Hybrid AI Models for Structured Mobility Prediction in Metropolitan Areas

Adrian M. P. Braşoveanu[1,2](✉), Lyndon J.B. Nixon[1,2], and Arno Scharl[1,3]

[1] Modul University Vienna, Am Kahlenberg 1, 1190 Vienna, Austria
{adrian.brasoveanu,lyndon.nixon,arno.scharl}@modul.ac.at
[2] Modul Technology, Am Kahlenberg 1, 1190 Vienna, Austria
[3] webLyzard technology, Liechtensteinstraße 41/26, 1090 Vienna, Austria
https://www.modul.ac.at, https://www.modultech.eu, https://www.weblyzard.com

Abstract. This paper introduces hybrid AI models for structured mobility prediction in metropolitan areas, focusing on Vienna, to guide citizens toward greener transportation options. The AI-CENTIVE project explores how AI can identify effective incentives by forecasting future trips using a combination of traditional machine learning and modern deep learning architectures. Trained on a dataset of commuter trips from the Ummadum app, the models predict transport mode, time, origin, destination, distance, and duration. The most accurate predictions trigger notifications suggesting sustainable alternatives. The evaluation of various hybrid architectures revealed that a graph convolutional network that uses statistical patterns achieved the best performance on the analyzed dataset. The presented research contributes to leveraging AI to promote sustainable mobility through targeted incentivization.

Keywords: Structured Prediction · Graph Convolutional Network · Transformers · Hybrid AI models · Mobility

1 Introduction

Supporting sustainable mobility is a complex challenge, as environmental concerns alone rarely prompt people to change their long-standing travel habits, such as using private cars. Mobility decisions are shaped by diverse factors, including affordability, convenience (e.g., overcrowded buses discourage use), available infrastructure (e.g., bike lanes), individual capabilities (e.g., driving licenses), or accessibility (e.g., inclusivity for people with disabilities).

This work presents findings from the AI-CENTIVE research project,[1], which explores how Artificial Intelligence (AI) can identify incentives to guide citizens toward greener transportation options, e.g., opting for bikes or public transport instead of private cars for commuting. The focus is on a metropolitan

[1] www.aicentive.eu.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
I. Rojas et al. (Eds.): IWANN 2025, LNCS 16008, pp. 628–639, 2026.
https://doi.org/10.1007/978-3-032-02725-2_49

area (Vienna) to understand commuting habits within and outside city limits. Our model evaluates hybrid architectures that combine traditional machine learning (ML) models with modern deep learning architectures, such as Transformers and graph neural networks. These models are trained to forecast future trips based on mode of transport, time, origin, destination, distance, and duration. The forecasting is done using structured prediction, an ML approach that predicts all these independent outputs simultaneously. Tree-based models (e.g., XGBoost) [23] and LSTMs demonstrated a good balance between performance and interpretability [4]. For routing problems, such as estimating the time of arrival (ETA), graph convolutional networks (GCNs) have also been considered [9]. Transformers are also increasingly applied to forecasting tasks due to their capacity to model complex dependencies [20]. Sometimes, it may be necessary to combine these architectures to achieve the best results for a particular route or user.

After establishing a baseline for mobility prediction models, we incorporate data from a real-world incentivization experiment using Ummadum, a mobile app that rewards users for sustainable trips in Austria. By comparing baseline predictions with outcomes from incentivized scenarios, we aim to assess the environmental benefits of behavioral change and provide practical guidance to policymakers and mobility providers. This paper focuses on an essential part of this process: the workflow for generating user notifications based on predicting their upcoming mobility behavior. This uses hybrid models based on Transformers or GCNs to create structured predictions for future trips.

The paper is organized as follows: Sect. 2 summarizes related work, Sect. 3 describes the data collection method and resulting AI models, whereas Sect. 4 presents the results from a first user pilot with predictions from our AI model. The paper concludes with a brief overview and outlook for future work.

2 Related Work

The number of works focused on mobility has increased exponentially since the introduction of deep learning architectures. It would be impossible to review all of them. Therefore, the following section highlights some well-known surveys and research articles to help us understand how to approach this field.

Tedjopurnomo et al. [19] provide a comprehensive survey on traffic prediction. They include datasets and conditions such as input lengths, forecast horizons, and model choices like LSTMs or Deep Belief Networks. The paper describes various prediction targets like traffic flow, speed, and crowd dynamics. The survey identifies key challenges, including capturing spatiotemporal patterns, improved benchmarking datasets that also encompass rural areas, online learning, and adaptability to new tasks. Lim et al. [12] investigate time-series forecasting from a broader perspective, proposing hybrid approaches to integrate auxiliary features. These include both deterministic and probabilistic methods for generating predictive distributions. For longer sequences, Chen et al. [2] extend similar strategies. Yin et al. [21] propose blending traditional models (e.g., statistical models, matrix factorization) with deep learning to improve interpretability and enhance representation power.

Transformer architectures have also been proposed for time series forecasting [1], leading even to a new class of TS-PTMs (Time-Series Pre-Trained Models) [14]. Such models are pre-trained on classification or traditional forecasting tasks and later fine-tuned on desired target domains to improve the performance of downstream tasks (e.g., anomaly detection, route prediction, ETA, time-series classification). Large datasets with multiple categories and time series are typically preferred for pre-training such models. This idea has been further extended by applying foundation models to time series, shaping another class of models called TSFMs (Time Series Foundation Models) [11].

In another survey, Jiang et al. [9] look at GCNs in traffic forecasting. They map various applications, such as road conditions, flow, or regional mobility data, and explore graph construction strategies (e.g., road-level, sensor-level) and multiple adjacency matrix types. The paper reviews models like graph neural models. Rahmani et al. [16] go one step further and provide a comprehensive review focused on the application of GCNs to all areas of transportation, including parking, safety, self-driving vehicles, or even urban planning. Beyond all the surveys that can help us identify current trends, the Google Maps GNN production model, focused on estimated time of arrival (ETA) [3], deserves special attention, as the paper provides notes on ablation studies and qualitative analyses on real-world data. Another important note regarding Google should be made for their willingness to adapt to ever-changing regulations (e.g., the recent move of personal travel histories from their servers to mobile devices).

Given the importance of understanding why AI models make certain predictions, explainability is vital for identifying and addressing biases (e.g., from imbalanced datasets). Schwalbe and Finzel [17] provide an extensive survey that reviews more than 50 surveys dedicated to this topic.

3 Method

This section describes the method developed to examine the metropolitan Vienna mobility dataset, as well as the overall architecture and workflows used to send notifications to users.

3.1 Dataset

To understand the Vienna metropolitan area, we examined a dataset created by Ummadum, a project partner of AI-CENTIVE. The dataset focuses on commuter communities who log trips taken in Vienna and its surrounding villages and cities. This dataset covers daily trips for users registered with the Ummadum mobile application. The dataset is updated monthly. Due to the application's growth, around 20,000 new trips are added monthly. The current dataset comprises approximately 540,000 data points spanning from April 2022 to May 2025. The dataset is built around several types of user activities, namely walking (WALK), biking (BIKE), public transport (PT), and ridesharing as a driver (CAR_DRIVER) or rider (CAR_RIDER). Data about activity status is also collected (e.g., STARTED, FINISHED, CANCELLED, etc.). User data is anonymised according to the GDPR. The location of each activity is also anonymized (e.g., segments were added to origin and destination so that a user's real location cannot be identified). The mapping from latitude/longitude to zip codes was made using an API based on OpenStreetMap (OSM) called Nominatim. In cases where coordinates did not point towards a precise location (e.g., a clear district), the nearest neighbor method was used. Data about each trip (origin and destination are expressed zip codes) and location type (e.g., home or office) is included in the dataset. Since one of the main goals of this experiment is to understand which incentives lead to choosing more sustainable mobility choices for trips, the list of active rewards is included in the data. This covers Ummadum Points, which are earned for making sustainable trips and may be converted to discounts in participating stores, community type (i.e., the type of community that provides the reward (e.g., company, municipality, marketing, management), carsharing (i.e., if the activity was rewarded with points for sharing a car trip with other passengers), (use of a) free parking space, or activity challenge where users earn additional rewards based on their activity levels.

3.2 Hybrid Models

After analyzing the data, we built a series of hybrid models for generating structured predictions for each trip. Instead of predicting each attribute separately, all the attributes are predicted simultaneously. The output contains the activity type, the time (date and hour), distance, duration, origin, and destination for each trip a user might perform. Structured prediction is an ML approach that simultaneously predicts multiple interdependent output variables while preserving their relationships and constraints.

While hybrid models are frequently used, they have different meanings in various contexts. For example, one can use the term *hybrid* to describe ensembles of lexical and deep learning models for sentiment analysis. Still, one can use the same term when combining traditional statistical and deep learning models. On a broader level, the term is also used to describe systems that combine symbolic (e.g., logic-based) reasoning and subsymbolic (e.g., data-driven) learning. For this paper, the term *hybrid models* combines traditional ML models with modern deep learning architectures. *Traditional models* refers to ML models typically used before the advent of deep learning between 2014 and 2016. The rest of this section briefly explains the terms we used for our definition.

Traditional methods described in [7] include linear or kernel methods. They are often considered to be parametric, interpretable (e.g., precise meanings for parameters), low data requirements (e.g., can work with limited samples), white-box (e.g., behavior and confidence intervals can be established analytically), and optimizable by theory. Linear regression or ARIMA (auto-regressive integrated Moving Average) models, which are heavily used for time series forecasting [8], belong to this category. In contrast, deep learning methods [6] are non-parametric and have low interpretability (e.g., while feature importance can be computed, it is often not enough), high data requirements (e.g., they need a lot of samples), black-box (e.g., behavior and confidence intervals are often messy and difficult to be established), and hard to optimize. Methods like decision trees or gradient boosting (e.g., XGBoost) [7] fall right in the middle between traditional and deep learning methods, as they are non-parametric, and they have medium interpretability (e.g., feature importance available through SHAP or similar libraries), medium data requirements, white-box, and somewhat easier to generalize.

In recent years, predictive analytics has increasingly shifted toward leveraging modern deep learning architectures (e.g., Transformers, Graph Neural Networks) due to their ability to capture complex, nonlinear patterns and significantly improve forecasting accuracy compared to traditional statistical methods. Such models are necessary when incorporating features from multiple domains, such as mobility, news media, or weather prediction. However, traditional models (e.g., decision trees or gradient boosting) still perform very well for time series forecasting tasks (see [18]). Ideally, we wanted the deep learning models to capture nonlinear patterns and for the traditional models to interpret the results easily. Due to this aspect, we combined traditional and deep learning architectures. This idea was proposed as early as 2003, before the advent of deep learning architecture, by Zhang's seminal paper [22], and recently reintroduced through articles like [5] and [15] using various types of neural architectures.

The next paragraphs describe how we analyzed our data and built the application workflows.

3.3 Architecture and Workflows

After examining the data, we have created several notification workflows linking model predictions to user incentives:

- *Classic* - includes success notifications acknowledging previous sustainable mobility efforts to reward and motivate participants;
- *AI* - recommends future sustainable mobility to participants using contextual information (e.g., behavior, mobility preferences, location, time) with the goal of (re-)activating participants;
- *Weather* - recommends future trips according to local weather forecasts.

We implemented the classic workflow and tested it for several months through a pilot with real users. The number of users who started using the application more frequently increased, and the overall number of users interested in the AI-CENTIVE community with these notifications has also increased. The notifications resulting from this workflow were sent weekly on a single day at the same time (e.g., typically Tuesdays at 8:00 a.m.).

Next, we implemented the weather workflow. Initially, these were not well received due to multiple factors. Because Vienna exhibits microclimates, it is often difficult to predict the weather in certain districts [13]. Additionally, designing sound user notifications and rewards in such conditions was also problematic, as it would not have been easy to send such messages to particular users without providing a lot of background information about how to interpret the notifications themselves. This was not ideal for us, as we wanted a simple system that everyone could easily understand and that followed the current European legislation (e.g., GDPR, the AI Act). We have ultimately decided to send notifications only for severe weather alerts.

In parallel, we have also implemented the AI recommendations. The workflow idea was to create an AI model for structured prediction (e.g., instead of simply predicting ETA or distance, we forecast multiple attributes simultaneously) and select the best predictions generated by this model to send daily notifications to users based on their habits. After an initial testing stage, we decided to reduce the number of notifications to several per week (e.g., two or three), as daily notifications were not well-received (e.g., faced with multiple notifications per day, users were more likely to consider them spam). Eventually, the workflow was tested with real users and was well-received.

The overall architecture, which combines these workflows, is presented in Fig. 1. The dataset and the three workflows are the centerpieces. Today, notifications are sent daily at various hours based on the predictions. Even the classic success notifications are spread throughout the week. This is achieved with the help of a mixer component that splits all notifications by day and user based on simple business logic (e.g., no more than one notification per day for each user and no more than three notifications per week for each user). A task scheduler is then used to send notifications directly to the users via the Ummadum notification API. All workflows are run once per week on Monday morning, and notifications are then scheduled for the remainder of the week.

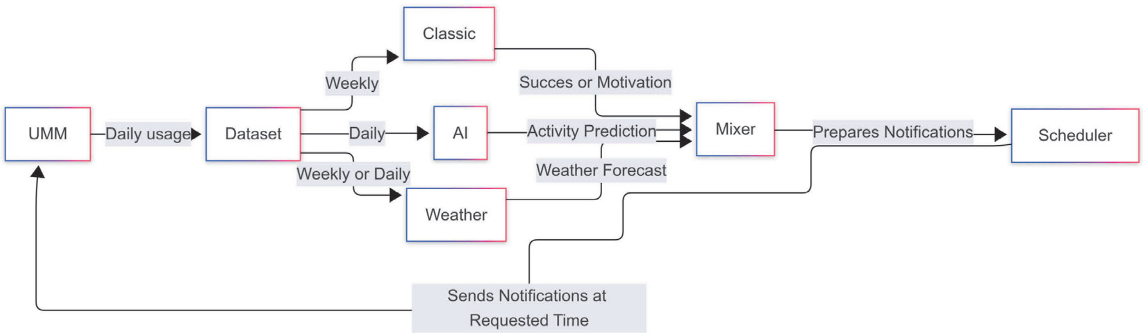

Fig. 1. Flow diagram of the notification architecture components.

Keeping everything simple and practical has enabled us to implement these workflows relatively quickly. However, selecting good hybrid models took longer, as the evaluation process was difficult, as seen in the next section.

4 Evaluation

The evaluation section describes the hybrid models we examined, the selection of the best models, and the evaluation strategies.

4.1 Models

We have examined multiple models based on Transformers and GCNs. The best model was selected based on the best combined loss, MSE, and RMSE test scores. The loss function combines normalized cross-entropy losses for categorical attributes and normalized MSE for numeric attributes, ensuring balanced multi-objective optimization across all prediction tasks.

The Transformer. Base model is a multi-output structured prediction baseline that forecasts multiple targets simultaneously. The model processes various feature types (categorical, numerical, and temporal) through an initial projection layer, followed by self-attention mechanisms that help capture complex relationships between features. Input data undergoes comprehensive preprocessing, including categorical encoding, feature scaling, and temporal feature extraction, before being fed through the model's transformer layers. The model processes sequence data through transformer layers and aggregates information using simple global average pooling before sending it to specialized prediction heads. These target-specific heads allow the model to handle multiple prediction tasks simultaneously, including regression (distance, duration), classification (activity types, locations), and timestamp forecasting. During training, the model optimizes multiple loss functions concurrently (MSE for regression, cross-entropy for classification) and employs early stopping to prevent overfitting. The model handles different prediction types with specialized output formats: regression values are exponentiated after prediction (reversing log transforms), classification outputs are converted to most-likely class labels, and datetime predictions are transformed back into proper timestamp format. The other Transformer models build upon this model and typically combine two architectures (e.g., a classic model and a deep learning model).

The hybrid Transformer + XGBoost model (T.XGB) implements structured prediction by generating multiple trip attributes simultaneously through a multi-head architecture. The transformer backbone processes sequential trip data to extract shared representations, which feed into specialized output heads for different attribute groups (location, time, activity, etc.) while producing global embeddings for XGBoost enhancement. These embeddings are combined with user-specific features to train dedicated XGBoost models for each prediction target, improving individual attribute forecasting while maintaining inter-attribute relationships. The model integrates these predictions, handling normalization conversions and ensuring proper generation of categorical attributes (activity types, postal codes) and numeric values (distance, duration). This architecture captures the dependencies between trip attributes through shared representations, enabling holistic trip prediction rather than treating each attribute as an independent target.

The hybrid Transformer + ARIMA (T.ARIMA) model processes input features through Transformer encoder layers, extracting shared representations that feed into specialized output heads for each attribute type—classification heads for categorical variables and regression heads for numeric ones. Temporal patterns are captured using the neural network's sequence modeling capabilities and dedicated ARIMA models for each attribute (e.g., activity type, distance, duration, etc.), with the final outputs representing a weighted blend (75% Transformer, 25% ARIMA). Training employs AdamW optimization with early stopping and learning rate scheduling. At the same time, predictions undergo careful post-processing to ensure valid outputs, including appropriately handling postal codes and activity types based on historical patterns.

The GCN.Base model processes trip data using a single graph convolutional layer that captures basic spatial relationships combined with simple user embeddings and temporal features. This baseline model employs a shared hidden representation followed by specialized prediction heads for each attribute, including separate components for hour and day prediction to generate start times accurately. It trains using a weighted loss function that balances the importance of each prediction component, with additional emphasis placed on temporal predictions. This model focuses on the core structured prediction task with minimal computational overhead.

The GCN.XAI model builds upon the GCN.Base model and adds a classic statistical model (user histories) and an explainable AI component. The architecture consists of multiple graph convolutional layers followed by global mean pooling, with dedicated embedding layers for users and activity types, incorporating temporal features through normalized time representations. The model's final linear layer outputs all the target attributes through a unified representation that captures their correlation. Training incorporates enhanced regularization techniques (L2 regularization for embeddings, dropout, and gradient clipping) with an AdamW optimizer and learning rate scheduling based on validation performance. A distinctive feature is the explanation framework that computes confidence scores for predictions based on historical user patterns, analyzing route frequencies, time patterns, and activity distributions. Trip predictions incorporate personalized commuting behaviors, differentiating between weekday patterns (with distinct morning/evening commutes) and weekend travel, with each prediction accompanied by an explanation detailing the rationale behind route selection and estimated confidence scores. The architecture considers numerical stability through standardization and outlier handling, while early stopping based on RMSE performance prevents overfitting.

4.2 Discussion

As already explained in Sect. 3.1, the dataset includes user and trip identification, activity details (e.g., transport mode, status), geolocation tagging (e.g., timestamp, distance, duration), incentivization system (e.g., communities, points, challenges, points budgets), and notification data (e.g., time, type).

For evaluation purposes, the training dataset contained several weeks of data (January–March 15th, 2024), the test dataset focused on the following two weeks. After several experiments, we decided that 10–12 weeks of training was ideal for our use case for multiple reasons, including the duration of training and trip recency. The average training time for 50 epochs was around 10 min, regardless of the model. This smaller evaluation dataset was needed due to the restriction of creating notifications in near real time (e.g., several minutes to run the workflows) using the latest data. The code ran on L4 and A100 GPUs.

Table 1. Comparison of Model Performance (values rounded to three decimals). T stands for Transformers. GCN abbreviates Graph Convolutional Networks. Best model highlighted in bold.

Model	Train Loss	Test Loss	Train MSE	Test MSE	Train RMSE	Test RMSE
T.Base	0.000	0.000	0.000	0.000	0.017	0.019
T.XGB	0.011	0.009	0.034	0.186	0.027	0.151
T.ARIMA	0.560	0.619	0.560	0.530	0.619	0.560
GCN.Base	0.000	0.000	0.000	0.015	0.000	0.019
GCN.XAI	**0.000**	**0.000**	**0.000**	**0.011**	**0.000**	**0.010**

Table 2. Generalization gap (Test - Train) for model performance (values rounded to three decimals). Best model highlighted in bold.

Model	Loss Gap	MSE Gap	RMSE Gap
T.Base	0.000	0.000	0.002
T.XGB	−0.002	0.152	0.124
T.ARIMA	0.059	−0.030	−0.059
GCN.Base	0.000	0.015	0.019
GCN.XAI	**0.000**	**0.011**	**0.010**

Table 1 presents the evaluation results. The classic metrics (normalized loss, MSE, RMSE) have been adapted for structured prediction. This was done via per-batch aggregation (e.g., batch_loss computes the average loss across all fields from a single batch of data) and per-epoch aggregation (e.g., by tracking how metrics evolve over the entire epoch). GCN models yield the best results for the smaller period discussed in this article and for any other periods from the larger dataset. The best model was selected based on the combined metrics.

Table 2 showcases the generalization gap [10] for the models. Defined as the difference between test and training performance metrics, it tracks how the model's performance degrades when exposed to unseen test data. A larger positive gap indicates overfitting, where the model has memorized training data patterns. This is a fundamental measure of model generalization ability, with smaller gaps indicating more robust models that perform consistently across different datasets.

The best model (GCN.XAI) includes a more sophisticated prediction approach that differentiates between weekday and weekend travel patterns and incorporates user-specific commuting behaviors. Temporal features are added as an input to the forward method (e.g., hour of day and day of week) and used in prediction. This model also employs more advanced regularization techniques, including dropout layers, L2 regularization specifically for embeddings, learning rate regularization, and explicit gradient clipping, making it more robust against overfitting. Similarly to the concept of explainability through verbalization, the predictions are explained based on the user's history. While the overall scores are second best, this model predicts the top scores for users with high confidence. Therefore, it was considered better only to send users the best predictions.

Route handling is the most critical difference between the GCN models. The GCN.Base model relies on randomized selection from known postal codes without considering route plausibility for new routes. The GCN.XAI model implements a second confidence metric specifically tailored for unseen routes. This metric considers: i) the user's pattern stability (how consistent their travel patterns are); ii) location familiarity (if the user has been to either the origin or destination before); iii) activity type; and iv) the time pattern reliability. Due to this aspect, this model can still leverage partial knowledge about new routes (e.g., past trips to the same destination from other starting points) to generate and assess predictions. At the time of writing, the GCN.XAI model is the main model used in the second AI-CENTIVE pilot focused on AI notifications. We will collect the study results during the Summer of 2025 and analyze the effects on sustainable mobility incentivization to improve the results further.

5 Conclusion

This paper focused on building hybrid models for structured mobility prediction in metropolitan areas like Vienna. The design space of hybrid models, however, is infinite. Therefore, the solutions we explored represent only a tiny fraction of what is possible to build when combining various models. We have focused on combining neural and classic models, as there can be a tendency to overlook traditional models' benefits when newer technologies emerge. Some statistics can go a long way and should not be dismissed simply because we have better models. Sometimes, better models come with additional problems like longer training times, more complex infrastructure, and difficult debugging. Combining a simple neural model with a classic model allows us to sidestep these kinds of issues and craft simple solutions that can help in near-real-time scenarios.

In upcoming work, we will integrate emissions data from the mobility CO_2 calculator of the *University of Natural Resources and Life Sciences* (BOKU) in Vienna to quantify the environmental impact of different incentivisation strategies. This will offer a concrete measure of emission reductions achieved through shifts in travel behaviour. We also plan to integrate weather features directly into the hybrid models to create more accurate predictions, and to evaluate total transportation emissions to understand mobility-related environmental costs.

Equally important is fostering a growing community committed to low-impact mobility choices. Tracking the rise in adopting sustainable transport options will help assess behavioral change. Additionally, we aim to address urban problem zones—areas lacking parking or reliable transport within walking distance. Measuring reductions in such zones will demonstrate improvements in accessibility and urban livability. These new metrics will deepen our predictive models and contribute to an inclusive, data-informed approach to sustainable transport and urban development. Through AI-CENTIVE, we hope to support citizens and stakeholders in building more adaptive, fair, and ecologically sound cities.

Acknowledgments. This work was partially funded through the AI-CENTIVE research project (BMK/FFG grant agreement number 892238). Special thanks to Günther Charwat and Egon Prünster from Ummadum for designing the sustainable mobility dataset and the notification messages.

Disclosure of Interests. The authors have no competing interests to declare relevant to this article's content.

References

1. Ahmed, S., Nielsen, I.E., Tripathi, A., Siddiqui, S., Ramachandran, R.P., Rasool, G.: Transformers in time-series analysis: a tutorial. Circuits Syst. Signal Process. **42**(12), 7433–7466 (2023). https://doi.org/10.1007/S00034-023-02454-8
2. Chen, Z., Ma, M., Li, T., Wang, H., Li, C.: Long sequence time-series forecasting with deep learning: a survey. Inf. Fusion **97**, 101819 (2023). https://doi.org/10.1016/J.INFFUS.2023.101819
3. Derrow-Pinion, A., et al.: ETA prediction with graph neural networks in Google Maps. In: Demartini, G., Zuccon, G., Culpepper, J.S., Huang, Z., Tong, H. (eds.) CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021, pp. 3767–3776. ACM (2021). https://doi.org/10.1145/3459637.3481916
4. Dunn, J., Mingardi, L., Zhuo, Y.D.: Comparing interpretability and explainability for feature selection. CoRR **abs/2105.05328** (2021). https://arxiv.org/abs/2105.05328
5. Gasparin, A., Lukovic, S., Alippi, C.: Deep learning for time series forecasting: the electric load case. CAAI Trans. Intell. Technol. **7**(1), 1–25 (2022). https://doi.org/10.1049/cit2.12060
6. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep Learning, vol. 1. MIT Press, Cambridge (2016). https://www.deeplearningbook.org/
7. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer Series in Statistics, Springer (2009). https://doi.org/10.1007/978-0-387-84858-7
8. Hyndman, R.J., Athanasopoulos, G.: Forecasting: Principles and Practice. OTexts, 3rd edn. (2021). https://otexts.com/fpp3/
9. Jiang, W., Luo, J.: Graph neural network for traffic forecasting: a survey. Expert Syst. Appl. **207**, 117921 (2022). https://doi.org/10.1016/J.ESWA.2022.117921

10. Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P.: On large-batch training for deep learning: generalization gap and sharp minima. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017). https://openreview.net/forum?id=H1oyRlYgg
11. Liang, Y., et al.: Foundation models for time series analysis: a tutorial and survey. In: Baeza-Yates, R., Bonchi, F. (eds.) Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024, pp. 6555–6565. ACM (2024). https://doi.org/10.1145/3637528.3671451
12. Lim, B., Zohren, S.: Time series forecasting with deep learning: a survey. CoRR **abs/2004.13408** (2020). https://arxiv.org/abs/2004.13408
13. Loibl, W., Vuckovic, M., Etminan, G., Ratheiser, M., Tschannett, S., Österreicher, D.: Effects of densification on urban microclimate — a case study for the city of vienna. Atmosphere **12**(4) (2021). https://doi.org/10.3390/atmos12040511
14. Ma, Q., et al.: A survey on time-series pre-trained models. IEEE Trans. Knowl. Data Eng. **36**(12), 7536–7555 (2024). https://doi.org/10.1109/TKDE.2024.3475809
15. Mathonsi, T., van Zyl, T.L.: A statistics and deep learning hybrid method for multivariate time series forecasting and mortality modeling. Forecasting **4**(1), 1–25 (2021). https://doi.org/10.3390/forecast4010001
16. Rahmani, S., Baghbani, A., Bouguila, N., Patterson, Z.: Graph neural networks for intelligent transportation systems: a survey. IEEE Trans. Intell. Transp. Syst. **24**(8), 8846–8885 (2023). https://doi.org/10.1109/TITS.2023.3257759
17. Schwalbe, G., Finzel, B.: A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. Data Min. Knowl. Discov. **38**(5), 3043–3101 (2024). https://doi.org/10.1007/S10618-022-00867-8
18. Taylor, S.J., Letham, B.: Forecasting at scale. Am. Stat. **72**(1), 37–45 (2018). https://doi.org/10.1080/00031305.2017.1380080
19. Tedjopurnomo, D.A., Bao, Z., Zheng, B., Choudhury, F.M., Qin, A.K.: A survey on modern deep neural network for traffic prediction: trends, methods and challenges. IEEE Trans. Knowl. Data Eng. **34**(4), 1544–1561 (2022). https://doi.org/10.1109/TKDE.2020.3001195
20. Wen, Q., et al.: Transformers in time series: a survey. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China, pp. 6778–6786. ijcai.org (2023). https://doi.org/10.24963/IJCAI.2023/759
21. Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., Yin, B.: A comprehensive survey on traffic prediction. CoRR **abs/2004.08555** (2020). https://arxiv.org/abs/2004.08555
22. Zhang, G.P.: Time series forecasting using a Hybrid ARIMA and neural network model. Neurocomputing **50**, 159–175 (2003). https://doi.org/10.1016/S0925-2312(01)00702-0
23. Zhu, L., Shu, S., Zou, L.: XGBoost-based travel time prediction between bus stations and analysis of influencing factors. Wirel. Commun. Mob. Comput. **2022**(1), 3504704 (2022). https://doi.org/10.1155/2022/3504704