Real-time RSET Prediction Based on Simulation Dataset Using Machine Learning: A Complex Geometry Case Study

Ondřej Uhlík

Brno University of Technology, Faculty of Civil Engineering
Veveří 331
Brno, 60200, 602 00, Czech republic
e-mail: ondrej.uhlik@vut.cz

ABSTRACT

Agent-based evacuation model simulations are not suitable for real-time estimates due to their complexity and computational demands. Machine learning models allow for the approximation of simulations through estimates, creating a metamodel whose outputs can be used in real-time for effective decision-making in object safety management. The article presents a case study demonstrating the process of training the metamodel on a dataset with seven input features and simulations of evacuation model generated by a quasi-random sequence. Among the compared machine learning regression models, the ANN metamodel achieved the best results.

INTRODUCTION

The current security situation in the world requires a systematic increase in the resilience of high-occupancy buildings. Simulation of the evacuation process through a numerical model enables the optimization of building designs and safety plans to mitigate the negative impacts of emergencies. Currently, numerical microscopic agent-based evacuation models prove most used for these simulations [1]. They allow for simulating any number of emergency scenarios during the design phase, and quantitatively evaluating their impacts, such as required safe egress time (RSET), congestion at critical locations, load on evacuation exits, exposure levels to fire effluents, etc. Based on simulation outputs, one can proceed to optimize building designs or evacuation plans aimed at reducing the adverse effects of emergencies.

Among the current challenges in enhancing the security of soft targets is the shift of focus toward an adequate immediate response by the facility management and subsequently by the relevant emergency response teams, based on valid real-time predictions of the evacuation process during an emergency, taking into account the current operational conditions of the facility. This task can be addressed by deploying Agent-Based Evacuation Model (ABEM) in combination with machine learning (ML) models, which are a subset of artificial intelligence (AI) models. ABEM are typically stochastic models that utilize the Monte Carlo method, running a series of simulations whose results are then statistically processed. In the case of soft targets, due to their high occupancy, the computational time for simulations can reach tens of minutes, and with calibration process according to current operational conditions, even hours. This computation time renders simulations practically unusable for real-time applications.

ML models have the potential to estimate simulation outcomes in real-time based on a training set of simulations and current calibration anonymous data from camera systems, thereby creating a so-called metamodel, which is an extension of ABEM. The metamodel is then capable of providing an immediate prediction of the evacuation process. These predictions can be used by the facility manager, security manager, emergency response teams, or other security personnel to increase the effectiveness of their response and reduce safety risks. The diagram in Figure 1 summarizes the basic idea of deploying the metamodel and places it in the context of evacuation modeling and machine learning.

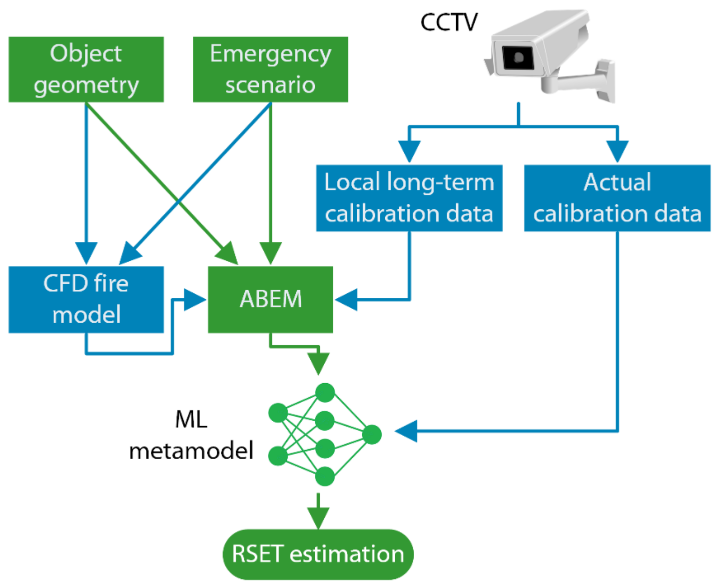

Figure 1: A diagram placing the metamodel into the overall context of the potential practical use of the system (the green parts are the subject of this work).

Green parts of this diagram are subject of this article. The diagram as a whole shows the potential of integrating ABEM calibration based on long-term data collection from camera systems, a CFD fire model (if fire scenario modeling is necessary), and real-time calibration data for input into a machine learning metamodel.

The aim of this article is to test the accuracy of 9 machine learning regression models trained on ABEM simulation dataset of complex geometry scenario using the Sobol Lpτ method [2] and subsequent Monte Carlo method [3]. The fundamental condition for the functioning of the mentioned metamodel is the training of the metamodel on a specific geometry with a specific evacuation scenario. Therefore, the work is not aimed at training a general model that can be applied to a wide range of tasks.

STATE OF THE ART

The possibility of using metamodels as an extension for agent-based models (not evacuation models) was tested by Angione [4] on an agent-based social care model called "Linked Lives." The results demonstrated the potential for deploying an ML model in the form of a metamodel, with the highest accuracy achieved using an artificial neural network (ANN). However, it is important to emphasize that each agent-based model utilizes different computational mechanisms, parameter sets, and degrees of stochasticity, making it necessary to carefully and independently test each complex agent-based model for its potential replacement with a metamodel.

The prediction of outputs from the Pathfinder evacuation agent model was addressed by Deng using Gradient Boost (GB), Extreme Gradient Boost (XGB), and Light Gradient Boost models [5]. Other algorithms were not tested in this work. Guo introduced a system that includes the prediction of RSET, density, and financial costs for potential building renovation. Only the RF algorithm was used for the prediction [6]. The prediction of RSET based on evacuation simulations in the Anylogic simulator was also addressed by Li's ANN classifier [7].

In none of the above-mentioned articles [5], [6], [7] is the pre-evacuation time, a key parameter in analyzing any evacuation process, considered. Additionally, these articles do not address the stochasticity of simulations, which results in different outcomes for the same set of input parameters due to the element of randomness introduced into the inputs and calibration parameters of the model [8], [9], [10]. Furthermore, none of the articles provides a detailed comparison of the available machine learning algorithms within the context of evacuation modeling tasks. This article adresses these mentioned issues.

METHODS

In this chapter, the methods used in this article are described as well as generation of simulation dataset and dataset used for metamodels training.

Simulator

Microscopic evacuation models were created using the simulation tool Pathfinder [11]. Pathfinder is an agent-based, stochastic, microscopic simulation tool that is the most widely used globally for modeling building evacuations [1]. Agent-based models are characterized by the fact that individual persons in the simulation (known as agents) have their own parameters and decision-making processes, similar to the real world, which determine their trajectory throughout the simulation. In the Pathfinder simulation, it is assumed that an agent can see all exits from the current room and the queues at those exits. The agent also knows the distance from these exits to the final destination. Based on this knowledge and other calibration parameters (e.g., the preference for a shorter route at the expense of longer waiting times in a queue), the agent chooses a path to the exit. This process is updated at each time step of the simulation, allowing the agent to change its trajectory if surrounding conditions change (e.g., a queue forms, an exit becomes blocked, etc.).

Dataset Generation

For generating the simulation dataset, a method for generating quasi-random numbers in a multidimensional space using the Sobol sequence, known as LPτ, was employed. This method ensures uniform coverage of the interval of each variable, which is advantageous for machine learning purposes. The use of this method follows on from previous research published in [12], where the training set was created based on a fixed step size for each variable in the set using the brute-force method. This approach involved covering all combinations of the training set variables, which is computationally very demanding. Utilizing the Sobol sequence, or other similar stochastic methods, allows for better testing of the required dataset size for the proper functioning of the metamodel in future research, while ensuring an optimal number of simulations is conducted. Based on simulation dataset the machine learning dataset is generated with 7 spatial features described in Table 1. This dataset is then used for machine learning processes. Dataset includes 100 samples, with 20 % designated for testing and 80 % for training as part of the cross-validation method described below.

Table 1: Set of input features for machine learning dataset.

Input variable	**Mathematic form**
Number of Occupants (N_{occ})	$N_{occ} = \sum i$
Minimum distance of occupants to exits (ExD_{min})	$ExD_{max} = \frac{1}{S}\sum_{i=1}^{S} \min(\lVert\vec{x}_k - \vec{y}_j\rVert)$
Mean distance of occupants to exits (ExD_{mean})	$ExD_{mean} = \frac{1}{S \cdot N_{occ} \cdot E}\sum_{i=1}^{S}\sum_{j=1}^{N_{occ}}\sum_{k=1}^{E}\lVert\vec{x}_k - \vec{y}_j\rVert$
Maximum distance of occupants to exits (ExD_{max})	$ExD_{max} = \frac{1}{S}\sum_{i=1}^{S} \max(\lVert\vec{x}_k - \vec{y}_j\rVert)$

Minimum distance between occupants ($OccD_{min}$)	$OccD_{min} = \frac{1}{S}\sum_{i=1}^{S} Q_{0,05}(\lVert\vec{y}_j - \vec{y}\rVert)$
Mean distance between occupants ($OccD_{mean}$)	$OccD_{mean} = \frac{1}{S \cdot N_{occ}}\sum_{i=1}^{S}\sum_{j=1}^{N_{occ}} \lVert\vec{y}_j - \vec{y}\rVert$
Mean walking speed (V_{mean})	$V_{mean} = \frac{1}{N_{occ}}\sum_{j=1}^{N_{occ}} v_j$
Output variable	**Mathematic form**
Mean required safe egress time ($RSET_{mean}$)	$RSET_{mean} = \frac{1}{S}\sum_{s=1}^{S} RSET_s$

Monte Carlo method is used for simulation. In the context of this work, it involves repeating an evacuation simulation with the same inputs (e.g., number of occupants and their distribution) and then statistically analyzing the results of the simulation set. For each agent, a pseudorandom number is generated in each repetition, which assigns new values to the agents within the calibration parameters (e.g., distribution of walking speed, pre-evacuation time, etc.), leading to different results in each repetition. The results are then statistically processed after conducting a set of simulations. Each simulation was performed in 20 repetitions to capture the stochasticity.

Machine learning techniques

The nested k-fold cross-validation method was used for metamodels training. The inner 5-fold cross-validation is employed to find the optimal hyperparameters of the metamodel. The number of iterations for random hyperparameter search was set to 300. The outer 5-fold cross-validation is intended for training and testing on the entire dataset and evaluating the performance of individual metamodels. The entire nested cross-validation process was repeated 20 times for robust statistical evaluation of the results. The metrics used for the evaluation of metamodels are described in Table 2, where N stands for whole sample, $\hat{y}_i$ means point estimate of y_i and $\bar{y}$ stands for mean estimate.

Table 2: Evaluation metrics for regression of machine learning metamodels.

Name	**Mathematical form**	**Description**
Coefficient of determination	$R^2 = \frac{\sum_{i=1}^{N}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$	Describes the proportion of variability in the dependent variable that the metamodel is able to capture.
Maximum Error	$ME = max(\lvert y_i - \hat{y}_i\rvert)$	This is an expression of the maximum deviation of the metamodel estimate from the training or test data set within the cross validation procedure. This metric is important with respect to estimating the RSET of a building, which is a safety task. For this reason, it is appropriate to check what maximum error can be expected from a given metamodel, and therefore whether the metamodel can be "trusted" to implement follow-up safety measures based on its estimate.
Mean Absolute Error	$MAE = \frac{1}{N}\sum_{i=1}^{N}\lvert y_i - \hat{y}_i\rvert$	This metric describes the average of the absolute deviations of the metamodel estimates from the RSET values in the dataset (residuals) and is therefore also a key indicator of its performance. This metric describes the average of the absolute deviations of the metamodel estimates from the RSET values in the dataset (residuals) and is therefore also a key indicator of its performance.

To test the ability of metamodels to generalize on this type of task, the following machine learning regression models were selected: ordinary least squares regressor (OLS), polynomial regressor (POLY), k-nearest neighbor regressor (KNN), random forest regressor (RF), Gradient Boost regressor (GB), extreme gradient boost regressor (XGB), support vector regressor (SVR), Gaussian process regressor (GP), and artificial neural network (ANN). Shapley analysis [13] was used to examine the contribution of each feature in the machine learning dataset.

EVACUATION MODEL DESCRIPTION

This is a part of the university building B, Faculty of Civil Engineering, Brno University of Technology. The classrooms are connected by a corridor equipped with three security grills, located near the staircase. The width of the doors in the security grilles is 0.8 meters. The width of the staircase flights is 2.8 meters, with a length of 5.25 meters. The height of the staircase steps is 0.16 meters, and their width is 0.35 meters. In the central part of the building, there is an elevator, which is not used for evacuation and, therefore, will not be included in the model. The classrooms are equipped with double doors. Figure 2 shows a bird's-eye view of the ABEM created in Pathfinder, with the evacuation ending on the 3rd floor.

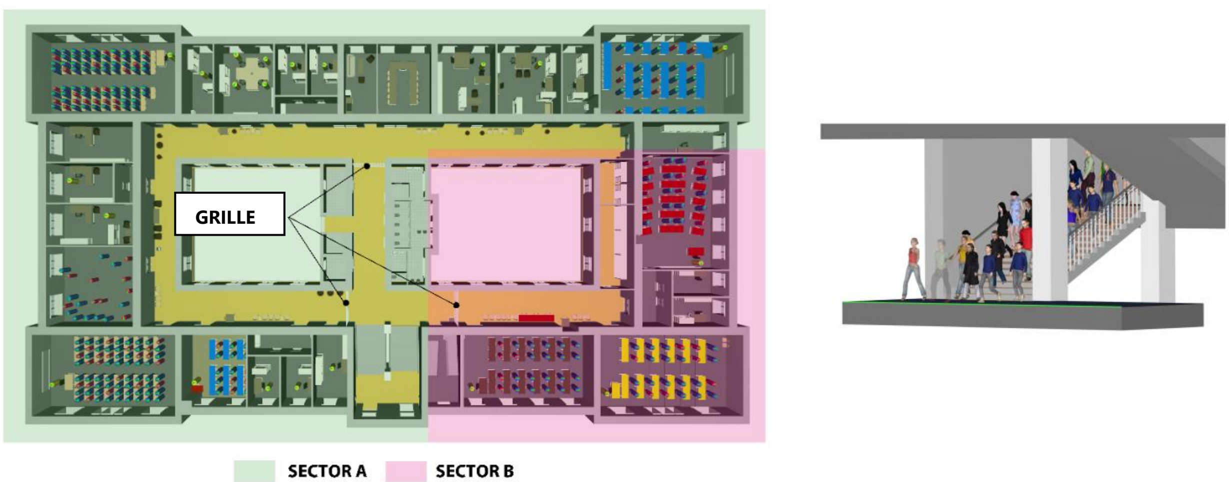

Figure 2: Pathfinder evacuation model scheme with sector selection (left) and detail of model exit in 3rd floor (right).

The model assumes that both wings of the classroom doors are open, providing a total door width of 1.3 meters. The floor contains a total of 8 classrooms, 17 offices, and one meeting room. Some classrooms are separated from the corridor by a staircase step. In the model, the evacuation of people ends when they leave the staircase flight leading from the landing to the 3rd floor. The geometric 3D model of the building was created in Revit Architecture software and subsequently imported into the Pathfinder user interface. Based on this, a movement navigation network for agents was then created. The floor contains a total of 8 classrooms, 17 offices, and one meeting room. The assumed maximum occupancy of the floor is 300 people.

Table 3: Specification of simulation dataset variables.

Dataset variable	Interval
Number of occupants in sector A	0–229
Number of occupants in sector B	0–71
Mean walking speed	0,9–1,4 $m.s^{-1}$

Pre-evacuation time was calibrated with log-normal distribution with parameters μ = 2,972 s, σ = 1,015 s, min: 5 s, max: 120 s. Distribution is based on the publication by Lovreglio [14], [15], which is compiled from several datasets related to the evacuation of university lecture halls which corresponds to given object. The histogram in Figure 3 shows the variability of RSET across all simulations.

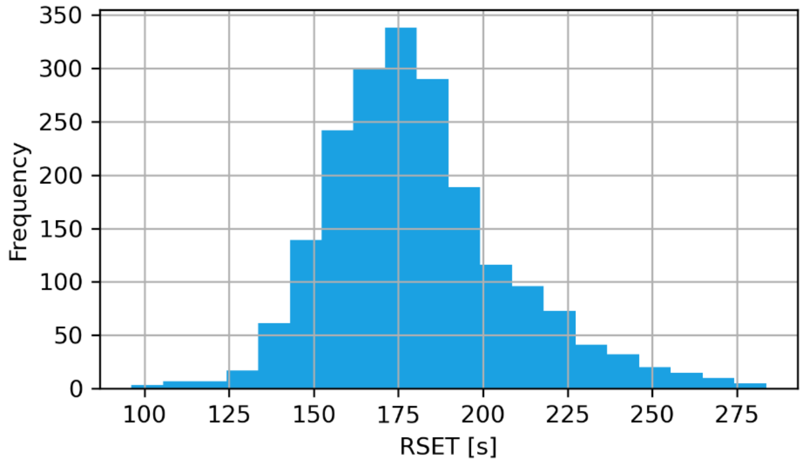

Figure 3: Histogram of RSET for all simulations.

The graph shows that the RSET in simulations ranges approximately from 100 to 280 seconds. It is evident that different combinations of the number of people in sectors A and B and the average movement speed can lead to results varying by up to 3 minutes. The task of the metamodels will be to replicate this variability as accurately as possible.

RESULTS

The results of the nested cross-validation for each metamodel are shown in Table 4. All metamodels show more or less overtraining, which is typical for machine learning models. Thus, the key will be to observe the ability of the metamodels to generalize on test data, not training data. The results show that the POLY metamodel cannot generalize on training data. The ANN metamodel achieved the best results on the test dataset in terms of all metrics - R^2: 0,94, ME: 12,77 s, MAE: 3,81 s. In terms of the ANN metamodel's ability to generalize on this type of training data and set of input attributes, the prediction results appear to be sufficiently accurate in the context of the evacuation model.

Table 4: Results of nested cross-validation for all 9 tested metamodels based on R2, ME, and MAE metrics.

ML	R^2		ME		MAE	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**
OLS	0,85 (±0,01)	0,80 (±0,08)	37,67 (±4,68)	26,67 (±9,61)	6,81 (±0,39)	7,63 (±1,30)
POLY	0,91 (±0,07)	-0,34 (±2,31)	22,16 (±9,89)	≫50 (±≫50)	5,32 (±1,69)	12,42 (±5,45)
KNN	0,93 (±0,02)	0,87 (±0,06)	23,91 (±5,77)	20,87 (±7,98)	4,75 (±0,46)	6,15 (±1,28)
RF	0,98 (±0,01)	0,87 (±0,05)	13,41 (±4,36)	22,47 (±6,44)	2,39 (±0,23)	6,05 (±1,51)
GB	1,00 (±0,00)	0,92 (±0,04)	1,44 (±1,54)	15,88 (±4,32)	0,45 (±0,40)	4,80 (±1,05)
XGB	1,00 (±0,01)	0,91 (±0,04)	4,00 (±4,44)	17,29 (±5,68)	0,82 (±0,66)	5,04 (±1,13)
SVR	0,98 (±0,01)	0,86 (±0,23)	10,51 (±2,39)	25,32 (±20,74)	2,13 (±0,46)	4,60 (±1,63)
GP	1,00 (±0,00)	0,92 (±0,04)	1,21 (±2,91)	20,17 (±10,54)	0,16 (±0,37)	4,46 (±1,24)
ANN	0,99 (±0,01)	0,94 (±0,05)	8,80 (±1,85)	12,77 (±7,14)	2,17 (±0,46)	3,81 (±0,75)

The graph in Figure 4 describes the distribution of residuals of the best ANN metamodel within nested cross-validation, with estimates made on a single test set. The graph shows that the vast majority of residuals are distributed in the range of -5 to 5 seconds. This fact once again confirms that the errors produced by the metamodel are on the order of a few seconds.

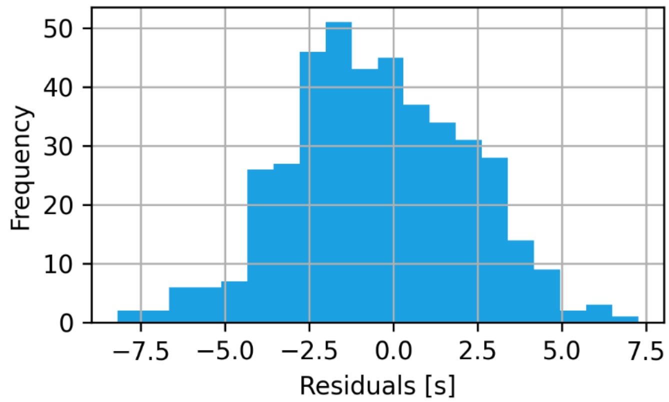

Figure 4: Histogram of ANN metamodel residuals.

In general, it is evident that the evacuation process involves a number of random variables (particularly behavioral ones, such as cognitive and decision-making processes), which evacuation models cannot capture due to a lack of calibration data. In this context, the errors generated by the ANN metamodel on the order of a few seconds are acceptable. The graph in Figure 5 shows the contribution of each machine learning dataset feature to the metamodel's estimation

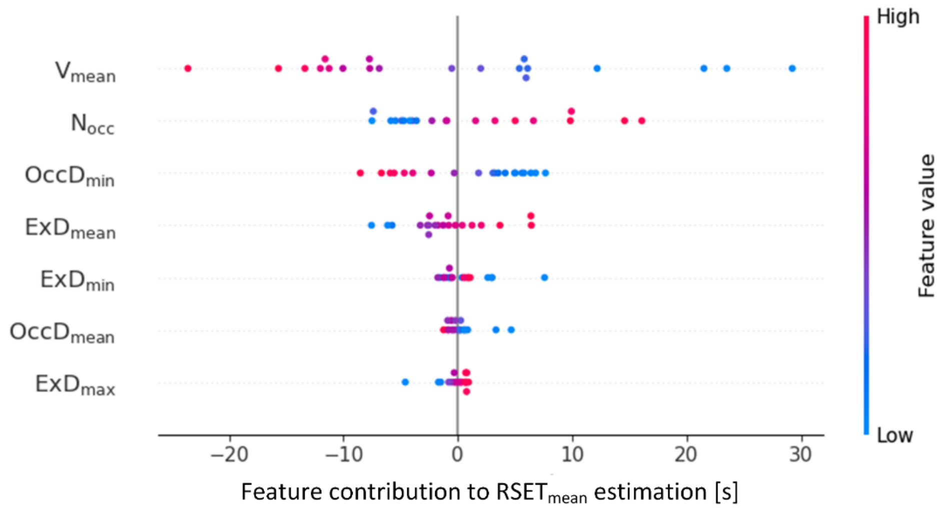

Figure 5: SHAP analysis of feature's contribution to ANN metamodel prediction.

Features are ordered from top to bottom according to their impact (higher means greater impact). It can be observed that the V_{mean} feature has the highest impact and contributes to $RSET_{mean}$ with a maximum of approximately 30 seconds. The second feature is the number of occupants, which is also intuitive. Then there is the mean distance between occupants; a lower distance (high density) leads to a higher $RSET_{mean}$, which makes sense since higher density leads to lower walking speeds. As the mean distance to the exit increases, $RSET_{mean}$ also increases, with the contribution being a few seconds.

DISCUSSION

Increasing the resilience of soft target buildings against threats such as fire or armed attacks is a current societal issue. This work addresses the topic and aims to explore the potential of utilizing machine learning methods and microscopic evacuation modeling to enhance the resilience of soft target buildings. These methods allow for efficient management and estimation of the evacuation process's development, with the goal of minimizing associated safety risks and the number of unexpected situations that need to be addressed during an emergency event. The work has verified that the deployment of a metamodel, as an overlay on a microscopic evacuation model, can be practically used for real-time decision-making based on local input conditions.

In the article, the ability of machine learning metamodels to estimate RSET was tested on a case study of a real object, based on a set of 7 input features and a simulation dataset. The results showed that the task of estimating RSET using machine learning metamodels can be solved with relatively sufficient accuracy in this manner. Of all 9 tested metamodels, the ANN metamodel achieved the best results in terms of all the metrics used. The aim of the article was to introduce and validate this approach to estimating RSET in real-time on a complex structure as a proof of concept. This work demonstrated that machine learning models can be a useful tool for real-time estimation based on simulations. However, we must be very careful when deploying them in real-world tasks, as this caution applies to any evacuation model. This work is a proof-of-concept pilot project and should be further expanded with follow-up research, particularly in the area of optimizing the simulation dataset in terms of its scope and the machine learning dataset in terms of the selection of input variables, with the aim of achieving even more reliable results.

REFERENCES

[1] R. Lovreglio, E. Ronchi, and M. J. Kinsey, “An Online Survey of Pedestrian Evacuation Model Usage and Users,” *Fire Technol*, vol. 56, no. 3, pp. 1133–1153, May 2020, doi: 10.1007/s10694-019-00923-8.

[2] I. M. Sobol’, “On the distribution of points in a cube and the approximate evaluation of integrals,” *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 4, pp. 86–112, Jan. 1967, doi: 10.1016/0041-5553(67)90144-9.

[3] P. L. Bonate, “A Brief Introduction to Monte Carlo Simulation,” *Clin Pharmacokinet*, vol. 40, no. 1, pp. 15–22, 2001, doi: 10.2165/00003088-200140010-00002.

[4] C. Angione, E. Silverman, and E. Yaneske, “Using machine learning as a surrogate model for agent-based simulations,” *PLoS One*, vol. 17, no. 2 February, Feb. 2022, doi: 10.1371/journal.pone.0263150.

[5] Q. Deng *et al.*, “Evacuation Time Estimation Model in Large Buildings Based on Individual Characteristics and Real-Time Congestion Situation of Evacuation Exit,” *Fire*, vol. 5, no. 6, Dec. 2022, doi: 10.3390/fire5060204.

[6] K. Guo and L. Zhang, “Simulation-based passenger evacuation optimization in metro stations considering multi-objectives,” *Autom Constr*, vol. 133, Jan. 2022, doi: 10.1016/j.autcon.2021.104010.

[7] J. F. Li, Y. L. Hu, and W. G. Zou, “Dynamic risk assessment of emergency evacuation in large public buildings: A case study,” *International Journal of Disaster Risk Reduction*, vol. 91, Jun. 2023, doi: 10.1016/j.ijdrr.2023.103659.

[8] E. Smedberg, M. Kinsey, and E. Ronchi, “Multifactor Variance Assessment for Determining the Number of Repeat Simulation Runs in Evacuation Modelling,” *Fire Technol*, vol. 57, no. 5, pp. 2615–2641, Sep. 2021, doi: 10.1007/s10694-021-01134-w.

[9] R. Lovreglio, E. Ronchi, and D. Borri, “The validation of evacuation simulation models through the analysis of behavioural uncertainty,” *Reliab Eng Syst Saf*, vol. 131, pp. 166–174, 2014, doi: 10.1016/j.ress.2014.07.007.

[10] E. Ronchi, P. A. Reneke, and R. D. Peacock, “A Method for the Analysis of Behavioural Uncertainty in Evacuation Modelling,” *Fire Technol*, vol. 50, no. 6, pp. 1545–1571, Oct. 2014, doi: 10.1007/s10694-013-0352-7.

[11] C. Thornton, R. O’Konski, B. Hardeman, and D. Swenson, “Pathfinder: An Agent-Based Egress Simulator,” in *Pedestrian and Evacuation Dynamics*, Boston, MA: Springer US, 2011, pp. 889–892. doi: 10.1007/978-1-4419-9725-8_94.

[12] O. Uhlík, P. Okřinová, A. Tokarevskikh, T. Apeltauer, and J. Apeltauer, “Real-time RSET prediction across three types of geometries and simulation training dataset: A comparative study of machine learning models,” *Developments in the Built Environment*, vol. 18, p. 100461, Apr. 2024, doi: 10.1016/j.dibe.2024.100461.

[13] L. S. Shapley, “ A Value for n-Person Games,” in *Contributions to the Theory of Games (AM-28), Volume II*, Princeton University Press, 1953, pp. 307–318. doi: 10.1515/9781400881970-018.

[14] R. Lovreglio, E. Kuligowski, S. Gwynne, and K. Boyce, “A pre-evacuation database for use in egress simulations,” *Fire Saf J*, vol. 105, pp. 107–128, Apr. 2019, doi: 10.1016/j.firesaf.2018.12.009.

[15] J. Chen *et al.*, “Pre-evacuation time estimation based emergency evacuation simulation in urban residential communities,” *Int J Environ Res Public Health*, vol. 16, no. 23, Dec. 2019, doi: 10.3390/ijerph16234599.