Time Series Analysis and Prediction for Hourly Bike Rentals Using LSTM Networks

December 23, 2024

Abstract

This work explores the prediction of hourly bike rental demand using data from the Capital Bikeshare system in Washington DC. Long Short-Term Memory (LSTM) networks were used to model temporal dependencies. By incorporating feature engineering, the model achieved an R^2 score of 0.807, effectively capturing the impact of temporal, environmental, and operational factors.

1 Introduction

Bike-sharing systems are becoming more popular as sustainable solutions for urban mobility, helping to reduce traffic congestion and promote environmental conservation. To ensure efficient operations, accurately predicting hourly bike rental demand is essential. Anticipating demand helps in effectively redistributing bikes across stations, improving user satisfaction and service reliability.

This study focuses on predicting hourly bike rentals using Long Short-Term Memory (LSTM) networks. LSTMs are ideal for sequential data because they can preserve temporal dependencies over long periods. Using data from the Capital Bikeshare system in Washington DC, we built a predictive model that integrates temporal, weather-related, and operational factors.

2 Dataset Description and Preparation

The dataset includes hourly and daily bike rental records from 2011 to 2012, provided by the Capital Bikeshare system. It contains features that influence bike rental behavior:

- **Temporal Features:** These include hour, day, month, and weekday, which capture temporal trends.
- **Weather Variables:** These include temperature, humidity, and wind speed.
- **Operational Indicators:** These represent seasonal classifications, holidays, and working days.

To prepare the dataset for analysis, we applied preprocessing and created features. Rolling averages over windows of 3, 7, and 24 hours were computed to smooth fluctuations and understand longer trends.

Lagged variables, such as a 1-hour lag for immediate past data and a 24-hour lag for daily patterns, were added to capture temporal dependencies. Interaction terms were created to model the combined effects of important factors. For example, a temperature-season interaction term highlighted how temperature sensitivity varied across seasons. Finally, all numerical variables were normalized to a [0, 1] range to ensure efficient model training.

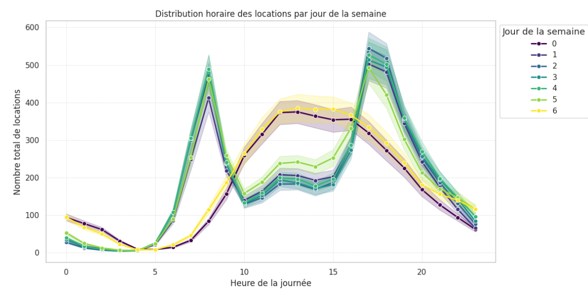

Figure 1: Temporal trends in bike rentals across different seasons and times of day.

We observe clear trends that differ significantly between weekdays and weekends. Weekdays exhibit more predictable patterns with morning and evening peaks, while weekends show a more evenly distributed demand throughout the day. Seasonal effects are less pronounced, though colder seasons, as expected, show a decrease in overall bike rental demand.

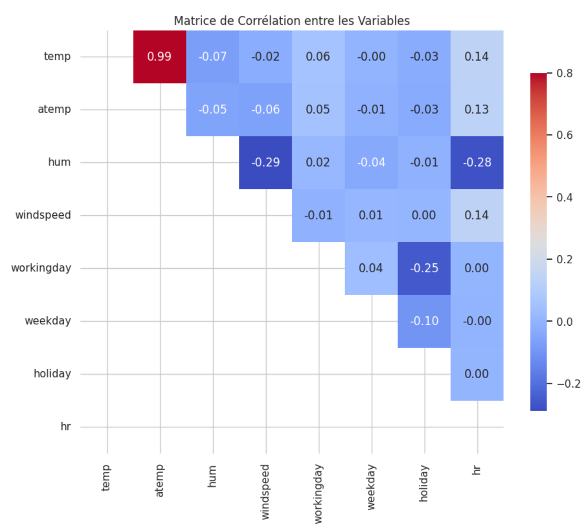

Figure 2: Correlation matrix of the features.

The correlation matrix in Figure 3 reveals that the features in the dataset are not strongly correlated, underscoring the complexity of the data. This lack of strong correlations makes it even more critical to capture temporal dependencies accurately. By decomposing time into meaningful components, such as hour of the day or type of day (weekday vs. weekend), the model can better understand the underlying patterns and improve its predictive capabilities.

3 Methodology

We implemented a Long-Short-Term-Memory (LSTM) network to predict hourly bike rentals. LSTMs are a type of recurrent neural network (RNN) designed to solve the vanishing gradient problem. They do this by using memory cells regulated by gates, which allow the network to decide what information to keep or forget.

3.1 Why LSTM?

LSTM networks are well-suited for time series prediction because they can capture long-term dependencies in sequential data. Unlike most RNNs, LSTMs use gated cells to selectively remember or forget information, solving the vanishing gradient problem. This makes them ideal for complex datasets like bike rentals, where factors such as time of day, weather, and weekdays affect demand. LSTMs also perform well with noisy or sparse data, focusing on important patterns for more accurate predictions.

3.2 LSTM Architecture

The architecture of the LSTM model consisted of the following (Géron, 2019) :

- **Input Layer:** This layer takes sequences of 10 hours, each containing 20 features, including temporal, weather, and interaction variables.
- **Two LSTM Layers:** Each layer has 150 units, which capture both short-term and long-term dependencies.
- **Dropout Layers:** These are applied after each LSTM layer, with dropout rates of 0.3 and 0.1, to prevent overfitting.
- **Dense Output Layer:** This layer produces a single continuous value predicting bike rentals for the next hour.

The equations for the LSTM cell are:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{(Forget Gate)} \tag{1}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{(Input Gate)} \tag{2}$$
$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad \text{(Candidate Update)} \tag{3}$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \quad \text{(Cell State Update)} \tag{4}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{(Output Gate)} \tag{5}$$
$$h_t = o_t \cdot \tanh(c_t) \quad \text{(Hidden State Output)} \tag{6}$$

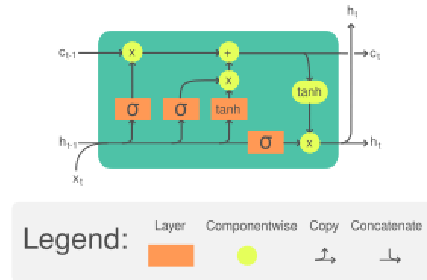

Figure 3: LSTM architecture (Wikipedia, 2024).

3.3 Training Process

The LSTM network was trained using the Adam optimizer, which adjusts learning rates dynamically to improve convergence (Dereich et al., 2024). The Mean Squared Logarithmic Error (MSLE) loss function was chosen because it penalizes large relative errors, ensuring accurate predictions even for smaller rental counts (Permetrics, 2024). Training was performed with a batch size of 32 for up to 50 epochs, but early stopping was used to stop training when validation loss stopped improving for 10 consecutive epochs. This method reduced overfitting while maintaining high accuracy.

To ensure realistic evaluation and to avoid information leakage, the dataset was split into training and testing sets in a time-aware manner. Specifically, 80% of the earliest data points were used for training, while the most recent 20% were reserved for testing. This chronological splitting reflects real-world forecasting scenarios, where models are trained on historical data and evaluated on unseen future data, eliminating the need for a separate validation set due to the use of time-aware cross-validation techniques.

3.4 Fitting Process

The fitting process involved a series of refinements to address the challenges posed by the dataset. Initially, the time series data, along with the created variables, were fitted to the model. These variables, such as lagged values and interaction terms, improved the fit by providing additional contextual information. However, it was observed that extreme values, particularly during peak hours, were underestimated. illustrates the initial training and validation loss curves during the first iterations.

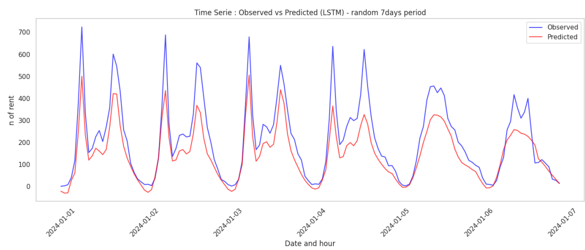

Figure 4: Observed vs. predicted values before refining the fitting.

To address this, the variable `is_peak_hour` was introduced to explicitly account for the high demand during critical periods (e.g., commute hours). This refinement improved the model's performance in capturing peak-hour demand.

Further, a loss compensation strategy was employed to reduce the impact of underestimations during extreme values. By tuning the training parameters, including additional epochs, more layers, and recreating features, the model's capacity to generalize was enhanced.

The architecture was extended with bidirectional LSTM layers to capture dependencies in both forward and backward sequences, and dropout layers were incorporated to prevent overfitting. The optimizer was reverted to Adam with these enhancements, and the combination of loss compensation and a network design ensured a more accurate and reliable model. These iterative improvements allowed the LSTM to better handle the complexities of the data set, effectively capturing both general trends and edge cases.

4 Results

The LSTM model achieved an R^2 score of 0.807 on the test set, showing that it can predict bike rental patterns accurately. Figure 5 shows the comparison between observed and predicted rentals, where the model successfully captures overall trends.

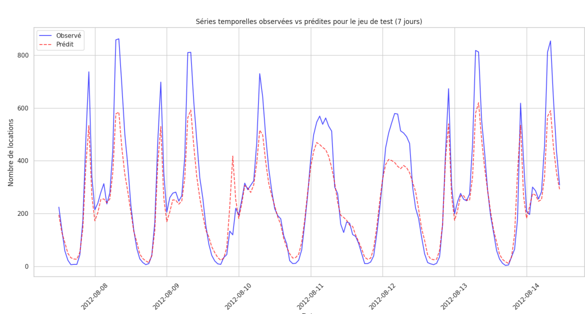

Figure 5: Comparison of observed vs predicted bike rentals on the test set.

Predictions during weekends are less accurate compared to weekdays. This is likely because weekend data shows higher variability and less consistent patterns, while weekdays have more regular trends (Makridakis et al., 1998). To evaluate the model's performance on new data, Time Series Cross-Validation was employed. This method splits the data chronologically, using earlier observations for training and later ones for testing. On average, the cross-validation resulted in an R^2 score of 0.62 across five folds. While lower than the test set score, this is expected due to smaller training sets and the presence of new trends or anomalies in the validation sets. This stricter evaluation highlights the model's ability to generalize to unseen data.

Figure 6 illustrates the observed vs predicted values during cross-validation, with most points aligning closely with the line $y = x$, indicating accurate predictions overall. Figure 7 displays the error distribution, which is centered around zero, confirming the model's unbiased performance. However, the LSTM's tendency to produce smooth predictions may lead to underestimation of sudden peaks or drops in bike rental demand.

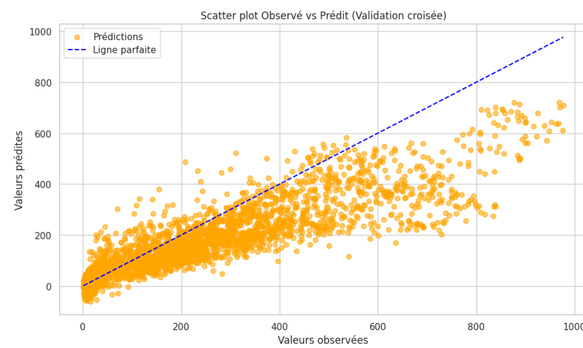

Figure 6: Scatter plot of observed vs predicted bike rentals during cross-validation.

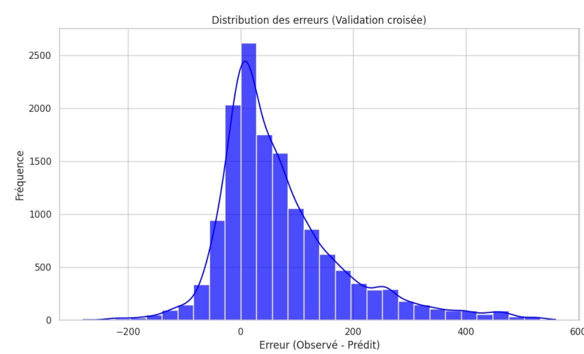

Figure 7: Error distribution during cross-validation.

5 Discussion

This study highlights the effectiveness of LSTM networks in predicting the demand for hourly bicycle rentals. The model successfully captured both short- and long-term dependencies, as well as external factors such as weather and operational conditions. While the training process required significant computational resources, the model's accuracy and flexibility justify its use.

Future improvements could include incorporating real-time data streams for dynamic predictions and extending the model to forecast over longer time horizons. Incorporating spatial dimensions into the model could further enhance its utility, by forecasting a demand by station.

6 Conclusion

The LSTM model developed in this work is a good tool for predicting bike rental demand. Using temporal modeling and feature engineering, we achieved good predictive accuracy. These findings demonstrate the value of machine learning models in optimizing bike-sharing operations and show how machine learning can be used for future innovations in urban mobility.

References

- Makridakis, S., Wheelwright, S. C., Hyndman, R. J. (1998). Forecasting Methods and Applications. Wiley.
- Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media. Chapter 15: RNNs and LSTMs.
- Dereich, S., Jentzen, A. (2024). Convergence rates for the Adam optimizer. Institute for Mathematical Stochastics, University of Münster; School of Data Science and Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen).
- Permetrics. (n.d.). Mean Squared Logarithmic Error (MSLE).

Additional Documentation

The complete code and associated resources for this project are available on GitHub at the following repository:
GitHub Repository

The dataset used in this project is available from the Capital Bikeshare system and can be accessed at:
https://capitalbikeshare.com/system-data