Evaluating Crowd Flow Forecasting Algorithms for Indoor Pedestrian Spaces
A Benchmark Using a Synthetic Dataset

Mai, Weiming; Duives, Dorine; Krishnakumari, Panchamy; Hoogendoorn, Serge

DOI
10.1109/TITS.2025.3559353

Publication date
2025

Document Version
Final published version

Published in
IEEE Transactions on Intelligent Transportation Systems

Citation (APA)
Mai, W., Duives, D., Krishnakumari, P., & Hoogendoorn, S. (2025). Evaluating Crowd Flow Forecasting Algorithms for Indoor Pedestrian Spaces: A Benchmark Using a Synthetic Dataset. *IEEE Transactions on Intelligent Transportation Systems*, *26*(6), 7953-7968. https://doi.org/10.1109/TITS.2025.3559353

Evaluating Crowd Flow Forecasting Algorithms for Indoor Pedestrian Spaces: A Benchmark Using a Synthetic Dataset

Weiming Mai, Dorine Duives, Panchamy Krishnakumari, and Serge Hoogendoorn

***Abstract*— Crowd management plays a vital role in urban planning and emergency response. Accurate crowd prediction is important for venue operators to respond effectively to adverse crowd dynamics during large gatherings. Although many studies have tried to predict crowd densities or movement dynamics with data-driven predictive models, their validation is often limited to data within the same scenario. As a result, the predictability of the data-driven model in unseen scenarios, such as evacuation scenarios, remains unknown due to the challenges of collecting out-of-distribution data regarding emergency conditions. To address this problem, we present an evaluation pipeline to evaluate different kinds of data-driven models. A method is proposed to generate realistic scenarios by simulation and collect synthetic data from these scenarios to acquire a comprehensive dataset. With these synthetic data, we evaluated different predictive models, from traditional machine learning methods to deep learning time-series prediction models, to explore their generalizability. Furthermore, we propose a weighted average metric, which is better suited to determine the performance of forecasting algorithms under adverse conditions. Through extensive experimentation, we showcase the heterogeneity and diversity of the simulation dataset. The evaluation results also revealed that all the data-driven models performed poorly in unseen scenarios, highlighting the urgent need to develop a robust and generalizable model for predicting crowd flow in indoor spaces.**

***Index Terms*— Simulation modeling, crowd flow prediction, data-driven methods.**

I. Introduction

In INDOOR spaces, such as train stations, airports, and shopping malls, the dynamics of the crowd can change instantaneously due to factors such as alterations in train schedules or the occurrence of significant events like evacuations or large gatherings. Grasping the intricacies of crowd movement dynamics and forecasting movement dynamics can prove vital in ensuring crowd safety. There are two primary approaches to predict crowd movement in indoor environments. The traditional method involves constructing offline simulation models to analyze crowd behavior and dynamics [1], [2], [3]. In order to analyze pedestrian movement dynamics, a myriad of simulation models have been developed that can simulate the walking dynamics of crowds. The Social Force Model (SFM) [4] is one of the representative models that focuses on microscopic pedestrian behavior. It is based on the principles of social interactions and the forces that influence individuals' movement in a crowd. The Cellular Automata (CA) [5] model is used to simulate complex systems consisting of a grid of cells that interact with their neighboring cells based on predefined rules. Macroscopic models focus on overall crowd movement and treat pedestrians as flows, which takes less computational time. The continuum models [6], [7] describe the behavior of a system as a continuous field or distribution rather than individual entities or particles. The network flow model [8] is another type of macroscopic crowd movement model. However, all these simulation models are designed to answer a what-if question rather than real-time prediction with the data observed by the sensors [9].

Instead of explicitly modeling the physical process, data-driven models can learn latent behavioral patterns from historical data to predict future crowd flow. Data-driven approaches can generally be categorized into video-based and non-video-based methods [10]. Video-based methods employ video datasets [11] for crowd counting or density estimation, they utilize object detection techniques [12], [13], [14] to predict the individual motion behavior of the pedestrian. On the other hand, non-video-based methods focus primarily on human trajectory prediction [15], [16]. Some approaches utilize Wi-Fi localization [17], [18] or Global Positioning System (GPS) data [19], [20] to identify high-density crowd areas. Additionally, Tordeux et al. [21] explored the use of artificial neural networks (ANNs) to predict microscopic crowd dynamics.

However, data-driven models rely on substantial amounts of data to effectively capture the inherent dynamics of mobility patterns. In reality, the availability of recorded data under extreme conditions is often severely limited, and such data typically lie outside the distribution of normal scenarios. Despite the success of all these fancy prediction models, their resiliency to abrupt scenarios remains unknown. Consequently, the use of simulators to generate data to train and test the models becomes crucial. Simulators offer the advantage of being controllable, enabling us to simulate various scenarios that might not be present in the limited real-world data. In [22], synthetic crowd datasets are generated using agent-based

Received 15 March 2024; revised 7 November 2024 and 24 January 2025; accepted 17 March 2025. Date of publication 16 April 2025; date of current version 2 June 2025. The Associate Editor for this article was R. Meneguette. *(Corresponding author: Weiming Mai.)*

The authors are with the Department of Transportation and Planning, Delft University of Technology, 2628 Delft, The Netherlands (e-mail: w.m.mai@tudelft.nl; d.c.duives@tudelft.nl; p.k.krishnakumari@tudelft.nl; s.p.hoogendoorn@tudelft.nl).

Digital Object Identifier 10.1109/TITS.2025.3559353

simulation for predicting crowd severity levels. The simulation output includes the agents' positions, speeds, and headings, which are then processed to obtain crowd density, speed, and heading direction as inputs for the prediction model. The particle filter simulation method is also studied in [9] and [23]. It is used for crowd-state prediction, using observation data to estimate latent parameters for real-time predictions. While both studies employ simulation tools for crowd analysis and prediction, they only focus on small indoor areas and require detailed data, such as agent speed, density, or precise positions, which are difficult to obtain in real-world scenarios.

In this work, we focus on evaluating data-driven models that predict crowd movement using pedestrian flow (count) data from sensors. Flow data is directional, making it a better representation of crowd movement than speed or density. In addition, it is easier and more accurate to collect. To assess model performance, we generate synthetic sensor flow data through simulation. This dataset not only addresses the lack of publicly available data but also serves as a key component of our evaluation framework for flow prediction models in indoor spaces. Source codes of the synthetic dataset and the pedestrian simulation model we used are publicly available.[1]

Our contribution could be summarized as follows:

- We design a set of conditions to reproduce realistic indoor crowding scenarios via microscopic pedestrian simulation. With this approach, we created a synthetic dataset encompassing a wide range of adverse conditions. To validate the quality of this dataset, we analyze the origin-destination (OD) matrix and the macroscopic fundamental diagram (MFD) of different scenarios to showcase the heterogeneity and diversity of the demand in the dataset. Quantitatively, the diversity of the data among these scenarios is evaluated by the prediction error of a data-driven predictor.
- We propose an evaluation pipeline to assess the generalizability of data-driven prediction models across various crowd dynamics. To ensure fair evaluation under different conditions, we introduce a safety-concerned metric that measures prediction error considering the quantile of the data. This data-centric pipeline allows researchers to easily evaluate predictive models in diverse scenarios.
- Using the synthetic datasets, we re-evaluate different prediction models from traditional machine learning methods, e.g. Multivariate Linear Regression (MLR) and Gradient Boosting Decision Tree (GBDT), to deep learning time-series predictive models, e.g. Recurrent Neural Networks (RNN) and Graph Neural Network (GNN) in the out-of-distribution (OOD) setting. The experimental results also affirm the robustness of the proposed metric.

The outline of this paper is as follows. Section II reviews various traffic and crowd flow prediction methods. Section III formally defines the indoor crowd flow prediction problem in the context of emergency management and outlines the process of simulation and synthetic dataset generation. Section IV introduces the predictive AI models used in our study and presents the evaluation pipeline diagram. Finally, Section VI discusses future research directions.

II. Related Work

A. Data-Driven Mobility Forecasting

With the emergence of machine learning and deep learning techniques, a lot of research studies how to develop a model to forecast or estimate the movement patterns and behaviors of individuals or vehicles across an entire network or transportation system [24]. It involves analyzing historical data, current conditions, and various factors influencing future mobility patterns. Compared to simulation techniques, data-driven approaches require more data and computational complexity. Benefits from the large amount of data, the models don't require making strict mathematical assumptions and go through a validation and calibration process.

Extensive research has been devoted to traffic prediction. Conventional statistical models such as history average (HA), auto-regressive-moving-average (ARIMA) [25], and vector auto-regression (VAR) have been widely used for time series traffic flow prediction. These models are limited to single target point prediction and are not suitable for network-wide mobility prediction. With the advent of complex model structures, e.g., convolutional neural network and graph neural network, researchers have developed different deep learning models to capture the complex dynamics of vehicular mobility [26], [27], [28] and human mobility [29], [30], [31]. These models aim to approximate the intricate dynamics function involved in predicting network-wide mobility patterns, mostly are in the city level.

B. Indoor Crowd Monitoring and Prediction

Some researchers have studied real-time monitoring to send out warnings of overcrowding. For instance, in the work by Zhang et al. [32], they develop a crowd management system by crowd density estimation. They propose a risk rating system and an early warning mechanism to manage the crowd. Similarly, Martani et al. [12] study monitoring techniques and apply them to pedestrian microsimulations to achieve crowd flow prediction. With the rise of deep learning, many studies have leveraged computer vision techniques [13], [33], [34] for crowd counting to enhance monitoring systems. The intrinsic drawback of monitoring is that they can only perceive the things that have happened, which lacks a proactive approach and does not offer a comprehensive solution.

Data-driven techniques have been widely applied to predict the crowd state proactively. In [34], the authors combined deep learning techniques and domain knowledge for predicting inbound and outbound metro passenger flow. Sudo et al. [18] make use of deep learning techniques to predict the crowd density with a Wi-Fi dataset in eight venues of an indoor environment. However, these works focus on a coarse time horizon prediction, in which the time interval is at least 10 minutes. Therefore they do not apply to instant indoor crowd movement prediction.

In a finer time granularity, Zhang et al. contribute a large-scale video dataset WORLDEXPO'10 [11] for crowd

[1] https://github.com/WaimenMak/Crowd-Prediction

counting or density estimation, and [35], [36] use human-trajectory datasets: ETH [37] and UCY [38] to predict individual pedestrian movement. The prediction of the video-based methods is highly dependent on the quality of the video, and these methods are not robust enough for large-scale crowd estimation and are restricted to a specific place where the camera can cover. Some used different non-video-based data e.g., GPS and Bluetooth for the crowd tracking [10]. And yet, with all these progressive works mentioned above for traffic prediction or indoor crowd prediction, there remains a research gap concerning the reliability of the prediction model in unseen scenarios. The prediction accuracy of the data-driven model could not be guaranteed without testing in a comprehensive dataset. Consequently, simulation is needed to generate diverse scenarios for evaluating these methods to thoroughly test data-driven methods.

III. Data-Driven Indoor Pedestrian Flow Forecasting and Synthetic Data Development

A. Research Approach

The state-of-the-art above identifies that there are two reasons why we cannot forecast flows in a building during adverse events at the moment, being 1. we do not have forecasting models that are trained for adverse conditions and, 2. There is no dataset that contains sufficient adverse event conditions to train. This research will benchmark existing (flow) forecasting techniques concerning their ability to forecast flow during adverse conditions. To do so, we develop a new evaluation pipeline. First, we create a synthetic dataset that features pedestrian flows in an indoor environment during normal and adverse conditions. This synthetic dataset is accordingly used to train existing flow forecasting models. Lastly, a new evaluation method is developed that specifically aims to assess models concerning their abilities to forecast flow under adverse conditions.

Underneath, first, a mathematical definition of the flow forecasting problem is provided. Accordingly, the simulation scenario variables are defined. Third, the Safety-concerned metric is introduced. This section ends with a brief introduction to the comprehensive evaluation pipeline.

B. Mathematical Definition of the Flow Forecasting Problem

Pedestrian flows are characterized by three fundamental traffic flow variables: density (ρ), speed (v), and flow (f) [3]. Density is defined as the number of pedestrians per unit of area at a certain moment. Speed is defined as the distance traveled by a person per unit of time. The crowd flow refers to the number of people crossing a line (i.e., cross-section) within a certain period. Here, both velocity and flow are vectors. To account for the directionality of the flow, we define the flow as inflow (Blue arrow in Figure 1 - direction towards the center of the building), and outflow (Orange arrow in Figure 1 - direction towards the nearest exit of the building). In practice, the flow data is collected by sensors located at the cross-sections shown in the figure. Please note that we record the flow in the corridors to reduce the directionality of the flow at a cross-section to II directions.

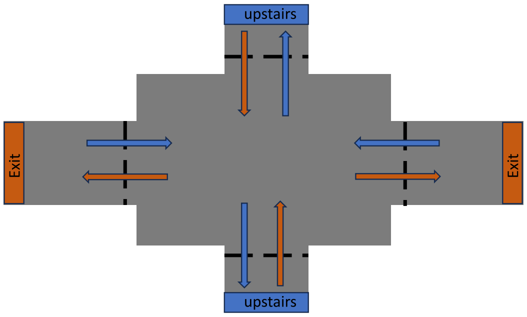


Fig. 1. Visualization of the flows in an intersection of a building.

In this paper, we focus on forecasting the flow rate. This crowd flow forecasting problem can be defined as: Given the T' lags of historical flow data $\mathbf{X}$ to forecast the future T horizons.

$$[\mathbf{X}^{t-T'+1}, \cdots, \mathrm{X}^{t}] \xrightarrow{\mathrm{f}(\cdot)} [\mathbf{X}^{t+1}, \cdots, \mathbf{X}^{t+T}].$$

where $\mathrm{f}(\cdot)$ is the function approximator that maps the historical time series signals to T prediction horizons. To capture the movement of the crowd throughout the infrastructure, there have been some works [8], [39] model the whole building as a network, in which rooms are vertexes and the corridors are edges. Different from these studies, we deploy the sensors in the corridors evenly, from that we then obtain a sensor network that monitors the spatial flow rates. The data collected from multiple sensors can be effectively represented as a graph signal denoted as $\mathbf{X} \in \mathbb{R}^{N \times C}$. Here, N represents the number of sensors, while C corresponds to the feature dimension of the data recorded by each sensor. In our simulation setup, each sensor is capable of recording bi-directional pedestrian flow rates, namely f_{in} and f_{out}. By summing these two flow values, we can obtain the overall flow f_{total}. Therefore, in our setting, the feature dimension C is equal to 3 since it encompasses the bidirectional flow components (f_{in}, f_{out}), as well as the combined overall flow (f_{total}).

C. Developing a Synthetic Crowd Flow Dataset

To create the synthetic dataset featuring sensor data we utilize a microscopic pedestrian simulation model. We run a wide variety of scenarios with varying conditions. During each simulated scenario, we capture the flow at a wide variety of sensor locations. The flow rate records form the synthetic dataset.

1) Introducing the Synthetic Dataset: We use the microscopic pedestrian dynamics simulation package **Pedestrian Dynamics**[2] to simulate the movement dynamics in the station, the underlying model used for this agent-based simulation is the Social Force model. Figure 2 shows the 2D view of the train station model, where the numbered yellow blocks represent the sensor locations. *Floor 1* represents the main hall of the station on the ground floor. *Floor 2* represents the

[2] https://www.incontrolsim.com/our-software/

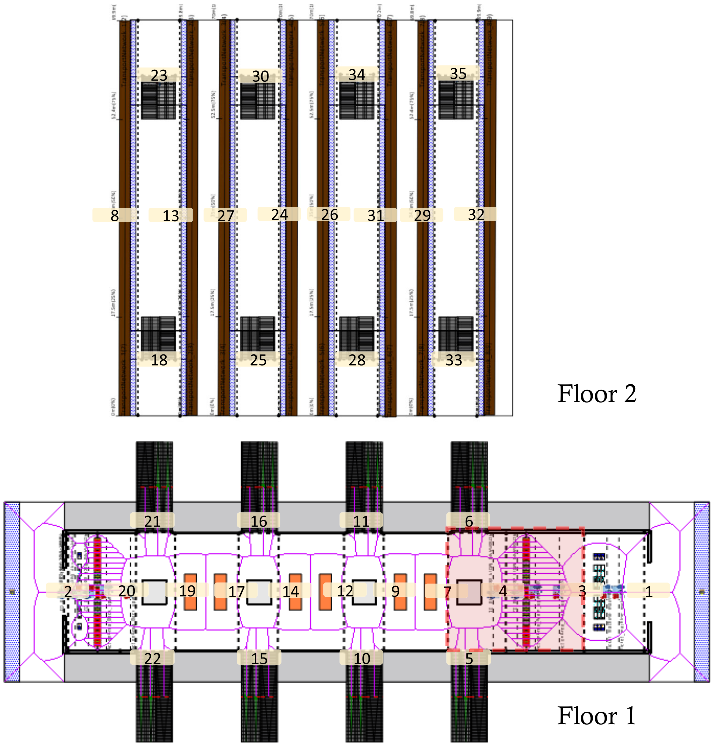


Fig. 2. 2D view of the train station.

train station platforms located on the first floor. Escalators and stairs connect Floor 1 and Floor 2. As shown in the figure, there are 8 train tracks. In our setting, there are a total of 4 lines and each train belongs to one of the lines.

Within the station, we designed a sensor network, featuring 35 sensors. The sensor in the simulation is the abstraction of all kinds of devices that capture pedestrian flow. In practice, the visual data collected by the camera can be used to estimate pedestrian counts using computer vision techniques. To ensure consistency, the time granularity of all sensors is set to 10s. Hence, the unit of the flow rate is peds/10s. Each sensor position is then treated as a node in the network. For each scenario, the simulation time is fixed at 1 hour, resulting in a total of 360 data points for the time-series. Under the setup described in [26], the time series data is divided into multiple samples using sliding windows. In this approach, the window width is consistently set to 24 time units. Thus, the input of the model can be denoted as $X = \{\mathbf{X}^{t-m}\}_{m=0:11}$, which is the previous 2-mins historical flow data. And the output of the model is $Y = \{\mathbf{X}^{t+n}\}_{n=1:12}$, which corresponds to the flow rate at the sensor location for the coming 2 minutes. In practice, the prediction horizon is determined according to the average travel time from the main entrance to the back exit.

In traditional time-series data prediction, the overall dataset is continuous and not time-correlated, thus the training and testing dataset can be obtained according to the given proportion. In contrast, in our study, we sample data from different scenarios to construct the training dataset and testing dataset. The testing is scenario-specific, with each scenario having its own set of parameters.

2) Scenario Development: In this section, we describe the process of developing scenarios for the synthetic dataset. In general, millions of scenarios can unfold in an indoor environment, with diverse infrastructures giving rise to distinct situations. For this study, we pick a train station as a case study. In our agent-based simulation model, agents primarily originate from two sources, from outside the train station and from inside the trains. There are 2 main normal activity routes for these agents: the agents from outside the station follow the route $Entry \rightarrow Wait \rightarrow Board$. To mimic the real scenario, there are also agents from outside who just walk through the station from one entrance to another exit $Entry \rightarrow Transfer \rightarrow Exit$. For the agents alighting from the train, their normal activities are $Alight \rightarrow Wait \rightarrow Transfer$ or $Alight \rightarrow Exit$. More detailed activity routes can be found in Table VI.

No day is the same in a train station. Therefore, we introduce variance in the scenarios. Table I summarize different exemplary conditions considered in our simulation. First, we consider the influence of passengers, i.e., the individuals boarding the transportation system. In particular, we consider the arrival frequency of the passenger, the route choice, and the walking speed. By setting the boarding probability of each line, we can generate different OD demands in the train station. The last variable in passenger represents the stochastic nature of pedestrian behavior, featuring, for instance, alternative route choice or unpredictable sudden movements. This variable introduces variability and noise into the dataset, reflecting real-world scenarios where pedestrian behavior deviates from the expected pattern. Secondly, we examine the impact of the

TABLE I

SCENARIOS GENERATION: *Factors* REPRESENT THE POTENTIAL FACTORS THAT INFLUENCE THE MOVEMENT OF THE CROWD. *Variables* ARE THE PARAMETERS OF EACH FACTOR THAT COULD BE TUNED TO GENERATE VARIOUS CONDITIONS. AND THE SCENARIO IS THE COMBINATION OF THE CONDITIONS

Factors	Variables	Conditions
Passenger	Arrival Rate	1. Rush hour 2. Off-peak period
	Demand	3. High demand for Line X 4. Low demand for Line X 5. Normal demand for Line X
	Pedestrian Speed	6. Low throughput 7. High throughput
	Pedestrian Activities	8. Alight/Board 9. Wait 10. Transfer
Transportation	Train Schedule	11. Different train schedules 12. Train X is delayed for T mins
Evacuation	Emergency Route	11. Evacuate from the main and back exit 13. Evacuate from the platform exit
Disruption	System Malfunctioning	14. Escalator E is malfunctioning

transportation system itself, such as the train schedule, whether the train is delayed, and for how long it will take.

In addition, two factors are included to simulate abnormal events, being an *Evacuation*, and a *Infrastructure disruption*. These two disruptions can significantly impact the mode choice and route choice [2] of the pedestrians, which creates large, very sudden changes in the sensor data. In the case of *Evacuation*, an emergency occurs, requiring a complete evacuation of all pedestrians from the station, leading to a complete alteration of their route towards a designated emergency exit (i.e., the nearest one). On the other hand, *Disruption* simulates infrastructure problems within the train station. In this case, we included a malfunctioning escalator.

We simulate a wide variety of scenarios by tuning the variables of each factor. This approach enables us to explore the effects of these factors on crowd flow and evaluate the models' performance in handling normal and abnormal scenarios. Many more conditions can be added to the synthetic dataset. Yet, the current synthetic dataset is sufficient for the purpose of benchmarking crowd flow forecasting models.

3) Parameter Settings for the Conditions: This section introduces the detailed parameter settings of different exemplary conditions. In each scenario, the agent generator generates a group of people at each time interval. The number of people in this group follows a uniform distribution $U(2, 5)$, meaning that 2 to 5 people will be in this group. The time interval is a random variable following an exponential distribution, such that the arrival of the passenger groups follows a Poisson distribution. We set the mean interval $s = 2$ for the rush hour condition, which means that on average every 2 seconds a group of passengers would arrive at the station. For off-peak hours, we set $s = 4$.

The demand for each train line is represented by the probability that passengers will choose that line. We categorize demand levels as high, normal, and low. Suppose there are four train lines, when the demand for Line 1 is high, the probability distribution is $[52\%, 16\%, 16\%, 16\%]$. For normal demand, it is $[25\%, 25\%, 25\%, 25\%]$, and for low demand, it is $[10\%, 30\%, 30\%, 30\%]$. The maximum walking speed follows a triangular distribution; in a high-throughput condition, the distribution is $Triangular(2, 1.5, 2.5)$, meaning that the average maximum walking speed of each agent is $2(m/s)$, the highest maximum walking speed is $2.5(m/s)$, and the lowest value is $1.5(m/s)$. The maximum walking speed follows the distribution $Triangular(1.35, 0.8, 1.75)$ in a normal condition. For all agents, the minimum walking speed is $0.06(m/s)$. More details of the agent profile are introduced in the Appendix.

In an evacuation scenario, the evacuation process starts at some point during the simulation and continues until all people are evacuated from the building. In our setting, we mainly change emergency routes as described in the table, while the other variables remain unchanged.

D. Data-Driven Predictive Models

We adopt a wide variety of prediction models ranging from traditional statistics-based methods to deep learning methods. Since there are currently no SOTA data-driven models specifically designed for crowd flow prediction, most of these models are data-driven and have been widely used in time-series prediction and widespread application in traffic or crowd forecasting. However, they have not yet been benchmarked in the context of indoor crowd flow prediction. The models are introduced as follows.

- **MLR**: *Multivariate Linear Regression* take multiple variables as input for prediction. We implement this model with the python package *scikit-learn*.[3]
- **VAR**: *Vector AutoRegression* is a forecasting algorithm that can be used when two or more time series influence each other. In our experiment, we tried different lags as input and found that in our dataset the model with a lag of length 3 (30s) performs the best.
- **MLP**: *Multi-layer Perceptron* which is a fully-connected artificial neural network. We set the hidden layers as 2, and in each layer, there are 50 neurons. This is also implemented based on *scikit-learn*.

[3] https://scikit-learn.org

- **XGBOOST**: *eXtreme Gradient Boosting* [40] is a scalable end-to-end tree boosting system, which is widely used by data scientists for many machine learning tasks. The early stop round is set to be 10 to prevent overfitting and RMSE is adopted as the evaluation metric.
- **N-BEATS** *Neural Basis Expansion Analysis for Time Series Forecasting* [41] is a deep learning model designed for time series forecasting. It uses a basis expansion method that allows for interpretable forecasts, offering insights into the underlying patterns in the data.
- **RNN**: *Recurrent Neural Network* is a mature and widely used deep learning model for time series or sequential data forecasting. We chose the Gated Recurrent Unit (GRU) as the model structure in this study. There are two RNN layers with 64 hidden units and one linear layer for the output. Note that in the experiment, we try two types of output format for multistep prediction: one vector output and a multi-step output in an autoregressive way, i.e., *Seq2seq* [42].
- **GAT-GRU**: *Graph Attention Network* [43] is a graph network that uses the attention mechanism to aggregate adjacent information. We combine the GRU with GAT to extract the spatial-temporal representation of the data.
- **DCRNN**: *Diffusion Convolutional Recurrent Neural Network* [26], a graph-based deep learning model for traffic forecasting that incorporates both spatial and temporal dependency in the traffic flow. It utilizes the spatial topology construction which is an adjacent matrix that captures the spatial dependency. Here, we adopt a binary graph [28]. The elements of the matrix are 1 or 0, where 1 identifies that two sensors are first-order neighbors.
- **STGCN**: *Spatio-Temporal Graph Convolutional Network* [44] uses graph convolutional layers to aggregate information from neighboring nodes and uses the convolutional layer to extract the temporal features.

E. Safety-Concerned Metric

Time-series crowd flow data often display frequent irregularities and sudden bursts, with data distributions frequently departing from a Gaussian pattern and showing considerable skewness. Typically, the majority of flow counts are zero or very low, with only a few instances of high counts. In the context of pedestrian flow prediction, commonly used time series prediction metrics are Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The definition of MAE and RMSE could be described by the following equations:

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

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

where N is the number of data samples, y_t is the ground truth value, and $\hat{y}_t$ is the prediction. These two metrics focus on evaluating the average prediction error of a model. However, in pedestrian flow prediction, the primary concern is whether the model can accurately predict large crowding moments, rather than predicting the average performance of the system. Hence, even if a model may not perform optimally during average conditions if it demonstrates the ability to capture crowding instances effectively, it is still considered a valuable prediction model. Figure 3 shows a case study and confirms our previous intuition, namely in both average metrics, Model 1 performs better than Model 2. However, the quantile loss of Model 1 is larger than Model 2, which means that Model 2 can better predict the 0.9 quantile of the flow data. Therefore, Model 2 has less underestimation on the crowding part framed by the red boxes in the figure.

To alleviate this issue, we introduce a composite metric that simultaneously considers average performance and edge performance (i.e., the error of the predicted quantiles). The first component of the new composite metric is designed to measure the average error between predictions and actual values. For instance, the MAE can be adopted for this purpose, serving as an unbiased estimator of the expected prediction error. On the other hand, RMSE can be employed, introducing a penalty factor for large differences between prediction and ground truth.

The second component considers the error between the prediction and the chosen quantile of the ground truth, which may correspond to the higher flow in the data. Therefore, we use the quantile loss as the second measurement:

$$L_\rho(y_t, \hat{y}_t) = 2|\hat{y}_t - y_t|(\rho\mathbb{I}_{\hat{y}_t < y_t} + (1-\rho)\mathbb{I}_{\hat{y}_t \geq y_t}). \tag{3}$$

Here, ρ represents the quantile, when the prediction $\hat{y}_t$ is equal to the ρth quantile of the ground truth distribution, the loss function could be minimized [45]. To reduce the noise of the data, and make the metric more robust, we adopt the $\rho - risk$ metric proposed in [46], with slight modifications tailored to suit our problem. In this paper, the $\rho - risk$ is defined as:

$$R^\rho[T;S] = |T|^{-1}\sum_{t\in T} L_\rho(Z_{t;S}, \hat{Z}_{t;S}). \tag{4}$$

where T denotes the number of training samples after the sliding window process. The aggregated count within a period is calculated by: $Z_{t;S} = \sum_{s=0}^{S-1} y_{t+s}$. Here, s denotes the prediction horizon of s steps and S is the time span. This summing process reduces the noise of the data and the sum value represents, in practice, the total flow in the future period S, which is more meaningful than the value of a single time step.

Finally, the weighted average error (WAE), featuring both components, can be described as the following equation:

$$WAE = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\left[(1-\gamma_i)L_{avg} + \gamma_i R^\rho[T;S]\right]. \tag{5}$$

Here, WAE is the weighted sum of the error L_{avg} and $\rho - risk$ average on each sensor i ($\mathcal{I}$ is the set of sensors/nodes). When L_{avg} is chosen to be MAE and the time span S is set to 1, the value of WAE represents the expected deviation between the prediction $\hat{Y}$ and the quantile $Q_\tau(Y|X)$ in terms of pedestrian flow (peds/10s). It is equivalent to $\tau - quantile$ loss. It can be proven that $\tau = \frac{1-\gamma+2\gamma\rho}{2}$, the proof is presented

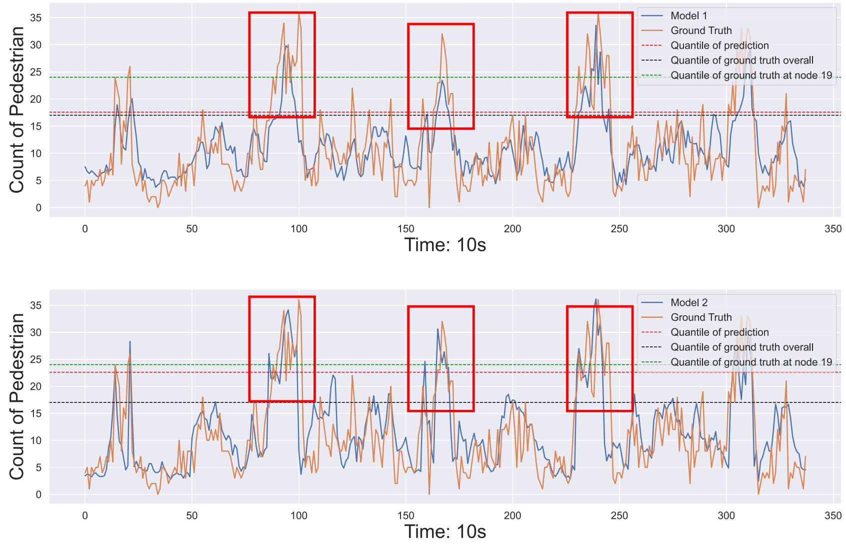


Fig. 3. Evaluation on different metrics: The figures show the prediction result of two models Model 1 (RNN) and Model 2 (DCRNN) at the $18th$ sensor. The MAEs are 4.15 $(peds/10s)$ (top) and 4.34 $(peds/10s)$ (bottom), respectively. And the RMSEs are 5.39 $(peds/10s)$ (top) and 6.04 $(peds/10s)$ (bottom). The quantile losses are 10.17 $(peds/10s)$ (top) and 8.60 $(peds/10s)$ (bottom) respectively.

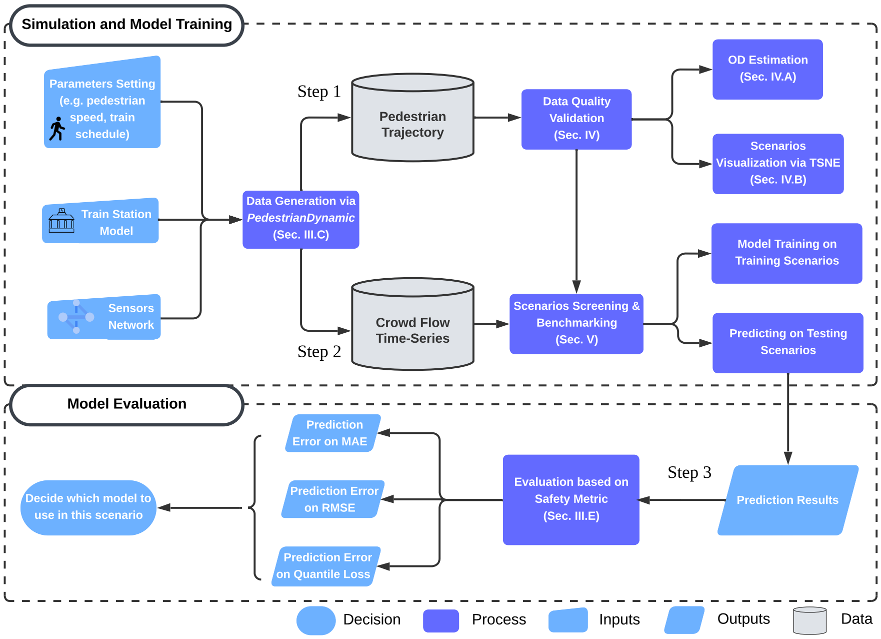

Simulation and Model Training
Parameters Setting (e.g. pedestrian speed, train schedule)
Train Station Model
Sensors Network
Data Generation via *PedestrianDynamic* (Sec. III.C)
Step 1
Pedestrian Trajectory
Step 2
Crowd Flow Time-Series
Data Quality Validation (Sec. IV)
OD Estimation (Sec. IV.A)
Scenarios Visualization via TSNE (Sec. IV.B)
Scenarios Screening & Benchmarking (Sec. V)
Model Training on Training Scenarios
Predicting on Testing Scenarios
Model Evaluation
Prediction Results
Step 3
Evaluation based on Safety Metric (Sec. III.E)
Prediction Error on MAE
Prediction Error on RMSE
Prediction Error on Quantile Loss
Decide which model to use in this scenario
Decision
Process
Inputs
Outputs
Data

Fig. 4. Evaluation pipeline.

in the appendix. When equal importance is given to both average and edge performance, i.e., γ and ρ are set to 0.5, with L_{avg} chosen as MAE. In this case, τ equals 0.5, reducing the weighted average metric in Equation 5 to MAE. This illustrates the flexibility of the metric without sacrificing generality.

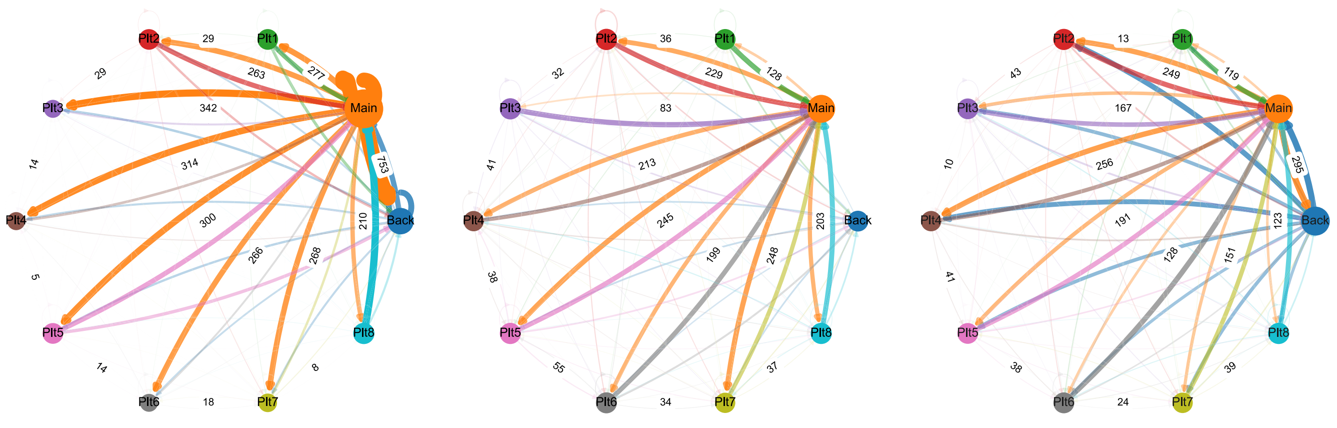
The next question is how to define the weight γ_i for each node. Since crowd flow patterns vary across sensors, the focus at each location may differ. For example, in crowded areas, we focus on prediction accuracy in the peak. In sparse areas, we focus on average prediction accuracy. To this end, we calculate the γ_i according to the following equation:

$$\gamma_i = l_\gamma + \frac{Diff_\rho(Y_i) - \min_i Diff_\rho(Y_i)}{\max_i Diff_\rho(Y_i) - \min_i Diff_\rho(Y_i)}(u_\gamma - l_\gamma). \quad (6)$$

where $Diff_\rho(Y_i) = Q_\rho(Y_i) - Q_\rho(Y)$, $Q_\rho(Y_i)$ is the ρth quantile of the historical flow data Y_i at the sensor i, and Y denotes all historical flow captured by all sensors. Equation 6 is basically a *min-max* scale transformation on the difference between $Q_\rho(Y_i)$ and $Q_\rho(Y)$. l_γ and u_γ are the predefined lower and upper bound of γ, i.e., $\gamma \in [l_\gamma, u_\gamma]$, in the experiment we set it to be [0, 1], and ρ is set to be 0.9, which means we focus on the prediction of 0.9 quantile [46].

F. Evaluation Pipeline

Figure 4 presents the full scenario-based evaluation pipeline. First, a set of parameters such as pedestrian walking speed, demand, and train schedules are selected as inputs for the



(a) Scenario 11

(b) Scenario 16

(c) Scenario 18

Fig. 5. The OD graph of scenario 11 (a), scenario 16 (b), and scenario 18 (c). In the graph, the vertexes are *Main Entry-Exit*, *Back Entry-Exit*, and *Platform* 1 to 8. Thicker edges indicate larger flows. Additionally, each origin is assigned a unique color, and edges with the origin color represent the flow from that origin. In contrast, edges with the destination's color represent the inflow to that destination.

simulation. A sensor network is then designed to capture pedestrian flows at various locations within the station model. The simulation, run via PedestrianDynamic, generates trajectory data for each pedestrian. This trajectory data includes details such as the *AgentID*, *ActivityID*, and the entry and exit times for each agent. Using this information, we construct an Origin-Destination (OD) matrix to analyze the diversity of each generated scenario. Based on these scenarios, we decide which crowd flow data will be used for training and which will be reserved for testing the model.

The data collected from different scenarios allow us to conduct both in-distribution and out-of-distribution testing. In in-distribution testing, the training and testing datasets come from different parts of the same distribution, while in out-of-distribution testing, the training data are derived from normal scenarios and the testing data from abnormal scenarios. These datasets are divided according to the specific scenarios used. The model is then evaluated using various metrics, such as MAE, RMSE, and quantile loss, to identify the most suitable predictive model for each testing scenario. This comprehensive pipeline ensures the model is rigorously tested across different conditions and configurations.

IV. Dataset Quality Validation

A. OD Visualization

In this section, we conduct qualitative analyses of the simulation data to verify its heterogeneity and diversity.

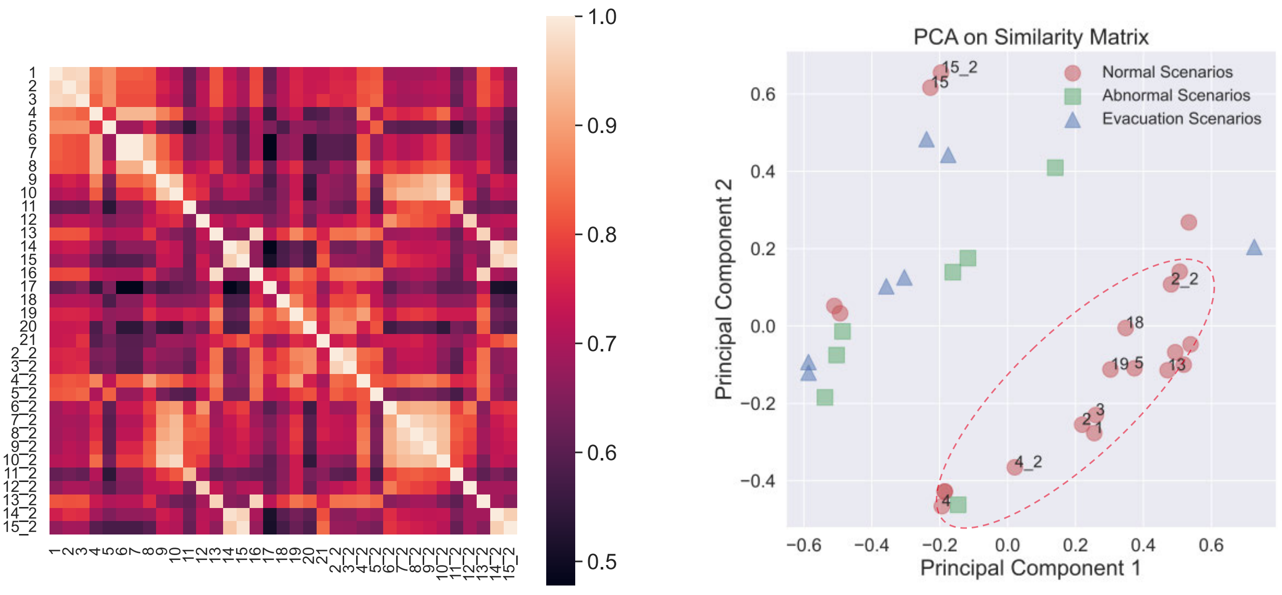
Based on our experience in train stations, we expect the OD matrix to vary across different simulated scenarios. The OD matrix represents the number of people traveling from an origin to a destination within a given period. Specifically, we anticipate changes in the OD matrix due to factors such as the proportion of demand at the main and back entrances, arrival frequency, and the occurrence of an evacuation. To verify this, we analyzed the OD matrices of three scenarios and visualized their OD graphs in Figure 5. Scenario 11 (Figure 5a) represents an evacuation scenario during rush hour. When the alarm is triggered, pedestrians evacuate the building, resulting in a significant flow from the main entrance to both the back exit and the main exit.

Scenarios 16 and 18 are normal scenarios with different demands on the train lines. In Scenario 18, the visiting probability distributions of $Main$ and $Back$ is set to $[50\%, 50\%]$ rather than $[75\%, 25\%]$ as in Scenarios 11 and 16. Hence, the flows from the back entry are more than Scenarios 11 and 16. Besides, the boarding probability distribution for train lines 1 to 4 is $[10\%, 30\%, 30\%, 30\%]$ in Scenario 16. Therefore, in Scenario 16 the flow from *Main* to *Plt1* and *Plt3* is much lower than the other flows.

B. Analysis of Scenario Variations

To gain an overview of the distribution w.r.t the OD demands of all scenarios, we compute the similarity between the ODs of different scenarios. Specifically, we choose structural similarity index (SSIM) as the similarity measure [47]. The resulting pairwise SSIM values are used to construct the similarity matrix presented in Figure 6a. We use the SSIM similarities feature vector as a representation of each scenario and perform the PCA dimension reduction, as shown in Figure 6b. As we expect, scenarios with similar OD demands tend to be close to each other. The scatter plot reveals the dispersion of scenarios in the feature space, illustrating their diversity in the OD demands. This indicates that the chosen factors in Table I are capable of generating data with diverse OD demands. The distinction between evacuation, abnormal, and normal scenarios may not be immediately clear, as their OD patterns can often appear similar. In essence, Figure 6b provides an overview of the OD distribution for each scenario. However, it falls short in capturing disparities across the temporal dimension.

We again pick Scenario 11, Scenario 16, and Scenario 18 to conduct a comparison of their temporal disparities and visualize the distribution of the network data of these scenarios. We conduct the t-SNE (t-distributed Stochastic Neighbor Embedding) technique to visualize the network time series



(a)

(b)

Fig. 6. (a) SSIM similarity matrix. (b) 2D visualization of each simulated scenario. Some scenarios with the same parameter settings but different random seeds could be located closely, e.g., scenario 15 and scenario 15_2 on the top.

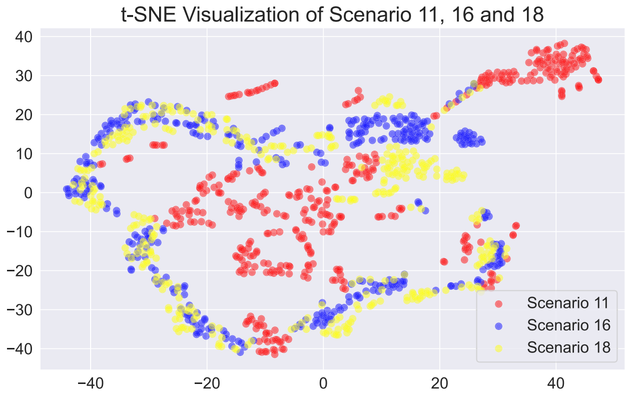


Fig. 7. Visualization of the distribution of the data from three scenarios.

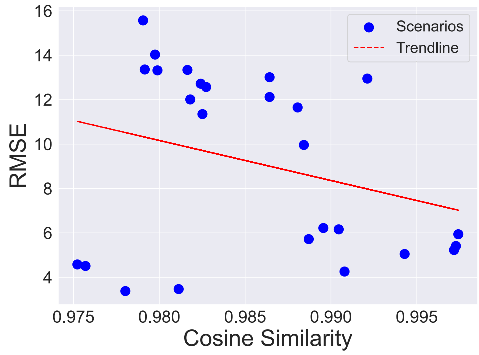


Fig. 8. In general, RMSE decreases as the cosine similarity increases.

data. t-SNE is a dimension reduction technique that can help visualize high-dimensional data in a low-dimensional space.

The result is shown in Figure 7. All data points from the three scenarios are generally concentrated within a single cluster. However, in Scenario 11, there is a bunch of data lying outside the cluster and corresponds to the data when evacuation occurred. Scenarios 16 and 18 exhibit similar data cluster shapes, whereas Scenario 11 shows most data points dispersed outside the circular region formed by the other two. This disparity reflects the fact that Scenarios 16 and 18 represent off-peak periods, while Scenario 11 occurs during rush hour. Consequently, there is a significant contrast in passenger volume between Scenario 11 and the other two scenarios.

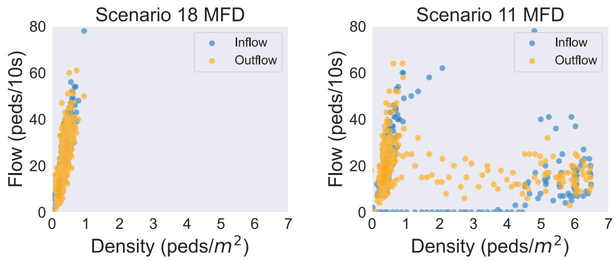
To validate the assumption that scenarios with less similar OD demands are harder to predict, we conducted a quantitative analysis. Using Figure 6b as a reference, we trained an MLP in nine scenarios sampled from the cluster in the lower right corner and evaluated its performance using OOD testing. The training scenarios are $[sc1, sc2, sc2\_2, sc3, sc4\_2, sc5, sc13, sc18, sc19]$. Essentially, most of these scenarios are in the off-peak period, but with different demands on each train line and a low volume of passengers. The trained MLP is then used to predict the other scenarios in our synthetic dataset. We calculate the mean SSIM representation vector of the training clusters and measure its cosine similarity to the other testing scenarios.

The validation results are presented in Figure 8. It is observed that the prediction errors generally decrease as the similarity increases. This observation underscores two key insights: (1) Scenarios with OD demands different from those of the training scenario tend to be less predictable. (2) Predicting scenarios with high passenger volume poses a significant challenge for the models, especially when high-volume data is scarce. Furthermore, analyzing the SSIM between the ODs of different scenarios allows us to have a ballpark estimation of the predictability of different scenarios.

TABLE II

PERFORMANCE OF MODELS ON IN-DISTRIBUTION AND OUT-OF-DISTRIBUTION TESTING SCHEME. THE ERROR IS CALCULATED AS THE AVERAGE OVER EACH SCENARIO, NODE, PREDICTION HORIZON, AND FEATURE, WITH UNITS IN (PEDS/10S). MODELS ANNOTATED WITH *v1* INDICATE THAT WE BUILD SEPARATE MODELS FOR EACH SENSOR, DECOUPLING THE SPATIAL INFORMATION THE MODELS PERCEIVE

Models	In-distribution			Out-of-distribution		
	RMSE	MAE	0.9-risk	RMSE	MAE	0.9-risk
VAR v1	6.43	3.86	10.00	6.45	4.01	9.56
VAR	5.24	3.11	7.25	6.52	3.80	9.38
Linear Regression v1	6.36	3.84	9.96	7.16	4.44	9.78
Linear Regression	4.91	3.13	6.95	8.22	5.01	11.42
MLP v1	6.40	3.77	8.77	7.32	4.26	8.95
MLP	4.84	3.04	6.62	5.78	3.48	8.24
XGBOOST v1	6.05	3.61	8.64	6.70	4.00	8.35
XGBOOST	**4.39**	**2.62**	**5.40**	5.40	**3.08**	**6.29**
N-BEATS	4.86	3.10	7.15	7.27	4.53	9.95
GAT-GRU	5.87	3.36	8.87	6.75	4.03	8.65
RNN	4.61	2.89	6.58	**5.34**	3.18	7.96
RNN Seq2seq	4.53	2.76	6.26	5.73	3.18	7.80
DCRNN	4.84	2.75	6.42	5.73	3.12	7.45
STGCN	4.96	2.99	6.48	7.03	3.88	10.79



(a) MFD in a free flow space.

(b) MFD in evacuation.

Fig. 9. Comparison of MFDs for two scenarios.

C. Macroscopic Fundamental Diagram

To better illustrate the characteristics of pedestrian flow in different scenarios, we present the density-flow macroscopic fundamental diagram (MFD) for Scenarios 18 and 11. To construct this MFD, we focus on the red-shaded area in Figure 2, which encompasses the gate machine and is bounded by sensors 5, 6, 7, and 3. The sum of the inflow and outflow data captured by these sensors is used to calculate the flow in the MFD. The number of passengers is estimated based on the inflow and outflow within this area, allowing us to infer the approximate density. The results are shown in Figure 9.

The MFD in Scenario 18 demonstrates the free-flow regime of the triangular density-flow fundamental diagram. In contrast, Scenario 11 captures both the free-flow and congestion regimes. During evacuation, people attempt to leave the station, but due to high density, they become stalled at the gate machine. The MFD is constructed using only sensor flow data, demonstrating that synthetic flow data can capture general traffic phenomena and thus reflect the authenticity of the synthetic dataset.

V. BENCHMARKING VARIOUS MACHINE LEARNING MODELS

In this section, we evaluate the data-driven prediction models using the synthetic dataset. Accordingly, we derive the answers to the following two questions:

- To what extent does a set of models featuring one sensor position predict better than one model featuring all sensor positions at once, given that different locations have different flow patterns?
- Which model performs best for a given set of scenarios?

We evaluated more complex learning-based prediction models on the proposed synthetic dataset to see whether the more advanced model can generalize better. To explore the model's performance under varying conditions, we conduct both in-distribution and out-of-distribution testing. In the in-distribution testing, we use all scenario data to generate training and testing sets. In out-of-distribution testing, the models are exclusively trained using normal scenarios, and their performance is assessed using abnormal and evacuation scenarios. The prediction errors for each model are presented in Table II.

A. Local or Global Inputs?

Upon analyzing the results in Table II, it is evident that the models trained employing individual sensor data generally perform worse than those trained using data from all sensors as input. This discrepancy may be due to the use of global sensor network data, combining information from multiple sensors can lead to better performance.

B. Which Model Is the Best?

Another key finding is that all models exhibit performance degradation when tested on out-of-distribution data. Analyzing the average error across all model types, we observe that *XGBOOST* achieves the best performance among the tested models in both evaluation settings.

As the average error alone does not capture edge-case performance, we analyze the model's performance on two scenarios of interest. Specifically, we focus on the performance in two abnormal scenarios: Scenario 11 and Scenario 21. Scenario 11 represents an evacuation where people exit through the main and back exits, while Scenario 21 simulates a group of people passing through the train station without boarding.

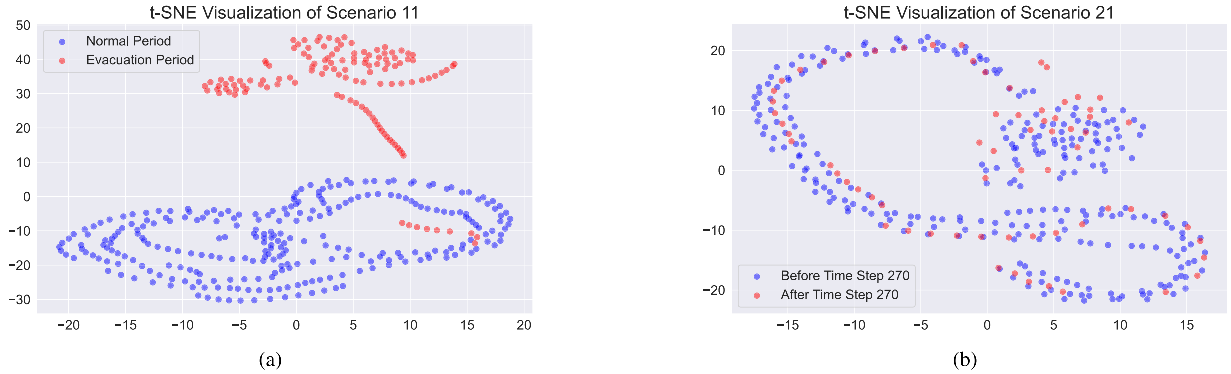


Fig. 10. (a) t-SNE visualization on Scenario 11. (b) t-SNE visualization on Scenario 21.

We analyze the behavioral differences between these two scenarios by projecting them into a 2-dimensional space using t-SNE (Figure 10). To differentiate data before and after the evacuation event, which occurs at time step 270 (45 minutes), data points after this time step are highlighted in red. In Figure 10a for Scenario 11, there is a significant shift in data distribution after evacuation. In contrast, Figure 10b for Scenario 21 shows that all data points remain within the same distribution.

We evaluated the performance of each model in both scenarios using multiple metrics. Specifically, for the WAE metric, the timespan S is set to 3 and $\rho = 0.9$, and we choose L_{avg} as the RMSE. The prediction errors for each evaluation metric are shown in Figure 11. From Figure 11a and Figure 11b, we observe that the best-performing models vary between the two scenarios based on MAE and RMSE. However, when evaluated using WAE, *XGBOOST* consistently achieves the lowest prediction error across both scenarios. Since WAE prioritizes accuracy at the 0.9 quantile, which is critical during overcrowding events, *XGBOOST* emerges as the top-performing model among those tested.

To delve deeper into the predictive capabilities of *XGBOOST*, we provide a 30-second future prediction visualization of *XGBOOST* at three sensor locations for Scenario 11 and Scenario 21 in Figure 12. Although *XGBOOST* performs the best among predictive models, it still struggles to accurately predict the evacuation period in Scenario 11.

Furthermore, although the prediction captures the overall flow trend, it underestimates the number of people using the $Back\ Exit$ in Scenario 21 (Figure 12(d)). This discrepancy arises because the model was trained on scenarios with fewer pedestrians exiting through the back. However, more people are moving from the main exit to the back exit in Scenario 21, therefore the model cannot accurately capture the increase in demand.

VI. Conclusion and Future Work

In this paper, we propose an evaluation pipeline and introduce a novel safety metric to assess the performance of various flow prediction models. To address the scarcity of collected data for abnormal scenarios, we develop a method to generate synthetic pedestrian flow data using a microscopic simulator that replicates the authentic crowd dynamics in rare events. The thorough experimental analysis reveals that our designed factors can generate heterogeneous and diverse indoor crowd flow datasets. The framework and synthetic dataset provide a platform for evaluating and validating predictive models prior to real-world data collection, model training, and deployment. Additionally, we thoroughly benchmark several widely used learning-based prediction models regarding the out-of-distribution generalization.

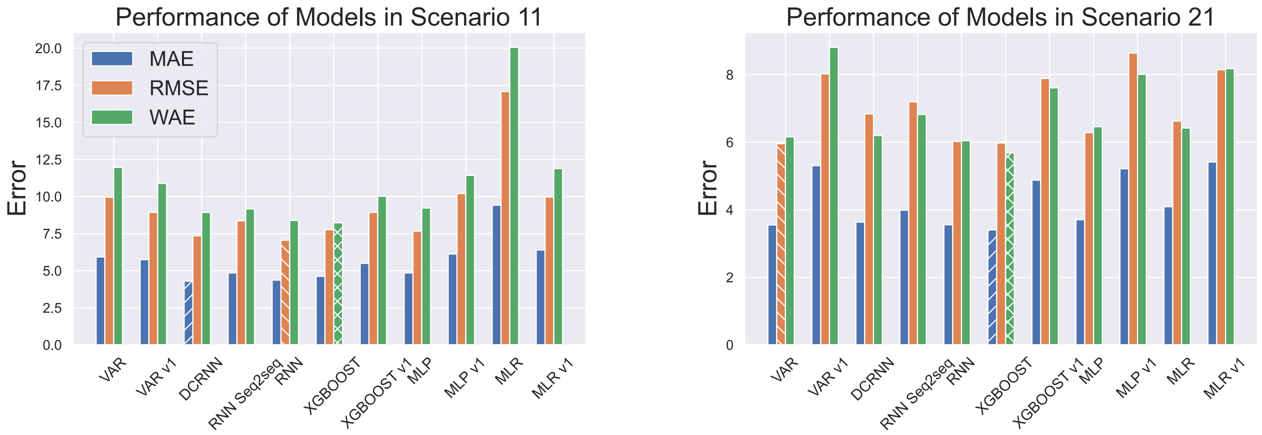
The results reveal the power of data-driven methods in predicting normal scenarios. They are particularly good at memorizing historical patterns. However, these SOTA time-series/traffic prediction models generate unreliable predictions in unseen scenarios, particularly during adverse events. Moving forward, future research should focus on enhancing the adaptability of predictive models to handle unpredictable and dynamic crowd behaviors. Exploring novel techniques such as online sequential learning could improve model performance under non-stationary crowd dynamics. Moreover, incorporating domain-specific knowledge and physics-based models into data-driven approaches may enhance interpretability and generalizability. Beyond model adaptability, real-world data could be collected from various scenarios to further calibrate the synthetic parameters, improving the diversity and realism of the datasets.

Appendix
Weighted Average Error

Proposition 1: For the weighted average error defined in Equation 5, when L_{avg} is chosen to be the mean absolute error, with time span $S = 1$, WAE is equivalent to $\tau-quantile$ loss, where $\tau = \frac{1-\gamma+2\gamma\rho}{2}$.

Proof: To prove that WAE is $\tau-quantile$ loss when $S = 1$ and L_{avg} is MAE, we take the derivative of the expectation of WAE and set it to 0 in order to determine the conditions for $\hat{y}$ that lead to the minimization of WAE. The expectation can be written as:

$$\mathbb{E}(WAE) = (1-\gamma)\underbrace{\mathbb{E}(|y-\hat{y}|)}_{MAE} + 2\gamma\underbrace{[\rho\mathbb{E}_{\hat{y}\leq y}(y-\hat{y}) + (1-\rho)\mathbb{E}_{\hat{y}\geq y}(\hat{y}-y)]}_{\rho-quantile\ loss}$$
$$= [(1-\gamma)+2\gamma\rho]\mathbb{E}_{\hat{y}\leq y}(y-\hat{y}) + [(1-\rho)$$



(a) (b)

Fig. 11. (a) Evacuation scenario. (b) Normal scenario with disruption. The minimum error of each metric is highlighted by different white-shaded patterns.

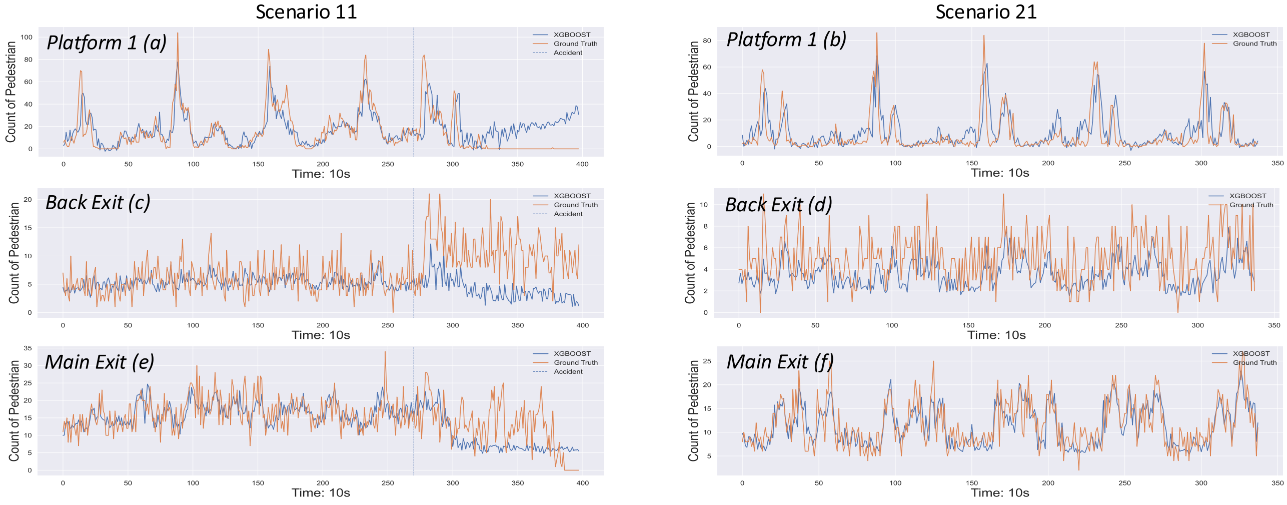


Fig. 12. Visualization of the predictions at three locations (blue line) and the ground truth (orange line). The vertical line highlights the time it starts to evacuate. From top to bottom: *Platform 1*, *Back Entrance_Exit*, *Main Entrance_Exit*. From left to right: Scenario 11 and Scenario 21.

$$
\begin{aligned}
&+2\gamma(1-\rho)]\mathbb{E}_{\hat{y}\geq y}(\hat{y}-y)\\
&=[(1-\gamma)+2\gamma\rho]\int_{\hat{y}}^{+\infty}(y-\hat{y})f(y)dy\\
&\quad+[(1-\rho)+2\gamma(1-\rho)]\int_{-\infty}^{\hat{y}}(\hat{y}-y)f(y)dy,
\end{aligned}
\tag{7}
$$

where $f(y)$ is the probability density function of the ground truth variable. Next, we take the derivative of Equation 7, let $m=[(1-\gamma)+2\gamma\rho]$ and $n=[(1-\rho)+2\gamma(1-\rho)]$, we have:

$$
\begin{aligned}
\frac{\partial\mathbb{E}(WAE)}{\partial\hat{y}}&=-m\int_{\hat{y}}^{+\infty}dF(y)+n\int_{-\infty}^{\hat{y}}dF(y)\\
&=-m(1-F(\hat{y}))+nF(\hat{y})\\
&=(m+n)F(\hat{y})-m.
\end{aligned}
\tag{8}
$$

Here, $F(y)$ denotes the cumulative distribution function of y, and $dF(y)=f(y)dy$. Let Equation 8 be 0, we can get $F(\hat{y})=\frac{m}{m+n}=\frac{1-\gamma+2\gamma\rho}{2}$. Thus, when $\hat{y}=Q_\tau(y)$, where $\tau=\frac{1-\gamma+2\gamma\rho}{2}$, WAE can be minimized, which indicates that WAE is equivalent to $\tau-quantile$ loss. □

TABLE III

STATISTICS OF THE SYNTHETIC CROWD FLOW DATASET

Num. of Scenarios	Num. of Samples	Num. Of Sensors	Frequency
35	11697	35	10 s

DATASETS

The statistics of the synthetic dataset are summarized in Table III.

SETTING OF THE SIMULATION

This section introduces the detailed setting of the simulation's input, including the settings of the agent and the transport element. Figure 13 illustrates the simulation process of the simulation platform. It consists of two generators to generate the agent and transport element (train). The generator

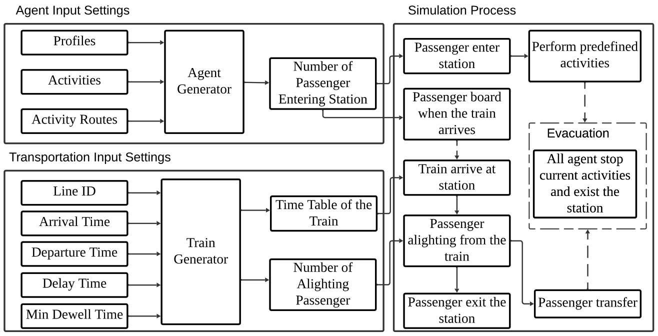

Agent Input Settings
Profiles
Activities
Activity Routes
Agent Generator
Number of Passenger Entering Station
Transportation Input Settings
Line ID
Arrival Time
Departure Time
Delay Time
Min Dwell Time
Train Generator
Time Table of the Train
Number of Alighting Passenger
Simulation Process
Passenger enter station
Passenger board when the train arrives
Train arrive at station
Passenger alighting from the train
Passenger exit the station
Perform predefined activities
Evacuation
All agent stop current activities and exist the station
Passenger transfer

Fig. 13. Framework of the simulation.

TABLE IV

PARAMETER SETTING OF THE PEDESTRIAN

Category	Parameters	Value
General	Radius (m) Max speed (m/s) Min speed (m/s)	0.239 Triangular(1.35, 0.8, 1.75) 0.06
Route Planning	Routing method Density delay weight Viewing distance (m)	LeastEffort Uniform(0.5, 1.5) 60
Route Following	Preferred clearance (m) Max shortcut distance (m) Side preference	0.3 0 Uniform(-1, 1)
Local Behavior	Viewing angle (°) FoV density range (m) FoV avoidance range (m) Personal distance (m) Relaxation time (s)	75 2 8 0.5 0.5

generates passenger and train according to the predefined parameters. During the simulation, the agents would perform the activities we designed. Once the evacuation is activated, all agents need to stop their current activities and need to exit the station.

Agent Profiles

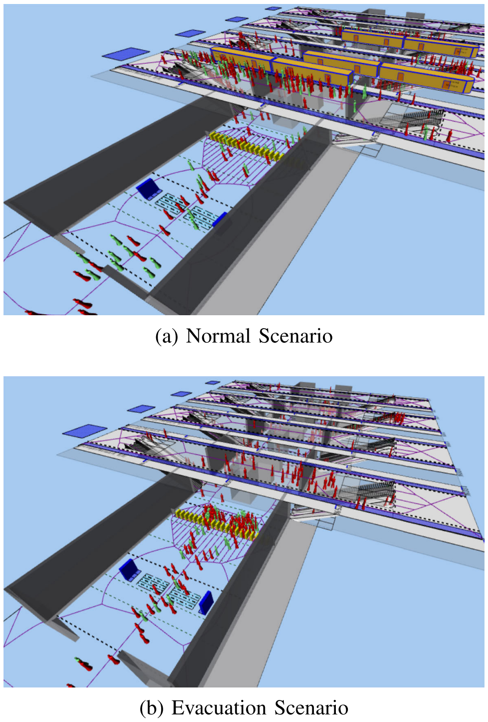
Table IV summarized the parameter setting for the pedestrian in the simulation. The explanations of the parameters are cited from the manual of Pedestrian Dynamics.

- **Radius**: The radius of the agent, in meters.
- **Routing method**: When the routing method is set to "least effort", the route is dynamically updated when new density information becomes available.
- **Density delay weight** : A multiplier for the density-based delay in the routing algorithm. Pedestrians generally have lower speeds when the density is high. When the value larger than 1 means that the agent is more sensitive to delays.
- **Viewing distance**: The distance (m) along which the agent can see edge densities in the ECM [48] network (≥0). A large value means that the agent is more aware of the environment's densities.
- **Preferred clearance**: The preferred minimum distance between the agent and obstacles (m).
- **Max shortcut distance**: When the attraction point is computed we find a point on the indicative route ahead of the agent. The maximum shortcut distance limits the distance that the attraction point can be from the agent, ≤ 0 means no restriction.
- **Side preference**: Bias towards a certain side of the corridor, between -1 (left) and 1 (right).
- **Field of view density range**: The distance in the field of view (m) that is used to determine the local density around the agent (≥ 0).
- **FoV avoidance range**: The distance in the field of view (in meters) that is used for agent collision avoidance (≥ 0).

TABLE V

PASSENGER ACTIVITIES

ID	Name	ActivityType	ActivityGroup
1	Entry_station_1	ENTRY_EXIT	station_entry_exit
5	Entry_platform_1	ENTRY_EXIT	platform_entry_exit_1
6	Entry_platform_2	ENTRY_EXIT	platform_entry_exit_2
7	Entry_platform_3	ENTRY_EXIT	platform_entry_exit_3
8	Entry_platform_4	ENTRY_EXIT	platform_entry_exit_4
9	Entry_platform_5	ENTRY_EXIT	platform_entry_exit_5
10	Entry_platform_6	ENTRY_EXIT	platform_entry_exit_6
11	Entry_platform_7	ENTRY_EXIT	platform_entry_exit_7
12	Entry_platform_8	ENTRY_EXIT	platform_entry_exit_8
16	Go_to	ENTRY_EXIT	***ALL***
17	Ticket_main	TICKET_FACILITY	ticket_facility_main
18	Ticket_back	TICKET_FACILITY	ticket_facility_back
19	Emergency_Exit	ENTRY_EXIT	station_entry_exit
21	Emergency_Exit_PLT	ENTRY_EXIT	platform_entry_exit_1
23	Alight	TRANSPORTATION	TransportGenerator_1
26	Entry_station_2	ENTRY_EXIT	***ALL***
27	Wait	WAITING	***ALL***
28	Board	TRANSPORTATION	***ALL***
29	Transfer	TRANSPORTATION	***ALL***
30	Cross	ENTRY_EXIT	station_entry_exit


(a) Normal Scenario

(b) Evacuation Scenario

Fig. 14. Visualization of passenger activities in normal and evacuation scenarios. Red agents represent passengers alighting from the train, while green agents are those heading to the platform.

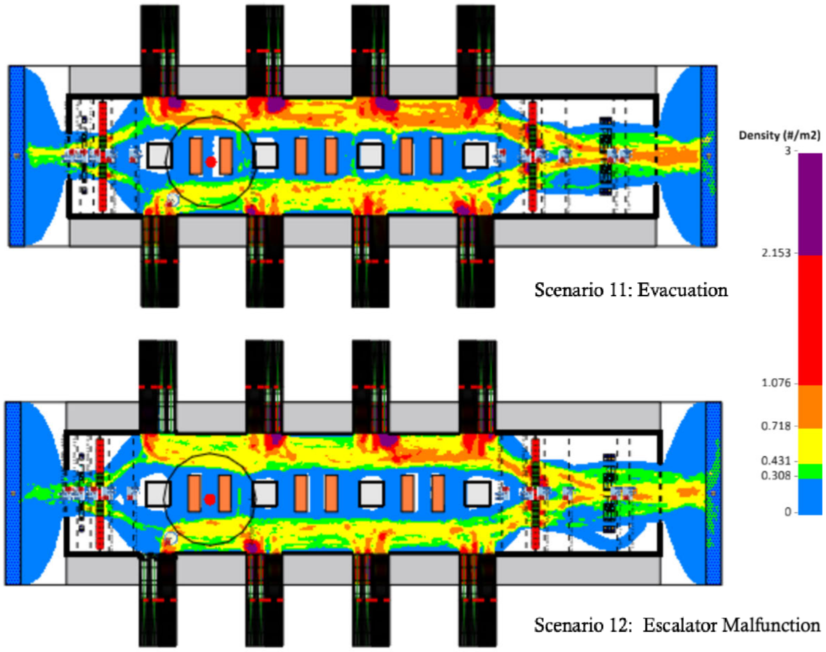


Fig. 15. Density map in two different scenarios.

TABLE VI
PASSENGER ACTIVITY ROUTES

Name	ID	Nr of activities	Activity 1	Activity 2	Activity 3	Activity 4
To_platform	1	4	1 | Entry_station	18 | Ticket_back	17 | Ticket_main	16 | Go_to
From_platform_1	2	2	5 | Entry_platform_1	16 | Go_to	-	-
From_platform_2	3	2	6 | Entry_platform_2	16 | Go_to	-	-
From_platform_3	4	2	7 | Entry_platform_3	16 | Go_to	-	-
From_platform_4	5	2	8 | Entry_platform_4	16 | Go_to	-	-
From_platform_5	6	2	9 | Entry_platform_5	16 | Go_to	-	-
From_platform_6	7	2	10 | Entry_platform_6	16 | Go_to	-	-
From_platform_7	8	2	11 | Entry_platform_7	16 | Go_to	-	-
From_platform_8	9	2	12 | Entry_platform_8	16 | Go_to	-	-
Emergency_Route	10	1	19 | Emergency_Exit	-	-	-
Alighting Route_1	12	2	23 | Alight	26 | Entry_station_2	-	-
Boarding Route	13	3	1 | Entry_station	27 | Wait	28 | Board	-
Alighting Route_2	14	3	23 | Alight	27 | Wait	29 | Transfer	-
Cross_station	15	2	1 | Entry_station	30 | Cross	-	-

- **Personal distance**: The desired personal distance ahead (in meters) between agents (≥ 0).
- **Relaxation time**: An agent is assumed to require the relaxation time to reach any desired velocity. It implies that an agent is required to keep a certain ‘distance’ from any static obstacle or agent.

Agent Activities.

Table V summarizes all possible activities agents can perform within the station, with their movement paths specified in Table VI. As shown, alongside primary activities (e.g., alighting, boarding, and transferring), additional activities are incorporated to introduce variability in the crowd flow data, such as passengers crossing through the station or entering and exiting the platform without boarding (*Entry_platform_x*). Additionally, activities for passengers purchasing train tickets are included.

Among these activities, *“GO_to”* manages the OD demand from station entries to platforms, while *“Board”* defines the percentage of passengers boarding specific train lines.

Figure 14 visualizes passenger activities in both normal and evacuation scenarios. During evacuation, all agents follow the *Emergency_Route* defined in Table VI, which is a predefined path guiding each agent to the nearest exit to leave the station.

Figure 15 visualizes the density map of an evacuation scenario and an escalator malfunctioning scenario. In the evacuation scenario, the density along the hallway and exit is higher than that in the scenario 12. We can also observe that due to the escalator malfunction, density around the left most escalator is less than 1 p/m^2

Transportation Input

The transport generator in the simulation uses a timetable to create transport elements (trains). Agents can only begin (alight) or complete (board) their journey with a transport

TABLE VII
TIMETABLE OF THE TRAIN STATION

ID	LineID	ArrivalTime	DepartureTime	Direction	DelayTime	MinDwellTime
Train1	Line1	60*0	60*0 + 90	0	0	0
Train2	Line2	60*2	60*2 + 90	1	0	0
Train3	Line1	60*4	60*4 + 90	0	0	0
Train4	Line2	60*6	60*6 + 90	1	0	0
Train5	Line3	60*3	60*3 + 90	0	0	0
Train6	Line4	60*5	60*5 + 90	1	0	0
Train7	Line3	60*6	60*6 + 90	0	0	0
Train8	Line4	60*1	60*1 + 90	1	0	0

activity. Table VII shows an example of the train schedule, with 8 tracks and 4 lines in the station. Each train operates on one of these lines. The *ArrivalTime* indicates the arrival time of each train during the simulation. The *DepartureTime* is the time that the train will leave the station, in the example, the train will depart after 90 seconds they arrive. In each scenario, the arrival list is repeated after a certain period until the simulation terminates.

In the example, the *DelayTime* parameter is set to 0, indicating no delays, but can be adjusted to simulate train delays if desired. The *MinDwellTime* specifies the minimum time a train must stay at the platform to allow passenger boarding and alighting, ensuring the train does not depart prematurely, even if its scheduled departure time has passed.

ACKNOWLEDGMENT

The authors acknowledge the support of NWO, VENI Project CrowdIT Space, and TU Delft AI Labs program. They also thank OpenAI for using their AI tool Chatgpt to improve the writing of this article.

REFERENCES

[1] D. C. Duives, W. Daamen, and S. P. Hoogendoorn, "State-of-the-art crowd motion simulation models," *Transp. Res. C, Emerg. Technol.*, vol. 37, pp. 193–209, Dec. 2013.

[2] A. Rasouli, "Pedestrian simulation: A review," 2021, *arXiv:2102.03289*.

[3] H. Dong, M. Zhou, Q. Wang, X. Yang, and F.-Y. Wang, "State-of-the-art pedestrian and evacuation dynamics," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1849–1866, May 2019.

[4] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 51, no. 5, p. 4282, 1995.

[5] V. J. Blue and J. L. Adler, "Emergent fundamental pedestrian flows from cellular automata microsimulation," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1644, no. 1, pp. 29–36, 1998.

[6] R. L. Hughes, "The flow of large crowds of pedestrians," *Math. Comput. Simul.*, vol. 53, nos. 4–6, pp. 367–370, Oct. 2000.

[7] S. P. Hoogendoorn, F. Van Wageningen-Kessels, W. Daamen, D. C. Duives, and M. Sarvi, "Continuum theory for pedestrian traffic flow: Local route choice modelling and its implications," *Transp. Res. Proc.*, vol. 7, pp. 381–397, May 2015.

[8] L. G. Chalmet, R. L. Francis, and P. B. Saunders, "Network models for building evacuation," *Manage. Sci.*, vol. 28, no. 1, pp. 86–105, Jan. 1982.

[9] F. Makinoshima and Y. Oishi, "Crowd flow forecasting via agent-based simulations with sequential latent parameter estimation from aggregate observation," *Sci. Rep.*, vol. 12, no. 1, p. 11168, Jul. 2022.

[10] A. M. Ibrahim, I. Venkat, K. G. Subramanian, A. T. Khader, and P. D. Wilde, "Intelligent evacuation management systems: A review," *ACM Trans. Intell. Syst. Technol.*, vol. 7, no. 3, pp. 1–27, Apr. 2016.

[11] C. Zhang, K. Kang, H. Li, X. Wang, R. Xie, and X. Yang, "Data-driven crowd understanding: A baseline for a large-scale crowd dataset," *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 1048–1061, Jun. 2016.

[12] C. Martani, S. Stent, S. Acikgoz, K. Soga, D. Bain, and Y. Jin, "Pedestrian monitoring techniques for crowd-flow prediction," *Proc. Inst. Civil Eng.-Smart Infrastructure Construct.*, vol. 170, no. 2, pp. 17–27, Jun. 2017.

[13] J. Gao, Q. Wang, and X. Li, "PCC net: Perspective crowd counting via spatial convolutional network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3486–3498, Oct. 2020.

[14] Z. Du, M. Shi, J. Deng, and S. Zafeiriou, "Redesigning multi-scale neural network for crowd counting," *IEEE Trans. Image Process.*, vol. 32, pp. 3664–3678, 2023.

[15] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 1–41, 2015.

[16] I. Karamouzas, N. Sohre, R. Hu, and S. J. Guy, "Crowd space: A predictive crowd analysis technique," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–14, Dec. 2018.

[17] L. Schauer, M. Werner, and P. Marcus, "Estimating crowd densities and pedestrian flows using Wi-Fi and Bluetooth," in *Proc. 11th Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Services*, 2014, pp. 171–177.

[18] A. Sudo, T.-H. Teng, H. C. Lau, and Y. Sekimoto, "Predicting indoor crowd density using column-structured deep neural network," in *Proc. 1st ACM SIGSPATIAL Workshop Predict. Hum. Mobility*, Nov. 2017, pp. 1–7.

[19] U. Blanke, G. Tröster, T. Franke, and P. Lukowicz, "Capturing crowd dynamics at large scale events using participatory GPS-localization," in *Proc. IEEE 9th Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process. (ISSNIP)*, Apr. 2014, pp. 1–7.

[20] S. Georgievska et al., "Detecting high indoor crowd density with Wi-Fi localization: A statistical mechanics approach," *J. Big Data*, vol. 6, no. 1, pp. 1–23, Dec. 2019.

[21] A. Tordeux, M. Chraibi, A. Seyfried, and A. Schadschneider, "Prediction of pedestrian dynamics in complex architectures with artificial neural networks," *J. Intell. Transp. Syst.*, vol. 24, no. 6, pp. 556–568, Nov. 2020.

[22] A. Bamaqa, M. Sedky, T. Bosakowski, B. B. Bastaki, and N. O. Alshammari, "SIMCD: SIMulated crowd data for anomaly detection and prediction," *Expert Syst. Appl.*, vol. 203, Oct. 2022, Art. no. 117475.

[23] A. Sudo et al., "Particle filter for real-time human mobility prediction following unprecedented disaster," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Oct. 2016, pp. 1–10.

[24] E. L. Manibardo, I. Laña, and J. D. Ser, "Deep learning for road traffic forecasting: Does it make a difference?" *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6164–6188, Jul. 2022.

[25] M. Levin and Y.-D. Tsao, "On forecasting freeway occupancies and volumes," *Transp. Res. Rec.*, pp. 47–49, no. 773, Oct. 1980.

[26] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. ICLR*, Jan. 2017, pp. 1–16.

[27] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial–temporal graph modeling," 2019, *arXiv:1906.00121*.

[28] F. Li et al., "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," *ACM Trans. Knowl. Discovery Data*, vol. 17, no. 1, pp. 1–21, Feb. 2023.

[29] J. Sun, J. Zhang, Q. Li, X. Yi, Y. Liang, and Y. Zheng, "Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2348–2359, May 2022.

[30] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, "DeepSTN+: Context-aware spatial–temporal neural network for crowd flow prediction in metropolis," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 1020–1027.

[31] F. Simini, G. Barlacchi, M. Luca, and L. Pappalardo, "A deep gravity model for mobility flows generation," *Nature Commun.*, vol. 12, no. 1, p. 6576, Nov. 2021.

[32] M. Zhang, Y. Yao, and K. Xie, "Prediction and diversion mechanisms for crowd management based on risk rating," *Engineering*, vol. 9, no. 5, pp. 377–387, 2017.

[33] X. Guo, K. Song, M. Gao, W. Zhai, Q. Li, and G. Jeon, "Crowd counting in smart city via lightweight ghost attention pyramid network," *Future Gener. Comput. Syst.*, vol. 147, pp. 328–338, Oct. 2023.

[34] Y. Liu, Z. Liu, and R. Jia, "DeepPF: A deep learning based architecture for metro passenger flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 101, pp. 18–34, Apr. 2019.

[35] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 961–971.

[36] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2255–2264.

[37] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 261–268.

[38] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese, "Learning an image-based motion context for multiple people tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3542–3549.

[39] G. G. Løvås, "Modeling and simulation of pedestrian traffic flow," *Transp. Res. B, Methodol.*, vol. 28, no. 6, pp. 429–443, Dec. 1994.

[40] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[41] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2019, pp. 1–21. [Online]. Available: https://openreview.net/forum?id=r1ecqn4YwB

[42] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27. Red Hook, NY, USA: Curran Associates, Jan. 2014, pp. 1–9. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf

[43] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

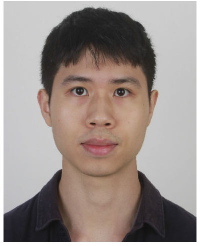
[44] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell., (IJCAI)*, 2018, pp. 3634–3640, doi: 10.24963/IJCAI.2018/505.

[45] R. Koenker and K. Hallock, "Quantile regression," *J. Econ. Perspect.*, vol. 15, no. 4, pp. 143–156, 2001.

[46] M. Seeger, D. Salinas, and V. Flunkert, "Bayesian intermittent demand forecasting for large inventories," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29. Red Hook, NY, USA: Curran Associates, Dec. 2016, pp. 4653–4661. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/03255088ed63354a54e0e5ed957e9008-Paper.pdf

[47] T. Djukic, S. P. Hoogendoorn, and H. van Lint, "Reliability assessment of dynamic OD estimation methods based on structural similarity index," in *Proc. Transp. Res. Board 92nd Annu. Meeting*, Jan. 2013, pp. 1–13.

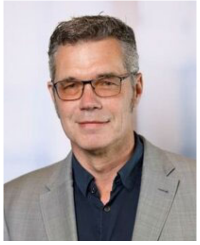
[48] R. Geraerts, "Planning short paths with clearance using explicit corridors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 1997–2004.



Weiming Mai received the M.Sc. degree in data analytics and artificial intelligence from Hong Kong Baptist University, Hong Kong, China, in 2022. He is currently pursuing the Ph.D. degree with the Department of Transport and Planning, Delft University of Technology, Delft, The Netherlands. His research interests include intelligent transportation systems, intelligent decision-making, and trustworthy machine learning.

Dorine Duives is currently an Associate Professor of active mode management and modeling. She is the Director of the Active Mode Laboratory, Delft University of Technology. Her main aim is to develop theories and models to support the design and evaluation of active mode infrastructures.

Panchamy Krishnakumari was an Assistant Professor of data-driven multiscale modeling for traffic and transportation and the Co-Director of the Artificial Intelligence for Mobility Laboratory, Department of Transport and Planning. Her research is on developing interpretable machine learning models for understanding the mobility dynamics of large-scale multimodal networks.



Serge Hoogendoorn is currently serving as one of four distinguished professors of smart urban mobility with Delft University of Technology. His research interests include smart urban mobility, covering theory, modeling, and simulation of various transportation networks, integrated management methods, the impact of travel behavior uncertainty, ICT on network operations, and urban data applications, addressing both recurrent and non-recurring situations.