Predicting occupant evacuation times to improve building design

J. Clever, J. Abualdenien, R. K. Dubey & A. Borrmann
Chair of Computational Modeling and Simulation
Technical University of Munich, Munich, Germany

ABSTRACT: Building design requires considering multiple requirements and must fulfill diverse regulations. Therefore, model analysis and simulations are fundamental parts of the design process to find the optimal solution for a given problem. Important decisions are based on a building's assessed final performance in the early design phases. In particular, the analysis of pedestrian flow dynamics is paramount for public facilities like train stations concerning occupants' comfort and evacuation behavior. Currently, it requires multiple steps, from preparing the BIM model to performing pedestrian flow analysis, including semi-automated, often manual work that demands high computation times. Therefore, to improve the building design efficiency in terms of time and pedestrian circulation, this paper proposes a framework applying Deep Learning methods. We propose a real-time pedestrian evacuation prediction to replace time-consuming pedestrian dynamics simulations. More precisely, a modular neural network architecture is designed, including a Convolutional Neural Network and a Multilayer Perceptron, that takes floorplan images and building and simulation parameters as input and predicts the crowd evacuation time for a given building model. As a result, a mean prediction accuracy of 15% could be achieved.

1 INTRODUCTION

Experts from various interconnected domains form a multidisciplinary design team in construction projects. The resultant design of a building and its performance is strongly influenced by multiple design decisions made by each discipline during the design process. In recent years, the Building Information Modeling (BIM) methodology has become an established and common tool that improves the collaborative work between the different disciplines in the project and provides information throughout the project beginning in the early phases (Borrmann et al. 2018).

The design process of a building consists of various stages, where the building is developed from a rough conceptual design to a complex model, including detailed information about all individual components. Especially in the early design phases, fundamental decisions are taken that have a significant impact on the final performance of the building (Knotten et al. 2015). Nevertheless, the required costs and efforts for changes in the design are relatively low (Abualdenien and Borrmann 2019).

By comparing the results of numerous simulations and analyses, architects and engineers explore several models and evaluate multiple design options regarding performance. Commonly, analysis and simulations include the structural system, embodied and operational energy during a building's life-cycle (Abualdenien et al. 2020), and pedestrians' evacuation behavior and comfort inside a building. BIM offers enormous information about different objects in the model (i.e., walls, stairs, zones). For each instance, a geometric representation and a set of properties are accessible (Abualdenien and Borrmann 2019). Moreover, individual simulation information can be added to the model, and, hence, a smooth workflow between BIM-authoring tools and simulation software can be provided.

To allow vendor-neutral data exchange, the open standard Industry Foundation Classes (IFC) (BuildingSMART 2020) is widely supported by various existing authoring tools and simulation software and allows an easy exchange of model data. So far, IFC BIM models show a promising possibility to work as a basis for simulation software as many researchers have confirmed (Mirahadi et al. 2019).

As the decision-making process highly influences the project outcomes, the application of simulations in that stage help estimate the building's performance (Abualdenien and Borrmann 2019). Especially, pedestrians' walking routes are essential when designing a building concerning emergency situations, for the pedestrians' behavior is strongly dependent on their environment (Low 2000). Specifically, the building's shape significantly influences efficient crowd routing considering safety and comfort (Hanisch et al. 2003). Thus, this paper aims to improve the integration of pedestrian dynamics simulations into the design phase, notably considering public buildings such as train stations, where emergency evacuation plays a vital role (Løvås 1994).

Typical results of pedestrian simulations can be comfort evaluation, walking routes visualization and insights about emergency situations. Nevertheless, integrating simulations into the workflow still consists of multiple steps, beginning with the building models' export from the BIM-authoring tool, converting and importing them into the simulation software, running the simulation, and postprocessing the final simulation results. Moreover, the usage of agent-based simulation models is computationally expensive, leading to long computation times, and is error-prone (Andriamamonjy et al. 2018). This time-consuming process may obstruct a full investigation of the design space.

To overcome these restrictions, the framework proposed in this paper employs Deep Learning (DL) methods to allow real-time predictions of pedestrians' behavior and walking times. To avoid expensive pedestrian dynamics simulations, especially Machine Learning (ML) approaches can serve as supportive tools or be used as complete replacements (Kim et al. 2019). Since BIM models include a massive set of information, we use them directly as input for the ML model and enable an immediate evaluation of pedestrians' behavior considering the interaction of multiple design options. The proposed method providing real-time evaluation allows interactive exploration of the solution space, thus enabling designers to find well-performing solutions in a shorter time. Since public buildings such as transport hubs must fulfill various requirements concerning evacuation time, this paper focuses on train stations.

The structure of this paper is as follows: Section 2 provides background information and related research. Section 3 introduces the concept of our approach step by step. In Section 4, details about the implemented neural network are given, whereas Section 5 presents the results. Finally, the last Section 6 summarizes the outcome and discusses future steps.

2 BACKGROUND AND RELATED WORK

2.1 *Performance-based building design*

The designing process of a building consists of many different steps, which result in various decisions and dependencies. Performance-based building design becomes a promising method to maximize the overall building's performance and reduce critical changes in the final project phases (Mehrbod et al. 2020). The accessibility of sufficient data and information is crucial, as decisions from early design phases can significantly impact the building's later performance and cost (Østergård et al. 2016). Especially BIM-based approaches allow the usage of comprehensive digital models within the design process, which helps improve the decision-making.

For the structural design of a building, A BIM-based optimization evaluation approach was developed by Hamidavi et al. (2020). With this, especially in the design phase, the coordination between architects and structural engineers is improved. Moreover, the authors of Röck et al. (2018) propose considering the building's materials for the Life Cycle Assessment and integrating parts into BIM. Hence, the potential effects of the building's materials become more comprehensible about their embodied energy.

2.2 *Pedestrian dynamics analysis and simulation models*

When designing public buildings such as shopping centers or train stations, especially emergency evacuation is essential (Løvås 1994). For efficient crowd routing inside a building, pedestrian dynamics analysis plays a vital role in safety and comfort while highly dependent on the building's shape (Hanisch et al. 2003). Research has shown that single pedestrians incline toward polygon-shaped walking routes, where visibility stimulates pedestrians to walk on straight paths for as long as possible. Moreover, while certain areas may appear crowded, pedestrians accept unknown detours and longer traveling times with or without intention (Helbing et al. 2001).

Furthermore, neither direct communication nor explicit concepts but intuitive awareness rule a crowd's self-organizational behavior, notably for crowds with unidirectional pedestrian flows (Helbing et al. 2005). When single pedestrians encounter stationary groups, they interpret them as obstacles and are prone to change their walking paths. Moreover, individual persons tend to adapt to the walking speed of other moving crowds within an overall crowded area (Yi et al. 2015).

The choice of the simulation model commonly depends on the number of virtual pedestrians (agents), where three main approaches were developed to model pedestrian behavior. On the one hand, individual agents and their reactions are modeled by microscopic approaches. On the other hand, macroscopic models reflect aggregated person streams. In addition, mesoscopic approaches can handle following individual agents and understanding group behavior (Ijaz et al. 2015). Regarding the findings that only rule-based methods may not necessarily lead to satisfactory results (Yang et al. 2020), Helbing et al. (2000) developed the more general (microscopic) social force model. In this approach, individual agents move with a certain velocity while their repulsive interaction forces consider obstacles and other agents.

When it comes to pedestrian crowds, the modeling instead follows a flow mechanism not considering the crowd's environment and individual agent's interplay, unlike modeling individual pedestrians' behavior. In particular, the authors of Hughes (2002) present the principle of continuum theory as the basis for crowd representation. In addition, using navigation- or guidance fields, the potential field model simulates multiple intentions of pedestrian crowds, introduced in Yang et al. (2020). Moreover, from fluid dynamics the aggregate dynamics, model is derived.

Although a common technique, strict cellular automata structuring leads to restrictions in representing reality, where obstacles or densities of pedestrian crowds may not be wholly cell-filling and, hence, lack accuracy (Biedermann et al. 2016). Hybrid models work as an alternative, where specific regions and areas can be assigned to particular modeling approaches representing individual behavior (Biedermann et al. 2021). Furthermore, the optimal steps model (OSM) is not focusing on a rigid spatial grid or dense crowds only. The OSM frees agents from a strict cell representation using continuous space, whereas a discrete stepwise movement is kept (Seitz and Köster 2012).

2.3 *Train stations and crowd dynamics*

In this research, we use train stations as an example of a facility whose design has a significant impact on the pedestrian flows, which has a major impact on the performant and save operation of that facility. It will thus serve as the subject of our investigations and as the basis for the proof-of-concept. Train station designs often vary concerning individual requirements. Usually, train stations provide waiting areas for pedestrians, where the uniform distribution of people over the respective spaces can be observed (Helbing et al. 2001). Besides, studies of crowd dynamics in train stations highlight a notable influence of waiting pedestrians. More precisely, inconveniently placed points of attraction and waiting pedestrians lead to an increase of up to 20% of walking time for arriving passengers leaving the platform area (Davidich et al. 2013).

As for the impact of different building elements on crowded areas in train stations, Ma et al. (2013) examined separation modules such as fences and pillars. Using pillars instead of other or no separation modules for non-unidirectional movements, the authors identified an increase in pedestrians' flow rate. Similarly, improvements in evacuation time could be observed for exit areas when placing pillars close to them (Frank and Dorso 2011).

2.4 *Deep learning*

Until now, the introduction of pedestrian behavior and various simulation models implied their complexity. As a result, performing pedestrian simulations for incredibly complex building designs can quickly yield high computation time. To overcome this issue, the research community more and more involves methods of Artificial Intelligence (AI). Naming a specific category of AI methods, predictive tools can replace time-consuming computations with the support of ML approaches. In doing so, a surrogate function is found and applied to the problem.

DL approaches became favored support, especially for dealing with distinct data types and various problems. More specifically, multiple architectures of Artificial Neural Networks (ANNs or NNs) exist to deal with different tasks and purposes. For instance, object detection and segmentation in images and as well as natural language processing attain different success rates.

A well-known feedforward NN is the Multilayer Perceptron (MLP), famous for solving various problems (Nielsen 2015). The MLP is arranged in multiple (hidden) layers that contain any number of connected computational nodes. These nodes store single values that are processed in one direction. A suitable number of nodes and layers for solving a given task sufficiently is essential for the resulting individual NN. The goal of the NN is mapping a given input to the desired output, also known as a classified label, training the network to customize the network to a particular problem. Commonly, applying a backpropagation algorithm to the network makes its parameters optimized and the accuracy improved (Nielsen 2015).

When it comes to matrix-like data such as images, Convolutional Neural Networks (CNNs) have become an efficient way of achieving results. Again, the CNN follows a feedforward architecture with multiple layers. Typically, each layer carries out a set of computations. In the first step, a kernel performs the convolution operation for a given input matrix and results in a feature map. Moreover, multiple feature maps can be computed by different kernels in parallel within the same layer returning a feature set, where a kernel can be described as a filter (Goodfellow et al. 2016).

Following the convolution, each element of the feature map is processed by a nonlinear activation function, for instance, the rectified linear unit function. Finally, down-sampling reduces the matrix dimensions, commonly done by a pooling operation such as maximum pooling. For all following layers, down-sampling helps reduce the computational effort. Additionally, for a given dataset, CNNs can detect and filter out patterns (features) (Goodfellow et al. 2016).

The training of a neural network is a process with various parameters and options. Although several optimization techniques help improve the training process, a sufficient amount of data is essential. Moreover, underfitting can occur by providing too little data. The same training data can often lead to overfitting since the network may adapt to the specifics of the examples to a too large extent. An intentional increase of uncertainty within the model can be applied to reduce overfitting by using regularization methods like the dropout. Thereby, the activated nodes are varied almost randomly, leading to the prevention of co-adaptions and, hence, to improved computations (Srivastava et al. 2014).

Furthermore, a network's training process can be enhanced by batch normalization (Santurkar et al. 2018). Each layer's inputs are normalized before the actual activation of the following computational nodes. Then again, deep dependencies between multiple layers may be partially relieved, also known as decreasing the covariate shift. On the other hand, the integration of batch normalization helps reduce the necessity of regularization methods like dropout (Ioffe and Szegedy 2015).

After training an NN architecture for a given task, the gained knowledge can also be used for other purposes. Moreover, a pre-trained network can be re-trained for a different dataset or partially reused for new tasks. This process is described as transfer learning and a standard procedure for improving network results, especially when using CNNs (Ribani and Marengoni 2019). Popular network architectures can be found online, such as the VGG16 pre-trained on the ImageNet dataset for classification problems (Simonyan and Zisserman 2014).

Giving some examples, the authors of Nishida and Hotta (2018) used CNNs to detect and distinguish cell particles from non-cell particles based on image data. In building design, Geyer and Singaravel (2018) developed a component-based model to estimate heating- and cooling energy within a building. In another approach, flow control, performance, and optimization of fluid dynamics calculations were improved using ML methods (Brunton et al. 2020). Finally, the understanding of pedestrians' walking routes and densities in a given environment could be predicted in (Clever et al. 2021).

3 METHODOLOGY

3.1 *Hypothesis and aim*

In this paper, the hypothesis is that real-time predictions can replace time-consuming pedestrian dynamics simulations with DL methods that relate the design information of a building model with individual simulation results. From here, two questions arise: (1) what is a suitable representation of the geometric and semantic design information? (2) Which aspects of the simulation results shall be predicted? These questions are essential for a suitable NN architecture, where layers and parameters must carefully be configured.

Contrary to Clever et al., 2021, in this paper we focus on the evacuation times. Thus, we propose a framework that automatically generates a training dataset and predicts results directly from the BIM model, considering specific simulation parameters.

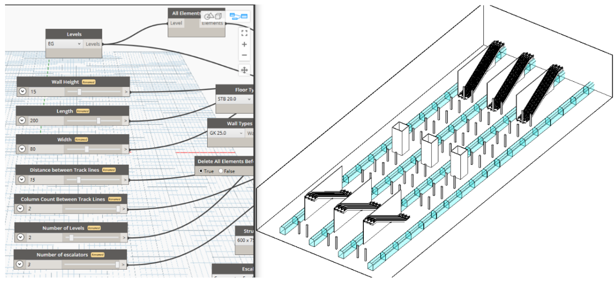

Figure 1: Example of parametric model (Revit/Dynamo)

Table 1: Parameter values for generic train station variation

Abbreviation	Meaning	Variations
F	No. of floors	2
T	Distance of tracks	15, 25
W	No. of tracks	2, 3, 4, 5
L	Station length	150, 200, 250, 300
H	Floor height	15, 25
E	No. of escalators	1, 2, 3
P	Agents per coach	5, 20, 50

Considering the work of Clever et al. (2021), the generic train station models created by a parametric model are reused for this paper. Instead of predicting heatmaps or tracing maps, the present paper focuses on predicting the respective evacuation times of a given train station design. Moreover, the corresponding IFC exports and pedestrian simulations are carried out. In the scope of this paper, we use the pedestrian dynamics simulator crowd:it (Accu:rate 2022), which is based on the optimal steps model (OSM) (Seitz and Köster 2012), for generating the training and validation data. With the available simulation results, post-processing is applied, and the agents' walking times are extracted for each model variation. Together with the train station models, the generated dataset is used for training a neural network.

3.2 *Parametric models*

The available train station models by Clever et al. (2021) were generated by a parametric model using Autodesk Revit (Autodesk 2022) and Dynamo (Autodesk 2021). By varying parameters of the design's geometry according to Table 1, 450 different generic train station models are considered for the dataset. The parametric platform presented in Figure 1 has three escalators at each end, four track lines, a row of two columns, and an elevator in between.

3.3 *Floorplan representation*

As mentioned earlier, specific zones are marked in the BIM models, necessary for the simulation. By assigning different colors to the different zones, a floorplan representation of the models is used as input for the neural network, alongside a vector containing the models' metadata, as presented in Table 1. Concerning the labeling, Figure 2 shows an example where the pink color represents spawning zones, and white spaces are walkable areas. The output of the simula-

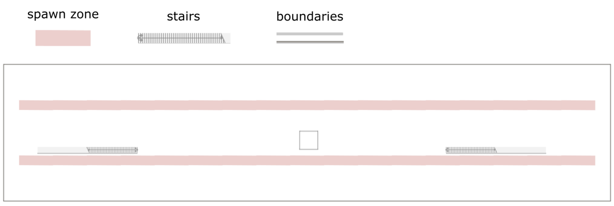

Figure 2: Colored floorplan example

tions is plain numbers serving as the models' overall evacuation times. In the simulation results (used for training and validation), each time step of each agent is given, whereas the very last time step would be the time needed to evacuate the building entirely.

4 NEURAL NETWORK ARCHITECTURE

As discussed in Section 2.4, multiple approaches exist to design a neural network architecture depending on the input and output data. This paper uses floorplan images and metadata of the underlying parametric building models as input information, while we predict a single value, i.e., the evacuation time, as output. Hence, we assemble different sub-networks according to the respective data structure to receive the final network architecture. The modular architecture is presented in Figure 3. In the simulation results, all individual time steps are saved for each agent, where the very last step counts as the building's evacuation time. To avoid including incorrect results, e.g., by agents got stuck, we omitted the last 5% of these time steps.

The metadata is structured as a vector of numbers containing various information about the building model. Thus, the metadata input will be processed by an MLP (MLP-in), with three hidden layers.

For the floorplan images, we use the basic CNN design of the VGG16 according to (Simonyan and Zisserman 2014) where the input layer is created considering an image resolution of 256 * 256 pixels. Moreover, the dense layers and the final classification layer of the VGG16 are excluded since no image classification is desired.

Finally, a concatenation layer (concat) combines the MLP-in and the CNN outputs to create a joined input for the predictive part of the network. Again, an MLP (MLP-out) is used to process the combined input information of the metadata and the floorplan images.

During network hyperparameter tuning, especially the CNN base model offers three different modifications we considered for each training setup: (1) the pre-trained weights of the VGG16 network are used as given. In contrast, an update of the weights during training is restricted. (2) The pre-trained weights of the VGG16 are used, but an update of the weights during training is possible. (3) The initial VGG16 weights are not explicitly defined and updated at each training epoch if necessary. The setup cases are summarized in Table 2.

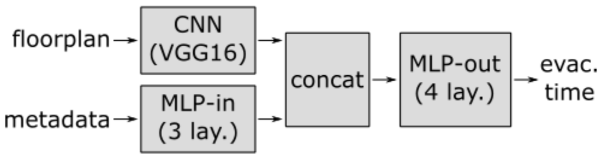

Figure 3: Structure of the neural network architecture

Table 2: Neural network training setup cases

Training case	Meaning for CNN
1	VGG16 pre-trained, no weight updates
2	VGG16 pre-trained, weight updates
3	VGG16 not pre-trained, weight updates

According to transfer learning, pre-trained weights shall help improve the overall network training and, thus, its efficiency.

5 NEURAL NETWORK RESULTS AND EVALUATION

In a first step, we split the dataset into two subsets of training and testing data. We created 450 different train station models, including their respective simulation results. We reserved 25% of the models (∼120) for testing. Due to the different parameters of the models, different resolutions for the respective floorplan images occur, which we adjusted to one equal resolution by resizing. Additionally, we used data augmentation for the training data, doubling the number of projects to 660 while applying random rotation and mirroring to the floorplan images.

Furthermore, 20% of the training data was used as validation data in every epoch. Various training setups lead to an optimal batch size of 16 and a total of 150 epochs to avoid overfitting, where the number of hidden layers for the MLP-in and MLP-out are three and four, respectively (see Figure 3). Due to a single value prediction as to the network's result, we used the mean squared error (MSE) as the loss function to compare the prediction with the ground truth during training and validation. Moreover, we used the Adam optimizer and a learning rate of 0.001.

Beginning with training case 1 (see Table 2), the MSE of both training and validation data is shown in Figure 4. The training loss went below an MSE of 1000 after 7 epochs and circle approx. 100 after approx. 20 epochs, while the validation loss needed approx. 20 epochs to go significantly below an MSE of 1000 and circle a loss of approx. 600.

A comparable difference between training and validation loss, as in Figure 4, occurred during all training iterations. Moreover, one must consider that the MSE implies squared error values. The evaluation of the testing dataset was performed on the difference between prediction and ground truth, normalized with respect to the ground truth. The mean of all differences yielded in 20%, while the ground truth's minimum and maximum evacuation times are approx. 90s and 480s, respectively. Training time was approx.

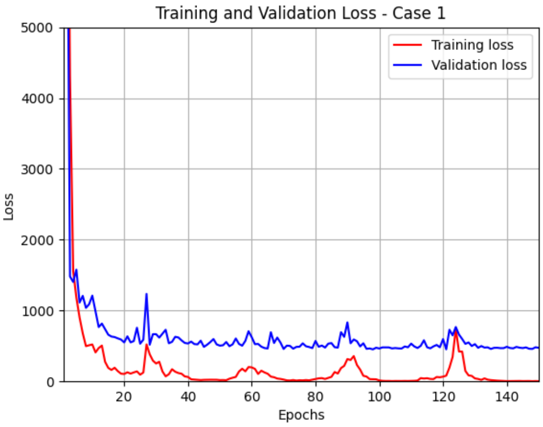

Figure 4: Case 1 - training and validation loss

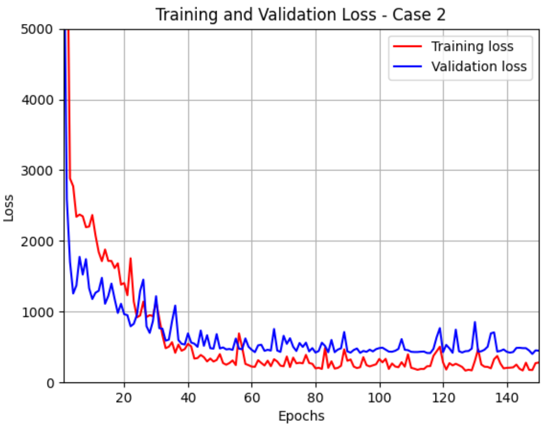

Figure 5: Case 2 - training and validation loss

6.5min, while the simulation time's minimum and maximum computation times are approx. 0.5min and 262.5min, respectively.

For training case 2, the training and validation loss needed approx. 30 epochs for the MSE to go below 1000, as shown in Figure 5. For one thing, the validation loss is slightly lower than in case 1. Then again, the training loss is higher than in case 1 while staying lower than the validation loss. Compared with case 1, the losses of case 2 need around 20 epochs more to decrease to similar loss values. One possible reason can be the update of the CNN's weights in case 2, where also the update of the MLPin and MLP-out are considered. Concerning the testing dataset, a mean of 19% was achieved. In this regard, a slight difference to case 1 exists. Moreover, training time is almost double with approx. 12.5min.

Lastly, Figure 6 shows the MSE for training and validation data of training case 3. Similar to case 2, for both losses, it took approx. 30 epochs to decrease to an MSE below 1000. Both graphs are reasonably comparable to case 2 concerning the loss values. However, the evaluation of the testing data yields a mean of 15%, which is significantly better than for cases 1 and 2. In addition, for case 3, Figure 7 shows a histogram of the testing data evaluation. Altogether, the accuracy of the prediction was 15% with a standard deviation of 8%. We noticed that the majority

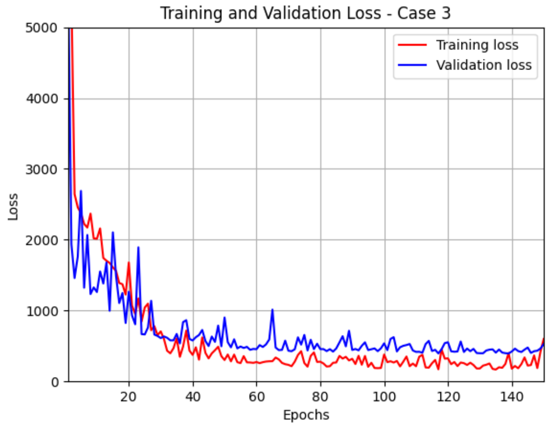

Figure 6: Case 3 - training and validation loss

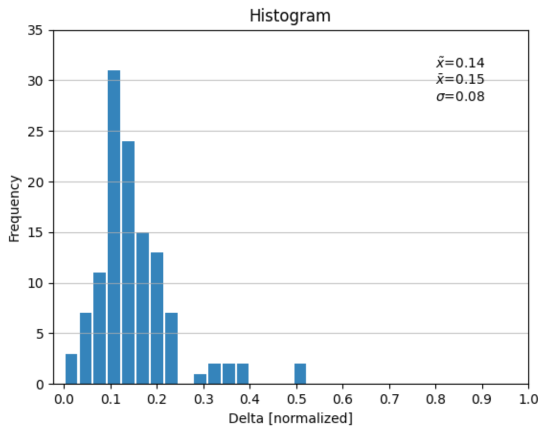

Figure 7: Case 3 – difference between prediction and ground truth

Table 3: Summary of results

Training case	Mean [%]	Training time [min]
1	20	**6.5**
2	19	12.5
3	**15**	12.5

(i.e. 80% out of 120 test samples) produce less than 20% deviation in the evacuation time (in secs) from the ground truth. As in case 2, training time is approx. 12.5min, thus, double the time as in case 1.

Overall, the comparison of the results shows differences between the three training setup cases. Although for case 1, the loss decreases faster than in cases 2 and 3, the accuracy measured by the mean values is best for case 3. Nevertheless, the final evaluation of a mean of 15% for case 3 depends on the particular use case, meaning the necessary individual accuracy for a given problem. Otherwise, for cases 2 and 3, the computation time of the network training is highest. Again, the importance of the necessary time training the network is to be assessed concerning a given problem. A summary of the results can be found in Table 3.

6 CONCLUSION AND FUTURE WORK

Conventional pedestrian simulations can easily lead to high computation times and effort concerning computational resources. Hence, a real-time prediction of the evacuation time for a given building model would improve the workflow significantly. In this paper, we designed a modular NN, combining different NN architectures for multiple input- and output data types, providing a real-time prediction of the evacuation time for a given train station geometry. The presented approach considered different neural network architectures, such as the MLP for metadata of the BIM model and the CNN for the corresponding floorplan images. Moreover, we created a new primary network architecture with an individual combination of subnetworks, including the possible option of using pre-trained weights for the CNN at training time.

The results of the approach allow the conclusion that pre-trained weights of the CNN while prohibiting their update during the entire network's training may be helpful when the training time of the network shall be reduced. Otherwise, depending on the particular problem, the accuracy of the entire network based on pre-trained weights for the CNN may not be sufficient, whereas an overall trained network without pre-trained weights shows better predictions.

Unlike the prediction of pedestrian trajectories in public buildings (Lui et al. 2021), and evacuation routes (Zhang et al. 2021), this paper presents the implementation of an NN-based real-time prediction of evacuation times. In the future, we aim to train a model that can predict both the overall evacuation time and the pedestrian trajectory. An immediate improvement of the presented work can be made by training on more complex data (e.g., multiple floors). In our case, a generic train station model may be relatively similar, leading to misinterpretations of changes in the building design.

We see the enormous advantage of including predictive tools in building design. Especially for exploring different building layouts in the early design stages, real-time predictions can significantly help explore the solution space while considering various performance criteria such as evacuation times. Consequently, project time and computational effort will be reduced while developing the optimal design solution that considers multiple factors and dependencies can be fulfilled.

7 ACKNOWLEDGMENTS

We gratefully acknowledge the support of mFUND – Bundesministerium für Digitales und Verkehr in Germany for funding the research project BEYOND.

REFERENCES

Abualdenien, J. & A. Borrmann (2019, apr). A meta-model approach for formal specification and consistent management of multi-LOD building models. *Advanced Engineering Informatics 40*, 135–153.

Abualdenien, J., P. Schneider-Marin, A. Zahedi, H. Harter, H. Exner, D. Steiner, M. Mahan Singh, A. Borrmann, W. Lang, F. Petzold, M. König, P. Geyer, & M. Schnellenbach-Held (2020). Consistent management and evaluation of building models in the early design stages. *Journal of Information Technology in Construction 25*, 212–232.

Accu:rate (2022). crowd:it – the software for state-of-the-art planners.

Andriamamonjy, A., D. Saelens, & R. Klein (2018, jul). An automated IFC-based workflow for building energy performance simulation with Modelica. *Automation in Construction 91*, 166–181.

Autodesk (2021). Dynamo BIM.

Autodesk (2022). Revit Software.

Biedermann, D. H., J. Clever, & A. Borrmann (2021, feb). A generic and density-sensitive method for multi-scale pedestrian dynamics. *Automation in Construction 122*, 103489.

Biedermann, D. H., C. Torchiani, P. M. Kielar, D. Willems, O. Handel, S. Ruzika, & A. Borrmann (2016). A Hybrid and Multiscale Approach to Model and Simulate Mobility in the Context of Public Events. *Transportation Research Procedia 19*, 350–363.

Borrmann, A., M. König, C. Koch, & J. Beetz (Eds.) (2018). *Building Information Modeling*. Cham: Springer International Publishing.

Brunton, S. L., B. R. Noack, & P. Koumoutsakos (2020, jan). Machine Learning for Fluid Mechanics. *Annual Review of Fluid Mechanics 52*(1), 477–508.

BuildingSMART (2020). IFC4 Documentation.

Clever, J., J. Abualdenien, & A. Borrmann (2021). Deep learning approach for predicting pedestrian dynamics for transportation hubs in early design phases. In *28th International Workshop on Intelligent Computing in Engineering*, Berlin, Germany.

Davidich, M., F. Geiss, H. G. Mayer, A. Pfaffinger, & C. Royer (2013, dec). Waiting zones for realistic modelling of pedestrian dynamics: A case study using two major German railway stations as examples. *Transportation Research Part C: Emerging Technologies 37*, 210–222.

Frank, G. A. & C. O. Dorso (2011). Room evacuation in the presence of an obstacle. *Physica A: Statistical Mechanics and its Applications 390*(11), 2135–2145.

Geyer, P. & S. Singaravel (2018). Component-based machine learning for performance prediction in building design. *Applied Energy 228*(July), 1439–1453.

Goodfellow, I., Y. Bengio, & A. Courville (2016). *Deep Learning*. MIT Press.

Hamidavi, T., S. Abrishami, & M. Hosseini (2020, nov). Towards intelligent structural design of buildings: A BIM-based solution. *Journal of Building Engineering 32*, 101685.

Hanisch, A., J. Tolujew, K. Richter, & T. Schulze (2003). Online simulation of pedestrian flow in public buildings. In *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*, pp. 1635–1641. IEEE.

Helbing, D., L. Buzna, A. Johansson, & T. Werner (2005). Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. *Transportation Science 39*(1), 1–24.

Helbing, D., I. Farkas, & T. Vicsek (2000). Simulating dynamical features of escape panic. *Nature 407*(6803), 487–490.

Helbing, D., P. Molnár, I. J. Farkas, & K. Bolay (2001). Self-organizing pedestrian movement. *Environment and Planning B: Planning and Design 28*(3), 361–383.

Hughes, R. L. (2002). A continuum theory for the flow of pedestrians. *Transportation Research Part B: Methodological 36*(6), 507–535.

Ijaz, K., S. Sohail, & S. Hashish (2015). A Survey of Latest Approaches for Crowd Simulation and Modeling using Hybrid Techniques. *17th UKSIM-AMSS International Conference on Modelling and Simulation*, 111–116.

Ioffe, S. & C. Szegedy (2015, feb). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

Kim, J., J. Moon, E. Hwang, & P. Kang (2019). Recurrent inception convolution neural network for multi short-term load forecasting. *Energy and Buildings 194*(2019), 328–341.

Knotten, V., F. Svalestuen, G. K. Hansen, & O. Lædre (2015). Design management in the building process - a review of current literature. *Procedia Economics and Finance 21*, 120–127.

Løvås, G. G. (1994, dec). Modeling and simulation of pedestrian traffic flow. *Transportation Research Part B: Methodological 28*(6), 429–443.

Low, D. J. (2000, sep). Following the crowd. *Nature 407*(6803), 465–466.

Lui, A. K.-F., Y.-H. Chan, & M.-F. Leung (2021, dec). Modelling of Destinations for Data-driven Pedestrian Trajectory Prediction in Public Buildings. In *2021 IEEE International Conference on Big Data (Big Data)*, pp. 1709–1717. IEEE.

Ma, J., S. M. Lo, W. G. Song, W. L. Wang, J. Zhang, & G. X. Liao (2013). Modeling pedestrian space in complex building for efficient pedestrian traffic simulation. *Automation in Construction 30*, 25–36.

Mehrbod, S., S. Staub-French, & M. Tory (2020, jan). BIM-based building design coordination: processes, bottlenecks, and considerations. *Canadian Journal of Civil Engineering 47*(1), 25–36.

Mirahadi, F., B. McCabe, & A. Shahi (2019, may). IFC-centric performance-based evaluation of building evacuations using fire dynamics simulation and agent-based modeling. *Automation in Construction 101*, 1–16.

Nielsen, M. A. (2015). *Neural networks and deep learning*, Volume 2018. Determination press San Francisco, CA.

Nishida, K. & K. Hotta (2018, oct). Robust cell particle detection to dense regions and subjective training samples based on prediction of particle center using convolutional neural network. *PLOS ONE 13*(10), e0203646.

Østergård, T., R. L. Jensen, & S. E. Maagaard (2016, aug). Building simulations supporting decision making in early design – A review. *Renewable and Sustainable Energy Reviews 61*, 187–201.

Ribani, R. & M. Marengoni (2019, oct). A Survey of Transfer Learning for Convolutional Neural Networks. In *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, pp. 47–57. IEEE.

Röck, M., A. Hollberg, G. Habert, & A. Passer (2018, aug). LCA and BIM: Visualization of environmental potentials in building construction at early design stages. *Building and Environment 140*, 153–161.

Santurkar, S., D. Tsipras, A. Ilyas, & A. Madry (2018, may). How Does Batch Normalization Help Optimization?

Seitz, M. J. & G. Köster (2012, oct). Natural discretization of pedestrian movement in continuous space. *Physical Review E 86*(4), 046108.

Simonyan, K. & A. Zisserman (2014, sep). Very Deep Convolutional Networks for Large-Scale Image Recognition.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, & R. Salakhutdinov (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research 15*(56), 1929–1958.

Yang, S., T. Li, X. Gong, B. Peng, & J. Hu (2020). A review on crowd simulation and modeling. *Graphical Models 111*(October 2019), 101081.

Yi, S., H. Li, & X. Wang (2015, jun). Understanding pedestrian behaviors from stationary crowd groups. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3488–3496. IEEE.

Zhang, Y., Z. Chai, & G. Lykotrafitis (2021, jun). Deep reinforcement learning with a particle dynamics environment applied to emergency evacuation of a room with obstacles. *Physica A: Statistical Mechanics and its Applications 571*, 125845.