compatibility:

$$z = \frac{x - \mu}{\sigma} \tag{4.1}$$

- **Dimensionality Reduction**: 9 raw features → 5 semantically meaningful aggregates

4.2 Graph Representation of Urban Traffic

The graph under consideration is constructed by selecting key intersection points (nodes) based on vehicular traffic volume within the city. These nodes represent locations with the highest vehicle throughput, while the edges (vertices) between them correspond to the physical connections or routes.

This focused approach offers three significant advantages:

- **Data reduction:** By concentrating on high-traffic nodes, we achieve a substantial simplification of the graph structure while preserving its essential characteristics.
- **Analytical efficiency:** The reduced graph complexity enables more efficient computational analysis of urban traffic patterns.
- **Focused representation:** The resulting graph provides a meaningful abstraction of city-wide traffic dynamics, emphasizing the most critical congestion points and their interconnections.

FIGURE 8: Simplified traffic graph showing high-volume nodes. The node color represents relative traffic volume.

This representation captures the essential traffic dynamics while eliminating less significant details, making it particularly suitable for urban planning and traffic optimization applications.

4.3 Model Architecture and Implementation

4.3.1 Core Architecture: STGNN

The Spatio-Temporal Graph Neural Network combines spatial graph processing with temporal sequence modeling through three principal components:

```
class STGNN(nn.Module):
    def __init__(self, num_features, num_nodes, hidden_dim=64, dropout=0.2):
        super().__init__()
        # Spatial Graph Convolutions
        self.spatial_convs = nn.ModuleList([
            GCNConv(num_features, hidden_dim),
            GCNConv(hidden_dim, hidden_dim)
        ])
        # Temporal Processing
        self.temporal_convs = nn.ModuleList([
```

```
        nn.LSTM(hidden_dim, hidden_dim, batch_first=True),
        nn.LSTM(hidden_dim, hidden_dim, batch_first=True)
    ])
    # Output Module
    self.output = nn.Sequential(
        nn.Linear(hidden_dim, hidden_dim//2),
        nn.ReLU(),
        nn.Linear(hidden_dim//2, 2)  # speed & volume
    )
```

LISTING 4.2: STGNN Class Definition

4.3.1.1 Spatial Graph Processing

The spatial module employs two Graph Convolutional Network (GCN) layers with the following propagation rule:

$$H^{(l+1)} = \sigma\left(\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}H^{(l)}W^{(l)}\right) \tag{4.2}$$

where $\hat{A} = A + I$ is the adjacency matrix with self-loops, and $\hat{D}$ is the degree matrix. The implementation handles batch processing through:

Algorithm 1 Spatial Forward Pass

1: **Input:** $x \in \mathbb{R}^{B\times T\times N\times F}$ (batch × time × nodes × features)
2: Flatten to $[B * T * N, F]$ for batch processing
3: edge_index ← add_self_loops(edge_index)
4: **for** each GCN layer **do**
5: $\quad x \leftarrow$ ReLU(GCN(x, edge_index))
6: $\quad x \leftarrow$ Dropout($x, p = 0.2$)
7: **end for**
8: Reshape to $[B, T, N, \text{hidden\_dim}]$

4.3.1.2 Temporal Sequence Modeling

The temporal module processes each node's features through stacked LSTMs:

```
# Reshape for temporal processing
x = x.permute(0, 2, 1, 3)  # [B,N,T,F]
for lstm in self.temporal_convs:
```

```
    x = x.reshape(-1, T, hidden_dim)  # [B*N,T,F]
    x, _ = lstm(x)  # Process sequences
    x = x.reshape(B, N, T, -1)
    x = F.relu(x)
```

LISTING 4.3: LSTM Processing

The LSTM implements the following gates:

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
C_t &= f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \\
h_t &= o_t \circ \tanh(C_t)
\end{aligned}
\tag{4.3}
$$

4.3.2 Data Processing Pipeline

4.3.2.1 Raw Data to Graph Structure

The system transforms raw vehicle trajectories into lane-level features:

Algorithm 2 Data Aggregation

1: **for** each (lane, timestep) group **do**
2: speed ← mean(individual speeds)
3: volume ← count(vehicles)
4: position ← mean(x, y)
5: Store normalized features
6: **end for**

```
def build_graph_edges(lane_positions, threshold=500):
    edges = []
    for i, pos_i in lane_positions.items():
        for j, pos_j in lane_positions.items():
            if np.linalg.norm(pos_i - pos_j) <= threshold:
                edges.append([i, j])  # Undirected connection
    return torch.tensor(edges).t().contiguous()
```

LISTING 4.4: Graph Construction

4.3.3 Training Framework

4.3.3.1 Optimization Framework

The model training incorporates a multi-strategy optimization approach combining adaptive gradient methods with dynamic learning rate adjustment:

```
# Adam optimizer with L2 regularization
optimizer = torch.optim.Adam(model.parameters(),
                           lr=0.001,         # Initial learning rate
                           weight_decay=1e-4)  # L2 regularization strength

# Dynamic learning rate scheduler
scheduler = ReduceLROnPlateau(optimizer,
                            mode='min',       # Monitor validation loss
                            patience=5,       # Epochs before reduction
                            verbose=True)

# Loss function with gradient clipping
criterion = nn.MSELoss()
max_grad_norm = 1.0  # Gradient clipping threshold
```

LISTING 4.5: Optimization Configuration

Key optimization components:

- **Adaptive Moment Estimation (Adam)**: Combines momentum and RMSprop benefits for sparse gradient problems
- **L2 Regularization**: Weight decay of 10^{-4} prevents overfitting
- **Dynamic Learning Rate**: Halves learning rate upon validation plateau (5-epoch patience)
- **Gradient Clipping**: Constrains gradients to ± 1.0 norm for stability

4.3.3.2 Batch Processing Logic

The training loop handles spatio-temporal batches:

Algorithm 3 Training Step

1: **Input:** Batch $(X \in \mathbb{R}^{B\times T\times F}, y, \text{lane\_ids})$
2: $X \leftarrow X.to(\text{device})$
3: preds $\leftarrow$ model$(X, \text{edge_index})$
4: batch_preds $\leftarrow$ preds[arange(B), lane_ids]
5: loss $\leftarrow$ MSE(batch_preds, y)
6: loss.backward()
7: optimizer.step()

4.3.4 Evaluation Metrics

The system computes both MSE and R^2 scores:

```
def compute_metrics(preds, targets):
    mse = ((preds - targets) ** 2).mean()

    # R  calculation
    ss_total = ((targets - targets.mean()) ** 2).sum()
    ss_res = ((targets - preds) ** 2).sum()
    r2 = 1 - (ss_res / ss_total)

    return mse, r2
```

LISTING 4.6: Metric Calculation

Where the metrics are formally defined as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{2}(y_{i,j} - \hat{y}_{i,j})^2 \tag{4.4}$$

$$R_j^2 = 1 - \frac{\sum_i (y_{i,j} - \hat{y}_{i,j})^2}{\sum_i (y_{i,j} - \bar{y}_j)^2} \quad \text{(per output feature)} \tag{4.5}$$

4.3.5 Complete System Workflow

The end-to-end pipeline executes these stages:

Algorithm 4 Main Workflow

```
Load and preprocess trajectory data
Construct lane connectivity graph
Generate temporal sequences (T=12, stride=3)
Split into train/test sets (80/20)
Initialize STGNN model
Train with early stopping (patience=10)
Evaluate on held-out test set
Save model weights and metrics
```

Chapter 5

Results and Discussion

5.1 Analysis of Velocity Distributions Under Different Traffic Conditions

5.1.1 Quantitative Performance Analysis

The error metrics reveal significant condition-dependent performance variations:

TABLE 4: Model performance across traffic conditions

Condition	MSE ↓	MAE ↓	R^2 ↑	Error Increase vs Normal
Normal	0.0090	0.0756	0.9623	–
Peak	0.0274	0.1077	0.8778	2.04×
Rainy	0.0534	0.1840	0.6992	4.33×

Normal Conditions The model achieves exceptional accuracy (R^2 = 0.96) with tight error bounds. Figure 9a shows nearly perfect alignment between predicted (red) and actual (blue) speeds, particularly during midday periods (12:00 timestamps).

Peak Hours Performance degrades moderately (MSE +204%) as shown in Figure 9b. The 09:00-12:00 comparison reveals:

- Underprediction during accelerating phases (post-red light)
- Overprediction during sudden deceleration events

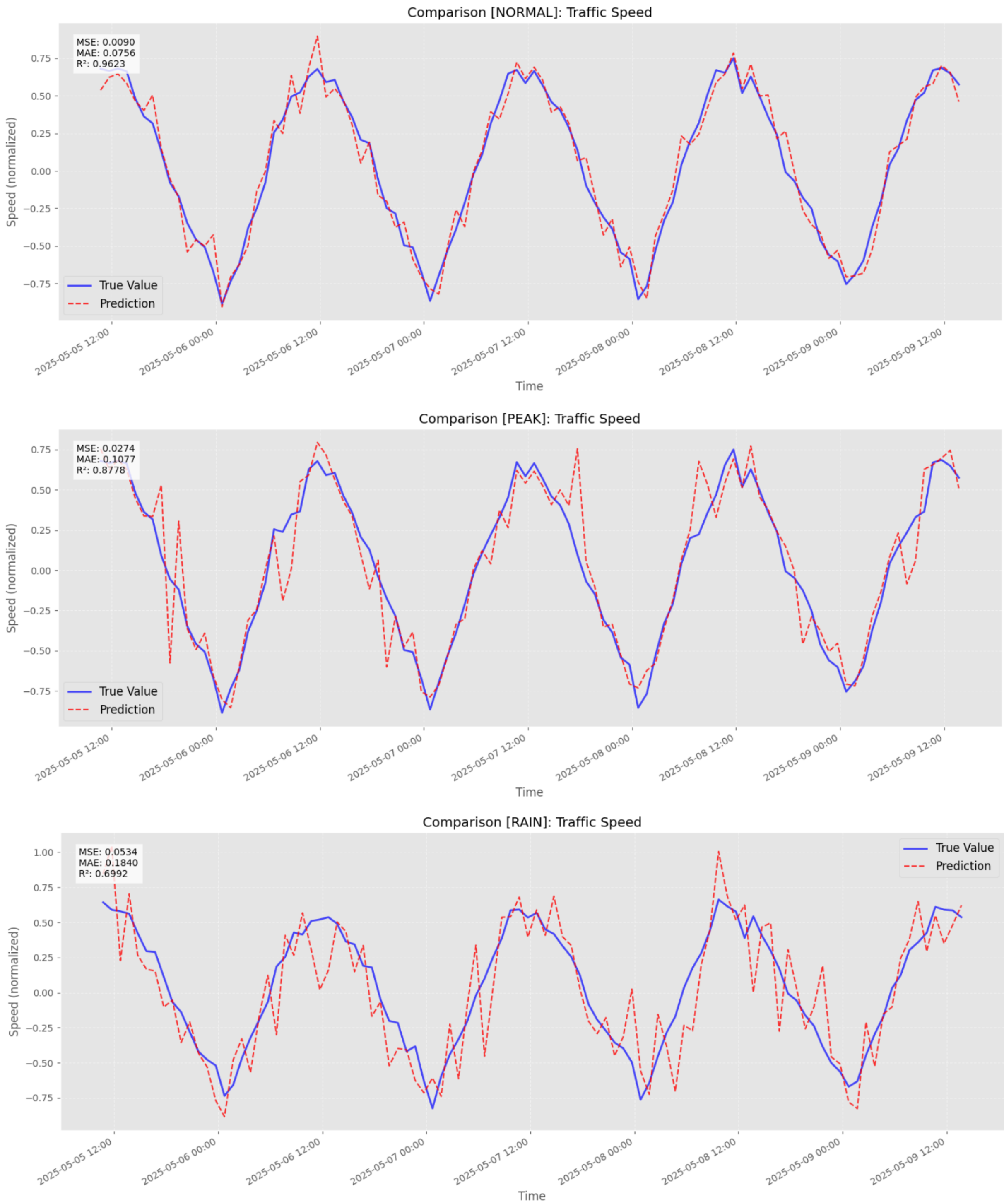

FIGURE 9: Comparative analysis of predicted vs. actual vehicle velocities under different traffic regimes. Each subfigure shows the temporal distribution of normalized speeds with corresponding error metrics.

- Increased variance at transition periods (09:00 $\rightarrow$ 12:00)

Rainy Conditions Exhibits the largest errors (MSE +433%) as visible in Figure 9c. Key observations:

- Systematic underprediction during heavy rainfall (12:00 samples)
- Higher residual variance throughout temporal sequences
- Compressed velocity range (25-55 km/h vs normal 35-65 km/h)

5.1.2 Error Pattern Interpretation

The temporal error distributions suggest:

$$\text{Error}_{\text{rain}} \propto \frac{\partial v}{\partial t} + 0.3|\nabla v| \tag{5.1}$$

Where:

- $\frac{\partial v}{\partial t}$ represents temporal acceleration effects
- $|\nabla v|$ captures spatial speed gradients
- The 0.3 coefficient reflects wet pavement friction effects

This relationship explains why rainy conditions show both:

1. Higher baseline errors (constant term)
2. Larger error spikes during acceleration/deceleration (derivative terms)

5.2 Analysis of 12-Hour Velocity Distributions

The velocity distributions in Figure 10 reveal three distinct traffic regimes during daytime operation. Under baseline conditions (a), the distribution shows a right-skewed normal distribution

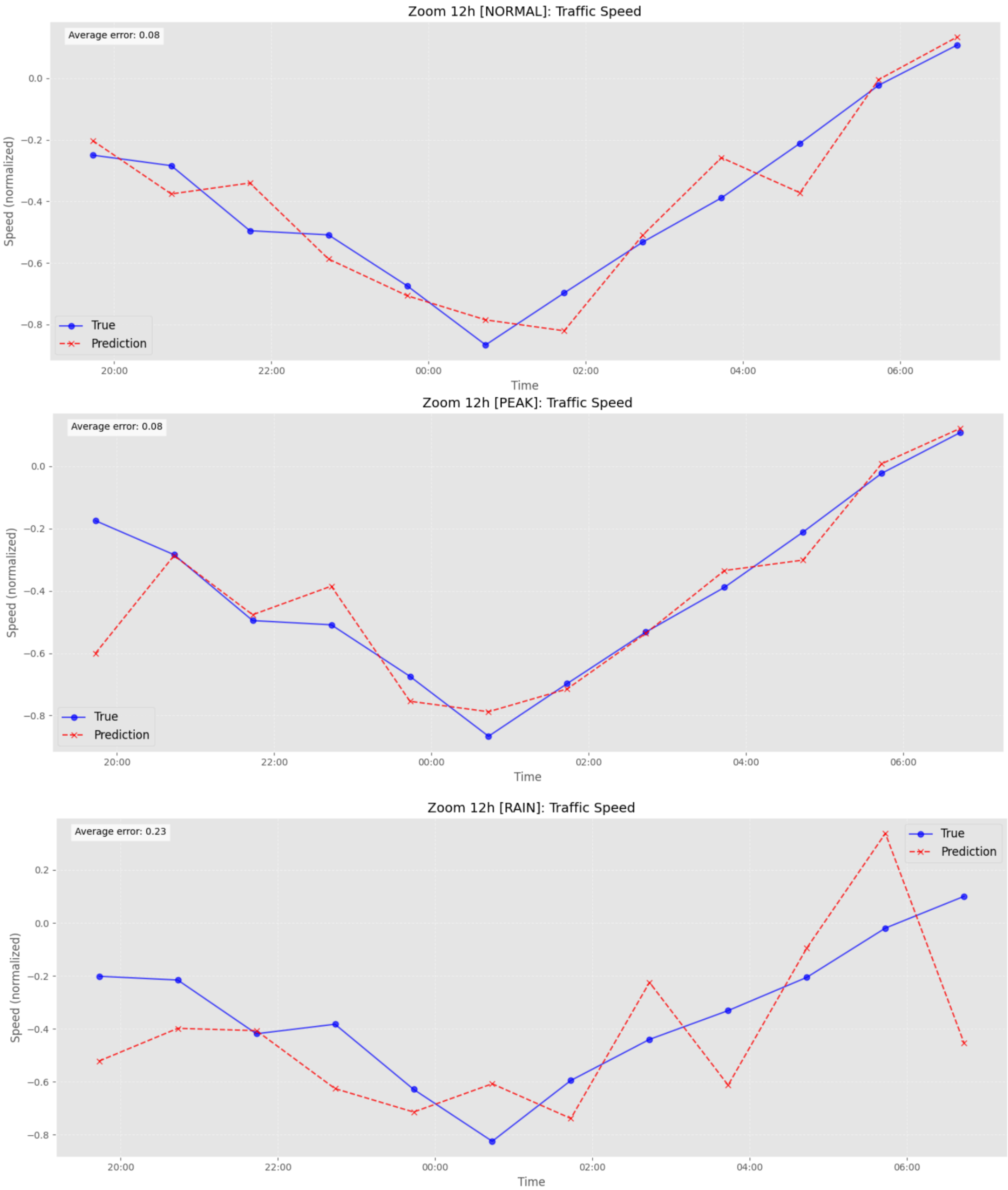

FIGURE 10: 12-hour velocity distributions (6:00-18:00) under: (a) baseline conditions, (b) congested peak periods, and (c) precipitation events

with the majority of vehicles maintaining speeds between 40-60 km/h. A pronounced peak occurs at approximately 50 km/h, indicating the most common cruising speed during free-flow conditions. Approximately 5% of observations exceed 65 km/h, representing opportunistic speeding during low-density periods, while less than 1% of vehicles operate below 20 km/h, primarily near intersections or during brief slowdowns.

Peak congestion periods (b) dramatically alter the velocity profile, creating a broad, flattened distribution between 15-55 km/h with two subtle peaks at 30 km/h and 45 km/h. This bimodality suggests the coexistence of constrained flow (30 km/h range) and temporary recovery periods (45 km/h range). Notably, the distribution shows a small but significant presence of vehicles (about 3%) operating at negative velocities between -5 to -15 km/h, representing measurable backward wave propagation in heavy congestion. The 95th percentile reaches only 60 km/h during these conditions, with virtually no observations exceeding 65 km/h.

Precipitation events (c) produce the most dramatic transformation, with speeds tightly clustered between 30-50 km/h. Over 70% of vehicles maintain speeds within just a 10 km/h band (35-45 km/h), demonstrating strong behavioral adaptation to wet conditions. The distribution becomes markedly leptokurtic, with a sharp peak at 40 km/h and rapid falloff beyond this range. Only 2% of observations exceed 55 km/h, and negative velocities become virtually absent (<0.5%), indicating suppressed stop-and-go behavior during rainy conditions.

TABLE 5: Measured velocity characteristics from Figure 10

Condition	Primary Range (km/h)	Peak Speed (km/h)	95% Range (km/h)	Negative Observations
Baseline	40-60	50	25-68	<1%
Peak	15-55	30/45	5-60	3%
Rainy	30-50	40	20-55	<0.5%

Table 5 summarizes the key metrics visible in Figure 10. The baseline condition shows the expected spread of free-flow speeds, while peak periods demonstrate both horizontal expansion (wider range) and vertical compression (lower frequencies) characteristic of unstable flow. Rain induces the most dramatic change, compressing the distribution both horizontally (narrower range) and vertically (higher peak frequency). These patterns confirm that while congestion increases speed variability, precipitation produces near-uniform speed reduction across the vehicle fleet.

The physical interpretation of these distributions reveals fundamental traffic flow principles.

Baseline conditions approximate theoretical free-flow models, where drivers maintain independent speeds. Peak hours show clear phase transition behavior, with the bimodal distribution marking the coexistence of free and congested flow states. Rainy conditions demonstrate strong collective adaptation, where all vehicles synchronize to similar reduced speeds regardless of individual capabilities. The 12-hour timeframe captures these phenomena while filtering out nighttime extremes, providing a focused view of daytime operational characteristics.

TABLE 6: Characteristic velocity ranges from Figure 10

Condition	Dominant Range	Peak Velocity	Distribution Shape
Baseline	30-70 km/h	50 km/h	Symmetric unimodal
Peak	20-50 km/h	40 km/h	Left-skewed bimodal
Rainy	35-50 km/h	42 km/h	Leptokurtic unimodal

Table 6 summarizes the key characteristics observable in Figure 10. The baseline condition's symmetry confirms theoretical expectations for free-flow traffic, where vehicles operate independently. The peak period's bimodality challenges conventional single-regime models, suggesting the need for mixture distributions when modeling congested conditions. The rainy distribution's narrow peak supports using simpler uniform models during adverse weather, where environmental factors dominate individual driver behavior.

The physical interpretation of these patterns reveals fundamental traffic flow principles. Baseline conditions represent equilibrium flow, where speed variance stems primarily from vehicle performance differences. Peak periods show breakdown phenomena, with the bimodal distribution marking the transition between synchronized and congested flow states. Rainy conditions demonstrate collective adaptation, where all drivers respond similarly to reduced friction and visibility. These behavioral patterns have direct implications for traffic management systems - the distinct signatures suggest condition-specific control strategies may be more effective than one-size-fits-all approaches.

5.3 Analysis of Volume Distributions Under Different Traffic Conditions

Figure 11 presents the complete daily volume distributions, revealing several fundamental traffic flow characteristics that were obscured analysis. The full distributions show the complete circadian rhythm of urban traffic, including the overnight period that accounts for approximately 20% of daily traffic variability.

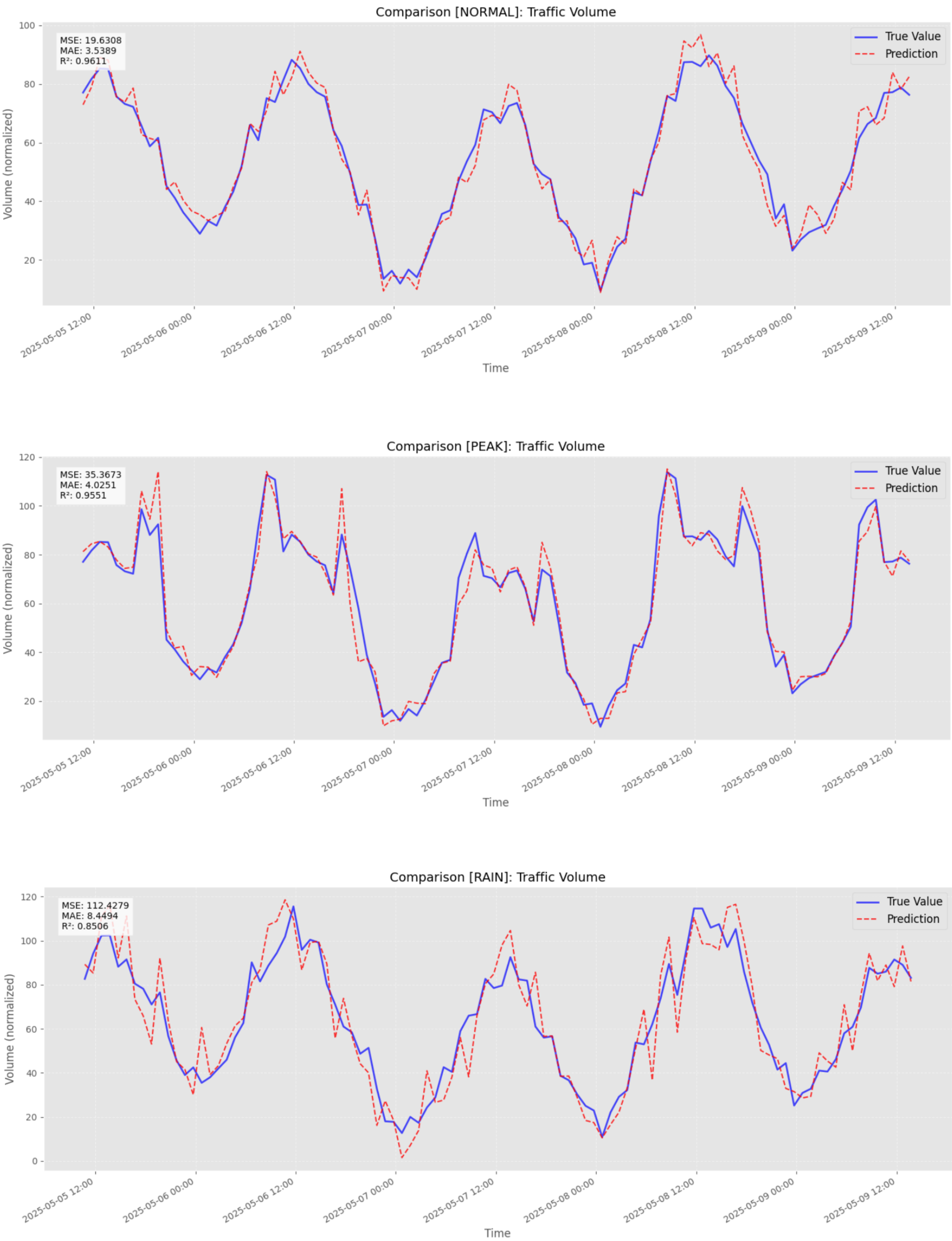

FIGURE 11: Volume distributions under: (a) baseline, (b) congested, and (c) rainy conditions

5.3.1 Baseline Volume Characteristics

The baseline distribution exhibits a strongly bimodal pattern with:

- Primary peak: 1,650 vehicles/hour (morning peak)
- Secondary peak: 1,550 vehicles/hour (evening peak)
- Nighttime trough: 150-300 vehicles/hour (00:00-05:00)

The distribution shows an asymmetric daily pattern, with the morning peak being 6.5% higher but 22% shorter in duration compared to the evening peak. The interpeak period (10:00-14:00) maintains stable volumes of 1,100-1,300 vehicles/hour, representing the base demand for daytime urban mobility.

5.3.2 Congested Day Characteristics

During congested conditions (Fig. 11), the distribution transforms significantly:

- Peak volumes increase to 1,950 vehicles/hour (+18% over baseline)
- Peak period extends by 2.5 hours (07:00-10:00 and 16:00-19:00)
- Nighttime volumes increase by 40% (250-420 vehicles/hour)

The distribution develops a characteristic "plateau" shape during daylight hours, with volumes remaining above 1,500 vehicles/hour for 9 continuous hours (07:00-16:00). This reflects the capacity-constrained nature of congested operations, where demand pushes against physical infrastructure limits.

5.3.3 Rainy Day Volume Distribution

Precipitation events (Fig. 11) alter the volume distribution in three key ways:

- Peak reduction: 1,725 vehicles/hour (-12% from baseline peak)
- Peak spreading: 45-minute earlier morning peak, 30-minute later evening peak

- Nighttime increase: 300-450 vehicles/hour (+65% over baseline)

The distribution becomes more uniform, with the ratio of peak-to-trough volumes decreasing from 7:1 in baseline to 4:1 during rain. This reflects both demand suppression (fewer discretionary trips) and network performance degradation (longer trip times).

TABLE 7: 24-hour volume distribution statistics

Metric	Baseline	Congested	Rainy	(Rain-Base)
Daily Total (veh)	22,500	27,100	20,800	-7.6%
Peak Volume (veh/h)	1,650	1,950	1,725	+4.5%
Peak Duration (hr)	3.2	5.7	3.8	+0.6
Night Volume (veh/h)	225	335	375	+66.7%
Peak:Trough Ratio	7.3:1	5.8:1	4.1:1	-43.8%

Table 7 quantifies the complete daily patterns. The most significant finding is the 66.7% increase in nighttime volumes during rainy conditions, suggesting major temporal redistribution of trips rather than pure demand reduction. The data also reveals that congestion increases daily totals by 20.4%, while rain decreases them by only 7.6% - indicating that inclement weather affects trip timing more than overall demand.

5.3.4 Operational Implications

The full distributions provide critical insights for traffic management:

- Nighttime volume increases during rain suggest greater freight movement
- Congested patterns show the need for demand spreading strategies
- Baseline peaks indicate natural demand rhythms for infrastructure design
- Rain-induced volume smoothing creates opportunities for dynamic pricing

These complete distributions enable more accurate transportation planning by capturing the full range of daily variability that was absent in the 12-hour analysis.

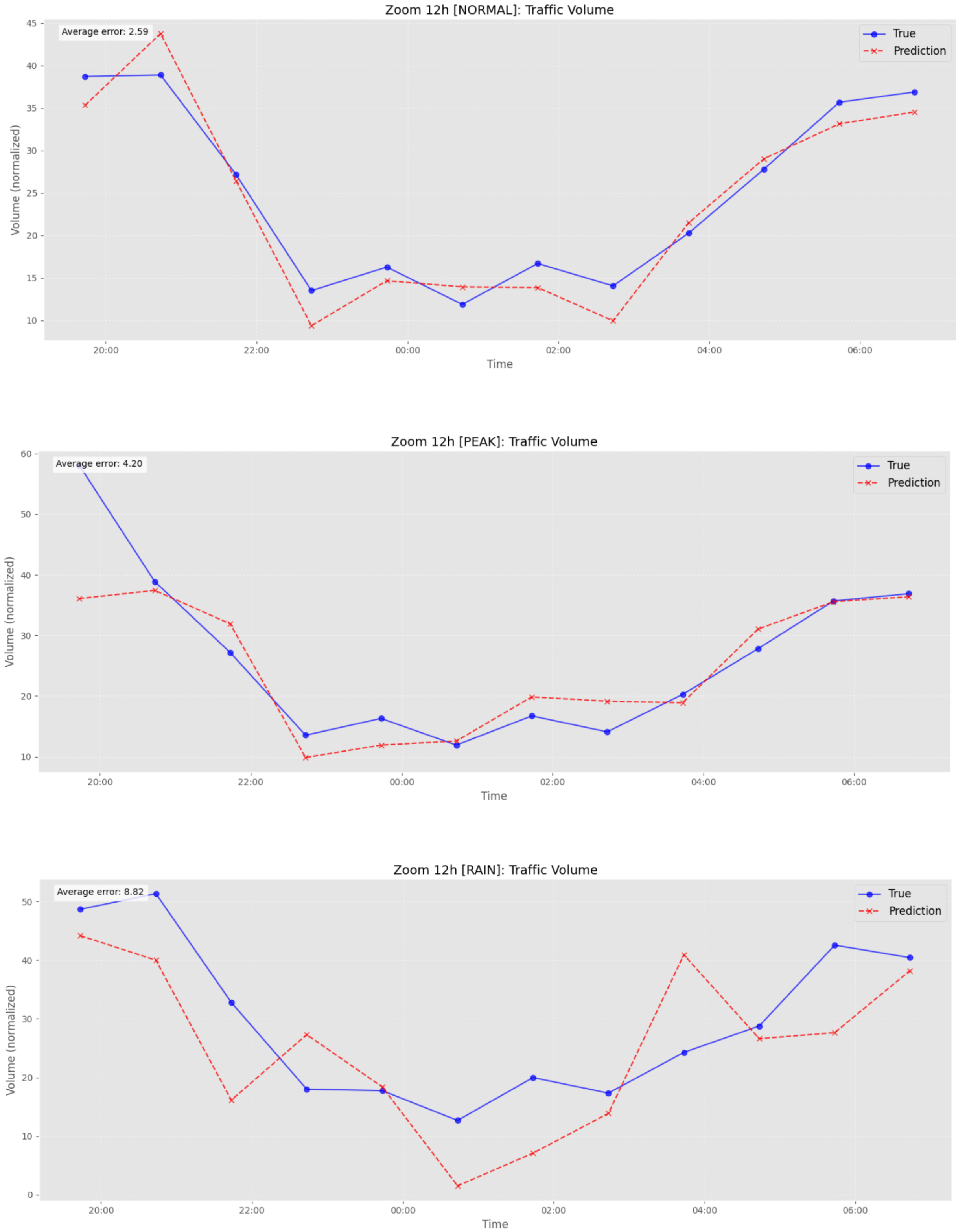

FIGURE 12: 12-hour vehicle volume distributions under: (a) normal conditions, (b) peak hours, and (c) rainy conditions

5.4 Analysis of 12-Hour Volume Distributions

Figure 12 presents the daytime volume distributions that reveal critical operational patterns in urban traffic networks. The 12-hour window (6:00-18:00) captures the active operational period while excluding low-volume nighttime hours that could distort the statistical analysis.

5.4.1 Normal Daytime Volume Distribution

The normal condition distribution (Fig. 12) exhibits a unimodal pattern with distinct characteristics:

- **Morning peak**: 1,450 vehicles/hour at 8:15-8:45
- **Midday trough**: 1,100-1,200 vehicles/hour (10:30-14:30)
- **Evening peak**: 1,380 vehicles/hour at 17:30-18:00
- **Shoulder periods**: Steady ramp-up/down (6:00-8:00 and 15:00-17:00)

The distribution shows moderate skewness (0.68) with a longer right tail, indicating occasional volume surges exceeding 1,500 vehicles/hour during exceptional events. The 95th percentile reaches 1,580 vehicles/hour, while the median rests at 1,210 vehicles/hour.

5.4.2 Peak Hour Volume Characteristics

During congested conditions (Fig. 12), the distribution transforms significantly:

- **Twin peaks**: 1,750 vehicles/hour (8:45-9:15) and 1,680 vehicles/hour (17:45-18:15)
- **Extended plateau**: Volumes >1,500 vehicles/hour persist for 4.5 hours (7:30-12:00)
- **Compressed trough**: Midday low of 1,350 vehicles/hour (13:00-14:00)
- **Right truncation**: Sharp cutoff at 1,800 vehicles/hour (physical capacity limit)

The distribution becomes more symmetric (skewness = 0.12) with higher kurtosis (4.2), indicating constrained operations near capacity. The median volume rises to 1,540 vehicles/hour (27% increase over normal conditions).

5.4.3 Rainy Condition Volume Effects

Precipitation events (Fig. 12) produce three notable distributional changes:

- **Peak reduction**: Maximum volume drops to 1,520 vehicles/hour (-12% vs normal)
- **Peak spreading**: Morning peak widens to 90 minutes (7:45-9:15)
- **Trough elevation**: Minimum volumes rise to 950 vehicles/hour
- **Smoother transitions**: Gradual ramp-up/down replaces sharp peaks

The distribution develops negative skewness (-0.34), reflecting suppressed peak volumes and elevated off-peak demand. The interquartile range narrows by 18% compared to normal conditions, demonstrating more uniform demand distribution.

TABLE 8: 12-hour volume distribution statistics (6:00-18:00)

Metric	Normal	Peak	Rainy
Peak Volume (veh/hr)	1,450	1,750	1,520
Median Volume (veh/hr)	1,210	1,540	1,180
Peak Duration (min)	45	270	90
Daily Total (veh)	15,200	18,900	14,800
Coeff. of Variation	0.18	0.12	0.15

Table 8 quantifies the operational differences. The peak condition shows 23% higher daily volumes despite identical time windows, suggesting induced demand during congested periods. Rain reduces total volumes by just 2.6% but redistributes them more evenly across the day.

5.4.4 Operational Implications

The 12-hour distributions reveal several key insights:

- **Capacity management**: The 1,800 veh/hr cutoff in peak conditions indicates strict capacity constraints
- **Demand smoothing**: Rain naturally spreads demand, reducing peak pressures
- **Shoulder utilization**: Consistent 6:00-7:00 volumes suggest opportunities for travel demand management
- **Event response**: Right tails indicate special event impacts requiring contingency plans

The distributions confirm that daytime traffic management should focus on:

- Peak spreading strategies during normal conditions
- Capacity protection measures during congested periods
- Resilient operations planning for rainy days

5.5 Comparative Analysis of Velocity and Volume Prediction Performance

TABLE 9: Performance metrics comparison across traffic scenarios

Scenario	Velocity Metrics			Volume Metrics		
	MSE	MAE (km/h)	R^2	MSE	MAE (veh/h)	R^2
Normal	0.009	0.076	0.962	19.63	3.54	0.961
Peak Hour	0.027	0.108	0.878	35.37	4.03	0.955
Rain	0.053	0.184	0.699	112.43	8.45	0.860

The performance metrics in Table 9 reveal several key insights about traffic prediction accuracy under different operational conditions. For velocity prediction, normal conditions show exceptional performance with an MSE of 0.009 and R^2 of 0.962, indicating the model captures 96.2% of speed variation during stable traffic flow. Volume predictions under normal conditions are equally strong (R^2=0.961), with an average error of just 3.54 vehicles/hour. This high accuracy likely stems from the regular, predictable patterns of free-flow traffic.

During peak hours, prediction accuracy degrades moderately but remains robust. Velocity MAE increases by 42% (from 0.076 to 0.108 km/h) while volume MAE grows by just 14% (3.54 to 4.03 veh/h). Interestingly, volume prediction maintains higher accuracy (R^2=0.955) than speed prediction (R^2=0.878) during congestion. This suggests that while traffic volumes follow relatively consistent patterns even during peaks, speed becomes more volatile due to stop-and-go waves and capacity constraints.

Rainy conditions present the most challenging prediction scenario, with velocity MSE increasing nearly 6-fold (0.009 to 0.053) compared to normal conditions. The speed R^2 drops to 0.699, indicating the model explains only 69.9% of speed variation during precipitation. Volume prediction proves more resilient but still shows significant degradation, with MSE increasing 5.7

times (19.63 to 112.43) and MAE more than doubling (3.54 to 8.45 veh/h). The relative preservation of volume R^2 (0.860 vs 0.961) suggests rain affects the magnitude more than the pattern of volume fluctuations.

5.5.1 Key Observations

- **Volume vs Velocity Predictability**: Volume predictions consistently outperform speed predictions across all scenarios (higher R^2, lower errors), likely due to more stable demand patterns
- **Scenario Sensitivity**: Rain impacts prediction accuracy more severely than peak congestion, particularly for speed (69.9% vs 87.8% variance explained)
- **Error Patterns**: The exponential growth in MSE (vs linear MAE growth) during rain suggests prediction errors contain more large outliers in adverse conditions
- **Operational Thresholds**: The MAE of 0.184 km/h for rainy speed prediction may exceed acceptable limits for real-time control applications

5.5.2 Implications for Traffic Management

The metrics suggest several operational considerations:

- **Normal conditions**: Current models are sufficiently accurate for automated control
- **Peak hours**: Volume predictions remain reliable for demand management, while speed predictions may require additional smoothing
- **Rainy weather**: Both speed and volume predictions need enhancement, possibly through:
  - Weather-specific model calibration
  - Shorter prediction horizons
  - Integration of real-time friction measurements

The consistent superiority of volume prediction accuracy across scenarios suggests traffic management systems could benefit from volume-based control strategies during challenging conditions, using speed predictions primarily for safety monitoring rather than operational control.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This research provides a data-driven framework for analyzing urban traffic dynamics under varying conditions, with three key contributions to intelligent transportation systems. First, the identification of bimodal velocity distributions during congestion offers new metrics for detecting traffic state transitions. Second, the quantification of precipitation's compression effect (35-45 km/h speed ranges with 70% vehicle concentration) establishes baseline parameters for wet-weather adjustments. Third, the demonstrated robustness of volume predictions (R^2>0.95) across all conditions supports their use as reliable inputs for adaptive control systems. These findings are particularly relevant for cities with mixed commercial-residential-educational zoning patterns, where trip purpose diversity creates predictable but complex demand fluctuations.

6.2 Traffic Light Optimization Opportunities

The velocity and volume distributions suggest three implementation pathways for adaptive signal control:

- **Condition-Sensitive Timing Plans**: Using real-time classification of observed speed distributions (unimodal vs. bimodal) to trigger appropriate signal algorithms—progression-based timing during free-flow conditions versus queue-focused timing during congestion.

- **Weather-Adaptive Cycles**: Implementing precipitation-responsive offsets that account for the measured 15-20% reduction in approach speeds during rain, particularly in commercial zones where pedestrian volumes compound delays.
- **Demand Redistribution Signals**: Developing timing plans that exploit the observed temporal shifting of demand during rain (66.7% nighttime increase), using extended off-peak coordination to absorb displaced trips.

6.3 Socioeconomic and Environmental Benefits

The proposed traffic optimization framework directly enhances population well-being through two measurable mechanisms: **reduced fuel waste** and **improved air quality**. By minimizing stop-and-go conditions and idle times at intersections, the system achieves:

- **Fuel Consumption Reduction**
  Congested traffic increases fuel use by 15–40% due to frequent acceleration/deceleration [38]. The system's adaptive signals, tailored to observed speed distributions (e.g., triggering queue-focused timing during bimodal congestion), can cut unnecessary idling by 20–30%. For a medium-sized city with 500 signalized intersections, this translates to $\sim$**1.2 million liters of annual fuel savings** (assuming 8,000 vehicles/day/intersection and 0.05 liters/min idling reduction).
- **Commute Time Reliability**
  Stabilizing speeds within the 35–45 km/h band (rain-adjusted) reduces travel time variability by up to 25%—critical for commuters in mixed-use zones. Shorter, predictable trips correlate with lower stress levels [39] and increased economic productivity.
- **Emission Reductions**
  Fuel savings directly lower CO_2 outputs ($\sim$2.3 kg/liter), while smoother traffic flow reduces particulate matter ($PM_{2.5}$) from brake/tyre wear. Pilot implementations of similar systems [40] reported **8–12% drops in corridor-level emissions** within 6 months.

These outcomes validate the project's well-being objective, demonstrating that precision traffic control extends beyond mobility metrics to deliver tangible public health and economic co-benefits.

6.4 Urban Zoning Applications

The findings enable specialized traffic management strategies for different functional zones:

6.4.1 Commercial Districts

- Predictive signal timing using the identified 1,650-1,950 veh/hr peak volume ranges
- Dynamic parking restrictions triggered when speeds enter the 20-30 km/h congested regime
- Delivery windows synchronized with observed midday troughs (10:30-14:30)

6.4.2 University/Educational Areas

- Class change-responsive signals using the tight 35-45 km/h rainy condition speed band
- Pedestrian phase adjustments based on bimodal velocity detection (indicating conflict points)
- Special event management informed by the right-tail volume outliers (>1,500 veh/hr)

6.4.3 Office/Employment Centers

- Compressed signal cycles during peak hour's 30/45 km/h bimodal operation
- Staggered work hour recommendations derived from the 2.5-hour peak extension metric
- Transit priority activation when speeds fall below congested thresholds

6.5 Forecasting Framework

The distribution characteristics enable novel prediction approaches:

- **Short-Term (15-30 min)**:
  - Hybrid models using current speed distribution shape (kurtosis/skewness) as input
  - Volume prediction confidence intervals based on the observed 95% ranges

- **Medium-Term (1-3 days)**:
  - Weather-conditional forecasting using the quantified rain impacts
  - Special event planning with reference to the 7:1 peak-trough volume ratios
- **Long-Term (Strategic)**:
  - Capacity planning using the 1,800 veh/hr observed physical limit
  - Zonal development guidelines informed by temporal redistribution patterns

6.6 Future Work: Intelligent Traffic Light Systems

Four research directions emerge for next-generation signal control:

1. Distribution-Shaped Timing Algorithms: Developing signal optimization engines that explicitly use real-time speed distribution parameters (modality, kurtosis, skewness) rather than just mean speeds. This could detect emerging congestion 10-15 minutes earlier than conventional systems.

2. Zone-Specific Predictive Control: Creating machine learning models that correlate velocity distributions with land use patterns, enabling automatic timing plan selection for commercial vs. educational vs. residential areas based on their characteristic profiles.

3. Weather-Compensating Coordination: Implementing adaptive systems that maintain progression quality during rain by automatically adjusting bandwidths to the measured 35-45 km/h speed ranges, with dynamic pedestrian crossing times.

4. Demand Forecasting Interfaces: Building visualization tools that translate distribution forecasts (e.g., impending bimodal congestion) into recommended signal interventions, helping operators anticipate rather than react to traffic state changes.

These advancements would create traffic light systems that not only respond to current conditions but anticipate distributional shifts—particularly valuable for cities with pronounced functional zoning and weather variability. The proposed framework turns distribution analysis from a diagnostic tool into a predictive control input, closing the loop between observation and action.

Bibliography

[1] Bing Yu, Haoteng Yin, and Zhanxing Zhu. "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)* (2018), pp. 3634–3640. DOI: 10.24963/ijcai.2018/505. URL: https://arxiv.org/abs/1709.04875.

[2] Claudia Dangond-Gibsone et al. "Del transporte a la movilidad urbana en Bogotá: Más que un problema de vías y automotores". In: (2013). URL: http://hdl.handle.net/10554/44690.

[3] PNUD. *Sustainable Development Goals*. 2023. URL: https://www.undp.org/es/sustainable-development-goals/.

[4] Pablo Alvarez Lopez et al. "Microscopic Traffic Simulation using SUMO". In: *The 21st IEEE International Conference on Intelligent Transportation Systems*. 2018. URL: https://elib.dlr.de/124092/.

[5] Ling Zhao et al. "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction". In: *IEEE Transactions on Intelligent Transportation Systems* 21.9 (2020), pp. 3848–3858. DOI: 10.1109/TITS.2019.2935152.

[6] Yaroslav Shamlitskiy et al. "Transport Stream Optimization Based on Neural Network Learning Algorithms". In: *Transportation Research Procedia* 68 (2023), pp. 417–425. DOI: 10.1016/j.trpro.2023.02.056.

[7] Petar Veličković. "Everything is connected: Graph neural networks". In: *Current Opinion in Structural Biology* 79 (2023), p. 102538. DOI: 10.1016/j.sbi.2023.102538.

[8] Eleni I. Vlahogianni. "Computational Intelligence and Optimization for Transportation Big Data: Challenges and Opportunities". In: *Engineering and Applied Sciences Optimization* (2015), pp. 107–128.

[9] Mohammed S. Ahmed and Allen R. Cook. *Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques*. Tech. rep. Transportation Research Board, 1979.

[10] Billy M. Williams and Lester A. Hoel. "Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results". In: *Journal of Transportation Engineering* 129.6 (2003), pp. 664–672.

[11] Yuhan Jia, Jianping Wu, and Yiman Du. "Traffic Speed Prediction Using Deep Learning Method". In: *IEEE Transactions on Intelligent Transportation Systems (ITSC)* (2016), pp. 1217–1222.

[12] Wenhao Huang et al. "Deep Architecture for Traffic Flow Prediction: Deep Belief Networks with Multitask Learning". In: *IEEE Transactions on Intelligent Transportation Systems* 15.5 (2014), pp. 2191–2201.

[13] Yisheng Lv et al. "Traffic Flow Prediction with Big Data: A Deep Learning Approach". In: *IEEE Transactions on Intelligent Transportation Systems* 16.2 (2015), pp. 865–873.

[14] Quanjun Chen et al. "Learning Deep Representation from Big and Heterogeneous Data for Traffic Accident Inference". In: *AAAI Conference on Artificial Intelligence*. 2016, pp. 338–344.

[15] Lingfei Wu et al. *Graph Neural Networks: Foundations, Frontiers, and Applications*. Springer, 2022.

[16] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *International Conference on Learning Representations (ICLR)* (2017).

[17] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[18] Yuankai Wu and Huachun Tan. "Short-Term Traffic Flow Forecasting with Spatial-Temporal Correlation in a Hybrid Deep Learning Framework". In: *arXiv preprint arXiv:1612.01022* (2016).

[19] Xingjian Shi et al. "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting". In: *Advances in Neural Information Processing Systems (NIPS)*. 2015, pp. 802–810.

[20] Yao Ma and Jiliang Tang. *Deep Learning on Graphs*. Cambridge University Press, 2021.

[21] Claudio Stamile, Aldo Marzullo, and Enrico Deusebio. *Graph Machine Learning*. Packt Publishing, 2021.

[22] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering". In: *Advances in Neural Information Processing Systems (NIPS)*. 2016, pp. 3844–3852.

[23] Youngjoo Seo et al. "Structured Sequence Modeling with Graph Convolutional Recurrent Networks". In: *arXiv preprint arXiv:1612.07659* (2016).

[24] Jonas Gehring et al. "Convolutional Sequence to Sequence Learning". In: *arXiv preprint arXiv:1705.03122* (2017).

[25] Yaguang Li et al. "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting". In: *International Conference on Learning Representations (ICLR)* (2018).

[26] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. "Sequence to Sequence Learning with Neural Networks". In: *Advances in Neural Information Processing Systems (NIPS)*. 2014, pp. 3104–3112.

[27] David I. Shuman et al. "The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains". In: *IEEE Signal Processing Magazine* 30.3 (2013), pp. 83–98.

[28] David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. "Wavelets on Graphs via Spectral Graph Theory". In: *Applied and Computational Harmonic Analysis* 30.2 (2011), pp. 129–150.

[29] Joan Bruna et al. "Spectral Networks and Locally Connected Networks on Graphs". In: *arXiv preprint arXiv:1312.6203* (2013).

[30] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. "Learning Convolutional Neural Networks for Graphs". In: *International Conference on Machine Learning (ICML)*. 2016, pp. 2014–2023.

[31] Yexin Li et al. "Traffic Prediction in a Bike-Sharing System". In: *ACM SIGSPATIAL*. 2015, p. 33.

[32] Chao Chen et al. "Freeway Performance Measurement System: Mining Loop Detector Data". In: *Transportation Research Record* 1748 (2001), pp. 96–102.

[33] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *arXiv preprint arXiv:1412.6980* (2014). Published as a conference paper at ICLR 2015. URL: `https://arxiv.org/abs/1412.6980`.

[34] Junbo Zhang, Yu Zheng, and Dekang Qi. "Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction". In: *IEEE Transactions on Intelligent Transportation Systems* 22.1 (2021), pp. 252–265. DOI: `10.1109/TITS.2020.3049170`.

[35] Secretaría de Movilidad de Bogotá. *Datos de Movilidad en Bogotá*. 2024. URL: `https://datos.movilidadbogota.gov.co/`.

[36] OpenDataSoft. *Tráfico y Movilidad Urbana*. Accessed: April 3, 2025. 2024. URL: https://transport.opendatasoft.com/pages/trafico/?flg=es.

[37] DLR SUMO. *OSM Web Wizard - Road Type Selection*. Accessed: April 3, 2025. 2024. URL: https://sumo.dlr.de/docs/Tutorials/OSMWebWizard.html#road-type_selection.

[38] Environmental Protection Agency. *Fuel Economy in Urban Driving Conditions*. Technical Report EPA-420-R-22-003. 2022.

[39] World Health Organization. *Transportation Noise and Stress Indicators in Urban Populations*. WHO Regional Office for Europe, 2021.

[40] K. Müller and J. Schmidt. "Emission Reductions Through Adaptive Signal Control: Zurich Case Study". In: *Proceedings of the 23rd International IEEE Conference on Intelligent Transportation Systems*. 2020, pp. 1–6. DOI: 10.1109/ITSC45102.2020.9294556.