Demand Forecasting with Machine-Learning Models for Bike-Sharing-Systems based on Open Data

Christian Wirtgen*[1], Matthias Kowald[2], and Johannes Luderschmidt[1]

[1]Department of Design, Computer Science, Media, RheinMain University of Applied Sciences, Germany
[2]Department of Architecture and Civil Engineering, RheinMain University of Applied Sciences, Germany

Short summary

The utilization of climate-friendly and cost-efficient bike-sharing systems (BSS) is gaining worldwide acceptance. For operators, rental demand constitutes a pivotal factor in operational and strategic decision-making, generating high-quality forecasts of this demand remains challenging. This study evaluates novel machine learning (ML) architectures, which have not yet been applied in the BSS domain, using open data. The study's findings indicate that Transformer and Long Short-Term Memory (LSTM) models demonstrate superiority in terms of forecast accuracy when compared to other models, including DLinear, Temporal Convolutional Network (TCN), and Time-series Dense Encoder (TiDe). Additionally, the study underscores the utility of open historical data sources in deriving pertinent features associated with BSS demand.
Keywords: Demand Forecasting, Machine Learning, Bike-Sharing-Systems, Big Data Analytics

1 Introduction

Bike-sharing systems (BSS) have been adopted in numerous cities worldwide, offering citizens an environmentally sustainable transportation option and many advantages over conventional modes of transport. As part of Vehicle Sharing Systems (VSS) and micromobility (Ataç et al., 2021), analyzing and predicting demand is becoming increasingly important for operators. Accurate demand forecasts support strategic decisions, including efficient bike rebalancing, service quality improvements, and station expansion (Collini et al., 2021; Jiang, 2022).

Predictive analytics facilitates forecasting future demand using historical data, with aggregated time series enabling time series forecasts (TSF). However, selecting appropriate models and generating high-quality predictions remains a challenge (Sun et al., 2020). Many studies evaluate models using only regional systems, raising concerns about transferability (Loidl et al., 2019). Additionally, factors like training time and model complexity are often overlooked (Jiang, 2022). This lack of standardization leaves operators uncertain about which models to adopt. Variations in datasets, metrics, and challenges complicate study comparisons, with many models focusing solely on short-term predictions, such as demand for the next minutes or hours (Ma et al., 2022).

Table 1: Overview of forecast durations

Task	Aggregation	Use case
Short-Term (STSF)	minutely (min), hourly (h)	Short-term forecasts for stations or districts, consideration of seasonalities such as times of day
Mid-Term (MTSF)	daily (D), weekly (W)	Medium-term developments of the system or for individual stations, consideration of seasonalities such as day of week
Long-Term (LTSF)	monthly (M), quarterly (Q), yearly (Y)	Longer-term developments of the system, consideration of seasonalities such as seasons

The performance of forecast models for tasks such as mid-term (MTSF) or long-term forecasts (LTSF) remains unclear. Table 1 provides an overview of forecast categories. Different factors

depend on the selected forecast type: short-term forecasts emphasize temporal features like times of day, while long-term forecasts require models to account for seasonalities.
Exogenous determinants of BSS can be incorporated into models to enhance forecast accuracy. This study addresses the following research questions:

- Which forecast models are suitable for specific durations or time periods?
- Which exogenous determinants are appropriate for different tasks?
- Can certain determinants be universally applied across systems or durations?

This study evaluates novel Machine Learning (ML) architectures, such as LTSF-Linear and Time-series Dense Encoder (TiDe), which have not yet been applied to BSS. It also examines time aggregations and horizons. The findings are relevant not only to BSS operators but also to the scientific community, as they can be adapted to systems like car-sharing (CSS) and (e-) scooter-sharing (SSS), which share similar determinants and dependencies.

2 Methodology

A data and ML pipeline was developed (see figure 1). The process begins with collecting and storing bicycle-sharing data, along with secondary data such as meteorological and sociodemographic information.

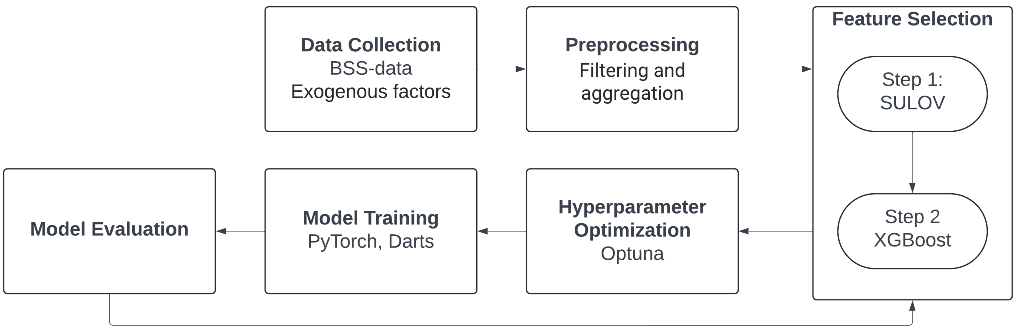


Figure 1: Overview of the Data & ML-pipeline

In the preprocessing step, data is filtered, checked for plausibility, and normalized (see Chapter Data Collection & Preprocessing). Input features are then generated, followed by a two-stage selection process to create a minimal, optimal feature subset. Model architecture details are provided in Chapter Machine Learning Models for Time Series Forecasting, and hyperparameter optimization using pruning is described in Chapter Model Optimization and Training. Model evaluation is conducted with various error metrics (see Chapter Model Evaluation & Metrics).

Data Collection & Preprocessing

Four publicly available BSS datasets from different countries were collected. The *Divvy* system, operated by Lyft in Chicago (USA), recorded over 16.2 million rentals since 2020. In Madrid (Spain), the city-owned *BiciMAD* achieved more than 22.3 million rentals since 2017, similar to *Bike Share Toronto* (Canada) with 22.6 million rentals. The *OsloBysykkel* system in Oslo (Norway) recorded 13.2 million rentals since 2016. Historical data is available on the respective corporate websites.

A BSS dataset must meet these criteria: operation for at least one year, current operational status, and inclusion of trip details with time stamps for departure and return. Most datasets include attributes like rental ID, station name/ID, GPS coordinates, and bike ID. However, due to data protection and privacy concerns, sociodemographic details such as age or sex are generally excluded. All records with incomplete or invalid GPS coordinates or missing start/destination data were removed. Stations with fewer than 10 rentals per year, including service or test stations, were excluded. Rentals were aggregated over time, producing a time series y_t with observed rents y_n for each time point $t = 1, ..., n$. Table 2 shows the sizes of the resulting time series.

Table 2: Summary of the individual time series and their number of time points

System	Frequency	#Timepoints
BikeChicago	h	32857
	D	1370
	W	196
	M	45
BikeMadrid	h	51585
	D	2151
	W	308
	M	71
BikeOslo	h	67916
	D	2831
	W	405
	M	93
BikeToronto	h	61321
	D	2556
	W	366
	M	84

Exogenous factors

Incorporating exogenous factors enhances model performance by including independent effects not explained by the model. External influences on BSS are diverse. Eren & Uz (2020) identifies determinants such as tourist areas and shopping districts, which attract high demand due to sights and POIs. Integration with public transport also plays a role, along with environmental factors like population density or offices, and temporal factors like public holidays or events impacting rental demand.

To ensure comparability, exogenous variables for all systems must come from the same publicly available data sources, which also provide historical data for integration as past-covariates. Historical and current weather data are sourced via the Open-Meteo API (CC BY 4.0) (Zippenfenig, 2023), offering metrics like temperature, humidity, wind speed, and precipitation. Data on POIs, land use, and public transport networks is retrieved from Open Street Maps (ODbL). Historical demographic data, including population numbers and density, is provided by WorldPop (CC BY 4.0). These variables across multiple categories serve as input features for applied forecasting models (see table 3).

Feature selection

A variety of exogenous variables is generated as potential input features, requiring evaluation to identify a minimal, optimal subset without redundancy or irrelevance. A two-stage process is applied. First, the SULOV method (Moulaei et al., 2024) eliminates highly correlated variables (correlation threshold: 0.7) and calculates Mutual Information Scores (MIS) for the target variable (number of rents at time t). Variables with low MIS are excluded, leaving only those with high explanatory value and low correlation. Second, the eXtreme Gradient Boosting (XGBoost) model refines the subset by recursively evaluating variables, selecting top features using the mRMR framework (Minimum Redundancy and Maximum Relevance). Feature selection was implemented via the Python package featurewiz (Seshadri, 2020).

Machine Learning Models for Time Series Forecasting

Bai et al. (2018) introduced the Temporal Convolutional Network (TCN) for sequence modeling. The TCN employs deep networks with augmented residual layers and dilated causal convolutions to prevent information leakage from future to past. Its architecture includes an input layer, one or more hidden layers for convolutions, and an output layer, enabling auto-regressive prediction

Table 3: Overview of data sources and determinants, grouped by category

Data source	Categories/Features	Example determinants
OpenStreetMaps	Gastronomy	Restaurants, bars or cafés
	Education	University, schools, kindergarten
	Tourism	POIs such as shops, museums, zoos and other attractions
	Healthcare	Hospital, clinic, pharmacies
	Culture & Entertainment	Cinemas, theatres, community and art centres
	Infrastructure	Highways, cycle paths, living streets
	Landuse	Area of farmland, forest and water surfaces
	Transport	stations of public transport, parking, car-sharing
Open-Meteo	Weather	temperature, humidity, Precipitation of rain and snow, cloudcover, windspeed
Worldpop Hub	Demographic	Age and sex structures, population counts and density

tasks with a large receptive field.

The Long Short-Term Memory (LSTM) model by Hochreiter & Schmidhuber (1997) builds on Recurrent Neural Networks (RNNs), which have internal memory states updated recursively. However, RNNs struggle with vanishing or exploding gradients, limiting their ability to learn long-term dependencies (Lim & Zohren, 2021). LSTMs address this by introducing a new cell state modulated by input, output, and forget gates to store long-term information. TCN and LSTM models are used as baselines.

The Transformer model, based on the attention mechanism by Vaswani et al. (2017), achieved state-of-the-art performance in natural language processing tasks. Its encoder-decoder structure enables inter-dependencies between input and output (encoder-decoder attention) and intra-dependencies within inputs and outputs (self-attention). For time series, it aggregates temporal features using dynamically generated weights (Lim & Zohren, 2021).

Zeng et al. (2023) introduced LTSF-Linear, a temporal linear layer that predicts future values via weighted sums over the temporal axis. Enhancements include DLinear, which decomposes time series into trend and seasonal components, applying LTSF-Linear separately and combining the outputs, and NLinear, which adjusts for distribution shifts by normalizing input sequences. Both significantly improve forecasts for time series with trends, seasonality, or distribution shifts.

Motivated by the success of Zeng et al. (2023) LTSF-Linear models, Das et al. (2023) introduced the Time-series Dense Encoder (TiDe) model. TiDe uses dense Multi-layer Perceptrons (MLP) instead of attention mechanisms, enabling better computational scaling with horizon length and context. On data sets like Electricity or ETTh1, TiDe outperforms DLinear and transformer-based models for long-term forecasting.

Model Optimization and Training

A pruning strategy was applied to optimize hyperparameters using the Optuna framework (Akiba et al., 2019). Based on the Successive Halving Algorithm Li et al. (2020), Optuna terminates unpromising trails early. A trail, dynamically generated from the search space, contains hyperparameters to evaluate. Trails not terminated advance to the next round, while historical results inform new promising hyperparameters. The process concludes after reaching the predefined limit of 100 trails.

The pruning strategy was tailored to each model, considering specific hyperparameters like kernel size (TCN) or encoder/decoder layers (Transformer), alongside learning rate, epochs, and dropout rate. Hyperparameters were optimized for each temporal aggregation, and models trained accordingly. The dataset was split 70/30 into training and validation sets. Metrics like training time and model size were recorded, ensuring comparability by running all pruning and training on the same workstation (i9-12900k, RTX A5000).

Model Evaluation & Metrics

To compare the forecasts results of the models, two absolute and two relative metrics are used. Mean Absolute Error (MAE) calculates the average magnitude of the absolute errors for the true series y and predicted series $\hat{y}$ of length T:

$$MAE = \frac{1}{T}\sum_{t=1}^{T}|y_t - \hat{y}_t| \tag{1}$$

Root Mean Square Error (RMSE) is a common metric and always non-negative. An RMSE of 0 indicates a perfect fit, but it is sensitive to outliers. RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2} \tag{2}$$

The Mean Absolute Scaled Error (MASE), introduced by Hyndman & Koehler (2006), is less sensitive to outliers. It compares the predicted series to an in-sample one-step naive forecast and is defined as:

$$MASE = \frac{\frac{1}{h}\sum_{t=n+1}^{n+h}|y_t - \hat{y}_t|}{\frac{1}{n-1}\sum_{t=2}^{n}|y_t - \hat{y}_{t-1}|} \tag{3}$$

Here, n is the training sample, and h is the forecasting period. Mean Absolute Percentage Error (MAPE) calculates average absolute percentage errors but fails when y includes zero or near-zero values, as seen in bike-sharing demand at night. To address this, symmetric Mean Absolute Percentage Error (sMAPE) is used and defined as:

$$sMAPE = 200 \cdot \frac{1}{T}\sum_{t=1}^{T}\frac{|y_t - \hat{y}_t|}{|y_t| + |\hat{y}_t|} \tag{4}$$

where y is the true series and $\hat{y}$ the predicted series.

3 Results and discussion

As noted, BSSs differ significantly due to city-specific factors like population, geography, and climate, affecting feature selection. Chicago's BSS features includes diverse demographics (e.g., age groups 15-75 by gender), while Madrid and Oslo uses limited demographic data. Demographics vary by temporal aggregation, rarely selected weekly. In contrast, land use features (e.g., retail, industrial, green spaces) are prominent in Madrid and Toronto, while Oslo and Toronto focus on infrastructure (e.g., education, administration, shops, transport). Weather features, such as temperature, humidity, and wind speed, are relevant across all systems and aggregations, with cloud cover and precipitation more important at lower temporal aggregations. System-level feature selection is essential, except for universal factors like temperature, which are critical to the accuracy of the prediction.

Table 4 highlights differences in training, tuning, inference times, parameter counts, and model sizes. LSTF models (DLinear, NLinear) have shorter training and tuning times due to fewer parameters, unlike Transformer and TiDe models, which require hours for optimization and training. Training times and model sizes peak at hourly aggregation due to longer time series with more time steps. Inference times are minimal, exceeding one second only for hourly or daily aggregation (e.g., Transformer: 1.27–2.49 seconds). NLinear, DLinear, and TCN models have fewer parameters

Table 4: Average training, tuning and inference times, as well as number of parameters and model size

Model	**AGG**	**Training-Time** (min)	**Pruning-Time** (min)	**Inference-Time** (sec)	**Parameter** (Amount)	**Size** (Mb)
NLinear	h	43.43	19.17	0.28	13,081	0.05
	D	1.43	1.91	0.36	1,489	<0.01
	W	0.25	0.82	0.29	3,375	0.01
	M	0.10	0.76	0.23	1,387	0.01
DLinear	h	128.86	25.48	0.85	54,917	0.84
	D	1.2	2.07	0.28	11,294	0.5
	W	0.2	0.76	0,11	5,165	0.02
	M	0.05	0.9	0,17	2,642	0.02
TCN	h	108.51	27.39	0.20	4,495	0.04
	D	1.24	2.91	0.36	1,940	0.01
	W	0.21	1.01	0.10	1,046	<0.01
	M	0.07	1.37	0.07	1,257	<0.01
LSTM	h	32.30	66.05	0.21	177,775	1.33
	D	1.92	2.61	0.32	109,455	0.83
	W	0.26	0.85	0.09	136,803	1.04
	M	0.18	1.15	0.34	133,290	1.01
Transformer	h	450.91	273.30	1.27	6,488,034	24.75
	D	12.02	14.21	2.49	3,478,788	13.27
	W	1.60	3.14	0.57	2,752,897	10.50
	M	0.92	2.18	0.22	2,068,674	7.89
TiDe	h	254.26	173.49	0.84	11,665,289	44.50
	D	6.34	23.20	1.20	4,369,694	16.66
	W	1.21	37.34	0.41	6,732,327	25.68
	M	0.76	42.75	0.27	13,866,923	52.89

and smaller overall model sizes due to their architecture.

Table 5 reports MAE, RMSE, MASE, and SMAPE scores for all models by time aggregation. All models show strong forecast quality. The Transformer model outperforms LTSF models by a factor of 3-4 for hourly forecasting, with RMSE, MASE, and SMAPE reduced by 55.55%, 64.64%, and 49.75%, respectively. The LSTM model is comparable, but has slightly worse MASE (+30.0%) and SMAPE (+7.54%). The TCN model has the poorest performance. For daily aggregation, the Transformer provides the best forecast quality, outperforming LTSF models in RMSE by 17.65% to 70.83%. In weekly aggregation, LSTM leads in all metrics, while LTSF models show better MASE and SMAPE than hourly or daily aggregations. The Transformer and TiDe models perform similarly. However, when evaluated on a monthly basis, no definitive conclusion can be drawn. According to the absolute error metrics MAE and RMSE, the LSTM model performs better, based on the relative metrics (MASE, SMAPE) the Transformer model. The TiDe Model demonstrates a 11.11% (RMSE) deficit in terms of forecast quality when compared to the LSTM model and exhibits a 21.64% (MASE) shortfall relative to the Transformer Model.

To assess forecast quality, RMSE values are calculated for each horizon step (see figure 2). The forecast horizon is defined as the number of future time steps for which the forecast is made. As expected, the Transformer and LSTM models show low RMSE values on an hourly basis. However, after horizon step 60, the Transformer's forecast quality decreases, while LSTM remains stable. NLinear and DLinear show fluctuating performance with substantial outliers. As the forecast horizon increases, the TCN model's quality approaches that of the Transformer, though it remains weaker overall, as shown in Table 5. Figure 2b indicates a shift in forecast quality from horizon step 18-20, with a decline for all models. NLinear performs poorly at steps 1-25 but surpasses

Table 5: Average error metrics based on all systems, categorized by time aggregation. Highlighted and underlined results represent the best value.

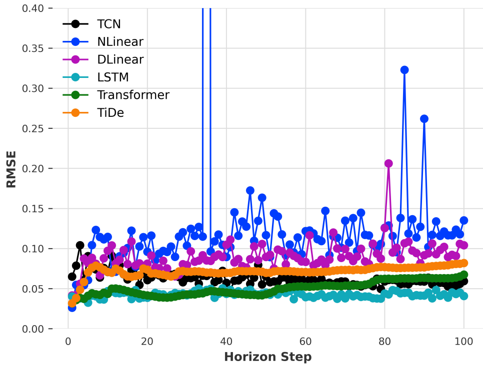
AGG	Model	MAE	RMSE	MASE	SMAPE
h	DLinear	0.08	0.11	2.37	86.83
	NLinear	0.10	0.14	2.94	101.44
	TCN	0.23	2.39	6.63	92.61
	LSTM	0.03	**<u>0.04</u>**	0.91	42.78
	Transformer	**<u>0.02</u>**	**<u>0.04</u>**	**<u>0.70</u>**	**<u>39.79</u>**
	TiDe	0.07	0.09	1.98	78.38
D	DLinear	0.25	0.32	3.68	81.03
	NLinear	0.26	0.48	3.93	66.54
	TCN	0.18	0.26	2.62	63.43
	LSTM	0.12	0.17	1.75	51.72
	Transformer	**<u>0.10</u>**	**<u>0.14</u>**	**<u>1.45</u>**	**<u>40.83</u>**
	TiDe	0.18	0.23	2.54	61.31
W	DLinear	0.14	0.18	1.39	47.53
	NLinear	0.15	0.20	1.48	47.03
	TCN	0.23	0.28	2.30	71.77
	LSTM	**<u>0.12</u>**	**<u>0.16</u>**	**<u>1.21</u>**	**<u>39.18</u>**
	Transformer	0.17	0.21	1.65	52.14
	TiDe	0.20	0.25	1.95	57.37
M	DLinear	0.22	0.29	2.46	62.95
	NLinear	0.24	0.31	1.94	59.34
	TCN	0.17	0.20	1.69	53.36
	LSTM	**<u>0.14</u>**	**<u>0.18</u>**	1.49	46.01
	Transformer	0.15	0.19	**<u>1.34</u>**	**<u>42.60</u>**
	TiDe	0.15	0.20	1.63	45.02

Transformer after step 30. DLinear and TiDe show fluctuating quality, with TiDe underperforming from step 37. In weekly aggregation, TCN and TiDe have the weakest performance (figure 2c), while LSTM outperforms Transformer from step 6. In monthly aggregation, the models show similar performance, except for DLinear, which contains outliers.

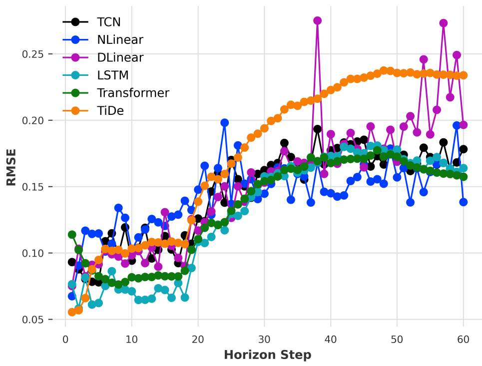
4 Conclusions

The analysis of the feature selection process has revealed the importance of incorporating weather data, including temperature, wind speeds, and humidity, into BSS demand forecasting models. These meteorological factors have been found to significantly contribute to the understanding of bike rental behavior. A key finding of the study is the necessity of considering the unique characteristics of both systems and cities when selecting input features. This approach is crucial to avoid oversimplification and ensure the accuracy of the predictions. The temporal aggregation of data has also emerged as a salient factor in this analysis. It has been observed that the selection of certain features, such as demographic characteristics, is more or less influenced by the aggregation method employed. This study underscores the potential of open historical data sources as a means to derive pertinent features for BSS demand forecasting.

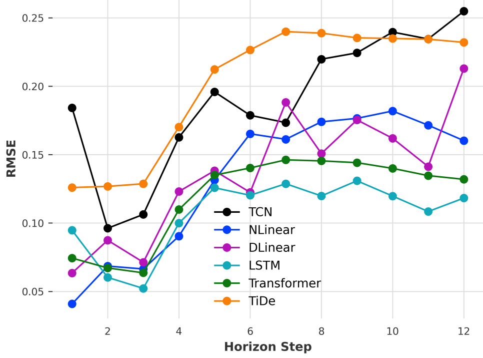
The Transformer and LSTM models outperform alternatives like DLinear, TCN, and TiDe in forecast quality. LTSF models. The performance results of the study by Zeng et al. (2023) could not be verified for the bike-sharing domain, nor could Das et al. (2023)'s TiDe model. As noted by Jiang (2022), training and inference times are often overlooked. This study shows significant differences in performance depending on the machine learning architecture. The Transformer and TiDe models have 2.0 to 13.8 million parameters, resulting in longer training times despite pruning.



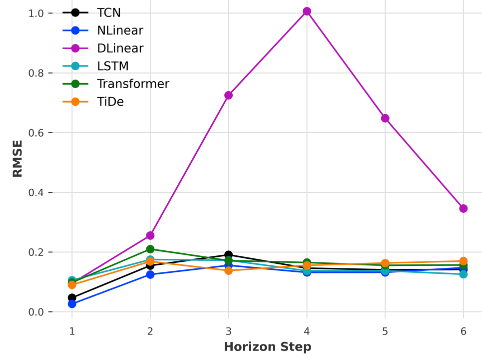
(a) Hourly aggregation



(b) Daily aggregation



(c) Weekly aggregation



(d) Monthly aggregation

Figure 2: Average RMSE value per horizon step

In contrast, LTSF models have fewer parameters, leading to faster training but limited forecast quality. The choice of model depends on the use case; for strategic decisions, forecast time may not be critical, but for real-time systems, it is. Based on these findings, the Transformer model is ideal for hourly or daily forecasts when training times and resources allow. The LSTM model is a good compromise between forecast quality, model size, and training duration.

To further substantiate the results, it would be beneficial to include more BSSs, such as those in London, Washington DC, and Munich, which also provide open data. Additionally, the impact of city size on feature selection should be explored between major cities, medium-sized towns and small municipalities, as OpenStreetMap data quality may vary for smaller cities. Further investigation into alternative ML models, such as Time-Series Mixer (Chen et al., 2023), N-HiTS (Challu et al., 2023), or N-Beats (Oreshkin et al., 2020), is also recommended.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining* (p. 2623–2631). ACM.

Ataç, S., Obrenović, N., & Bierlaire, M. (2021). Vehicle sharing systems: A review and a holistic management framework. *EURO Journal on Transportation and Logistics*, *10*, 100033.

Bai, S., Kolter, J. Z., & Koltun, V. (2018). *An empirical evaluation of generic convolutional and recurrent networks for sequence modeling.* Retrieved from `https://arxiv.org/abs/1803.01271`

Challu, C., Olivares, K. G., Oreshkin, B. N., Garza, F., Mergenthaler-Canseco, M., & Dubrawski, A. (2023). N-hits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 37, pp. 6989–6997).

Chen, S.-A., Li, C.-L., Yoder, N., Arik, S. O., & Pfister, T. (2023). *Tsmixer: An all-mlp architecture for time series forecasting.* Retrieved from `https://arxiv.org/abs/2303.06053`

Collini, E., Nesi, P., & Pantaleo, G. (2021). Deep learning for short-term prediction of available bikes on bike-sharing stations. *IEEE Access*, *9*, 124337–124347. doi: https://doi.org/10.1109/ACCESS.2021.3110794

Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., & Yu, R. (2023). Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*.

Eren, E., & Uz, V. E. (2020). A review on bike-sharing: The factors affecting bike-sharing demand. *Sustainable cities and society*, *54*, 101882. doi: https://doi.org/10.1016/j.scs.2019.101882

Hochreiter, S., & Schmidhuber, J. (1997, 11). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780. Retrieved from `https://doi.org/10.1162/neco.1997.9.8.1735` doi: 10.1162/neco.1997.9.8.1735

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, *22*(4), 679–688. doi: 10.1016/j.ijforecast.2006.03.001

Jiang, W. (2022). Bike sharing usage prediction with deep learning: a survey. *Neural Computing and Applications*, *34*(18), 15369–15385. doi: https://doi.org/10.1007/s00521-022-07380-5

Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Ben-Tzur, J., Hardt, M., ... Talwalkar, A. (2020). A system for massively parallel hyperparameter tuning. *Proceedings of Machine Learning and Systems*, *2*, 230–246.

Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, *379*(2194), 20200209.

Loidl, M., Witzmann-Müller, U., & Zagel, B. (2019). A spatial framework for planning station-based bike sharing systems. *European Transport Research Review*, *11*, 1–12.

Ma, X., Yin, Y., Jin, Y., He, M., & Zhu, M. (2022). Short-term prediction of bike-sharing demand using multi-source data: a spatial-temporal graph attentional lstm approach. *Applied Sciences*, *12*(3), 1161. doi: https://doi.org/10.3390/app12031161

Moulaei, K., Afrash, M. R., Parvin, M., Shadnia, S., Rahimi, M., Mostafazadeh, B., ... others (2024). Explainable artificial intelligence (xai) for predicting the need for intubation in methanol-poisoned patients: a study comparing deep and machine learning models. *Scientific Reports*, *14*(1), 15751.

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International conference on learning representations.* Retrieved from `https://openreview.net/forum?id=r1ecqn4YwB`

Seshadri, R. (2020). *Github - autoviml/featurewiz: Use advanced feature engineering strategies and select the best features from your data set fast with a single line of code.* Retrieved from `https://github.com/AutoViML/featurewiz`

Sun, D., Feng, Z., Chen, Y., Wang, Y., Zeng, J., Yuan, M., ... Qu, H. (2020). Dfseer: A visual analytics approach to facilitate model selection for demand forecasting. In *Proceedings of the 2020 chi conference on human factors in computing systems* (pp. 1–13). doi: 10.1145/3313831.3376866

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (p. 6000–6010). Red Hook, NY, USA: Curran Associates Inc.

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are transformers effective for time series forecasting? In *Proceedings of the aaai conference on artificial intelligence* (Vol. 37, pp. 11121–11128). doi: https://doi.org/10.1609/aaai.v37i9.26317

Zippenfenig, P. (2023). *Open-meteo.com weather api.* Retrieved from https://open-meteo.com/ doi: 10.5281/zenodo.7970649