TU Delft

Delft University of Technology

Document Version
Final published version

Licence
CC BY

Citation (APA)
Hari, T., Daamen, W., Hoogendoorn-Lanser, S., Steenbakkers, J., & Hoogendoorn, S. P. (2026). Evaluation of Machine Learning and Deep Learning Models for Multi-Horizon Crowd Forecasting at Scheveningen Beach, The Netherlands. *Transportation Research Record*. https://doi.org/10.1177/03611981261422125

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

Check for updates

TRR
JOURNAL OF THE TRANSPORTATION RESEARCH BOARD

Research Article

Transportation Research Record
1–25
© The Author(s) 2026

Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/03611981261422125
journals.sagepub.com/home/trr
Sage

Evaluation of Machine Learning and Deep Learning Models for Multi-Horizon Crowd Forecasting at Scheveningen Beach, The Netherlands

Theivaprakasham Hari[1], Winnie Daamen[1], Sascha Hoogendoorn-Lanser[2], Jeroen Steenbakkers[3], and Serge Paul Hoogendoorn[1]

Abstract

Efficient crowd management is crucial for municipalities to ensure public safety and enhance visitor experience, particularly in tourist-centric areas, such as Scheveningen Beach. Scheveningen Beach faces challenges because of the limited precision of visitor count data and the lack of accurate forecasts. Currently, crowd safety managers use their professional experience to forecast based on factors such as weather, events, and holidays, leading to inaccuracies, highlighting the need for accessible data and advanced analytics to enhance crowd management strategies. This study evaluates machine learning and deep learning models for multi-horizon hourly pedestrian crowd count forecasting, addressing the limitations of current manual prediction methods. Historical crowd data, weather, and holidays were integrated to train eXtreme gradient boosting, categorical boosting (CatBoost), light gradient boosting machine (LightGBM), long short-term memory (LSTM), and Temporal Fusion Transformer models for short-term (1-day), mid-term (7-day), and long-term (30-day) horizons. Models were developed for individual locations and as a unified multilocation approach. Performance was assessed using the coefficient of determination, root mean square error, normalized root mean square error, symmetric mean absolute percentage error, mean absolute error, and normalized mean absolute error metrics. The results showed that CatBoost was best for short-term forecasts, CatBoost and LightGBM for mid-term forecasts, and LSTM and LightGBM for long-term forecasts. Forecast performance decreases over longer time horizons in many locations, suggesting different applications: short-term forecasts for immediate operational decisions and long-term predictions for general trend analysis and strategic planning. Individual location models generally outperformed the unified approach, but at a higher computational cost. This study reveals significant spatial and temporal variability in crowd dynamics, which is crucial for optimizing resource allocation and enhancing preparedness in crowd management at Scheveningen Beach and similar tourist destinations.

Keywords

multi-horizon crowd count forecasting, crowd management, machine learning, deep learning, time series forecasting

Introduction

The prevalence of crowd-related accidents has increased significantly over the past 120 years, highlighting a growing concern in urban and event management (*1*). This increase can be attributed to various factors, including increased population, rapid urbanization, and the inherent limitations of existing infrastructural capacities (*2*). Crowds are a common feature at events across various contexts, including religious, sports, entertainment, and political events, posing unique challenges for safety and risk management (*3*). Crowds are inherently prone to various hazards, such as stampedes, collisions, panic

[1]Department of Transport & Planning, Delft University of Technology (TU Delft), Delft, The Netherlands
[2]Mobility Innovation Centre Delft, Impact & Innovation Center, TU Delft, Delft, The Netherlands
[3]Argaleo B.V., 's-Hertogenbosch, The Netherlands

Corresponding Author:
Theivaprakasham Hari, t.hari@tudelft.nl

reactions, trampling, suffocation, and dehydration. Moreover, the complexity and variability in crowd dynamics can occur in any location or circumstances, making accurate estimation and forecasting of crowd-related risk particularly challenging (*4*).

Managing thousands of visitors at Scheveningen Beach, The Hague, Netherlands, is complex. This complexity arises from the need to balance visitor enjoyment with safety measures and the availability of emergency services. Accurate forecasting of pedestrian numbers over short-term (1-day), mid-term (7-day), and long-term (30-day) horizons is critical for optimizing daily, weekly, and monthly resource allocation and enhancing crowd management preparedness. For example, short-term hourly forecasting can assist crowd safety managers in the immediate real-time distribution and allocation of personnel and resources, such as security staff, medical teams, and crowd control barriers across different beach zones, which are crucial for identifying and managing immediate potential risks and preventing incidents. Similarly, mid-term and long-term hourly forecasting can help effectively plan resources and mitigate traffic congestion, parking issues, and overcrowding.

A significant challenge in crowd management at Scheveningen Beach (and other crowded areas) is the limited availability of precise data on visitor numbers. Crowd safety managers primarily rely on their professional experience to create manual crowd predictions based on weather information, scheduled events, and holiday calendars. This reliance can lead to inaccuracies in anticipating visitor volume and movement. To improve crowd management and planning strategies, there is a pressing need for accessible, precise data and advanced data analytics. Adopting a data-driven approach will enable crowd managers to plan, allocate, and manage crowds more effectively.

Numerous studies have focused on applying statistical modeling, machine learning, and deep learning techniques for traffic forecasting in the transportation sector. There is a notable gap in multi-horizon pedestrian crowd forecasting in outdoor settings across various locations. Most existing research has been tailored toward predicting vehicular traffic patterns, which, although similar, pose different challenges and require distinct approaches compared with pedestrian crowd forecasting. Pedestrian dynamics in environments such as beaches, parks, and urban plazas are inherently more complex and influenced by variables, including weather conditions, special events, holidays, and social behavior patterns (*5*). This complexity is especially noticeable in these environments, where the factors influencing crowds are more variable and less predictable than in structured road networks. There is an evident need for forecasting models that provide accurate predictions and are generalizable across diverse environments and time horizons.

In this study, various machine and deep learning models are evaluated for forecasting hourly pedestrian crowd counts at different locations on Scheveningen Beach for the short-term (1-day-ahead), mid-term (7-day-ahead), and long-term (1-month-ahead) horizons. Historical crowd data, weather information, and holiday periods are integrated to train machine learning (eXtreme gradient boosting [XGBoost], categorical boosting [CatBoost], and light gradient boosting machine [LightGBM]) and deep learning models (long short-term memory [LSTM] and Temporal Fusion Transformer [TFT]) for individual locations and a unified multilocation model to assess the generalizability of the model. The models' performance was evaluated using the coefficient of determination (R^2), symmetric mean absolute percentage error (SMAPE), root mean square error (RMSE), mean absolute error (MAE) metrics, normalized mean absolute error (NMAE) and normalized root mean square error (NRMSE) by evaluating varying input and prediction window sizes.

The main contributions of this study are as follows.

1. Comprehensive multi-horizon model evaluation: Machine and deep learning models are evaluated for short, mid, and long-term pedestrian crowd count forecasting and their generalizability across different locations and forecasting horizons is demonstrated.
2. Integrated data sources: Historical crowd data, weather information, and holiday periods are integrated to enhance the forecasting models' performance.
3. Performance insights and resource optimization: Best-performing models for different forecasting horizons (CatBoost was best for short-term forecasts, CatBoost and LightGBM for mid-term forecasts, and LSTM and LightGBM for long-term forecasts) are identified, and insights are provided for crowd management and planning.

This study progresses through key sections: Related Works, Research Approach, Results, Discussion and Conclusion. A comprehensive literature review is performed, and then details of the methodological framework, including data set analysis, feature selection, engineering, and experimental design, follow. The Results section presents and analyzes experimental outcomes, and the final section interprets the findings, discusses implications, summarizes this study, and proposes future research directions, aligning with the initial objectives and discoveries.

Related Works

Previous studies on pedestrian crowd forecasting have employed a range of methods, from traditional statistical approaches to advanced machine learning and deep learning models. This section reviews the relevant literature on pedestrian crowd forecasting, focusing on methodologies that are promising for handling multi-horizon predictions and multivariate outputs.

Traditional statistical methods have long been used as the baseline for time series forecasting, including pedestrian crowd forecasting. Among these, the Naive Seasonal Model (weekly and yearly) stands out as a simple yet effective baseline for comparison (*6*). This model assumes that the forecast for a given period will equal the last observed value from the same season (e.g., the same day of the week or the same day of the year). It often serves as a strong benchmark, especially in data sets with strong weekly or annual seasonality. Traditional statistical methods, such as linear regression, autoregressive integrated moving average (ARIMA), and seasonal autoregressive integrated moving average (SARIMA), have been widely used for short-time series forecasting in various domains, including pedestrian crowd count (*7*, *8*). However, these models assume linear relationships between past and future values, which do not always hold in pedestrian crowd dynamics on beaches, where dynamics can be highly nonlinear because of sudden events, weather changes, or social interactions. In addition, ARIMA and SARIMA rely on the target variable's historical values and struggle to incorporate external variables. These methods provide a solid baseline, but they often struggle to capture nonlinear relationships in crowding data, limiting their effectiveness for multi-horizon forecasting.

Machine learning approaches have demonstrated superior performance in capturing nonlinear patterns and handling multiple input and output variables. Random forest has been successfully applied to pedestrian count prediction in Melbourne, Australia (*9*), demonstrating the potential of ensemble methods in this domain. Building on this success, gradient-boosting algorithms have emerged as powerful tools for time series forecasting. XGBoost has shown significant performance in predicting pedestrian crowd counts (*10*) at Scheveningen Beach when trained with weather attributes. Similarly, LightGBM was used to forecast short-term traffic flow at several road sections in Changsha City (*11*). This success suggests that other advanced gradient boosting algorithms, such as CatBoost, may also be effective.

Deep learning models have also shown significant potential for capturing complex temporal dependencies in time series data. LSTM has been successfully applied to long-term pedestrian crowd forecasting and traffic flow prediction (*12–14*) because of its ability to capture long-term dependencies in sequential data. Other models, such as spatiotemporal graph convolutional networks (*15*) and graph-convolution-embedded LSTM (*16*), have shown strong performance in traffic forecasting. However, they often have complex data requirements that may not be readily available or applicable in all pedestrian crowd forecasting scenarios. Recent advances in deep learning have led to the development of more sophisticated architectures. The TFT (*17*) represents a significant step forward in multi-horizon forecasting. The TFT combines the strength of transformers in capturing long-range dependencies with the ability to handle different inputs and outputs for multi-horizon forecasting.

Several studies have highlighted the importance of integrating diverse data sources for improved forecast accuracy. For instance, incorporating environmental conditions, such as wind speed, temperature, and rainfall, alongside traffic flow data has enhanced prediction quality (*9–11*). This study builds on this approach by integrating historical crowd data, weather information, and holiday periods.

The literature reveals a gap in studies that evaluate model performance across multiple forecasting horizons (short, mid, and long-term) and assess generalizability across different locations. Significant progress has been made in pedestrian crowd forecasting, but further research is needed on multi-horizon predictions, multivariate outputs, and the integration of diverse data sources to improve model accuracy and applicability across various environments.

Research Approach

This study employs a comprehensive research approach to forecast hourly pedestrian crowd counts at Scheveningen Beach. First, data collection, feature selection, and engineering are performed, utilizing historical pedestrian data, weather information, and holiday calendars. Then, multiple forecasting models are implemented and compared. Model training methods are elaborated on, focusing on multi-horizon, multivariate output forecasting using a sliding window approach. Finally, model performance is evaluated using multiple metrics. This structured approach allows different forecasting techniques to be analyzed and compared systematically.

Data Set

This study uses an extended version of the historical pedestrian crowd count data set for Scheveningen Beach, the weather data set (temperature, precipitation rate, precipitation chance, wind speed, wind direction, global

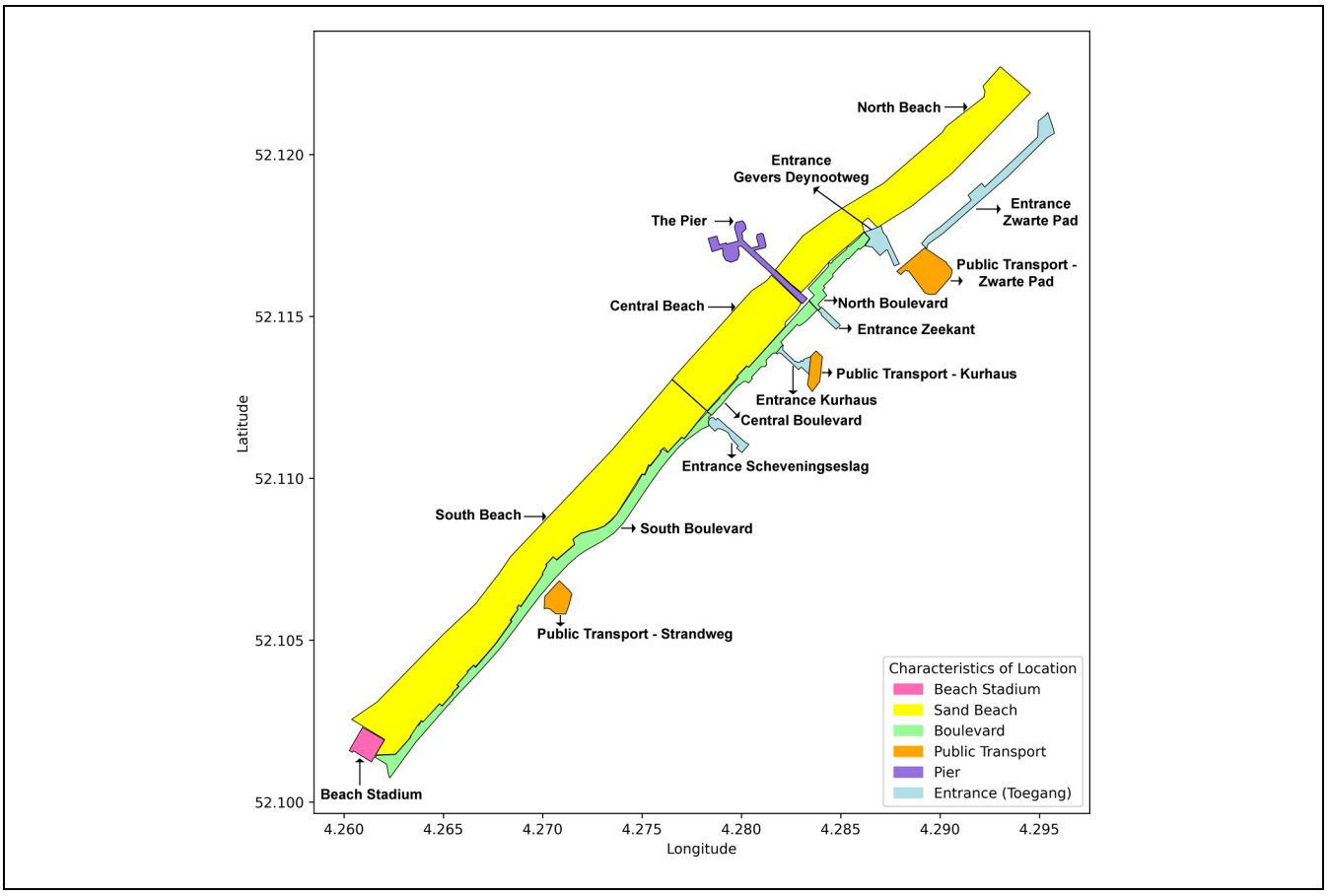

Figure 1. Map of Scheveningen Beach with locations based on different characteristics.

radiation, and cloud cover) used by the study (*10*), and the Netherlands Public Holiday calendar for forecasting pedestrian counts. This study utilizes a data set with an hourly resolution from April 1, 2021, to November 17, 2023. The pedestrian crowd count data set, generated by RESONO (https://reso.no/), includes counts collected, preprocessed, and aggregated based on the number of smartphones detected at Scheveningen Beach using anonymous information from various mobile applications. Of note, these counts serve as an approximate proxy and do not represent real-time pedestrian crowd counts because they only account for smartphone detections. The weather data set was collected from the Royal Netherlands Meteorological Institute. The data set includes data for 17 locations at Scheveningen Beach, The Hague, Netherlands, namely the Beach Stadium, Central Boulevard, North Boulevard, South Boulevard, The Pier, Public Transport (Kurhaus), Public Transport (Strandweg), Public Transport (Zwarte Pad), Entire Scheveningen, Central Beach, North Beach, South Beach, Entrance Gevers Deynootweg, Entrance Kurhaus, Entrance Scheveningseslag, Entrance Zeekant, and Entrance Zwarte Pad, as shown in Figure 1. The location, Entire Scheveningen, spans the area of Scheveningen Beach, and the choice of the other 16 locations was strategic, ensuring diverse pedestrian social dynamics. The sites spanned various recreational areas and restaurants (The Pier), boulevards, several beach locales, entrances, public transportation, and the Beach Stadium.

Feature Selection and Engineering

The selection of features for training influences forecasting performance. A systematic approach is followed to identify the most relevant features and create new ones to enhance the model's performance. The feature selection and engineering process involves identifying potential influencing factors from the literature, assessing data availability, performing Spearman's rank correlation analysis, selecting promising features, and engineering additional features.

Based on previous studies, several factors that potentially influence pedestrian crowd counts at Scheveningen Beach were identified. These include temporal factors such as time of day, day of the week, month, season, and

holidays; weather conditions such as temperature, precipitation rate, precipitation chance, wind speed, wind direction, global radiation, and cloud cover; and location-specific characteristics. Additional temporal features were created from the data set's time stamps, including day, month, year, hour, week, weekday, day of the year, and whether it is a weekend. To provide a clearer understanding of the data set used, Table 1 summarizes the independent (temporal and weather) and dependent variables used in this study, reporting summary statistics and ranges for continuous weather, crowd count, and temporal variables.

Spearman's rank correlation analysis was performed to quantify the relationship between potential features and pedestrian crowd counts. This analysis helps to identify the strength of linear and nonlinear monotonic relationships between features and the target variable. Figure 2 shows a heatmap of Spearman's correlation for specific features that affect various areas of Scheveningen Beach.

Key findings from the correlation analysis revealed the highest correlations (close to 0.6) between the hour of the day and crowd counts at locations such as Public Transport (Kurhaus), The Pier, Entire Scheveningen, and Central Beach, suggesting that crowd levels are significantly influenced by the time of day. The positive correlations of temperature, global radiation, and whether it is a weekend with crowd counts reflect that warmer, sunnier weather and weekends tend to attract more people to Scheveningen Beach. Precipitation rate, precipitation chance, and cloud cover negatively correlate with crowd counts because rainy, overcast conditions typically discourage beach attendance. Day, month, year, and weekday showed weak correlations because crowd patterns are more directly influenced by immediate factors such as weather and weekends rather than by these broader

Table 1. Data Set Description Table

Features	Description	Minimum	Maximum
Independent variable (temporal features derived from hourly time stamps)			
Day	Day of the month	1	31
Month	Month of the year	1	12
Year	Year of observation	2021	2023
Hour	Hour of the day	0	23
Week	Week of the year	1	52
Weekday	Day of the week	0	6
Day of the year	Day number within the year	1	365
Is it a weekend	Weekend indicator	0	1
Is it a Dutch holiday	Dutch public holiday indicator	0	1

Features	Description with units	Minimum	Mean	Median	Maximum	Standard deviation
Independent variable (weather [hourly resolution])						
Temperature	Hourly average temperature (°C)	−6.2	12.08	12.1	34.9	6.09
Precipitation rate	Hourly total precipitation (mm)	0	0.11	0	23.6	0.58
Precipitation chance	Hourly chance of precipitation (%)	0	19.54	0	100	39.65
Wind direction	Hourly mean wind direction (°)	0	201.14	210	359	125.94
Wind speed	Hourly mean wind speed (m/s)	0	4.32	4	20	2.56
Cloud cover	Hourly mean cloud cover (%)	0	71.49	96	100	36.37
Global radiation	Hourly global solar radiation (W/m^2)	0	52.55	4	332	81.33
Dependent variable (crowd count [hourly aggregation])						
Beach Stadium	Hourly count of people detected at Beach Stadium	0	22.71	0	3,593	115.98
Central Boulevard	Hourly count of people detected at Central Boulevard	0	1,265.03	573	24,795	1,903.90
North Boulevard	Hourly count of people detected at North Boulevard	0	657.12	172	13,009	1,103.92
South Boulevard	Hourly count of people detected at South Boulevard	0	550.83	222	9,882	823.69
The Pier	Hourly count of people detected at The Pier	0	466.99	68	12,747	847.42

(continued)

Table 1. (continued)

Features	Description	Minimum		Maximum		
Public Transport (Kurhaus)	Hourly count of people detected at Public Transport (Kurhaus)	0	885.31	615	8,729	941.98
Public Transport (Strandweg)	Hourly count of people detected at Public Transport (Strandweg)	0	196.26	100	4,324	256.02
Public Transport (Zwarte Pad)	Hourly count of people detected at Public Transport (Zwarte Pad)	0	186.73	63	5,791	317.16
Entire Scheveningen	Hourly count of people detected at Entire Scheveningen	0	1,476.61	645	20,616	2,161.02
Central Beach	Hourly count of people detected at Central Beach	0	734.46	221	12,694	1,232.68
North Beach	Hourly count of people detected at North Beach	0	423.49	81	10,520	754.06
South Beach	Hourly count of people detected at South Beach	0	535.39	169	10,252	863.12
Entrance Gevers Deynootweg	Hourly count of people detected at Entrance Gevers Deynootweg	0	291.70	121	4,130	411.72
Entrance Kurhaus	Hourly count of people detected at Entrance Kurhaus	0	840.84	516	11,711	1,016.75
Entrance Scheveningseslag	Hourly count of people detected at Entrance Scheveningseslag	0	425.40	236	4,483	507.15
Entrance Zeekant	Hourly count of people detected at Entrance Zeekant	0	410.20	206	6,310	546.55
Entrance Zwarte Pad	Hourly count of people detected at Entrance Zwarte Pad	0	105.92	28	4,734	241.45

temporal categories. Features with consistently weak correlations (day, month, year, and weekday) were discarded to reduce dimensionality and mitigate potential overfitting.

Then, to capture comprehensive temporal dependencies, lag features were created for all dynamic variables in the data set. Lag peaks identified with the autocorrelation function often yield the most accurate results when used as the input window size for forecasting (*7*). Autocorrelation analysis on the crowd count data revealed significant daily (24-h) and weekly (168-h) cycles, as shown in Figure 3.

Based on these findings, two distinct sets of lagged input features were constructed to test their effectiveness. The first set included all hourly lag values from the previous 24 h (i.e., lags 1, 2,…, 24 h), and the second set included all hourly lag values from the last 168 h (i.e., lags 1, 2,…, 168 h) of the crowd count, weather conditions, and temporal indicators. This approach allowed the empirical determination of whether a shorter or longer history of the target variable was more predictive across different forecasting horizons. The final feature set used for model training combined one of these two lag configurations with the engineered temporal features and weather-related features. This extensive feature engineering provides the models with a rich, multivariate historical context, enabling them to implicitly learn complex daily, weekly, and seasonal patterns from the collective behavior of all these variables, rather than relying solely on the recent history of the crowd count itself.

All the features (temporal, weather, and crowd count) were transformed using Z-score standardization, with parameters μ and σ computed on the training set to prevent data leakage, as shown in the following equation:

$$x^* = \frac{x - \mu}{\sigma} \tag{1}$$

where μ and σ represent the mean and standard deviation of data x to compute the standardized value x^*. Z-score scaling is preferred over normalization (*18*) because the latter compresses heavy-tailed, unbounded variables, such as crowd counts, into a narrow [0, 1] interval, hindering interpretability and transferability. Z-score standardization preserves linear relationships, yields dimensionless

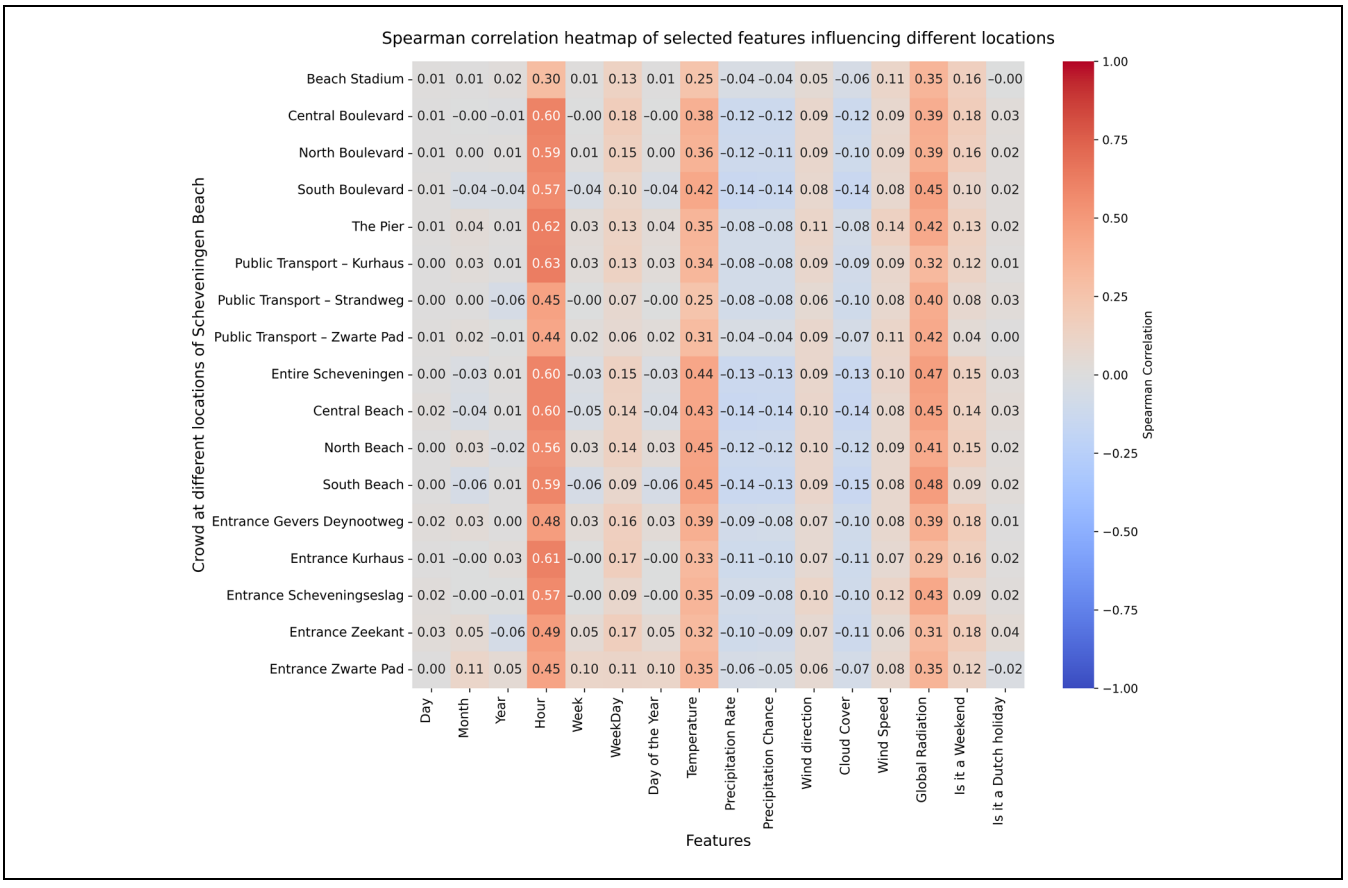

Figure 2. Heatmap illustrating Spearman's correlation between selected features and crowd data from different locations on Scheveningen Beach.

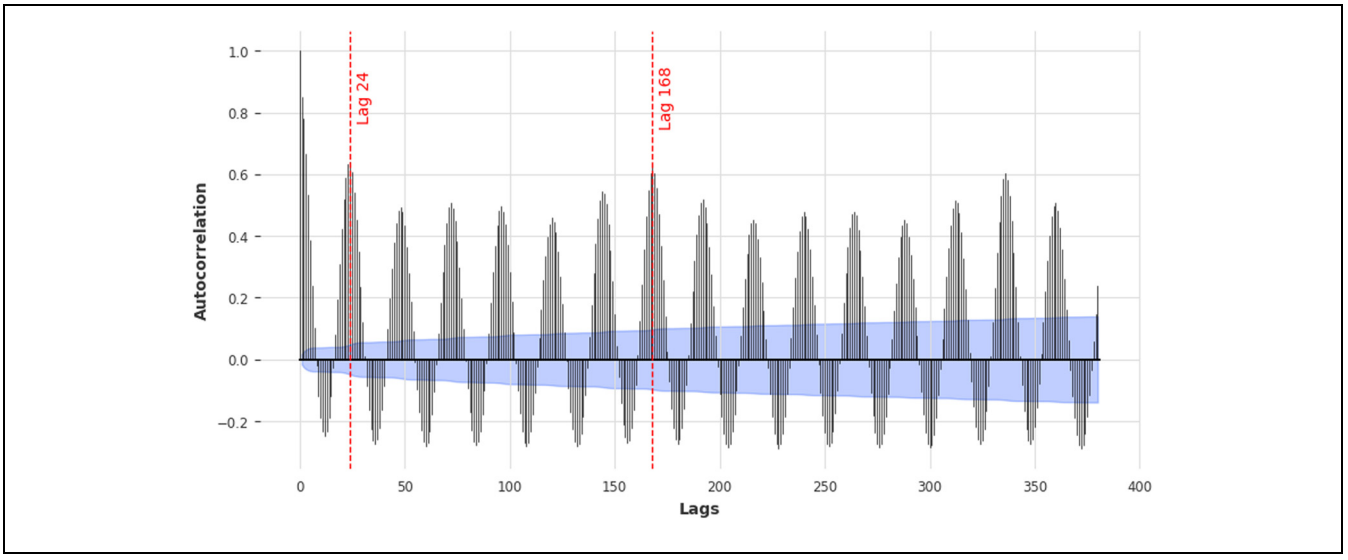

Figure 3. Autocorrelation plot highlights peak daily (lag 24) and weekly (lag 168) cycles in pedestrian crowd count at Entire Scheveningen, with a 95% confidence interval shown in blue.

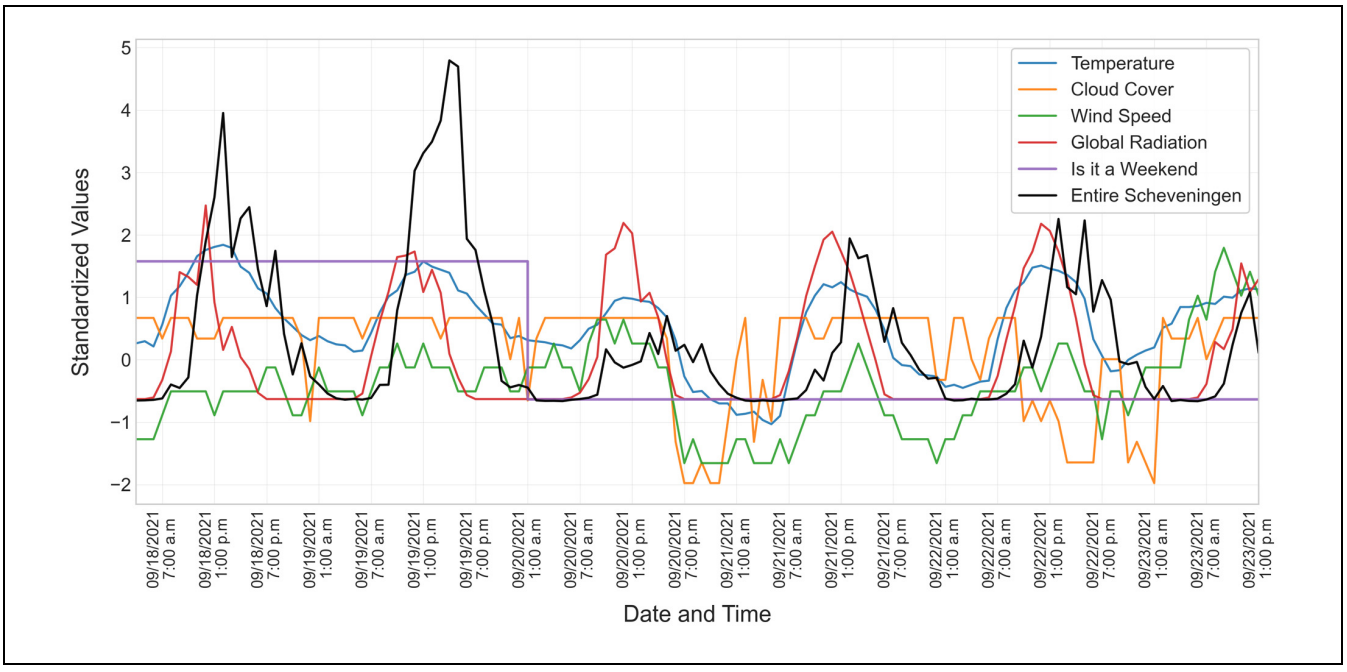

Figure 4. Standardized pedestrian crowd counts and weather data for 5 days for Entire Scheveningen Beach, with values standardized to a consistent scale (mean = 0) for comparison.

values, and remains fixed once μ and σ are set. Figure 4 shows a sample of the standardized weather and pedestrian crowd data for Entire Scheveningen Beach, plotted over 5 days (September 18–23, 2021). In Figure 4, the values are from roughly -2 (less crowded nights) to $+4$ (a busy, sunny afternoon on September 19), and most weather covariates lie within ± 2. In addition, the interval $[-2, 4]$ shown in Figure 4 arises naturally from the selected subset and is not an imposed scaling limit.

The final feature set included the lagged features (24 and 168-h lags) of selected temporal features (hour, week, day of the year, is it a weekend, and is it a Dutch holiday), weather-related features (temperature, precipitation rate, wind direction, cloud cover, wind speed, and global radiation), and crowd count data for different locations.

Model Implementation

This section provides an overview of the methods used to forecast pedestrian crowd counts, specifically tailored to the data set used in this study. Three techniques are employed to train and evaluate crowd forecasting results: (1) statistics-based models; (2) machine learning-based models; and (3) deep learning-based models. Based on the literature review, as discussed in the Related Works section, several models have been identified that warrant further investigation: (1) statistics-based models (Naive Seasonal Model); (2) machine learning models (XGBoost, CatBoost, and LightGBM); and (3) deep learning models (LSTM and TFT).

Statistics-Based Forecasting Model. Historically, statistics-based methods have formed the foundation of time series analysis and remain relevant for their simplicity, interpretability, efficiency, and ability to capture fundamental patterns in data. They offer a point of comparison for more complex methods and help to highlight their added value. In this study, seasonal naive forecasting methods are used as statistical baselines to benchmark machine learning and deep learning models.

Naive Seasonal Forecasting. Naive forecasting methods serve as essential benchmarks to validate the effectiveness of more complex models. A robust model should significantly outperform these simple, rule-based approaches. Based on the intense daily, weekly, and yearly cycles identified in the autocorrelation analysis, three naive models were implemented to serve as benchmarks for different forecast horizons.

- Naive Daily Forecast: This model is used as the benchmark for the short-term (24-h) forecast horizon. It assumes that the crowd count for a specific hour will be the same as the crowd count at the same time on the previous day. For example, the forecast for 02:00 p.m. on Wednesday is the observed value from 02:00 p.m. on Tuesday (a 24-h lag).
- Naive Weekly Forecast: This model serves as the primary benchmark for the mid-term (168-h) forecast horizon. It operates on the assumption that

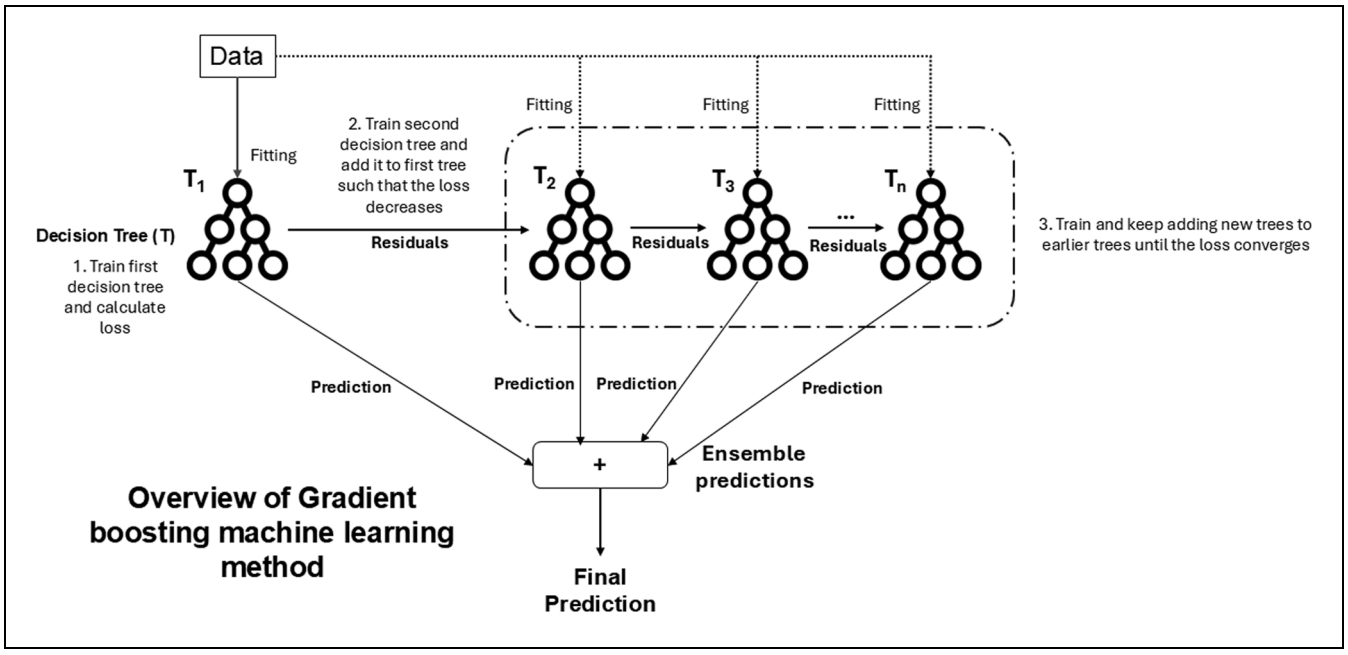

Figure 5. Overview of the gradient boosting process used in the machine learning models.
Note: T_1 = initial decision tree; T_2 = decision tree trained on the residuals (errors) of the first tree; T_3,…, T_n = additional decision trees.

crowd patterns repeat every week. The forecast for a given hour and day of the week is the observed value from the same hour and day of the previous week (a 168-h lag). This benchmark is particularly relevant for capturing recurring patterns on weekends and weekdays.

- Naive Yearly Forecast: To provide a broader context on long-term seasonality, a yearly model was implemented. The forecast for a specific hour and day is the observed value from the same hour and day in the previous year. The machine learning models in this study were not trained for yearly predictions; this benchmark helps illustrate the strength of annual patterns related to holidays and seasons.

It is important to note that a Naive Monthly Forecast (using a 30-day lag) was not considered because this is not a standard time series benchmark, and preliminary analysis showed a very weak correlation between hourly crowd counts with a 30-day lag. The Naive Daily and Weekly models provide the most direct and meaningful performance baselines for the short-term and mid-term forecasting scenarios, respectively.

Machine Learning-Based Forecasting Model. The machine learning-based approach started gaining attention because it can handle larger data sets and uncover more complex patterns than traditional statistical methods without requiring explicit specification of the underlying processes. Among various machine learning techniques, gradient-boosted methods, such as XGBoost, CatBoost, and LightGBM, were selected for this study because of their proven effectiveness in time series forecasting and their ability to handle large data sets with high-dimensional features efficiently.

Gradient boosting methods, such as XGBoost, CatBoost, and LightGBM, are ensemble learning techniques that build predictive models sequentially by combining multiple decision trees. The core idea behind gradient boosting is to iteratively reduce prediction errors by focusing on residuals, the difference between observed and predicted values. As shown in Figure 5, the process begins with training an initial decision tree T_1 using the input data and calculating the loss (Step 1). A subsequent decision tree T_2 is trained on the residuals (errors) of the first tree, aiming to minimize the loss further (Step 2). This process continues, where each new tree focuses on reducing the remaining residual errors from the previous iteration, and additional trees (T_3,…, T_n) are added sequentially until the loss converges (Step 3). The predictions from all the trees are then combined to produce the final ensemble prediction. The iterative nature of gradient boosting enables the method to capture complex, nonlinear patterns in the data while progressively improving accuracy. This makes it particularly suitable for crowd forecasting

tasks, where factors such as weather conditions, temporal variations, and crowd dynamics exhibit strong nonlinear relationships.

XGBoost, CatBoost, and LightGBM all share this foundational boosting principle; their internal architectures and optimization strategies differ significantly. These differences in tree construction, data handling, and computational efficiency are critical to their performance in different scenarios.

- XGBoost (*19*) is a highly optimized and versatile implementation that is often considered a benchmark in machine learning competitions. It employs a level-wise (breadth-first) tree growth strategy, building the tree layer by layer and splitting all nodes at the current depth before proceeding to the next. This systematic approach can result in more balanced trees and may be less prone to overfitting on smaller datasets. XGBoost's power stems from its advanced features, including built-in L1 (Lasso) and L2 (Ridge) regularization to control model complexity, a novel sparsity-aware algorithm that allows for the efficient handling of missing data, and a theoretically justified weighted quantile sketch for finding optimal split points on large data sets.
- CatBoost's (*20*) main distinguishing feature is its advanced and native handling of categorical features, which are prevalent in time series datasets (e.g., day of the week or holiday status). Traditional methods often require extensive preprocessing, such as one-hot encoding, which can lead to a "curse of dimensionality" with high-cardinality features. CatBoost addresses this with a novel algorithm called ordered boosting, a permutation-driven approach that significantly reduces prediction shift and overfitting caused by target leakage. Furthermore, CatBoost builds symmetric (or oblivious) decision trees, where all nodes at the same depth level share the same splitting criterion. This symmetric structure acts as a form of implicit regularization, preventing the model from becoming too complex and enabling extremely fast prediction times once the model is trained.
- LightGBM (*21*): In contrast to XGBoost, LightGBM is engineered for exceptional speed and efficiency, particularly with large-scale data. Its primary architectural difference is the use of a leaf-wise (or best-first) tree growth strategy. Instead of growing horizontally, LightGBM expands the tree by splitting the leaf that yields the largest reduction in loss. This approach allows the model to converge much faster and often achieves a lower overall loss, although it can risk creating overly complex and deep trees that may overfit if not properly regularized. To achieve its remarkable efficiency, LightGBM introduces two innovative techniques: (1) gradient-based one-sided sampling, which reduces the data set size by focusing on instances with larger gradients (i.e., those that are poorly trained); and (2) exclusive feature bundling (EFB), which groups mutually exclusive features to reduce the feature space dimensionality without information loss.

Deep Learning-Based Time Series Forecasting. Recently, deep learning-based models have gained prominence for their ability to capture complex long-term dependencies in time series data. Unlike traditional methods or machine learning models that rely heavily on manual feature engineering, deep learning approaches can extract features directly from raw input data, making them highly effective for multi-horizon crowd forecasting tasks. In this study, we focus on two prominent deep learning architectures: LSTM networks and TFT, to assess their multi-horizon forecasting capabilities.

Long Short-Term Memory. LSTM (*22*) networks are particularly well-suited for time series forecasting because they can remember previous information for extended periods. LSTM networks are a type of recurrent neural network (RNN) designed to overcome the limitations of traditional RNNs, particularly the vanishing gradient problem, which hinders the learning of long-term dependencies. LSTMs achieve this by incorporating specialized memory cells and three gating mechanisms, input, forget, and output gates, that regulate the flow of information throughout the network. The input gate determines which new information is added to the memory cell, the forget gate decides which information is discarded, and the output gate controls what information is propagated to the next time step. This ability to selectively "remember" and "forget" information over long sequences makes LSTMs particularly effective for crowd forecasting, where short-term variations (e.g., daily crowd peaks) and long-term trends (e.g., weekly patterns) need to be captured. This enables LSTMs to capture short- and long-term patterns in pedestrian crowd data, which is crucial for accurate multi-horizon pedestrian crowd forecasting.

Temporal Fusion Transformer. TFT (*17*) represents a significant advancement in deep learning for time series forecasting, particularly relevant to multi-horizon pedestrian crowd prediction at Scheveningen Beach. This architecture ingeniously combines elements from traditional forecasting methods with the attention mechanism

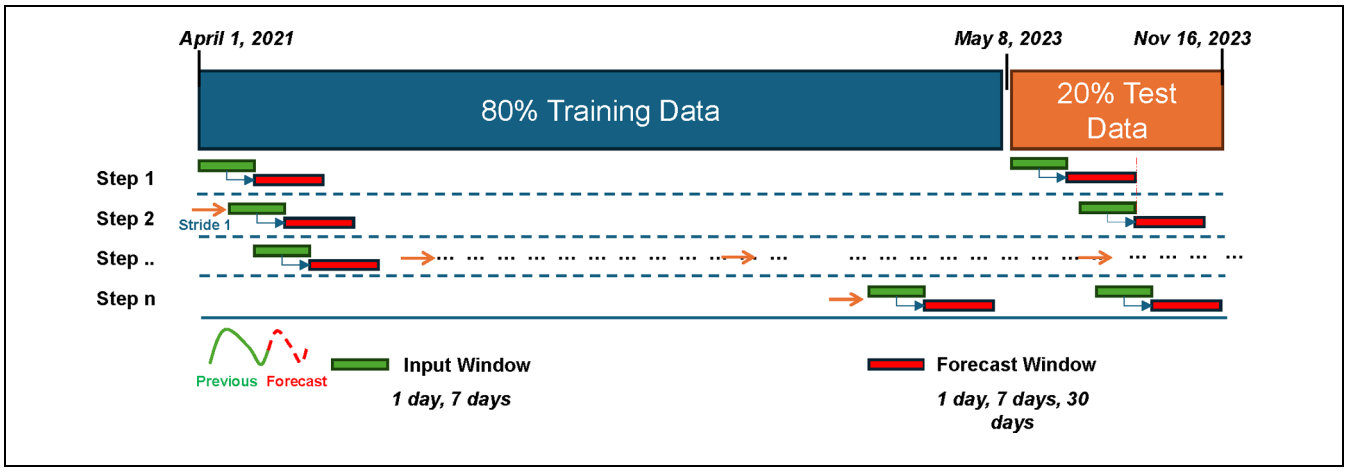

Figure 6. Sliding window method for training and testing crowd count forecasting models.

popularized by transformer models in natural language processing. The architecture includes several key components that enhance its performance. Variable selection networks dynamically select the most relevant input features, ensuring that the model focuses on the most influential predictors while reducing computational overhead. Static covariate encoders capture fixed attributes, such as location-specific characteristics, which remain constant over time. Temporal self-attention layers allow the model to identify critical time steps within the input sequence, enabling it to focus on short-term events or anomalies that influence future forecasts. In addition, gated residual networks facilitate smooth information flow throughout the model, improving its ability to capture nonlinear dependencies and enhancing overall interpretability. The TFT can process diverse inputs, including static, dynamic, and temporal features, and generate forecasts across multiple horizons. Its attention-based mechanism also provides insights into feature importance, helping identify which variables, such as temperature, wind speed, or day of the week, contribute most to the predictions. While computationally intensive, TFT's flexibility, accuracy, and interpretability make it well-suited for multi-horizon forecasting, particularly valuable for predicting crowd sizes at future time steps at Scheveningen Beach, facilitating improved resource allocation and crowd management strategies.

Model Training

A sliding window methodology was adopted to prepare and process temporal data to train and test the forecasting models, as shown in Figure 6. The sliding window approach is essential for handling time series data, ensuring models learn patterns effectively using historical input windows to predict future outputs.

The data set was split chronologically into 80% (18,400 hourly time steps) for training and 20% (4,631 hourly time steps) for testing, with the split boundary on May 8, 2023. During the training period, a time-ordered validation split was created by setting aside the last 15% of the training data for validation and the first 85% for model fitting, to support hyperparameter tuning and early stopping without leaking future information. This split ensures that the models are trained on historical data and tested on unseen future data to assess generalizability. The sliding window technique works by systematically generating overlapping input and forecast windows. The fixed-size input (green) represents a fixed segment of historical data that serves as the model's input features. Following the input window is the forecast window (red), which is the target segment the model aims to forecast. At each step, the input window advances through the time series by a fixed stride, set to one time step (1 h) in this study. The process continues sequentially, step by step, until the entire data set has been covered. This systematic movement ensures that all data points contribute to model training and evaluation, while allowing the model to learn patterns across different sections of the time series.

The lengths of the input and forecast periods are crucial to forecasting accuracy. The size of the input and forecast windows was chosen based on the temporal cycles observed in the data using autocorrelation analysis (Figure 3). For the input window, two lengths were used: (1) 24 time steps (representing 1 day of hourly data); and (2) 168 time steps (representing 7 days, or a weekly cycle). For each input window, the models were provided with a comprehensive set of historical inputs. This included all hourly lagged values (up to 24 or 168 h, respectively) for the crowd count itself, as well as all hourly lagged values for weather conditions and temporal indicators. This

extensive multivariate input ensures that the models are equipped with detailed historical context and the relationships between lagged weather patterns and temporal dynamics. This capability is crucial for understanding recurring daily and weekly seasonality, enabling models to make robust predictions for horizons that may extend beyond the direct historical sequence of the target variable alone.

The forecast window was varied to assess the model's performance across different forecasting horizons. The short-term forecast window spanned 1 day (24 time steps), the mid-term forecast covered 7 days (168 time steps), and the long-term forecast extended to 30 days (720 time steps). For the tree-based models (XGBoost, CatBoost, and LightGBM), a direct forecasting strategy was used by training a separate model for each forecast horizon (24, 168, and 720 h) to predict the full horizon directly from the input window. These window lengths were selected to effectively capture daily, weekly, and monthly patterns within the data. During model training, the input window serves as the input features, and the model learns to predict the corresponding forecast window. At each iteration, the model parameters are adjusted to minimize the error between predicted and actual values within the forecast window. This process is repeated for all overlapping windows in the training set, enabling the model to learn the underlying patterns in the data. Once training is complete, the model is evaluated on the test set using the same sliding window logic. The trained model generates predictions on unseen test data, and its performance is assessed by comparing the predicted values with the actual observations. By systematically moving the input and forecast windows across the data set, the sliding window methodology ensures that the model is exposed to various temporal patterns and dependencies.

Hyperparameter Tuning

To further enhance forecasting performance, Bayesian optimization was applied for hyperparameter tuning. The Bayesian optimization technique was implemented using the Optuna library (*23*), which efficiently identifies the optimal hyperparameters by modeling the model's performance as a probabilistic function. Bayesian optimization streamlines the search for optimal settings by reducing the required function evaluations.

For each model, hyperparameters were selected that significantly influence predictive performance, complexity, and generalization. In all gradient-boosting models, the number of trees (boosting rounds) was not a fixed hyperparameter but was determined automatically via an early-stopping mechanism based on the model's performance on a validation set. This prevents overfitting by stopping training once performance no longer improves.

Light Gradient Boosting Machine

- Learning rate [learning_rate]: Governs the step size during each iteration, ensuring a balance between convergence speed and accuracy.
- Number of trees [num_boost_round]: Sets the maximum number of trees to be built. The final number of trees was determined using an early stopping mechanism, selecting the iteration that yielded the lowest validation set error to prevent overfitting.
- Number of leaves [num_leaves]: Controls the maximum complexity of the decision tree structure, with higher values increasing model flexibility but risking overfitting.
- Maximum depth [max_depth]: Limits the tree depth to prevent overfitting on training data.
- Feature fraction [feature_fraction]: Determines the proportion of input features randomly selected for training each tree, aiding in generalization.
- Bagging fraction [bagging_fraction]: Specifies the fraction of data randomly sampled for each boosting iteration to enhance model robustness.

eXtreme Gradient Boosting

- Learning rate [learning_rate]: Regulates the size of updates applied to model parameters during training.
- Number of trees [num_boost_round]: Sets the maximum number of trees to be built. The final number of trees was determined using an early stopping mechanism, selecting the iteration that yielded the lowest validation set error to prevent overfitting.
- Maximum depth [max_depth]: Sets the depth of trees to control model complexity and avoid overfitting.
- Minimum child weight [min_child_weight]: Defines the minimum sum of instance weights required for a node to split, reducing the likelihood of overfitting small subsets.
- Subsample [subsample]: Determines the fraction of training samples used in each boosting round to enhance generalization.
- Column sample by tree [colsample_bytree]: Restricts the number of features considered during tree construction, which can improve diversity among trees.

Categorical Boosting

- Learning rate [learning_rate]: Controls the step size for updating model parameters to balance training speed and accuracy.
- Depth [depth]: Specifies the depth of each tree, balancing expressiveness and overfitting.
- Number of trees [num_boost_round]: Sets the number of boosting trees. The final number of trees was determined using an early stopping mechanism, selecting the iteration that yielded the lowest validation set error to prevent overfitting.
- L2 regularization [l2_leaf_reg]: Penalizes large leaf values to prevent overfitting.
- Border count [border_count]: Defines the number of bins used for splitting numerical features, affecting model precision.

Long Short-Term Memory

- Learning rate: Adjusts the rate of weight updates during optimization.
- Batch size: Sets the number of training samples processed in each batch before updating the model.
- Number of layers: Determines the depth of the LSTM network, which affects the model's ability to capture sequential dependencies.
- Neurons per layer: Controls the number of hidden units in each layer, influencing model capacity.
- Dropout rate: Reduces overfitting by randomly deactivating units during training.
- Number of Epochs [epochs]: Sets the maximum number of training epochs. The optimal number of epochs was determined using an early stopping mechanism based on validation loss to prevent overfitting.

Temporal Fusion Transformer

- Learning rate: Governs the step size of parameter updates.
- Hidden size [hidden_size]: Specifies the dimensionality of the hidden layers, which affects model capacity.
- Number of attention heads [num_heads]: Controls the number of parallel attention mechanisms for capturing dependencies.
- Dropout rate: Prevents overfitting by randomly zeroing out neurons during training.
- Attention head size: Determines the size of each attention head, affecting how information is processed across attention layers.
- Number of epochs [epochs]: Sets the maximum number of training epochs. The optimal number of epochs was determined using an early stopping mechanism based on validation loss to prevent overfitting.

These hyperparameters were chosen because they directly influence the models' ability to balance learning complexity, training efficiency, and generalization to unseen data. The final optimized hyperparameter values for all models are provided in Tables S1 and S2 in the Supplementary Materials.

Model Evaluation: Performance Metrics

On completion of the training phase, the best-performing model was reported based on its forecasting performance on the test data set. Several metrics were used to evaluate performance, including R^2, RMSE, SMAPE, and MAE. The R^2 and RMSE assess model performance from a statistical viewpoint. The R^2 is a statistical measure that quantifies the proportion of variance in a dependent variable that is explained by the independent variables. In addition, the RMSE indicates the magnitude of prediction errors. Higher R^2 and lower RMSE values suggest better model accuracy. From a process perspective, SMAPE and MAE are crucial. The SMAPE evaluates the relative accuracy of forecasts in percentage terms, making it easy to interpret. The MAE provides the average absolute difference between predicted and actual values, indicating the average prediction error. To facilitate a fair comparison of model performance across different locations with varying crowd count scales, the MAE and RMSE were normalized (NMAE and NRMSE). The NMAE and NRMSE were calculated by dividing the MAE and RMSE by the mean of the observed crowd counts for each respective location (as listed in Table 1). These normalized metrics provide a relative error measure, which is more suitable for comparing model performance across data sets with different scales. These metrics ensure a comprehensive evaluation, capturing the statistical and practical aspects of model performance, which leads to a robust pedestrian crowd count forecasting model.

Results

In this section, a comprehensive evaluation of the trained crowd forecasting models is conducted, such as XGBoost, LightGBM, CatBoost, LSTM, and TFT, for the various horizons of 1, 7, and 30 days ahead on the test data set, evaluating the 24-h and 168-h input windows for each forecast horizon. To assess the multivariate output capabilities of the models, the forecasting models were trained in two distinct ways: (1) individually for each location (individual location models); and (2) collectively across all locations (unified multilocation

models). In the unified approach, a single model per model family (XGBoost, CatBoost, LightGBM, LSTM, and TFT) is trained using data from all locations to learn generalized patterns and forecast crowd counts for all respective locations. Then, the performance of different model architectures was compared within this unified framework to identify the most effective approach.

Baseline Forecasting Model's Performance across Various Locations

Table 2 presents the comparative performance of individual location crowd forecasting models trained on Naive Daily, Naive Weekly, and Naive Yearly models, evaluated using the MAE, NMAE, RMSE, NRMSE, and R^2 metrics. The performance of these baseline models differs across locations and time horizons; however, they generally indicate the difficulty of capturing complex crowd dynamics with simple methods.

The Naive Daily model exhibits high absolute errors in high-traffic areas, such as Central Boulevard (MAE = 937.72) and the Entire Scheveningen area (MAE = 1,128.52). For locations with lower but more erratic crowd counts, such as Beach Stadium and Entrance Zwarte Pad, the NMAE is particularly high (1.38 and 1.41, respectively), indicating that the model's predictions are poor relative to the average crowd size. This is further confirmed by negative R^2 in many areas (e.g., -0.76 for Beach Stadium), signifying that the forecasts are less accurate than simply using the historical mean.

The Naive Weekly model offers slightly better performance in locations with strong weekly patterns, such as Public Transport (Kurhaus), where the R^2 improves to 0.39. However, for most other locations, the improvements are marginal, or the performance degrades. The Naive Yearly model consistently performs the worst, with the highest MAE and RMSE across most locations, and a profoundly negative R^2 (e.g., -2.19 for Beach Stadium and -0.41 for Central Boulevard). This indicates that simple naive forecast baselines fail to capture the evolving dynamics of crowd behavior. The high error metrics and predominantly low or negative R^2 across all three naive models underscore their limited accuracy, establishing a clear need for more advanced forecasting techniques.

Multi-Horizon Forecasting Performance of Individual Location Models

Table 3 presents the performance metrics of the best models trained individually in various locations at Scheveningen Beach at various horizons, along with optimal input window, training time, and forecast time. The dominance of CatBoost in short-term forecasting across most locations suggests its strength in capturing immediate trends and local patterns. This could be attributed to its ordered boosting approach, which may be especially effective in modeling the day-to-day variations in crowd patterns at Scheveningen Beach. In addition, the preference for a 168-step input window at most locations in short-term forecasting suggests that weekly patterns play a crucial role in daily crowd forecasting and may capture effects such as weekday–weekend variations and weekly recurring events. The exceptions (Boulevard areas and Entire Scheveningen) preferring a 24-step input might indicate more rapid changes in crowd patterns in these areas because of the presence of recreational areas and restaurants.

For mid-term forecasting, CatBoost was the best, followed by LSTM (Entire Scheveningen, Central Beach, North Beach, and Entrance Zwarte Pad) and LightGBM (Entrance Kurhaus, Entrance Scheveningseslag, and Entrance Zeekant). The continued strong performance of CatBoost in mid-term (1-week-ahead) forecasting, along with the emergence of LSTM and LightGBM in certain locations, reveals the increasing complexity of forecasting as the time horizon extends. The consistent preference for a 168-step input window in mid-term forecasting reinforces the importance of weekly patterns in crowd behavior. However, the exceptions (e.g., Public Transport [Strandweg], Public Transport [Zwarte Pad], South Beach, and Entrance Gevers Deynootweg) that prefer a 24-step input are intriguing and might indicate locations where very recent trends are more predictive of future crowds than longer-term patterns.

The shift toward LightGBM as the top performer for long-term forecasting, with LSTM excelling in some locations, followed by the TFT model, indicates a changing landscape of forecasting factors as the time horizon extends. The notable performance of LightGBM could be attributed to the gradient-based one-sided sampling and EFB techniques that better capture unique characteristics or data patterns for longer-term predictions. The variation in preferred input window sizes for long-term forecasting (split between 24 and 168 steps) suggests that different locations have distinct long-term crowd dynamics. This variability underscores the complexity of long-term crowd prediction and the need for location-specific modeling approaches.

Multi-Horizon Forecasting Performance of Unified Multilocation Model

For the unified multilocation approach, several model types were evaluated (XGBoost, CatBoost, LightGBM, LSTM, and TFT). Each model type was trained as a

Table 2. Comparative Performance of Individual Location Crowd Forecasting Models Trained on Naive Daily, Weekly, and Yearly Models on Test Data Set

	Naive Daily					Naive Weekly					Naive Yearly				
Location	MAE	RMSE	NMAE	NRMSE	R^2	MAE	RMSE	NMAE	NRMSE	R^2	MAE	RMSE	NMAE	NRMSE	R^2
Beach Stadium	31.23	119.79	1.38	5.27	−0.76	24.50	101.25	1.08	4.46	−0.25	41.02	161.57	1.81	7.11	−2.19
Central Boulevard	937.72	1,720.06	0.74	1.36	0.05	990.50	1,676.27	0.78	1.33	0.10	1,138.75	2,100.53	0.90	1.66	−0.41
North Boulevard	575.28	1,085.83	0.88	1.65	−0.02	580.90	1,024.04	0.88	1.56	0.09	656.20	1,222.82	1.00	1.86	−0.30
South Boulevard	477.58	884.36	0.87	1.61	0.00	471.71	818.02	0.86	1.49	0.15	518.61	921.24	0.94	1.67	−0.08
The Pier	432.30	856.21	0.93	1.83	0.04	445.98	821.48	0.96	1.76	0.12	469.63	940.91	1.01	2.01	−0.16
Public Transport (Kurhaus)	512.20	838.04	0.58	0.95	0.26	466.37	758.84	0.53	0.86	0.39	600.26	961.88	0.68	1.09	0.03
Public Transport (Strandweg)	178.80	312.46	0.91	1.59	−0.17	178.85	288.90	0.91	1.47	0.00	211.94	346.75	1.08	1.77	−0.44
Public Transport (Zwarte Pad)	234.80	465.31	1.26	2.49	−0.53	195.41	398.29	1.05	2.13	−0.12	251.58	459.01	1.35	2.46	−0.49
Entire Scheveningen	1,128.52	2,041.15	0.76	1.38	0.23	1,208.06	2,057.34	0.82	1.39	0.21	1,313.85	2,379.74	0.89	1.61	−0.05
Central Beach	579.48	1,112.57	0.79	1.51	0.11	674.18	1,151.04	0.92	1.57	0.05	669.75	1,286.90	0.91	1.75	−0.19
North Beach	395.37	762.84	0.93	1.80	0.07	407.35	764.55	0.96	1.81	0.06	458.01	891.93	1.08	2.11	−0.27
South Beach	495.36	966.41	0.93	1.81	0.05	563.99	983.75	1.05	1.84	0.02	544.77	1,042.47	1.02	1.95	−0.10
Entrance Gevers Deynootweg	262.24	439.26	0.90	1.51	0.21	271.14	469.55	0.93	1.61	0.10	298.74	495.05	1.02	1.70	−0.01
Entrance Kurhaus	475.48	795.11	0.57	0.95	0.27	507.35	811.51	0.60	0.97	0.24	621.69	1,082.28	0.74	1.29	−0.35
Entrance Scheveningseslag	299.69	499.88	0.70	1.18	0.30	300.23	486.18	0.71	1.14	0.34	314.29	521.84	0.74	1.23	0.24
Entrance Zeekant	325.07	558.37	0.79	1.36	0.01	285.3	467.55	0.70	1.14	0.31	356.22	614.02	0.87	1.50	−0.19
Entrance Zwarte Pad	149.81	359.65	1.41	3.40	−0.67	115.23	295.41	1.09	2.79	−0.12	158.83	360.74	1.50	3.41	−0.68

Note: MAE = mean absolute error; RMSE = root mean square error; NMAE = normalized mean absolute error; NRMSE = normalized root mean square error; R^2 = coefficient of determination.

Table 3. Comparative Performance of Individual Location Crowd Forecasting Models Trained on Different Prediction Windows

Prediction window (time steps)	Best model	Best input window (time steps)	MAE	RMSE	NMAE	NRMSE	SMAPE (%)	R^2	Training time (s)	Forecast time (s)
Beach Stadium										
24	CatBoost	168	35.78	94.11	1.58	4.14	176.91	−0.07	2,088.21	147.81
168	CatBoost	168	34.66	96.20	1.53	4.24	176.92	−0.11	2,833.19	67.62
720	LightGBM	24	42.23	103.76	1.86	4.57	182.18	−0.06	1,406.26	11.79
Central Boulevard										
24	CatBoost	24	688.92	1,142.99	0.54	0.90	83.75	0.58	535.65	22.18
168	CatBoost	168	750.28	1,211.18	0.59	0.96	85.88	0.54	3,046.18	71.69
720	LSTM	24	789.93	1,266.26	0.62	1.00	91.64	0.51	61.72	4.45
North Boulevard										
24	CatBoost	24	438.49	755.97	0.67	1.15	101.66	0.51	538.66	21.43
168	CatBoost	168	466.64	801.58	0.71	1.22	102.64	0.45	3,064.4	75.23
720	LightGBM	24	510.01	846.95	0.78	1.29	100.54	0.38	1,435.4	10.16
South Boulevard										
24	LightGBM	24	339.79	590.79	0.62	1.07	87.40	0.56	43.53	0.47
168	CatBoost	168	373.07	644.47	0.68	1.17	90.13	0.48	3,069.45	72.85
720	LightGBM	24	426.35	723.79	0.77	1.31	87.42	0.40	1,422.13	10.50
The Pier										
24	CatBoost	168	311.28	577.40	0.67	1.24	105.53	0.58	2,207.72	149.03
168	CatBoost	168	327.71	610.02	0.70	1.31	105.18	0.53	3,019.86	72.90
720	LightGBM	168	351.02	642.16	0.75	1.38	97.92	0.50	2,192.06	11.42
Public Transport (Kurhaus)										
24	CatBoost	168	351.95	531.24	0.40	0.60	54.24	0.71	2,245.74	150.00
168	CatBoost	168	367.62	564.95	0.42	0.64	57.22	0.67	3,054.68	72.94
720	LightGBM	168	398.83	634.41	0.45	0.72	54.08	0.61	2,385.37	12.56
Public Transport (Strandweg)										
24	CatBoost	168	138.13	210.66	0.70	1.07	94.60	0.48	2,226.97	146.50
168	CatBoost	24	151.66	235.39	0.77	1.20	97.98	0.35	2,767.44	85.94
720	LSTM	168	167.04	261.97	0.85	1.33	118.80	0.29	78.80	4.40
Public Transport (Zwarte Pad)										
24	CatBoost	168	199.09	345.77	1.07	1.85	109.23	0.18	2,143.50	146.41
168	CatBoost	24	200.96	349.54	1.08	1.87	110.18	0.17	2,724.86	83.56
720	LightGBM	168	220.93	361.00	1.18	1.93	112.36	0.14	2,262.30	12.78
Entire Scheveningen										
24	CatBoost	24	713.55	1,191.96	0.48	0.81	69.64	0.74	532.44	21.73
168	LSTM	168	891.63	1,386.24	0.60	0.94	86.37	0.65	80.21	4.69
720	LSTM	168	949.71	1,480.22	0.64	1.00	82.32	0.64	79.05	4.86
Central Beach										
24	CatBoost	168	420.14	721.30	0.57	0.98	85.14	0.64	2,203.68	147.11
168	LSTM	168	462.86	780.97	0.63	1.06	100.34	0.57	79.52	4.92
720	LSTM	168	494.27	821.19	0.67	1.12	94.64	0.53	77.50	4.38
North Beach										
24	CatBoost	168	291.74	499.14	0.69	1.18	95.45	0.61	2,202.97	145.65
168	LSTM	168	319.40	549.54	0.75	1.30	110.14	0.53	78.17	4.43
720	LSTM	168	342.37	582.02	0.81	1.37	102.65	0.53	79.13	4.99

(continued)

Table 3. (continued)

Prediction window (time steps)	Best model	Best input window (time steps)	MAE	RMSE	NMAE	NRMSE	SMAPE (%)	R^2	Training time (s)	Forecast time (s)
South Beach										
24	CatBoost	168	359.76	637.15	0.67	1.19	88.35	0.60	2,183.37	146.40
168	CatBoost	24	399.96	729.57	0.75	1.36	91.84	0.48	2,755.23	84.78
720	LightGBM	24	430.84	784.06	0.80	1.46	86.96	0.46	1,452.41	11.52
Entrance Gevers Deynootweg										
24	CatBoost	168	223.84	371.21	0.77	1.27	84.66	0.45	1,217.08	53.57
168	CatBoost	24	237.86	392.67	0.82	1.35	87.59	0.39	2,778.23	83.79
720	LightGBM	24	252.07	395.46	0.86	1.36	85.65	0.41	1,567.81	12.00
Entrance Kurhaus										
24	CatBoost	168	365.50	575.18	0.43	0.68	61.10	0.63	1,225.38	53.62
168	LightGBM	168	365.21	582.95	0.43	0.69	59.85	0.61	576.23	3.54
720	TFT	24	426.32	658.98	0.51	0.78	69.09	0.51	1,824.44	42.57
Entrance Scheveningseslag										
24	CatBoost	168	214.33	344.04	0.50	0.81	69.48	0.68	1,247.23	52.52
168	LightGBM	168	225.58	373.10	0.53	0.88	67.04	0.62	577.11	3.54
720	LightGBM	168	249.98	407.92	0.59	0.96	70.13	0.58	2,435.66	12.87
Entrance Zeekant										
24	CatBoost	168	247.32	382.81	0.60	0.93	81.97	0.55	1,213.02	52.35
168	LightGBM	168	270.26	437.07	0.66	1.07	81.64	0.41	572.88	3.55
720	LightGBM	24	305.87	458.24	0.75	1.12	87.67	0.29	1,513.00	12.14
Entrance Zwarte Pad										
24	CatBoost	168	128.70	265.47	1.22	2.51	104.48	0.12	1,146.59	52.43
168	LSTM	168	128.23	274.13	1.21	2.59	111.96	0.09	79.31	4.73
720	LightGBM	24	146.92	282.84	1.39	2.67	109.73	0.07	1,449.90	11.86

Note: MAE = mean absolute error; RMSE = root mean square error; NMAE = normalized mean absolute error; NRMSE = normalized root mean square error; R^2 = coefficient of determination; SMAPE = symmetric mean absolute percentage error; CatBoost = categorical boosting; LightGBM = light gradient boosting machine; LSTM = long short-term memory; TFT = Temporal Fusion Transformer.

single, unified model on the combined data set from all locations, generating forecasts for all locations simultaneously. Table 4 presents the performance metrics of the best unified multilocation models trained on various locations at Scheveningen Beach at various horizons, with an optimal input window, training time, and forecast time. In short-term forecasting, CatBoost maintained its strong performance, comparable with individual location models. In addition, LightGBM and LSTM showed effectiveness for certain locations, indicating that varied approaches might benefit different beach areas within a unified framework. This preference suggests that recent trends are more universally informative when attempting to generalize the entire beach area. Location-specific observations revealed varying success. Beach Stadium showed poor performance with a negative R^2, indicating challenges in generalizing crowd patterns. In contrast, high-traffic areas, such as Central Boulevard and the Entire Scheveningen area, maintained relatively good performance ($R^2 > 0.5$), suggesting that the model captures general trends well in busier locations with more consistent patterns.

In mid-term forecasting (1-week-ahead), a shift in model effectiveness is observed. LightGBM emerged as the top performer in most locations, and LSTM excelled in several areas. This transition from CatBoost to LightGBM suggests that LightGBM's leaf-wise growth strategy allows for more nuanced tree structures, potentially capturing subtle mid-term trends more effectively than other models. LightGBM favored 24-step input windows, and LSTM preferred 168-step input windows. This suggests that LightGBM effectively extracts features from recent data, and LSTM captures extended temporal dependencies. Performance metrics showed a decrease in R^2 and an increase in the MAE and RMSE values compared with short-term forecasts, indicating increased difficulty in predicting further into the future, with varying predictability across different beach areas.

Long-term forecasting (1-month-ahead) showed that LSTM consistently outperforms other models across all locations, highlighting the deep learning model's ability to capture long-term dependencies and trends. Input window preferences varied, with about half the locations favoring 24 steps and the other half 168 steps, indicating distinct long-term patterns that benefit from different data ranges, even within a unified modeling approach. Performance metrics indicated a further decrease in R^2 compared with mid-term forecasts, accompanied by an increase in the MAE and RMSE values, suggesting higher uncertainty in long-term predictions. In Beach Stadium, the high variability might be primarily because of the irregular patterns introduced by the frequent and diverse events hosted there, making long-term trend prediction challenging.

To visually compare the performance of the different modeling strategies, Figure 7 shows the 7-day-ahead hourly forecasts for the Entire Scheveningen Beach location. This figure shows two distinct visualizations. The results from the individual location models are shown first, followed by the forecasts from the unified multilocation models. In the initial plot of the individual models, the forecasts accurately capture daily visitor patterns, with clear daytime peaks. As noted in Table 3, the LSTM model was the best performer for this mid-term horizon, and the visualization confirms that its predictions closely track the actual visitor counts. In contrast, the TFT model appears more volatile, occasionally overestimating the peaks significantly. The second plot, which illustrates the unified approach, demonstrates that this method also successfully captures the overall weekly trends. For the unified models, Table 4 identifies LightGBM as the top performer. Visually, the LightGBM and LSTM models provide robust forecasts that align well with the actual data. Comparing the two visualizations reveals that while the best individual model (LSTM) offers a slightly more accurate fit, the best unified model (LightGBM) remains highly competitive. This direct comparison highlights the trade-off between the higher precision of specialized individual models and the generalized performance of a single unified model.

Comparative Analysis of Baseline, Individual, and Unified Multilocation Models

The comparative analysis of baseline, individual, and unified multilocation models revealed performance variations across different forecasting approaches for Scheveningen Beach crowd forecasting.

Naive baseline models, both weekly and yearly, consistently underperformed, often yielding negative R^2 values (e.g., -0.25 for Beach Stadium in the weekly model), highlighting their inability to capture the complex spatiotemporal patterns in beach crowd dynamics. In contrast, machine learning and deep learning models demonstrated superior capabilities for locations driven by regular temporal and weather-related patterns. However, for locations with highly irregular, event-driven dynamics, such as the Beach Stadium, all models struggled to produce meaningful forecasts, as evidenced by consistently negative R^2 in the individual (Table 3) and unified (Table 4) models. This indicates that the crowd dynamics at the stadium are governed by factors not present in the current feature set, such as a specific event schedule, which makes its attendance patterns fundamentally harder to predict without this crucial external data. For

Table 4. Comparative Performance of Unified Multilocation Crowd Forecasting Models Trained on Different Prediction Windows

Prediction window (time steps)	Best model	Best input window (time steps)	MAE	RMSE	NMAE	NRMSE	SMAPE (%)	R^2	Training time (s)	Forecast time (s)
Beach Stadium										
24	LightGBM	168	38.89	96.89	1.71	4.27	179.28	−0.14	1,799.66	1.12
168	LSTM	168	54.77	100.26	2.41	4.41	182.03	−0.20	1,41.18	9.12
720	LSTM	168	58.64	108.40	2.58	4.77	183.15	−0.15	1,40.03	9.63
Central Boulevard										
24	CatBoost	24	713.91	1,171.85	0.56	0.93	83.05	0.56	5,229.38	259.50
168	LightGBM	24	815.89	1,352.96	0.64	1.07	82.07	0.43	3,039.33	4.93
720	LSTM	168	1,221.03	1,624.63	0.97	1.28	103.48	0.19	140.03	9.63
North Boulevard										
24	CatBoost	24	444.08	764.01	0.68	1.16	102.00	0.50	5,229.38	259.50
168	LightGBM	24	490.89	854.19	0.75	1.30	100.23	0.38	3,039.33	4.93
720	LSTM	168	660.35	889.38	1.00	1.35	110.84	0.31	140.03	9.63
South Boulevard										
24	CatBoost	24	339.29	587.78	0.62	1.07	88.62	0.56	5,229.38	259.50
168	LSTM	168	451.66	655.57	0.82	1.19	99.92	0.46	141.18	9.12
720	LSTM	168	472.94	685.83	0.86	1.25	95.63	0.46	140.03	9.63
The Pier										
24	LightGBM	24	303.10	575.31	0.65	1.23	98.40	0.57	474.81	0.79
168	LightGBM	24	328.33	622.30	0.70	1.33	98.93	0.51	3,039.33	4.93
720	LSTM	168	527.38	737.18	1.13	1.58	119.35	0.34	140.03	9.63
Public Transport (Kurhaus)										
24	CatBoost	24	360.81	541.79	0.41	0.61	57.64	0.69	5,229.38	259.50
168	LightGBM	24	368.23	580.84	0.42	0.66	53.62	0.65	3,039.33	4.93
720	LSTM	24	676.13	826.34	0.76	0.93	82.10	0.33	104.27	10.09
Public Transport (Strandweg)										
24	LightGBM	24	147.52	228.19	0.75	1.16	98.16	0.38	474.81	0.79
168	LSTM	24	166.92	234.12	0.85	1.19	103.02	0.36	103.08	9.38
720	LSTM	24	172.46	247.72	0.88	1.26	99.88	0.37	104.27	10.09
Public Transport (Zwarte Pad)										
24	CatBoost	24	204.36	346.32	1.09	1.85	111.87	0.16	5,229.38	259.50
168	LightGBM	24	207.13	354.16	1.11	1.90	109.12	0.15	3,039.33	4.93
720	LSTM	168	204.96	358.11	1.10	1.92	111.13	0.15	140.03	9.63
Entire Scheveningen										
24	CatBoost	24	727.18	1,217.75	0.49	0.82	69.87	0.73	5,229.38	259.50
168	LightGBM	24	905.24	1,547.28	0.61	1.05	71.05	0.57	3,039.33	4.93
720	LSTM	168	1,227.25	1,656.12	0.83	1.12	90.60	0.55	140.03	9.63
Central Beach										
24	CatBoost	24	419.38	731.87	0.57	1.00	87.24	0.62	5,229.38	259.50
168	LightGBM	24	499.79	861.26	0.68	1.17	83.84	0.48	3,039.33	4.93
720	LSTM	168	652.18	927.60	0.89	1.26	99.92	0.41	140.03	9.63
North Beach										
24	CatBoost	24	293.25	507.32	0.69	1.20	97.77	0.59	5,229.38	259.50
168	LightGBM	24	337.85	584.58	0.80	1.38	95.31	0.47	3,039.33	4.93
720	LSTM	168	448.17	617.42	1.06	1.46	106.15	0.47	140.03	9.63

(continued)

Table 4. (continued)

Prediction window (time steps)	Best model	Best input window (time steps)	MAE	RMSE	NMAE	NRMSE	SMAPE (%)	R^2	Training time (s)	Forecast time (s)
South Beach										
24	CatBoost	24	355.12	657.57	0.66	1.23	87.60	0.57	5,229.38	259.50
168	LSTM	168	430.21	719.06	0.80	1.34	97.81	0.50	141.18	9.12
720	LSTM	168	458.45	769.03	0.86	1.44	92.97	0.48	140.03	9.63
Entrance Gevers Deynootweg										
24	CatBoost	24	218.11	369.77	0.75	1.27	86.33	0.44	5,229.38	259.50
168	LightGBM	24	231.31	401.88	0.79	1.38	84.22	0.36	3,039.33	4.93
720	LSTM	24	298.67	428.62	1.02	1.47	95.72	0.31	104.27	10.09
Entrance Kurhaus										
24	LightGBM	168	338.91	554.81	0.40	0.66	57.60	0.65	1,799.66	1.12
168	LightGBM	24	389.25	618.53	0.46	0.74	61.41	0.56	3,039.33	4.93
720	LSTM	24	687.05	885.95	0.82	1.05	86.49	0.11	104.27	10.09
Entrance Scheveningseslag										
24	CatBoost	24	228.64	357.63	0.54	0.84	75.23	0.65	5,229.38	259.50
168	LightGBM	24	230.12	386.38	0.54	0.91	67.87	0.60	3,039.33	4.93
720	LSTM	168	334.51	417.32	0.79	0.98	90.96	0.56	140.03	9.63
Entrance Zeekant										
24	CatBoost	24	256.80	394.23	0.63	0.96	84.66	0.51	5,229.38	259.50
168	LightGBM	24	266.27	437.01	0.65	1.07	80.75	0.41	3,039.33	4.93
720	LSTM	24	339.39	444.08	0.83	1.08	95.75	0.33	104.27	10.09
Entrance Zwarte Pad										
24	LSTM	24	115.93	261.55	1.09	2.47	102.81	0.13	101.86	8.08
168	LSTM	168	123.86	268.45	1.17	2.53	104.69	0.13	141.18	9.12
720	LSTM	24	126.40	276.81	1.19	2.61	102.57	0.11	104.27	10.09

Note: MAE = mean absolute error; RMSE = root mean square error; NMAE = normalized mean absolute error; NRMSE = normalized root mean square error; R^2 = coefficient of determination; SMAPE = symmetric mean absolute percentage error; CatBoost = categorical boosting; LightGBM = light gradient boosting machine; LSTM = long short-term memory.

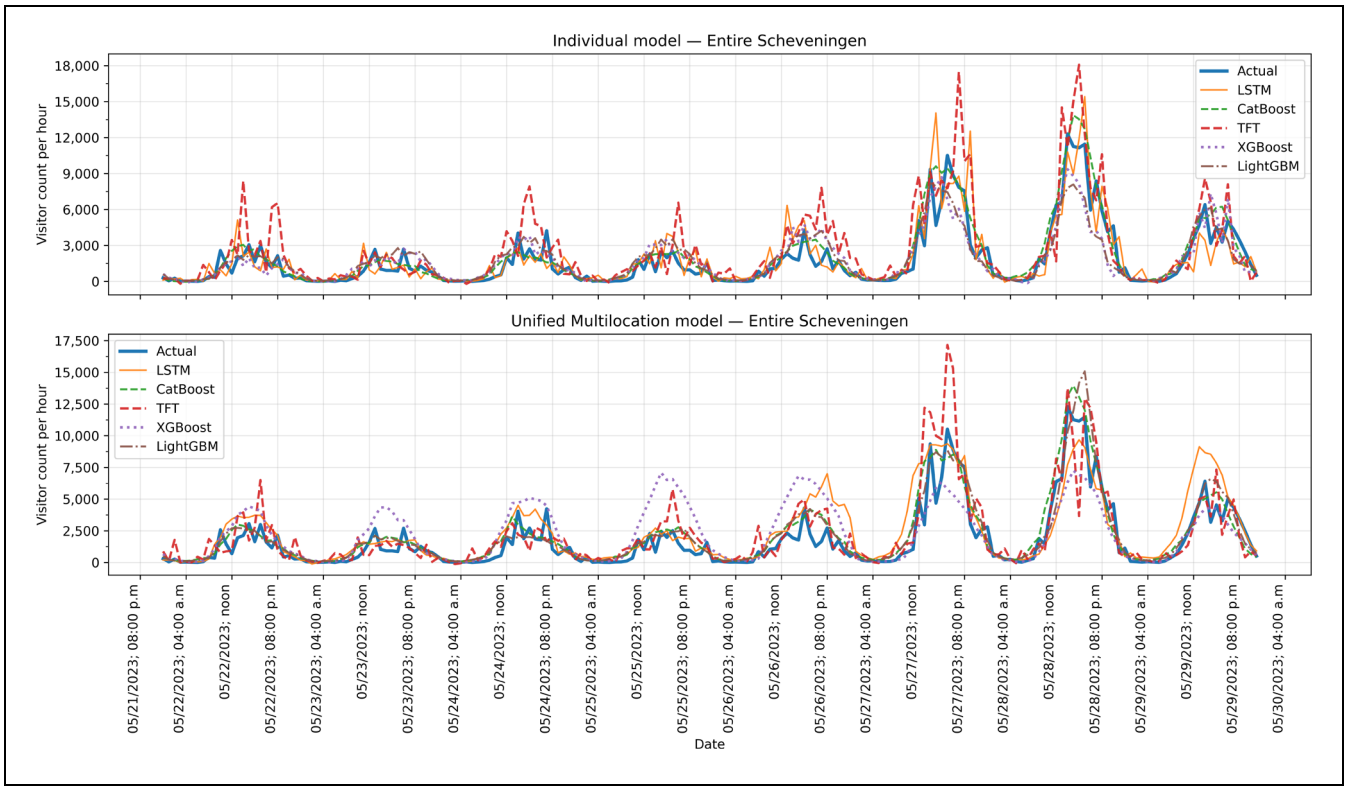

Figure 7. Visualization of 7-day-ahead hourly crowd forecasts for Entire Scheveningen location using individual and unified location models.

Note: LSTM = long short-term memory; CatBoost = categorical boosting; TFT = Temporal Fusion Transformer; XGBoost = eXtreme gradient boosting; LightGBM = light gradient boosting machine.

instance, the individual CatBoost model for Central Boulevard achieved an R^2 of 0.58 for 24-h predictions, substantially outperforming the corresponding naive daily baseline (R^2 of 0.05). Likewise, for the 168-h forecast horizon, the individual CatBoost model (R^2 of 0.54) demonstrated a significant performance improvement over the naive weekly baseline (R^2 of 0.1). The unified multilocation approach, while slightly less accurate than individual models in some cases, offered computational efficiency and more consistent performance across various locations. For the entire Scheveningen area in the 24-h prediction window, the unified CatBoost model attained an R^2 of 0.73, compared with the individual model's R^2 of 0.74, demonstrating the unified model's potential to leverage cross-location correlations and shared influencing factors. Figure 8 shows the MAE performance comparison between individual and multilocation models for forecasting crowd counts at different locations across three prediction windows: 24, 168, and 720 h. This demonstrates the varying performance of both approaches across various locations and time horizons. While individual models generally show lower MAE values, especially for shorter prediction windows, the unified multilocation model maintains competitive performance, particularly for longer-term forecasts in some locations.

Analysis of the training and prediction times revealed variations between models and approaches. CatBoost models in the unified approach typically had longer training times (e.g., 5,229.38 s for Central Boulevard's 24-h prediction) but faster prediction times (259.5 s). LightGBM models showed shorter training times (e.g., 474.81 s for The Pier) and extremely fast prediction times (0.79 s). The LSTM models in the unified approach showed moderate training times (around 140 s) and prediction times (around 9 s) for 720-h forecasts. In general, LightGBM and TFT provided a better balance between time efficiency and forecasting performance, and LSTM and CatBoost, although capable of high R^2, required more time for training and forecasting. From a practical perspective, the choice between unified and individual models presents a classic trade-off between precision and operational efficiency. Individual models offer the highest accuracy and the ability to capture nuanced local patterns; they require more computational resources, more complex management systems, and potentially higher maintenance efforts. The unified model, although sometimes less accurate, provides a more streamlined, cost-

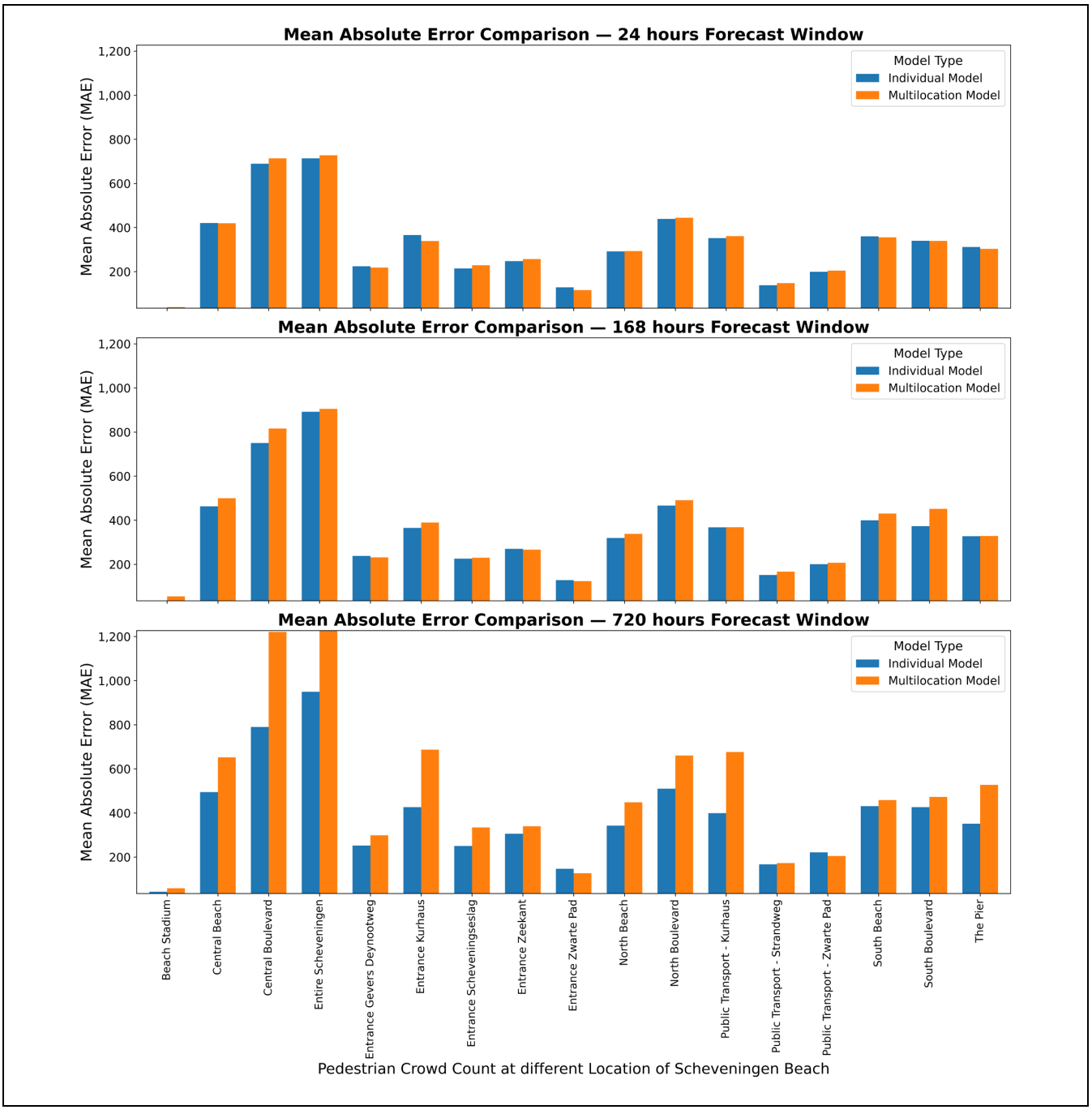

Figure 8. MAE performance comparison between individual and multilocation models for forecasting crowds at different locations across various prediction windows.
Note: MAE = mean absolute error.

effective solution that might be more feasible for large-scale implementation.

Discussion and Conclusion

This study presents a comprehensive evaluation of multi-horizon crowd forecasting models for Scheveningen Beach, revealing complex spatiotemporal patterns in crowd dynamics with significant implications for crowd management strategies and planning. The findings demonstrate a clear temporal hierarchy in model performance. Gradient-boosting models, particularly CatBoost, excelled in short-term forecasting (24 h-ahead), and the LightGBM and LSTM models showed increased efficacy

for mid-term predictions (168 h-ahead). The LSTM models dominated long-term forecasting (720 h-ahead), highlighting the importance of capturing long-range temporal patterns for extended predictions.

The analysis revealed spatial heterogeneity in crowd dynamics, highlighting a trade-off between the statistical precision of location-specific modeling and the operational consistency of a unified approach. First, in short-term (24-h) forecasts, individual models often provided a tighter fit; for instance, Entrance Zeekant achieved a lower NMAE (0.60) compared with the unified model (0.63). Second, this performance gap became more pronounced as the forecast horizon increased. In the mid-term (168-h) forecast for Central Boulevard, the individual model produced $R^2 = 0.54$ compared with the unified model's 0.43. While the individual model captures local variance more accurately, the unified model still maintains reasonable predictive power and captures the general trend. Therefore, while individual models are statistically superior at specific high-variance locations, the error magnitudes are often comparable, suggesting that the unified approach remains a robust and computationally efficient alternative for general deployment across the beach.

Furthermore, the optimal model architecture often differed across locations, suggesting unique underlying data characteristics. For the mid-term forecast for Entire Scheveningen, the best-performing individual model was the LSTM; however, the best unified model was LightGBM. This divergence implies that a model optimized on data from all locations is not necessarily the best at capturing the specific temporal dynamics of a single area. The most extreme case of this heterogeneity was Beach Stadium, where event-driven crowds produced negative R^2 across all models, indicating that its patterns cannot be generalized to other beach locations. Despite these variations, a unified model offers significant benefits for model maintenance and understandability. As the prediction horizon extended, forecast accuracy consistently decreased, highlighting the increasing uncertainty in long-term forecasts.

This suggests that an optimal forecasting strategy must align model selection with specific management goals. Short-term forecasts generally possess high accuracy; however, long-term planning must account for greater uncertainty. Therefore, short-term forecasts should guide immediate tactical decisions, such as opening specific beach entrances or deploying medical units. In contrast, long-term forecasts, despite higher uncertainty, are valuable for strategic baseline planning, such as scheduling waste management services or determining seasonal staffing budgets. Areas with better predictability might benefit from automated, data-driven resource allocation, and less predictable areas may require more flexible management approaches.

The proposed machine learning and deep learning models provide practical, actionable tools to enhance forecasting accuracy and improve real-world crowd management strategies. These models can be integrated into existing decision-support systems used by crowd safety managers, such as dashboards that consolidate historical crowd data, weather forecasts, and event schedules. Short-term forecasts (e.g., 1-day) generated by models such as CatBoost enable immediate operational decisions, including optimizing the allocation of security personnel, medical teams, and crowd-control infrastructure to specific zones. Mid-term forecasts (7-day horizon) using models such as LightGBM facilitate planning for peak-demand periods, including weekends and public holidays, by improving resource scheduling and anticipating traffic congestion. For long-term predictions (30-day horizon), models such as LSTM are particularly useful for strategic planning, including budgeting for staff, coordinating with public transportation services, and ensuring adequate infrastructure to accommodate expected crowds. As demonstrated at Scheveningen Beach, these methodologies can be extended to other crowded environments, such as large-scale events (concerts, festivals, and sports matches) and regularly crowded areas, such as inner cities. In both cases, crowd dynamics are influenced by similar factors. By training these models on location-specific data, the approach remains hiadaptable, offering accurate, data-driven predictions to support effective crowd management, resource allocation, and safety interventions.

Future research directions include exploring hybrid models and integrating additional data sources such as real-time public transportation schedules, parking occupancy data, and social media trends. This is particularly relevant for locations subject to event-driven volatility, such as Beach Stadium, where this study revealed a sparse time series characterized by low crowd counts and sharp, intermittent spikes. To address this, future studies should incorporate comprehensive event calendars to differentiate between standard daily patterns and exceptional gatherings. By adding variables that capture the occurrence, type, and expected attendance of scheduled events, models can better anticipate the extreme fluctuations associated with large-scale events. Furthermore, developing probabilistic methods to quantify forecast uncertainty and evaluate the potential effect of spatial autocorrelation, especially in complex beach environments, could enhance model performance. Creating models that dynamically adjust to real-time data inputs could improve forecasting accuracy and responsiveness.

In conclusion, this study validates the effectiveness of machine learning and deep learning for multi-horizon crowd forecasting in dynamic beach environments. Based on the findings, a strategy is recommended that balances

precision with operational efficiency. For implementations where computational resources and maintainability are the priorities, a unified multilocation model is a robust, scalable solution. However, for critical locations requiring maximum precision, individual models (specifically CatBoost for short-term and LightGBM or LSTM for longer horizons) are recommended. Adopting a data-driven, multimodel framework provides a powerful tool to enhance public safety, optimize resource allocation, and improve the overall visitor experience at popular tourist destinations such as Scheveningen Beach.

Acknowledgments

We would like to acknowledge Dr. Yanan Xin, Assistant Professor, Department of Transport and Planning, TU Delft, for her valuable assistance in reviewing the manuscript during the revision process. We acknowledge the use of GPT-4o LLM for refining the text in the Introduction, Methodology, and Results sections.

Author Contributions

The authors confirm contributions to the paper as follows: study conception and design: T. Hari, W. Daamen, S. Hoogendoorn-Lanser, S.P. Hoogendoorn; data collection: J. Steenbakkers, S. Hoogendoorn-Lanser; analysis and interpretation of results: T. Hari, W. Daamen, S.P. Hoogendoorn; draft manuscript preparation: T. Hari, W. Daamen, S.P. Hoogendoorn. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was funded by the EMERALDS project, which has received financial support from the European Union's Horizon Europe program (Grant Agreement No. 101093051).

ORCID iDs

Theivaprakasham Hari https://orcid.org/0000-0002-0603-8830
Winnie Daamen https://orcid.org/0000-0003-4518-4184
Jeroen Steenbakkers https://orcid.org/0009-0002-5371-3672
Serge Paul Hoogendoorn https://orcid.org/0000-0002-1579-1939

Data Accessibility Statement

The data sets and code used in this study are available from the corresponding author on request.

Supplemental Material

Supplemental material for this article is available online.

References

1. Feliciani, C., A. Corbetta, M. Haghani, and K. Nishinari. Trends in Crowd Accidents Based on an Analysis of Press Reports. *Safety Science*, Vol. 164, 2023, p. 106174. https://doi.org/10.1016/j.ssci.2023.106174.
2. Zomer, L. B., W. Daamen, S. Meijer, and S. P. Hoogendoorn. Managing Crowds: The Possibilities and Limitations of Crowd Information During Urban Mass Events. In *Planning Support Systems and Smart Cities* (G. Stan, J. Ferreira, Jr., R. Goodspeed, and J. Stillwell, eds.), Lecture Notes in Geoinformation and Cartography. Springer International Publishing, Cham, 2015, pp. 77–97. https://doi.org/10.1007/978-3-319-18368-8_5.
3. Subramanian, G. H., and A. Verma. Crowd Risk Prediction in a Spiritually Motivated Crowd. *Safety Science*, Vol. 155, 2022, p. 105877. https://doi.org/10.1016/j.ssci.2022.105877.
4. Earl, C., E. Parker, and M. F. Capra. The Management of Crowds and Other Risks at Outdoor Music Festivals: A Review of the Literature. *Environmental Health*, Vol. 5, No. 1, 2005, pp. 34–49.
5. Yu, R., Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu. Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting. In *Proc., 17th SIAM International Conference on Data Mining, SDM 2017* (C. Nitesh, and W. Wang, eds.), Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017, pp. 777–785. https://doi.org/10.1137/1.9781611974973.87.
6. Novak, H., F. Bronić, A. Kolak, and V. Lešić. Data-Driven Modeling of Urban Traffic Travel Times for Short- and Long-Term Forecasting. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 24, No. 10, 2023, pp. 11198–11209. https://doi.org/10.1109/TITS.2023.3287980.
7. Chatziantoniou, I., S. Degiannakis, B. Eeckels, and G. Filis. Forecasting Tourist Arrivals Using Origin Country Macroeconomics. *Applied Economics*, Vol. 48, No. 27, 2016, pp. 2571–2585. https://doi.org/10.1080/00036846.2015.1125434.
8. Zhang, Y., T. Cheng, Y. Ren, and K. Xie. A Novel Residual Graph Convolution Deep Learning Model for Short-Term Network-Based Traffic Forecasting. *International Journal of Geographical Information Science*, Vol. 34, No. 5, 2020, pp. 969–995. https://doi.org/10.1080/13658816.2019.1697879.
9. Asher, M., Y. Oswald, and N. Malleson. Predicting Pedestrian Counts Using Machine Learning. *AGILE: GIScience Series*, Vol. 4, 2023, pp. 1–6. https://doi.org/10.5194/agile-giss-4-18-2023.
10. Krishnakumari, P., S. Hoogendoorn-Lanser, J. Steenbakkers, and S. Hoogendoorn. Crowd Safety Manager: Towards Data-Driven Active Decision Support for Planning and Control of Crowd Events. 103rd TRB Annual Meeting, Washington, D.C., 2024. https://doi.org/10.48550/arXiv.2308.00076

11. Qu, D., K. Chen, S. Wang, and Q. Wang. A Two-Stage Decomposition-Reinforcement Learning Optimal Combined Short-Time Traffic Flow Prediction Model Considering Multiple Factors. *Applied Sciences*, Vol. 12, No. 16, 2022, p. 7978. https://doi.org/10.3390/app12167978.
12. Tian, Y., K. Zhang, J. Li, X. Lin, and B. Yang. LSTM-Based Traffic Flow Prediction with Missing Data. *Neurocomputing*, Vol. 318, 2018, pp. 297–305. https://doi.org/10.1016/j.neucom.2018.08.067.
13. Yang, B., S. Sun, J. Li, X. Lin, and Y. Tian. Traffic Flow Prediction Using LSTM with Feature Enhancement. *Neurocomputing*, Vol. 332, 2019, pp. 320–327. https://doi.org/10.1016/j.neucom.2018.12.016.
14. Kong, D., and F. Wu. HST-LSTM: A Hierarchical Spatial-Temporal Long-Short Term Memory Network for Location Prediction. *Proc., 27th IJCAI International Joint Conference on Artificial Intelligence*, Stockholm Sweden, AAAI Press, 2018, pp. 2341–2347. https://doi.org/10.24963/IJCAI.2018/324.
15. Yu, B., H. Yin, and Z. Zhu. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *Proc., 27th IJCAI International Joint Conference on Artificial Intelligence*, Stockholm Sweden, AAAI Press, 2018, pp. 3634–3640. https://doi.org/10.24963/ijcai.2018/505.
16. Chen, J., X. Wang, and X. Xu. GC-LSTM: Graph Convolution Embedded LSTM for Dynamic Network Link Prediction. *Applied Intelligence*, Vol. 52, No. 7, 2022, pp. 7513–7528. https://doi.org/10.1007/s10489-021-02518-9.
17. Lim, B., S. Ö. Arık, N. Loeff, and T. Pfister. Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting. *International Journal of Forecasting*, Vol. 37, No. 4, 2021, pp. 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012.
18. Alis, C., E. F. Legara, and C. Monterola. Generalized Radiation Model for Human Migration. *Scientific Reports*, Vol. 11, No. 1, 2021, p. 22707. https://doi.org/10.1038/s41598-021-02109-1.
19. Chen, T., and C. Guestrin. XGBoost: A Scalable Tree Boosting System. *Proc., 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, Association for Computing Machinery, New York, NY, 2016, pp. 785–794. https://doi.org/10.1145/2939672.2939785.
20. Prokhorenkova, L., G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. CatBoost: Unbiased Boosting with Categorical Features. *Proc., 32nd International Conference on Neural Information Processing Systems*, Curran Associates, Inc., Montréal, Canada, 2017, pp. 6639–6649. https://dl.acm.org/doi/10.5555/3327757.3327770.
21. Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems*, Vol. 30, 2017, pp. 3146–3154. https://dl.acm.org/doi/10.5555/3294996.3295074.
22. Hochreiter, S., and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, Vol. 9, No. 8, 1997, pp. 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.
23. Akiba, T., S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. *Proc., 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, AK, Association for Computing Machinery, New York, NY, 2019, pp. 2623–2631. https://doi.org/10.1145/3292500.3330701.