Leveraging Feature Engineering and Model Enhancements to Improve Bicycle Travel Time Estimation

Veena Madhu[1]

Supervisor(s): Elvin Isufi[1]**, Ting Gao**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfillment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Veena Madhu
Final project course: CSE3000 Research Project
Thesis committee: Elvin Isufi, Ting Gao, Jing Sun

Abstract

Accurate prediction of bicycle travel time is critical for efficient urban mobility and sustainable transport planning. However, real-world datasets are noisy, imbalanced and lack rich contextual features. This limits the effectiveness of current graph-based neural network models. This research aims to explore how feature engineering and model enhancements can improve the performance of a Graph Convolutional Neural Network (GCNN) in the context of travel time prediction. Building on a currently existing DG4b architecture, the input data is enriched with temporal, spatial and traffic-related features. Architectural enhancements are integrated by employing techniques such as graph data augmentation, and multi-scale graph learning. Using a dataset from Berlin, the improvements are evaluated primarily in terms of prediction accuracy across varying trip lengths, which implicitly reflect speed variability and route diversity. The goal is to explore how targeted feature engineering and graph-based modeling techniques influence the accuracy of bicycle travel time estimation, especially across different trip durations that reflect real-world cycling variability. The results show that optimal feature engineering improved the model up to 6% and a combination of the model enhancement techniques improved the model up to 2%.

1 Introduction

Urban cycling has gained a lot of popularity over the years as cities seek to reduce congestion and promote sustainable transportation. As more people make the switch to cycling as the preferred mode of transportation, especially for short-distance trips, accurate bicycle travel time estimation (TTE) is becoming increasingly critical for route planning, infrastructure planning, and traffic management. However, unlike motorized transport, bicycle traffic presents a different set of challenges. Cyclists travel at more variable speeds, often deviate from prescribed routes, exhibit far more diverse behaviors than their motorized counterparts. This complicates predictive modeling [3]. These factors make traditional travel-time prediction techniques - many of which were designed for car traffic - less effective when directly applied to cycling.

Gao et al. [3] proposed DG4b, a dual-graph convolutional neural network (GCNN) that models both static road networks and trip-specific structures to estimate bicycle travel times. However, this approach still faces limitations as it relies on limited features (e.g., segment length, peak/off-peak flag) and often struggles to generalize across trip lengths and temporal patterns [2].

Moreover, the previous work [3] does not systematically explore the role of feature engineering or model architecture tuning in improving the GCNN-based bicycle TTE. Given the known challenges of cyclist behavior variability, data sparsity, and road network complexity, there is a need to examine whether rich spatial and temporal features and architectural additions can help GCNNs capture speed variability and trip diversity more effectively.

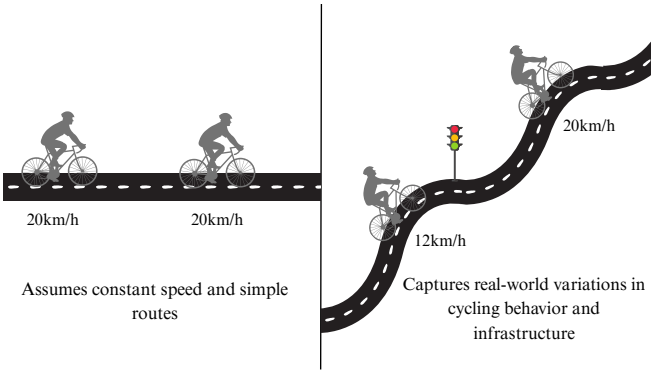

Figure 1: Traditional Travel Time Estimation Models vs DG4b Travel Time Estimation Model

To address these challenges, the following overarching research question is investigated: **How do feature engineering and model enhancements help models learn patterns that reflect speed variability and trip length differences in bicycle travel time estimation?**. This is explored through four sub-questions:

1. Which trip-specific and spatial features best capture speed variability and trip length diversity?
2. How do different feature sets impact GCNN model performance across trip length categories?
3. How do architectural enhancements affect prediction accuracy across trips of varying durations, which may reflect different patterns of speed variability?
4. How does combining different architectural techniques influence the model's performance compared to individual methods?

Our feature engineering experiments reveal that careful selection and combination of features significantly impact the accuracy of bicycle travel time estimation. The results highlight that augmenting baseline features (such as log-transformed distance, speed coefficient of variation, and detour ratio) with detailed network-based and temporal features consistently improves model performance across short, medium, and long trips.

The model was enhanced by employing two different methods: graph data augmentation and multi-scale graph learning. The experiments show that on its own, both methods minimally improve the prediction errors. However, when both methods are combined, the performance of the model shows significant improvement across all trip-length categories.

The paper is organized as follows. Section 2 reviews the relevant literature in graph-based mobility modeling and travel time estimation and then explores the different methods we can employ to improve the data as well as the model. Section 3 describes the methodology, including feature extraction and model enhancements. Section 4 outlines the experimental setup and datasets. Section 5 presents results and analysis based on trip length categories and feature ablation. Section

6 discusses implications and limitation. Section 7 reflects on the ethical aspects of the research and the reproducibility of the methods. Section 8 concludes the paper and proposes directions for future research.

2 Literature Review: Exploring current solutions in feature engineering and model enhancements

This section discusses the recent solutions we can use to implement feature engineering and model enhancements. We first talk about the current solutions in the realm of feature engineering and how spatial and temporal features play a role in machine learning performance. We then shift our focus on two methods that we can use to enhance our model - as per the scope of this research - for better generalization and representation: graph data augmentation and multi-scale graph learning.

While car travel time estimation has matured significantly over decades [8] [9], modeling for bicycles is comparatively nascent. Bicycles introduce unique challenges such as personalized pacing, route flexibility, and frequent deviations from traffic norms [12] [15]. These factors contribute to more stochastic and less structured movement patterns, complicating the modeling landscape.

Feature Engineering. Feature engineering plays a pivotal role in enhancing machine learning model performance, especially in domains like travel time estimation where heterogeneous data and dynamic conditions exist. Traditional feature engineering relies heavily on domain expertise to craft meaningful transformations [7]. However, with increasingly complex data types (e.g., spatiotemporal graphs), automated methods like SAFE [16] and deep feature synthesis [10] have emerged, allowing scalable construction of feature sets.

In transport research, temporal encodings (e.g., time-of-day, weekday, seasonal cycles) have been widely used to capture periodic patterns [19], while network-specific features, such as edge density or segment connectivity, have been shown to enhance graph-based models' sensitivity to local topology. Beyond these methods, studies such as Ye et al. [19] have explored categorical approximations of travel time distributions through adaptive temporal encoding, which can help address long-tailed distributions common in cycling data. Similarly, Han et al. [6] demonstrated that embedding both semantic and physical path characteristics significantly improves estimation accuracy.

Recent works emphasize learning adaptive feature interactions rather than manual crafting. Cheng et al. [1] introduced adaptive factorization networks that dynamically learn feature interaction orders, while Jin et al. [9] applied spatiotemporal dual graph networks to jointly learn temporal signals and graph connectivity for travel time estimation.

Model Enhancements. To improve model generalization and robustness, graph data augmentation [12] has emerged as a powerful tool. This includes techniques like edge perturbation, node feature masking, and subgraph sampling, which expand training distributions and improve resilience against overfitting [20]. Such methods have been applied successfully in traffic estimation [2], where graph augmentation improves the model's ability to generalize to rare or noisy travel patterns. Furthermore, graph contrastive learning has emerged as a promising self-supervised approach to improve representation robustness under augmentation. Methods like graph contrastive learning [20] and implicit augmentation schemes [12] have been successfully applied to traffic graphs, suggesting untapped potential in bicycle-specific GCNN training.

However, most augmentation and contrastive learning methods stem from homophily-assuming traffic graph contexts. They typically don't account for cyclist-specific complexities—like variability in speed due to terrain or individualized pacing. Standard augmentations (e.g., edge-drop, node masking) can inadvertently obscure these cues, leading to loss of critical physical signals. Because of this, instead of adopting them wholesale, we design bordered augmentation schemes that preserve speed-length integrity.

Despite the rise of static and dynamic GNN (DGNN) methods, they fall short in cycling contexts. GNNs often suffer from over-smoothing — where deep layers cause node representations to blur — a critical issue for preserving local speed variance (e.g., uphill vs flat segments) [4]. Moreover, cycling graphs are inherently heterophilous: adjacent segments may have radically different physical attributes. This breaks the homophily assumption many GNNs rely on [21]. Moreover, dynamic GNNs are often tuned to snapshot tasks (e.g., traffic at 5-minute intervals) and bench-marked on standard datasets, not route length variability seen in cycling. They also struggle with scaling to long graph sequences reflective of multi-kilometer trips [5]. Because of these physical and methodological mismatches, static or off-the-shelf DGNNs are not directly applied.

Another promising direction is multi-scale graph learning, which incorporates hierarchical representations across different spatial resolutions [13]. By learning both local (segment-level) and global (network-level) structures, these methods better capture long-range dependencies and aggregate information effectively. For example, Yan et al. [18] proposed multi-task dual graph neural networks that jointly model intersections and segments for improved travel time predictions, while Jin et al. [9]) extended this to spatiotemporal settings, demonstrating superior performance over single-scale models.

Despite recent advances, no prior work has systematically evaluated the combined impact of engineered spatiotemporal features and architectural augmentations on GCNN performance for bicycle travel time estimation. In particular, the dual reliance on a static road graph and a trip-specific subgraph [3] has not been leveraged in conjunction with feature ablation strategies or augmentation techniques to address noise, bias, and overfitting in cycling data. In summary, current graph approaches for travel estimation—whether feature-engineering, augmentation, or architectural—all struggle to preserve physical interpretability, local speed distinctions, and long-route dynamics appropriate for cycling.

3 Methodology

This study proposes an integrated framework for bicycle travel time estimation that combines robust feature engineering with advanced graph-based model enhancements. Our methodology consists of two primary components:

1. feature engineering,
2. model enhancements through graph data augmentation and multi-scale graph learning

Background

The GCNN used in Gao et al. [3] will be our baseline. The engineered features will be added to the model and architectural enhancements will be used to modify it. It is a dual-graph network architecture, where two graph streams are processed in parallel:

- Static road graph: captures invariant structural properties of the urban cycling network, including segment connectivity, availability for bicycles, and traffic light presence.
- Trip-specific graph: models the directional, dynamic properties of each individual trip, such as route sequence, temporal context (peak/off-peak), and remaining distance.

A static graph is constructed once for the entire road network. The nodes represent road segments and edges represent adjacency or connectivity. A graph neural network (GNN) propagates information across multiple hops, learning embeddings that capture network-wide spatial dependencies, typical traffic patterns and structural bottlenecks. These embeddings serve as a stable contextual prior for any trip.

For each individual trip, a local subgraph is extracted. This comprises only the sequence of segments traversed during that specific trip. The node features incorporate relative position, remaining distance, and temporal flags. A parallel GNN processes this subgraph to produce embeddings reflecting real-time, sequential, and ephemeral factors (e.g. sudden slowdowns).

Both graphs are encoded using graph convolutional layers with an encode-process-decode pipeline. The outputs are combined to estimate segment-wise speed likelihoods, which are then aggregated to compute the total travel time of the trip.

Feature Engineering

Based on prior findings [7] [16] [1], feature sets combining statistical, spatial, and temporal elements will be designed. The engineered features include:

- Distance-based features: log-transformed travel distance, detour ratio.
- Speed-based features: coefficient of variation of speed, standard deviation of speed.
- Network-based features: edges per kilometer, representing route density and complexity.
- Temporal features: time-of-day and month encodings using sinusoidal and cosine transformations to capture periodic traffic patterns [12] [18].

Figure 2: Feature Extraction Pipeline

For every route r let:

Symbol	Description
d_r	Travelled distance of the first segment of route r (in meters)
σ_{s_r}	Population standard deviation of all speeds on route r
$\bar{s}_r$	Mean of all speeds on route r
D_r	Total travelled distance of all segments on route r
E_r	Total edge count of all segments on route r
t_r	Start time of route r, in seconds since midnight
m_r	Month of start of route r, as an integer from 1 to 12
p_r	Peak hour flag of the first segment of route r, either 0 or 1

Table 1: Basic Route Features

and let every engineered feature be defined as follows:

Symbol	Description
ld_r	Log-distance of the first segment
cs_r	Coefficient of variation of speed
dt_r	Detour ratio
st_r	Standard deviation of speeds
ekm_r	Number of edges per kilometer
$tods_r$	Sine of time-of-day
$todc_r$	Cosine of time-of-day
mc_r	Sine of month
ms_r	Cosine of month
ph_r	Peak hour indicator

Table 2: Engineered Route Features

Then every engineered feature is calculated using the following formulae.

$$\text{ld}_r = \log(d_r + 1),$$

$$\text{cs}_r = \frac{\sigma_{s_r}}{\bar{s}_r},$$

$$\text{dt}_r = \frac{D_r}{E_r},$$

$$\text{st}_r = \sigma_{s_r},$$

$$\text{ekm}_r = \frac{E_r}{D_r/1000}$$

$$\text{tods}_r = \sin\left(2\pi \cdot \frac{t_r \bmod 86400}{86400}\right),$$

$$\text{todc}_r = \cos\left(2\pi \cdot \frac{t_r \bmod 86400}{86400}\right),$$

$$\text{ms}_r = \sin\left(2\pi \cdot \frac{m_r}{12}\right),$$

$$\text{mc}_r = \cos\left(2\pi \cdot \frac{m_r}{12}\right),$$

$$\text{ph}_r = p_r.$$

The value 86400 is used in the formulae to normalize the time to a full day. It represents the number of seconds in a day. The value 1000 is used to convert the travelled distance from meters to kilometers.

Through systematic feature ablation experiments, we will evaluate the performance contributions of these combinations, using root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) across short, medium, and long trip categories as done in Gao et al. [3].

Model Enhancements

To improve generalization, we incorporate graph data augmentation techniques [20] [12], including edge perturbation and node masking which expand the training set and mitigate overfitting.

- **Edge drops**: Edges are randomly dropped with a probability of 10% to mitigate over-smoothing and over-fitting [15].
- **Edge additions**: Edges may also be randomly added with a probability of 1% to diversify the connectivity of the graph.
- **Node masking**: This zeros out node feature vectors with a probability of 10% to improve robustness.

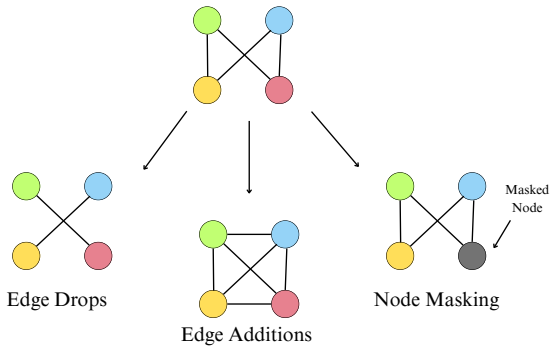

Figure 3: Visualization of edge drops, edge additions and node masking from left to right

Additionally, we apply multi-scale graph learning [13] [18] to capture both fine-grained local relationships (e.g., within neighborhoods or intersections) and global network patterns (e.g., across the entire city network). This hierarchical approach allows the model to learn robust representations across spatial scales, improving resilience to data sparsity and variability.

For each road segment in a trip, its static (global) embedding is concatenated with its local (trip-specific) embedding. This fusion yields a multi-scale feature vector that jointly encodes infrastructure context and trip dynamics. The fused vector is mapped via a multilayer perceptron (MLP) to a discrete set of speed bins and normalized with a softmax, producing a probability distribution over speeds. The expected speed under this distribution balances the model's certainty about both large-scale patterns (e.g., typical segment speeds) and small-scale variations (e.g., temporary congestion) [8]. Segment-level expected speeds are converted into travel-time estimates (distance divided by speed) and then summed to obtain the total trip time. This decomposition allows losses to be applied at both segment and trip levels.

Through experiments, we will see how each of the two enhancement techniques affect the performance of the model. This will be done in a manner that is similar to the experiments done with feature engineering, that is, using root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) across short, medium, and long trip categories.

4 Experimental Setup

The experimental setup for feature engineering and model evaluation was implemented in Python using PyTorch, scikit-learn, and pandas. We use a crowd-sourced dataset from Berlin [11], which includes GPS-tracked bicycle trip data and map-matched road network data, and split the data into five folds using K-Fold cross-validation.

First, we engineer three main sets of features:

- Baseline features: including raw trip-level attributes like start time, average speed, edge count, and spatial coordinates.
- Distance and speed features: such as log-transformed

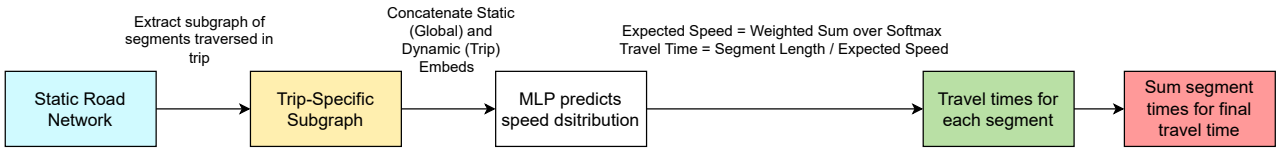

Figure 4: Flowchart detailing Multi-scale Learning

distance, detour ratio, edges per kilometer, coefficient of variation of speed, and speed standard deviation.

- Temporal features: sinusoidal and cosine encodings of time-of-day and month, as well as peak/off-peak hour indicators.

The feature sets are defined as follows (Refer to Table 2 for variable definitions):

- F0: Baseline features
- F1: F0+(ld, cs, dt)
- F2: F1+(st, ekm, tods)
- F3: F2+(todc, ms, mc, ph)
- F4: F0+(ld, dt, ekm)
- F5: F4+(cs, st)
- F6: F0+(tods, todc, ms, mc, ph)

For graph data augmentation, we will apply perturbation techniques to the graph inputs, such as edge deletion, edge addition, node feature masking, and subgraph sampling, following the approaches of You et al. [20] and Liang et al. [12]. This will increase the diversity of the training set and improve model robustness. Augmented graphs will be generated during batch preparation and fed into the graph-based neural network architecture during training.

For multi-scale graph learning, the architecture was inspired by the dual-graph and multi-resolution strategies proposed by Yan et al. [18] and Jiang & Luo [13]. This is achieved by modifying the static graph encoder, that processes the global road network structure, and the trip-specific graph encoder, that models local, sequential patterns along the trip trajectory.

The node-level representation $K_{local} \in \mathbb{R}^{n \times d}$ is combined with a global embedding:

$$K_{\text{global}} = \frac{1}{n} \sum_{i=1}^{n} K_i$$

The global mean vector is broadcast and concatenated to each local node embedding before downstream prediction:

$$K_{\text{multi}} = [K_{\text{local}} \mid K_{\text{global}}] \in \mathbb{R}^{n \times 2d}$$

This approach mimics hierarchical GNN concepts seen in Yan et al. [18], where information at the intersection level (fine-grained) and the road segment level (coarse-grained) is jointly used. However, instead of building separate node- and edge-based graphs, we preserve scalability by reusing the global feature statistics instead of constructing new coarse graphs.

For each feature set and model enhancement, we run experiments through a structured pipeline:

- Data preparation: combine baseline and engineered features, filling missing values, and aligning feature slices for each experiment.
- Model training: using the graph-based neural network, a learning rate of 0.001, batch size of 1024, and 10 training epochs. The network architecture ingests both graph structure (edge index, graph-level features) and the feature-engineered route-level inputs. For the model adjustment we will decide whether to include graph data augmentation, multi-scale graph learning or both.
- Evaluation: we collect Root Mean Squared Error (RMSE), Mean Average Error (MAE), and Mean Absolute Percentage Error (MAPE) on the validation folds, both overall and separately for short (less than 8 minutes), medium (8–16 minutes), and long (over 16 minutes) trips. Evaluation results will be aggregated across folds, and summary statistics (mean and standard deviation) will be reported.

The pipeline will be implemented for efficiency and reproducibility, using pre-saved pickle objects for graph data structures, GPU acceleration where available, and modular code for extending feature sets and evaluation metrics.

The K-Fold cross-validation protocol and using the same evaluation metrics (RMSE, MAE, MAPE) allows direct comparison between the baseline, feature-engineered models, the graph-augmented models and multi-scale models. It will allow us to evaluate which solutions are the most effective and can be combined for better model performance.

5 Results and Analysis

This section presents empirical findings from the two experiments outlined in 4. The two tables Table 3 and Table 4 summarize the results across three distance-based bins - short (less than or equal to 8 minutes), medium (8-16 minutes) and long (over 16 minutes) - as well as overall metrics. We report the RMSE, MAE and MAPE for each setting.

The baseline results of the model rely mostly on raw route geometry and minimal temporal cues as shown in both Table 3 and Table 4. Relatively, it yields high errors on short trips (MAPE = 26.5%). This large error can be attributed to minor delays (e.g., waiting at one light) represent a large fraction of a trip less than 8 minutes.

Feature Engineering. The introduction of three features in F1 initially worsens prediction on short and medium trips. Only for long trip does the RMSE improve slightly, but MAE is higher. This indicates that the initial feature additions,

Feature sets	Short (< 8 min)			Medium (8–16 min)			Long (> 16 min)			Total		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
F0	89.81	61.65	26.51	175.14	124.09	17.66	737.67	255.78	15.25	445.45	147.65	19.87
F1	102.62	68.73	28.82	187.16	130.89	18.66	721.47	263.98	16.06	438.94	155.02	21.26
F2	**83.10**	**57.15**	24.90	**160.30**	**112.61**	**16.05**	729.77	**249.97**	**14.63**	438.90	**140.52**	**18.59**
F3	91.75	59.95	25.80	181.33	123.34	17.57	720.93	255.30	15.16	437.66	146.75	19.58
F4	91.92	60.94	26.25	174.23	120.65	17.17	726.38	253.77	15.08	439.38	145.65	19.56
F5	96.27	60.86	25.99	180.73	120.09	17.15	**701.85**	251.06	14.90	**427.12**	144.52	19.41
F6	88.11	59.89	**24.72**	173.92	121.82	17.31	717.28	254.43	15.17	433.84	145.93	19.12

Table 3: Error metrics (RMSE, MAE, MAPE) for each feature set across different trip-length categories. Minimum values in each column are highlighted in bold red. Refer to Section 4 for feature set definitions.

Model Enhancement Technique	Short (< 8 min)			Medium (8–16 min)			Long (> 16 min)			Total		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
No Enhancement	**89.81**	61.65	26.51	175.14	124.09	17.66	737.67	**255.78**	15.25	445.45	147.65	19.87
Graph Data Augmentation	98.76	63.14	25.55	171.51	119.15	17.06	720.69	262.75	15.64	**436.36**	148.95	19.49
Multi-scale graph learning	99.91	62.44	26.33	184.34	124.26	17.75	**719.30**	258.93	15.45	437.16	149.07	19.91
Combined	92.15	**60.22**	**24.31**	**162.37**	**112.47**	**16.20**	730.04	258.73	**15.04**	439.86	**144.49**	**18.58**

Table 4: Model enhancement results: error metrics (RMSE, MAE, MAPE) across different trip-length categories. Minimum values in each column are highlighted in bold red.

without normalization or combination may be injecting noise. Overall MAPE climbed to 21.26%, which suggests that F1's extra features do not outweigh the variance they introduce yet. Hence, these features may not linearly relate to trip time in shorter routes where rider behavior varies more sharply due to individual choices or frequent interruptions.

By including measures of speed variation (speed_std), route density (edges_per_km) and a sinusoidal temporal feature (tod_sin), F2 corrects much of F1's degradation. For short and medium trips, MAPE reduces to slightly below baseline. MAPE for long trips and overall data remains also shows a slight improvement. These features better represent the fluctuating pace of cyclists and the stop-and-go nature of urban environments. The model's improved performance here—lower errors across all trip lengths—suggests that incorporating variability and structural complexity directly improves realism in time estimation.

Adding more temporal features yields negative effects. Across all trip lengths RMSE, MAE and MAPE spike up. The feature set now has 20 features, which could be causing noise and overfitting. Moreover, this may indicate that cycling behaviors are not strongly correlated with these characteristics. Real-world cycling times are only weakly influenced by long-term seasonality or time granularity, especially when such patterns are diluted by individual rider variability and environmental unpredictability (e.g., wind, traffic).

F4 is a combination of the baseline features plus engineered features that rely on distance. It reduces MAPE on long trips and short trips but the total MAPE shows minimal fluctuation as the medium-trip errors barely show improvement. Hence, while F4 helps the longest trips - suggesting that cold degrees and route degradation metrics matter more on extended trips - its overall benefit is offset by increased short-trip RMSE and medium-trip MAE.

F5 is a feature set comprised of F4 (engineered features that rely on distance) and engineered features that rely on speed. It shows major improvement in the RMSE of long trips thereby drastically improving the overall RMSE as well. F5 shows that carefully adding statistical dispersion can reduce errors, especially when combined with distance-related features.

F4 and F5 highlight how physical attributes like route length and speed dispersion affect long trips. For longer durations, minor behavioral noise is averaged out, and geometric factors—such as detours and route density—play a more deterministic role. F5, in particular, shows a marked reduction in RMSE for long trips, indicating that combining route structure with variability in cyclist speed captures the real dynamics more accurately.

F6 focuses on the addition of only temporal features to the existing feature set F0. Interestingly, it reduces error for short trips—the most time-sensitive to local urban patterns like rush hour—but yields little improvement for longer ones. The result is a new minimal MAPE for short trips (24.72%) which is the lowest MAPE for this trip category. However there are very minimal improvements across other trip-lengths and metrics. This reflects that short trips may be more influenced by time-dependent factors like congestion, while longer trips' variability depends more on the route and rider endurance.

According to Table 3, F2 emerges as the best all-rounder, balancing absolute and relative error across all trip lengths. This is likely because it captures both the variable pace of cyclists and the influence of route complexity without overcomplicating the model. It has an optimal combination of distance, speed and temporal features.

Model Enhancements. The implementation of graph data augmentation, which randomly drops, adds or masks perturbations on the raw graph, slightly improves the model's absolute errors by 2-3%. However the improvements are only visible for medium and long trips. The RMSE of short trips increase from 89.8% to 93.8%. The net effect is a modest overall reduction in RMSE, showing that this enhancement technique alone helps reduce large-route errors at the expense of some noise in short trips.

The improvement in medium and long trips suggests it helps the model generalize better in the face of infrequent route choices or network irregularities—realistic scenarios where cyclists take unfamiliar detours or where GPS errors occur. However, performance on short trips declines, likely because the minor disturbances disproportionately affect already brief durations.

Multi-scale graph learning on its own yields nearly no gain over the baseline results. The only visible improvement is in the RMSE for long trips, where understanding both local road segments and broader network context (e.g., arterial vs. residential routes) can help disambiguate routing choices. In other words, the introduction of hierarchical graph layers alone does not improve overall predictive accuracy.

When the two methods are combined, the model outperforms nearly every other setting. The graph augmentation improves generalization to unpredictable conditions, while multi-scale learning contextualizes local behavior within broader spatial patterns. This synergy confirms that augmenting the graph and introducing layers in the graph complement each other, enabling the enhanced model to more accurately reflect the multiscale, nonlinear nature of urban cycling—where trip time depends on both immediate obstacles (e.g., a busy intersection) and macro-patterns (e.g., route topology, typical flow). While RMSE for short trips do increase, it is offset by the massive gains in other metrics and trip-length categories, especially medium trips. This leads to the best overall MAPE observed in Table 4.

6 Discussion: Implications and Limitations

Our findings demonstrate clearly that thoughtfully engineered features - especially those capturing speed variability, route geometry, and temporal context - significantly enhance bicycle travel time estimation. Specifically, the feature set F2, as seen in Table 3 yielded the lowest overall error across all trip lengths. This align with prior studies emphasizing the importance of feature design, such as Heaton's empirical results on feature engineering [7] and Shi et al.'s SAFE framework for automates feature generation [16].

The enhanced model demonstrates lower prediction errors, particularly for medium and long trips, indicating improved capacity to model complex route dynamics. While the results do not explicitly measure generalization or robustness (e.g., on out-of-distribution data or under adversarial noise), the improved performance across different trip lengths suggests the model may better accommodate variation in cycling patterns.

Architecturally, while graph augmentation and multi-scale graph learning provided modest gain on their own, their combination produced substantial error reductions, particularly for medium-length trips. This mirrors broader GNN trends, where augmentation and hierarchical multi-scale approaches drive performance improvements [14]. It suggests that augmenting both local and global representations is essential for capturing the heterogeneous dynamics of cyclist behavior.

Regarding computational complexity, the DG4b-based approach achieves competitive performance with a relatively small parameter count compared to heavier models, This echoes the trend in Intelligent Transport Systems toward lightweight, efficient GNN models that prioritize scalability and interpretability. The architecture balances accuracy, complexity, and deployment feasibility which is particularly relevant for real-time mobility and bike-sharing application in smart cities.

While there are many improvements, there are also some limitations when it comes to the implementation of feature engineering and model enhancements in the DG4b model.

Returning to Table 3, F3 shows that adding excessive temporal features introduces overfitting, especially in noisy or sparse contexts. To ensure that the data were kept private and secure, individual cyclist traits are missing. As a result, the model does not account for these features in a trip and there may be some merit to exploring the effect of these features on the model. Enhancing the model with individual techniques also seemed to cause higher errors for some metrics, suggesting that those techniques may not be suitable for certain trip-length categories. Lastly, the model may show performance drops in regions where the cycling patterns and infrastructure are different from those of Berlin. Since this study does not include cross-city generalization or stress testing with noisy data, the claims of robustness and generalization should be interpreted cautiously. Future work should include such evaluations to validate model performance under more varied conditions.

7 Responsible Research

This section outlines the principles of responsible research that have guided this study. It focuses on the ethical conduct and the importance of reproducibility throughout the research process.

7.1 Ethical Considerations

The dataset used in this research is a German crowd-sourced dataset collected via a mobile app. Collecting route data from citizens may be considered sensitive data. Hence, the data collected was anonymized to preserve the privacy of the citizens.

This was achieved through three mechanisms: Delaying recording allows users to define a time and a distance threshold after which a recording will start, users were allowed to crop their ride manually to hide where they started or arrived, and per-record pseudonymization stores demographic and ride data separately so that rides cannot be connected to individual users. Furthermore, each ride is pseudonymized separately [11].

7.2 Reproducibility of Research

This research adheres to the FAIR (Findable, Accessible, Interoperable and Reusable) principles [17] to ensure reproducibility and transparency. This paper provides a detailed explanation of the methodology and experimental setup along with its results and analysis. All the libraries used to produce the results are open-source and freely available for public use.

8 Conclusions and Future Work

This research demonstrated that feature engineering and architectural enhancements significantly improve the performance of GCNNs for bicycle travel time estimation. The DG4b model, augmented with carefully designed features and multi-scale graph learning, outperformed traditional baselines across all trip-length categories thereby answering the primary research question.

We explored the effects of feature engineering across six different feature sets and saw that a careful combination of distance, speed variation and time-encoding features yielded the lowest prediction errors across all categories. Too many features seemed to cause too much noise and led to overfitting.

Model enhancements work best when combined. Graph augmentation and multi-scale learning complement each other, yielding better generalization and robustness, especially for long and medium-length trips. This confirms that combining local and global spatial information enables the model to handle heterogeneous trip dynamics more robustly.

Performance–complexity trade-off remains favorable. The model achieves state-of-the-art accuracy with lower complexity and faster training times compared to more complex alternatives. Despite low parameter counts and minimal tuning, the model surpassed more complex methods, demonstrating practical viability.

The experiments in this paper only explored the effect of feature engineering and model enhancements individually. In the future, combining the engineered features with the enhanced model may yield even better predictions, thereby improving its robustness and ability to generalize.

We could also expand the feature set further by incorporating weather, elevation, and land-use data. This could improve predictions particularly for long and uphill routes. Another set of features worth exploring may be rider profiles. If we can find a way to use identifiers that preserve privacy, we can explore how the model can be improved to take into account the cycling experience of a rider. This could enable more personalized estimations.

Applying the framework to new cities with minimal retraining could validate how well the model is able generalize. Another solution may also be to feed the model with more than one road network and train it over a set of trips from different cities to see how well it can generalize.

In summary, this research demonstrates that integrating domain-informed features or robust, lightweight GNN enhancements provides a powerful and efficient solution for bicycle travel time estimation. By extending these insights through combination, personalization and richer contexts, future research can further advance this mode of mobility.

References

[1] Minhao Cheng, Simranjit Singh, Patrick Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack, 2020.

[2] Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, Peter W. Battaglia, Vishal Gupta, Ang Li, Zhongwen Xu, Alvaro Sanchez-Gonzalez, Yujia Li, and Petar Velickovic. ETA Prediction with Graph Neural Networks in Google Maps. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pages 3767–3776, New York, NY, USA, 2021. Association for Computing Machinery. event-place: Virtual Event, Queensland, Australia.

[3] Ting Gao, Winnie Daamen, Elvin Isufi, and Serge Hoogendoorn. Bicycle Travel Time Estimation via Dual Graph-Based Neural Networks.

[4] Jhony H. Giraldo, Konstantinos Skianis, Thierry Bouwmans, and Fragkiskos D. Malliaros. On the trade-off between over-smoothing and over-squashing in deep graph neural networks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 566–576, New York, NY, USA, 2023. Association for Computing Machinery.

[5] Kai Guo, Xiaofeng Cao, Zhining Liu, and Yi Chang. Taming over-smoothing representation on heterophilic graphs. *Information Sciences*, 647:119463, 2023.

[6] Liangzhe Han, Bowen Du, Jingjing Lin, Leilei Sun, Xucheng Li, and Yizhou Peng. Multi-semantic path representation learning for travel time estimation. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):13108–13117, 2022.

[7] Jeff Heaton. An empirical analysis of feature engineering for predictive modeling. In *SoutheastCon 2016*, page 1–6. IEEE, March 2016.

[8] Xiaohui Huang, Junyang Wang, Yuanchun Lan, Chaojie Jiang, and Xinhua Yuan. Md-gcn: A multi-scale temporal dual graph convolution network for traffic flow prediction. *Sensors*, 23(2):841, 2023.

[9] Guangyin Jin, Huan Yan, Fuxian Li, Jincai Huang, and Yong Li. Spatio-temporal dual graph neural networks for travel time estimation. *ACM Trans. Spatial Algorithms Syst.*, 10(3), October 2024.

[10] James Kanter and Kalyan Veeramachaneni. Deep feature synthesis: Towards automating data science endeavors. pages 1–10, 10 2015.

[11] A. Karakaya, J. Hasenburg, and D. Bermbach. Simra: using crowdsourcing to identify near miss hotspots in bicycle traffic. *Pervasive and Mobile Computing*, 67:101197, 2020.

[12] Huidong Liang, Xingjian Du, Bilei Zhu, Zejun Ma, Ke Chen, and Junbin Gao. Graph contrastive learning with implicit augmentations. *Neural Networks*, 163:156–164, 2023.

[13] Wei Liang, Yuhui Li, Kun Xie, Dafang Zhang, Kuan-Ching Li, Alireza Souri, and Keqin Li. Spatial-temporal aware inductive graph neural network for c-its data recovery. *Trans. Intell. Transport. Sys.*, 24(8):8431–8442, August 2023.

[14] Saeed Rahmani, Asiye Baghbani, Nizar Bouguila, and Zachary Patterson. Graph Neural Networks for Intelligent Transportation Systems: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 24(8):8846–8885, August 2023.

[15] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification, 2020.

[16] Qitao Shi, Ya-Lin Zhang, Longfei Li, Xinxing Yang, Meng Li, and Jun Zhou. Safe: Scalable automatic feature engineering framework for industrial tasks, 2020.

[17] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J. Boiten, L. O. B. d. S. Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. W. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. González-Beltrán, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. '. Hoen, R. Hooft, T. Kuhn, R. Kok, J. N. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. v. Schaik, S. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. v. d. Lei, E. M. v. Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, and B. Mons. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3, 2016.

[18] Huan Yan, Guangyin Jin, Deng Wang, Yue Liu, and Yong Li. Jointly modeling intersections and road segments for travel time estimation via dual graph convolutional networks. In H Wu, Y Liu, J Li, J Meng, Q Guan, X Song, G Liao, and G Li, editors, *SPATIAL DATA AND INTELLIGENCE, SPATIALDI 2022*, volume 13614 of *Lecture Notes in Computer Science*, pages 19–34. ACM SIGSPATIAL China Branch; ACM SIGMOD China Branch; China Univ Geosciences; Wuhan Univ; Huazhong Univ Sci & Technol; China Geogr Informat Ind Assoc Theory & Method Working Comm; Int Chinese Geog Informat Sci Assoc, 2022. 3rd International Conference on Spatial Data and Intelligence (SpatialDI), Wuhan, PEOPLES R CHINA, AUG 05-07, 2022.

[19] Jiexia Ye, Juanjuan Zhao, Kejiang Ye, and Chengzhong Xu. How to Build a Graph-Based Deep Learning Architecture in Traffic Domain: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(5):3904–3924, May 2022.

[20] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations, 2021.

[21] Xin Zheng, Yi Wang, Yixin Liu, Ming Li, Miao Zhang, Di Jin, Philip S. Yu, and Shirui Pan. Graph neural networks for graphs with heterophily: A survey, 2024.