



A data-driven neural network approach to simulate pedestrian movement

Xiao Song*, Daolin Han, Jinghan Sun, Zhengui Zhang

Beihang University (BUAA), Beijing 100191, China

HIGHLIGHTS

- A multi-scenario adaptive neural network is proposed to model pedestrian behavior.
- The network inputs include normalization of relative positions among pedestrians and speed direction transfer algorithm.
- Both non-competitive and competitive tests show the validity of the proposed network.

ARTICLE INFO

Article history:

Received 1 February 2018
Received in revised form 28 April 2018
Available online 21 June 2018

Keywords:

Pedestrian flow
Artificial intelligence
Neural networks
Social force

ABSTRACT

As artificial intelligence becomes a research hotspot, more and more researchers are trying to apply it to numerous domains. It is therefore interesting and challenging to apply data-driven neural network technology to pedestrian movement modeling to test its effect against traditional social force model which can be applied to manifold pedestrian scenarios. Recent neural network based pedestrian movement simulation studies often train the network with only one scenario and then test within this scenario with various parameters. To make a more adaptive neural network, we propose a four layer network to learn multiple scenario data by normalization of relative positions among pedestrians, transferring velocity vector to scalar and incorporating more path planning information, and thus to make it more adaptive to various scenarios. Simulation results show that the proposed neural network approach can be applied to several typical pedestrian scenarios including counterflow and evacuation. Moreover, it shows more realistic speed-density curve and generates less trajectory fluctuations compared with social force model. Therefore, the proposed method is capable of generating more realistic pedestrian flow in multiple scenarios.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Pedestrian model has long been a problem faced by many government officers, physicists, and computer scientists. From the perspective of pedestrian model granularity, current works can be divided into two classes: macroscopic models and microscopic models based on a simple or complex topology. The two resolution models can be applied to situations such as tall buildings [1,2], cities [3,4], airports, and subway stations, with the goals of evacuation guidance, finding exits [1,2], or evacuation training [5]. Generally, these pedestrian modeling approaches are knowledge-driven, which involve formal models and simulating them to understand more about the system. This means these models' parameters are mostly hand-crafted designed and implemented according to experts' observations [6]. For crowd dynamics, it is very difficult to exact

* Corresponding author.

E-mail address: songxiao@buaa.edu.cn (X. Song).

each individual's characteristics and build a universal yet specific pedestrian model. Therefore, most existing methods often focus on flow dynamics instead of characterizing each figure's specific psychological and physical parameters [7]. And the parameter values are chosen based on statistical results and regression functions. This modeling process is often slow and introduces expert bias [8–10].

To tackle these problems of traditional methods, artificial intelligence oriented neural network (ANN) approaches have emerged and prevailed. As long as we have a large volume of data, we are able to train the network to learn realistic human behavior without much expertise knowledge. For pedestrian model, we believe the movement of each people is mainly affected by his own will to the goal and neighboring figures and obstacles. By collecting data from real-life videos using recent high resolution cameras, we can record detailed video sequences and then build a neural network based prediction model to simulate pedestrian maneuver behaviors. In this paper, a neural network will be designed to learn the movement of each pedestrian's neighbors and predict his velocity at the next step. Using multi-scenario data to train the network, the intelligent behavior of pedestrians will be learned and compared with the knowledge-driven social force model (SFM).

This data based learning approach might not only make the pedestrian simulation more realistic but also avoid building the model from scratch. Moreover, another important property of the neural network is adaptability, which means what the network learned is not just constrained to a specific scenario but an adaptive behavior which can estimate pedestrian movement in various scenarios including counterflow, evacuation etc. To fulfill this, we will design a four layer network to learn multiple scenario data by normalization of relative positions among pedestrians, transferring velocity vector to scalar and incorporating more path planning information, and thus is to make it more adaptive to various scenarios.

The outline of this paper is as follows. Section 2 discusses the state-of-the-art works. Section 3 presents the design of our artificial neural network. Section 4 presents the data acquisition and network training. Section 5 presents the simulation results compared with SFM. Section 6 draws the conclusion.

2. Related works

For the past two decades, there are two main methods modeling pedestrian behaviors, which are social force model (SFM) [11], cellular automata model (CAM) [12] and velocity model (VM) [13,14]. Basically these three modeling approaches solve the collision avoidance problem. CAM often employs floor field model to lead the pedestrian to his goal and game theory [15] to solve the conflict when two pedestrians want to move to the same cell [16,17]. Velocity model is employed in robotic applications, which derive appropriate conditions for collision-free motion by solving a certain low-dimensional linear program. Meanwhile, SFM creates a repulsive force oriented equation to avoid touching and needs path planning algorithm to people's target place [18].

From the perspective of position and velocity updating, SFM and VM are continuous, while CAM is discrete. VM assumes that each robot acts fully independently and does not communicate with other robots. This assumption is to some extent not suitable for crowd simulation because people can communicate with each other easily. Continuous SFM often needs more computation than CAM but it is more precise to describe pedestrian movement and is continuous to present direction and magnitude of speed while CAM is discretized. Along with the ever increasing computation power of computers, it might be more desirable to employ the more accurate continuous model to generate pedestrian flows [19,20].

The continuous force based social pedestrian model often needs to define a set of maneuver rules including how to move to the goal and how the pedestrians interact with each other. Typical scenarios include a crosswalk counterflow or room evacuation.

There have been numerous studies that tried to improve the original social force model since it was proposed in Helbing [11,21]. For example, Fernando [22] proposes a spheropolygon shape model to simulate a dense counterflow during a music event. Farina and Fontanelli [18] include heading force and torque to improve the realism of the trajectories. Langston [23] used a three-circle model to simulate the pedestrians' rotation behavior. Although these improvements are meaningful to study crowd dynamics, few of them incorporate realistic people data to calibrate and validate their motion models due to various reasons such as lack of concrete data that corresponds to their proposing models.

Meantime, it is still challenging to develop a more realistic pedestrian model due to the intrinsic complexity of human behaviors. For instance, the force based model may generate unrealistic backward force [8,24]. Furthermore, force based models are generally handcrafted using expert knowledge (knowledge-driven). However, this process is slow and introduces modeler bias [16,24].

To overcome these shortcomings, some researchers have tried to introduce artificial intelligence approaches to solve pedestrian movement modeling problems. Ma and Lee [8] proposed an artificial intelligence-based approach for simulating pedestrian movement of counter-flow. In this approach, a large volume of microscopic pedestrian movement behavior types were collected and encapsulated into an artificial neural network via network training. Their results show their neural network is capable of simulating human-like zebra crossing. Meantime, Alahi and Geol [25] follow the recent success of Recurrent Neural Network (RNN) models for sequence prediction tasks and propose an LSTM (Long Short-Term Memory) model which can learn general human movement and predict their future trajectories. Their results show that their social-LSTM is often more closer to ground truth than social force model. Zainuddin [26] incorporates a probabilistic neural network for determining the decision-making ability of the pedestrians and simulate an exit-selection phenomenon. Szemes [27] investigates the obstacle avoidance based mobile robot control system, in which the observation of human walking behavior is applied to train fuzzy-neural networks.

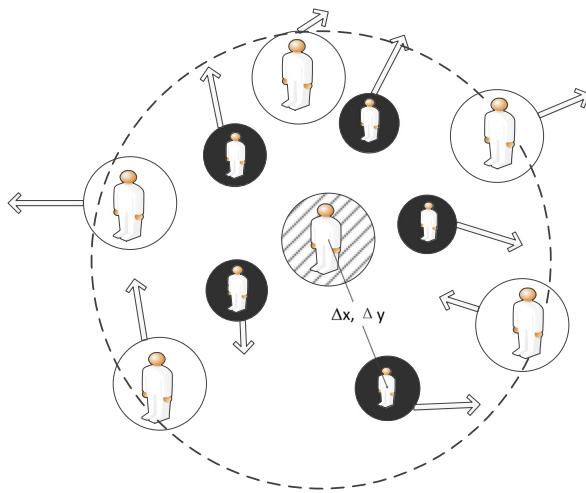


Fig. 1. Within the 3 m circle, five nearest neighbors are considered. The radius of the dotted circle is 3 m, which is the scope of SFM.

All these works are useful but most of their ANNs actually learned one specific scenario, which means they trained their neural network model to learn a concrete model. Although the results are promising, these studies are not adaptable to other scenarios because they did not incorporate multi-scenario data, corresponding inputs and effective ways to promote learning efficiency. In fact, training ANN with another scene might incur the problem of diminishing the learning effect of learned scenes because the network weights may be neutralized. Thus, one emerging challenge is to design and train a network with data of various environments yet make the network show more realistic behavior than popular force-based methods such as SFM. Meantime, we found it is impossible to design an ANN which can learn both competitive and non-competitive pedestrians. As such, our idea is to design two ANNs that have the same inputs and outputs but have different weights to adapt to competitive and non-competitive scenarios respectively. Both ANNs are trained with real data and applied to counterflow and evacuation scenarios to test their validity.

3. Design of artificial neural network (ANN)

3.1. Basic inputs and outputs of ANN

To conduct more realistic and efficient pedestrian dynamics, this paper tries to develop an artificial neural network (ANN) [17,26,28–31] pedestrian model which not only possesses the main functions of the well-known social force model (SFM) [23,27,32–34] but also use the real-life data to learn the movement behaviors of crowds. When designing the ANN, it is important to determine the input parameters and we believe the social force model is a good reference, which is employed to compute the interaction among pedestrians and obstacles. The inputs of the proposed ANN are designed as follows.

- (1) The desired moving direction computed by A^* algorithm, which is normed in this paper and denoted by \vec{A}^* . This direction can represent the desired direction to the goal and the planned direction to avoid obstacles. It represents \vec{F}_i^d .
- (2) The current speed vector includes x and y values, i.e., $\vec{v}_t = (v_{t,x}, v_{t,y})$.
- (3) An array W is employed to store the subject figure's surrounding crowd information. To determine the number of interacting neighbors, the works of Ma [8], Bialeka and Cavagna [35] are considered. These two works suggested 5 and 7 nearest neighbors respectively. Correspondingly, we implemented experiments to evaluate this number with respect to our ANN method. Results show that both the two trajectories in Fig. 10 and density curves in Fig. 11 are very similar. To achieve a trade-off between the computation complexity and fidelity of the model, five nearest neighbors are considered to compute the interaction forces of F_i^r . (See Fig. 1.)

$$W = \{\Delta x_i, \Delta y_j, \Delta Vx_k, \Delta Vy_l | 1 \leq i \leq 5, 6 \leq j \leq 10, 11 \leq k \leq 15, 15 \leq l \leq 20\} \quad (1)$$

where, W is the neighbor array, which has 20 elements. Note that ANN needs fixed number of input and the number of neighbors might be less than 5. To tackle this, we propose to use normalized Δx and Δy to represent the relative position. When the neighbor is near to the subject person, w_i and w_j is near 1, shown respectively in Eqs. (2) and (3). When the number of the near neighbors is less than 5, w_i and w_j is 0. The devised equations are as follows.

$$x_i \geq x, w_i = \frac{x_i - x}{3} = \frac{\Delta x_i}{3}; x_i < x, w_i = -(1 + \frac{\Delta x_i}{3}), 1 \leq i \leq 5 \quad (2)$$

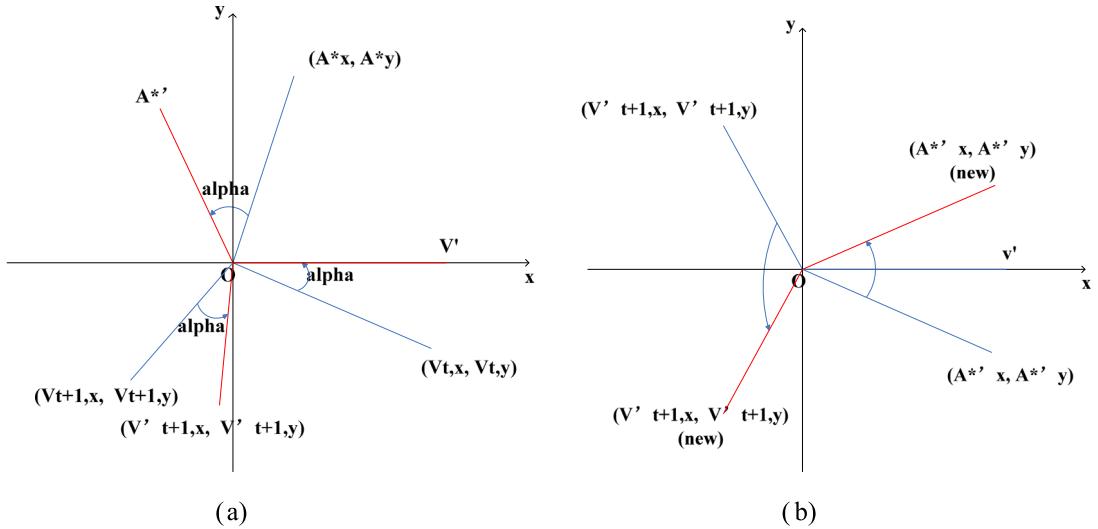


Fig. 2. (a) Step 1 to 3, to rotate \vec{v}_t and related parameters. (b) Step 3-1 and 3-2, to rotate \vec{A}'^* and \vec{v}'_{t+1} if \vec{A}^* is located in the third or fourth quadrant.

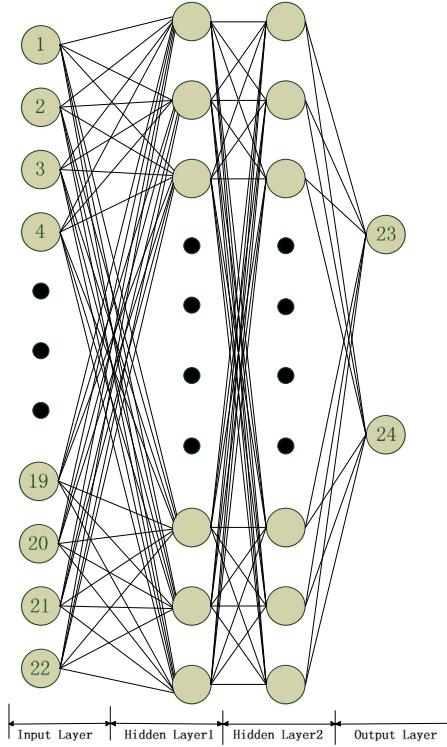


Fig. 3. Structure of the proposed ANN. The parameters are explained in Table 1.

$$y_i \geq y, w_j = 1 - \frac{\Delta y_i}{3}; y_i < y, w_j = -(1 + \frac{\Delta y_i}{3}), 6 \leq j \leq 10 \quad (3)$$

$$w_k = \Delta Vx_i = Vx_i - Vx, 11 \leq k \leq 15 \quad (4)$$

$$w_l = \Delta Vy_i = Vy_i - Vy, 16 \leq l \leq 20 \quad (5)$$

where $(x, y), (V_x, V_y)$ are the position and speed of the subject person, respectively. $(x_i, y_i), (V_{x_i}, V_{y_i})$ are the position and speed of the neighbor i . The relative distance, $x_i - x$, is divided by 3 m to scale to $(0, 1)$, where 3 m is the radius of the distance considered by the subject person.

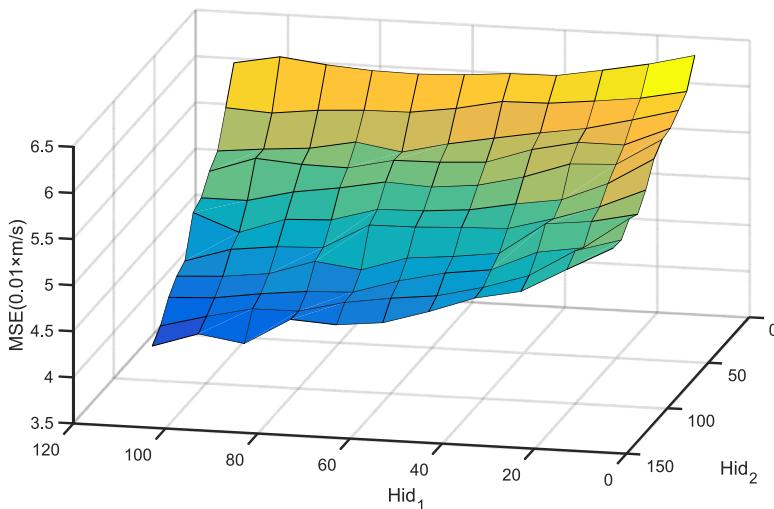


Fig. 4. The corresponding mean square error (MSE) against various number of the neurons of two hidden layers.

Table 1

Inputs and outputs of the proposed ANN.

Layer	No.	Meaning	Unit
Pre-input	1	\vec{v}_t	m/s
	2	The angle between \vec{A}^* and unit vector $(1, 0)$	rad
	3-7	$W[1-5]$, Horizontal relative distance between the subject person and the nearest 5 neighbors	m
	8-12	$W[6-10]$, Vertical relative distance between the subject person and the nearest 5 neighbors	m
	13-17	$W[11-15]$, Horizontal relative velocity between the subject person and the nearest 5 neighbors	m/s
	18-22	$W[16-20]$, Vertical relative velocity between the subject person and the nearest 5 neighbors	m/s
Input	1	V' , a scalar that stores the speed magnitude	m/s
	2	The angle between $\vec{A}^{*'}$ and unit vector $(1, 0)$	rad
	3-22	W'	
Output	23	V_{t+1}', x	m/s^2
	24	V_{t+1}', y	
Post-output	1	V_{t+1}, x	m/s^2
	2	V_{t+1}, y	

Table 2

Training Scenarios. Each evacuation scenario is competitive or non-competitive, with different obstacles and door widths. As such, the number of trained evacuation scenarios is $2^*3^*2 = 12$. Meantime, the counterflow of crossing road is non-competitive and the experiments were repeated twice.

Scenario	Comp/Non-comp	Obstacle (Rectangle with top-left and bottom-right coordinate)	Door width	No. of scenarios
Evacuation in a room	Comp, Non-comp	•Rect 1 (9.75, 7.25), (10.93, 6.67); •Rect 2 (10.26, 7.5), (10.84, 6.32); •Rect 3 (9.33, 7.5), (9.91, 6.32)	0.5 m, 1.0 m	12
Counterflow	Non-comp	-	-	2

3.2. Transfer layer

The walking or running directions of pedestrians are continuous. This incurs a problem that it is impossible for ANN to learn all the possible directions because it is continuous. For instance, when the ANN learns some data the pedestrian walks toward east and the speed vector is $(1, 0)$. When the direction of the person changes to $(0, -1)$, $(-1, 0)$ or any other directions, we cannot apply the result of $(1, 0)$ directly to this case. As such, we propose a transfer layer to solve this problem. Note that this layer also decreases the learning epochs enhancing the performance of ANN learning.

In this transfer layer, all the pedestrians' directions are rotated counterclockwise to $(1, 0)$. Meantime, all the other input parameters are also rotated with the same angle. Moreover, note that we can further rotate A^* to decrease the learning samples. For instance, when a pedestrian comes to an intersection node and he can turn left or right, the only difference between the two choices is the direction of A^* . And they are symmetry along the horizontal axis direction. This is explained in Step 3-1 and 3-2.

The detailed transfer steps are as follows.

Step 1. As

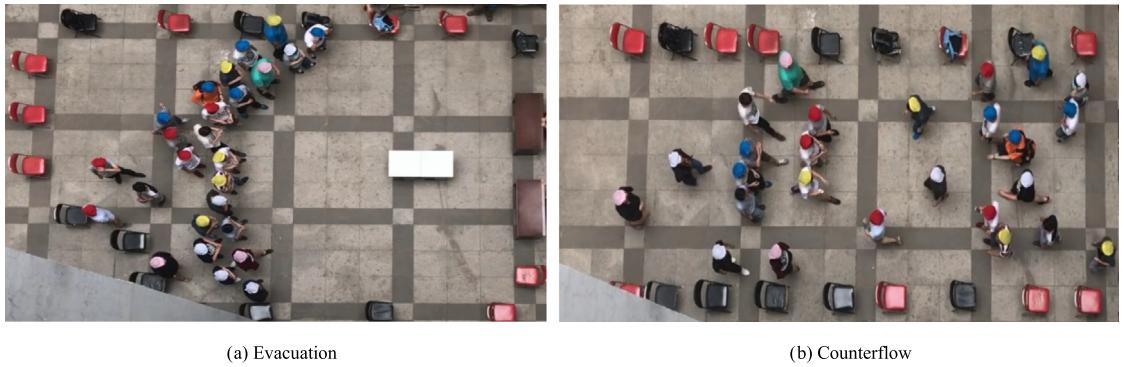
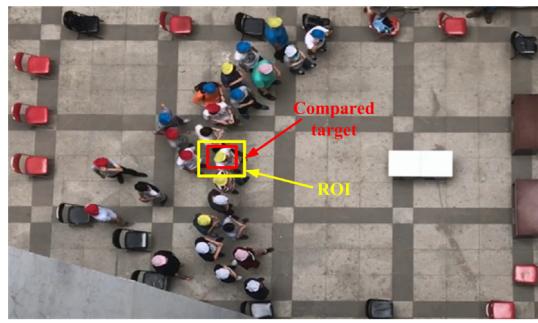
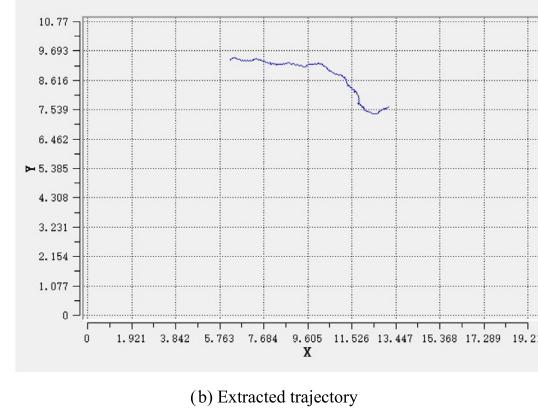


Fig. 5. Snapshot of the real-life pedestrian experiment. (a) is an evacuation scenario in which pedestrians move toward the exit on the right. Obstacle positions and the door widths are changeable. (b) is a counterflow scenario, which is similar to a zebra crossing situation.



(a) Image processing



(b) Extracted trajectory

Fig. 6. Snapshot of the application to automatically extract the pedestrian trajectory. (a) is snapshot of the program. Draw a rectangle, i.e. the target, focusing on the target person. Compare the region of interest in the current frame with the target in the previous frame, if they are similar, finds the target and then draw the target with a rectangle. (b) is the result trajectory obtained regarding the rectangular target person.

Fig. 2(a) shows, the vector \vec{v}_t is rotated counterclockwise to direction of unit vector $(1, 0)$ and it is changed to a scalar v'_t . Note that ANN only accepts scalars. And the rotation angle is α .

Step 2. Rotate the normed vector \vec{A}^* counterclockwise with angle α and $\vec{A}^{*\prime}$ is obtained. Meantime, rotate v_{t+1} counterclockwise with angle α .

Step 3. Rotate all the relative position and speed vectors counterclockwise with angle α and store them into array W' . If \vec{A}^* is located in the first or second quadrant, the rotation is finished. However, if it is located in the third or fourth quadrant, two more steps are needed as follows.

Step 3-1. Make $\vec{A}^{*y} = \vec{A}^{*x}$, and $V_{t+1,y'} = -V_{t+1,x'}$, i.e., rotate vector \vec{A}^{*x} and \vec{v}_{t+1}' along the horizontal axis, as Fig. 2(b) shows.

Step 3-2. Obtain the minus value of all elements in W .

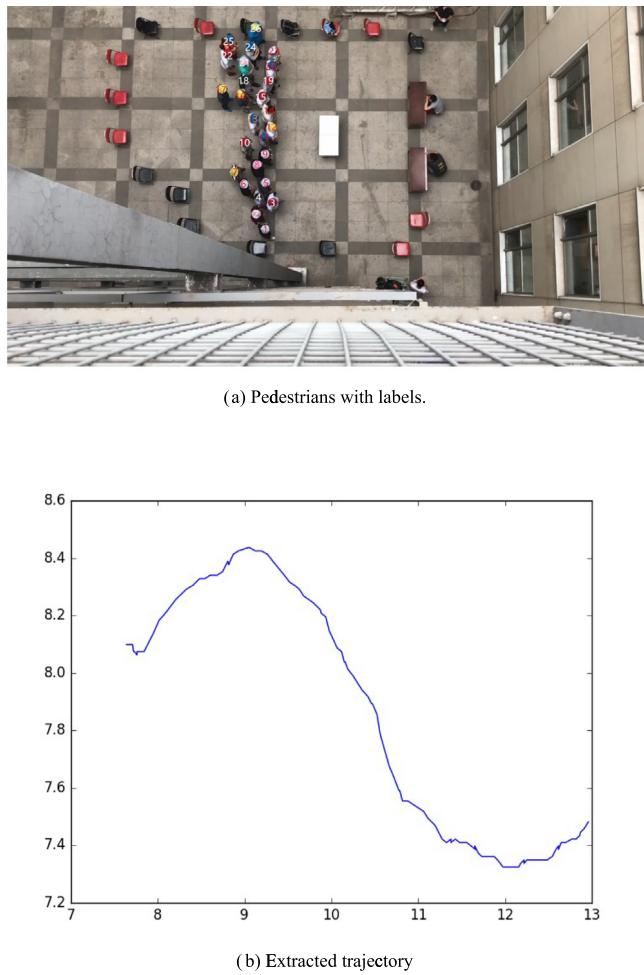


Fig. 7. Snapshot of the data extraction application using MDnet [36].

Table 3
Test Scenarios.

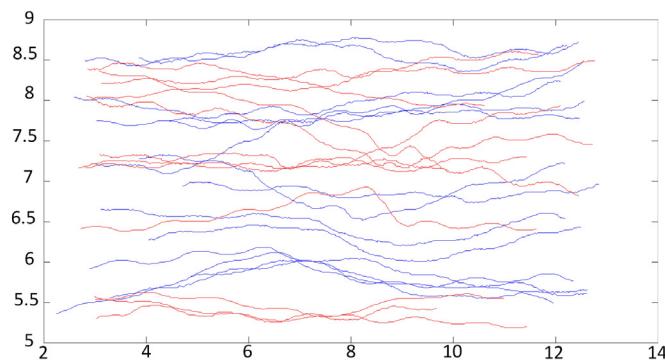
No.	Scenario	Competitive/Non-comp	The number of pedestrians	The number of obstacles	Door width
1	Counterflow	Non-comp	26	–	–
2	Evacuation	Non-comp	26	1	1.0 m
3	Evacuation	Non-comp	35	2	0.7 m
4	Evacuation	Comp	35	2	0.7 m

To determine the number of neurons in hidden layers, experiments were carried out to compute the corresponding mean square error (MSE). The results are shown in Fig. 4. We can observe that more complex network brings less MSE. When the numbers of hidden neurons are both 110, MSE drops to about 0.04 m/s, which is the mean error compared with real pedestrian speed. To make a balance between the computational cost and reliability, the number of neurons in the hidden layer 1 and layer 2 are designed as 110.

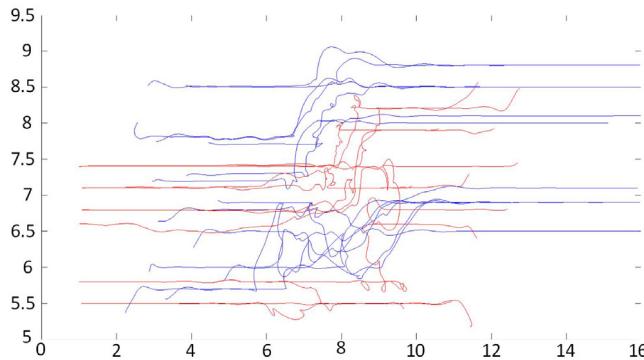
4. Data acquisition and network training

4.1. Data acquisition

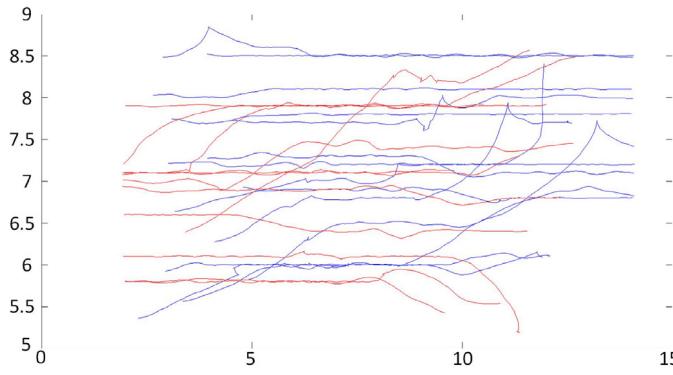
Data are very important for training a good ANN which can simulate more realistic pedestrian movement behaviors. To achieve sufficient and clear video data of typical real-life scenarios, experiments were carried out with a crowd of 26 university graduates and undergraduates as Fig. 5 shows. To make target tracking easier, the students were asked to wear colorful hats. The initial positions are randomly chosen by the volunteers and thus to enable ANN to learn different relative



(a) Real-life model. Self-organized lane formation is observed. Some people are very close to one or two persons, which means the relative distance in realistic model shows some random characteristics due to complex personal and environmental properties.



(b) Social force model. Self-organized lane formation is observed. Relative distances among pedestrians are almost even. To keep these even distances in crowd, the left trajectories show some unrealistic position fluctuations.



(c) Artificial neural network model. Self-organized lane formation is observed. Relative distances are more random than SFM. Unrealistic fluctuations are not observed.

Fig. 8. Comparison of pedestrian trajectories. Blue is heading right, red is toward the left. Left figures are pedestrian trajectories, right are the position distributions at $t = 3$ s. (a) Trajectories in real-life. (b) Trajectories with social force model. (c) Trajectories with ANN.

positions of the crowd dynamics. Meanwhile, the cameras are located in a higher place to record the trajectories of students with a top-down view.

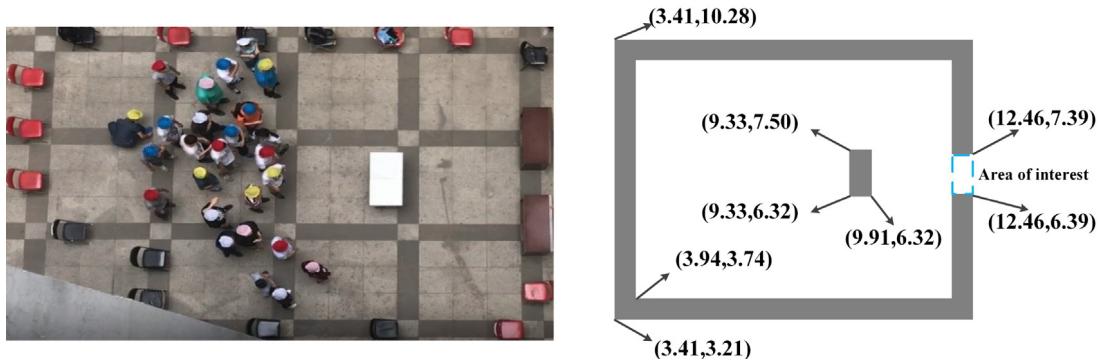


Fig. 9. Scenario of evacuation. Left is the test scenario, which is different from the training scenario in Fig. 5(a). Note that the obstacle position and shape are changed. Right shows the coordinate values of this scenario.

To enable our ANN to learn various surrounding conditions, two types of training scenarios were implemented as Table 2 shows. The first is evacuation from a room. The second is a counterflow to cross a road.

The road-crossing situation is normally non-competitive, while the evacuation scenario is either competitive or non-competitive. In the competitive evacuation case, a bonus was paid for the first 16 students. In the non-competitive case, no bonus was paid and the students often choose to walk more leisurely to pass the door.

Note that the proposed ANN does not have the desirable speed as its input because the speed is actually learned by the ANN. As such, non-competitive and competitive data are used, respectively, to generate two ANNs. One is for non-competitive, and the other is for competitive scenario. Correspondingly, the competitive and non-competitive scenarios are used to train competitive and non-competitive ANN respectively.

As Fig. 5 shows, each recorded video of a scenario is actually a series of video frame sequences. Each video frame is a learning sample of ANN and we had 61 305 samples in total. The video sequence step is 1/30 s and the time step of each learning frame is chosen as 1/15 s considering both tracking position accuracy and computation cost.

Please also note that it is impossible to collect the position data manually for this large number of samples. And we found that it is relatively easy to capture a pedestrian movement in non-competitive case while it is hard to obtain his position in competitive case.

For non-competitive case, a frame difference tracking application [37] is developed to compare the region of interest in the current frame to the target in the previous frame as Fig. 6 shows. If there is a similar feature in these two frames, the target pedestrian is found and then he is drawn with a rectangle. In addition, please note that the top view of the camera is not 100% vertical so that we need to develop a simple coordinate transfer algorithm to obtain the correct positions of pedestrians.

For competitive case, the normal foreground detection method [37] is often invalid because pedestrians ran so fast that the difference between video clips is too big to be recognized. To tackle this problem, recent convolutional neural network based MDnet [36] method is used to obtain the fast moving pedestrians data. MDnet is a network designed for object tracking, in which hard negative mining and bounding box regression technique are implemented to guarantee its accuracy. Its processing is about 3 fps and its accuracy is 99.8%. It also has a high requirement of display card and we used GPU GTX1060, GPU Mem: 6G. As Fig. 7 shows, all volunteers were labeled with a sequence number and each figure is extracted with a moving trajectory.

4.2. Network training

As mentioned in Section 3, the number of ANN input is 22 and the number of output is 2. There are two hidden layers and both have 110 neurons. The main learning method of our ANN is Back Propagation (BP) [38], which compares the predicted result with realistic data and feedback to adjust the weights to minimize the error. This procedure is repeated numerous times until the pre-defined upper limit of a certain number of epochs or if the mean square error (MSE) drops below a certain threshold.

In this paper, the upper limit of the number of epochs is 4000 and the threshold of the MSE error is 0.05. In the proposed ANN, 90% samples (55 175) are for train set and 10% for test set. Other key training parameters and functions are as follows.

- Optimization Algorithm: Stochastic gradient descent.
- Learning rate: 0.1. To make gradient descent have a good trade-off of accuracy and performance, it is desirable to set the learning rate to a suitable value. As our learning output is pedestrian's velocity, this value is to some extent large compared with other method.
- Activation function is Rectified Linear Unit (ReLU): $\text{ReLU}(x) = \max(x, 0)$.

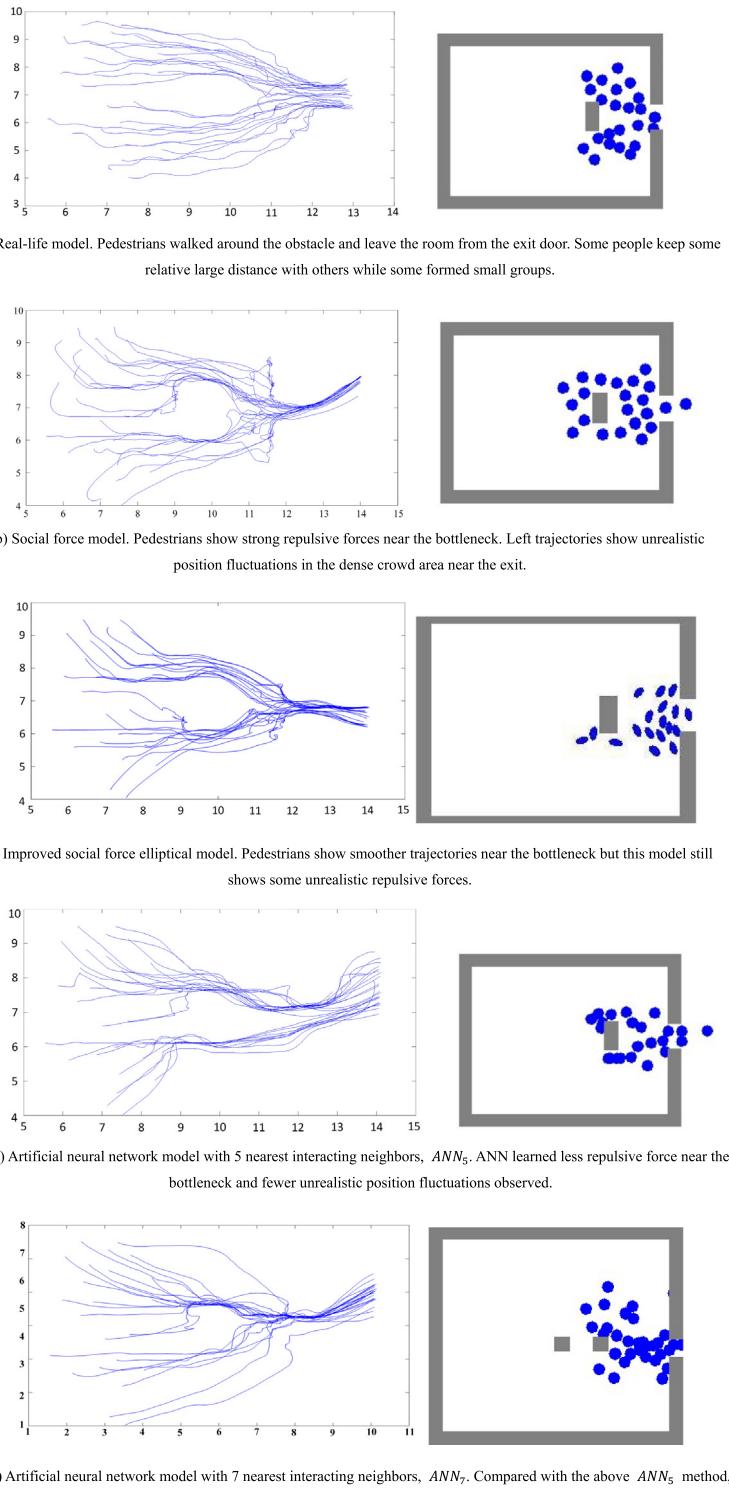


Fig. 10. Evacuation scenario. Left figures are pedestrian trajectories; right are the position distributions at $t = 5$ s. For SFM, the desired speed $v_i^0 = 1.7$ m/s..

- Loss function $C(\theta)$ is using the mean-square error (MSE) method, i.e., $C(\theta) = \sum_k \|\hat{y}_k - f(x_k; \theta)\|$. Here θ is the ANN parameter set, i.e., $\theta = \{w^1, b^1, w^2, b^2, \dots, w^L, b^L\}$. w is the weight, b is the bias.

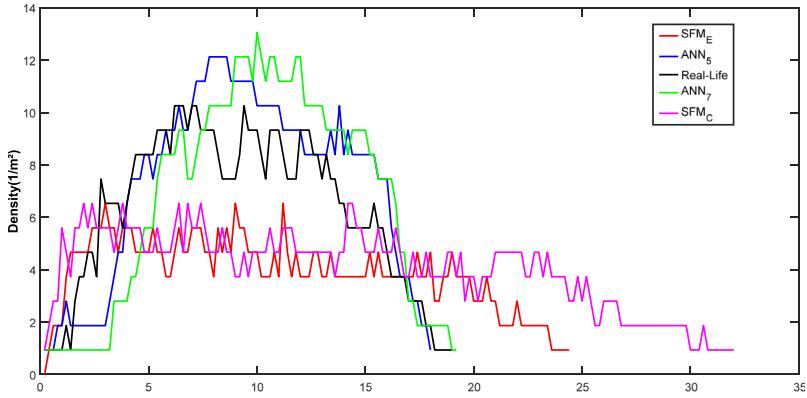


Fig. 11. Comparisons of density changing with time. Various density values, ranging from 0–12/m², are observed in real-life and ANN models, but relatively less value, 0–6/m² is observed for SFM. This means ANN is better in line with the real-life in pedestrian distributions. Also, SFM_E is the improved model of elliptical shape. SFM_C is the original model of circle. ANN_5 and ANN_7 is the influences of 5 and 7 nearest neighbors are simulated respectively. We can observe that the two numbers of interacting neighbors show similar curve patterns. Because of this, we make a tradeoff between computation cost and fidelity and choose to use ANN_5 method in the following experiments.

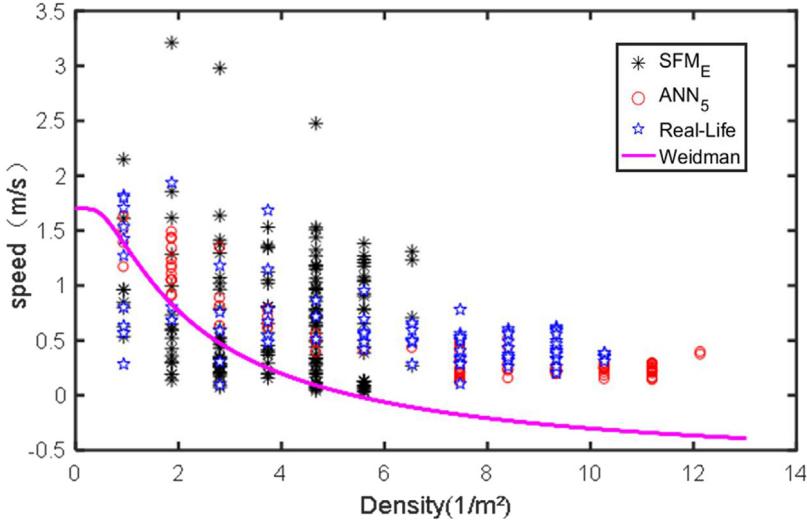


Fig. 12. Comparisons of speed-density diagram with real-life, SFM, ANN and Weidmann [39] models. Similar with Fig. 11, the overall data points of ANN shows a better matching tendency with real-life scattering points compared with clustered points of SFM.

5. Simulation and neural network validation

One essential purpose of this study is to apply the trained ANN to more than one scenario to test its potential of outperforming and replacing SFM. The two validation scenarios include counterflow and evacuation respectively. In each scenario, the result of the three models, i.e., real-life, SFM and ANN, will be compared and analyzed. Note that SFM and its parameter values applied in this paper have been discussed in Section 3. In Table 3, please note that for the 3rd and 4th scenario, the number of pedestrians are 35, which means they are another group of student volunteers and the evacuation experiments were taken in a different date.

In this paper, SFM is the change of velocity at time t given by the acceleration with Eqs. (6)–(9).

$$\vec{F}_i = m_i \frac{d\vec{v}_i}{dt} = \vec{F}_i^d + \vec{F}_i^r + \vec{F}_i^b \quad (6)$$

$$\vec{F}_i^d = \frac{m_i}{\tau_i} (v_i^0 e_i^0 - \vec{v}_i) \quad (7)$$

$$\vec{F}_i^r = \sum_{j(j \neq i)} \vec{F}_{ij} = \vec{F}_{ij}^a = A_i \exp(d_{ij}/B_i) \vec{n}_{ij} \quad (8)$$

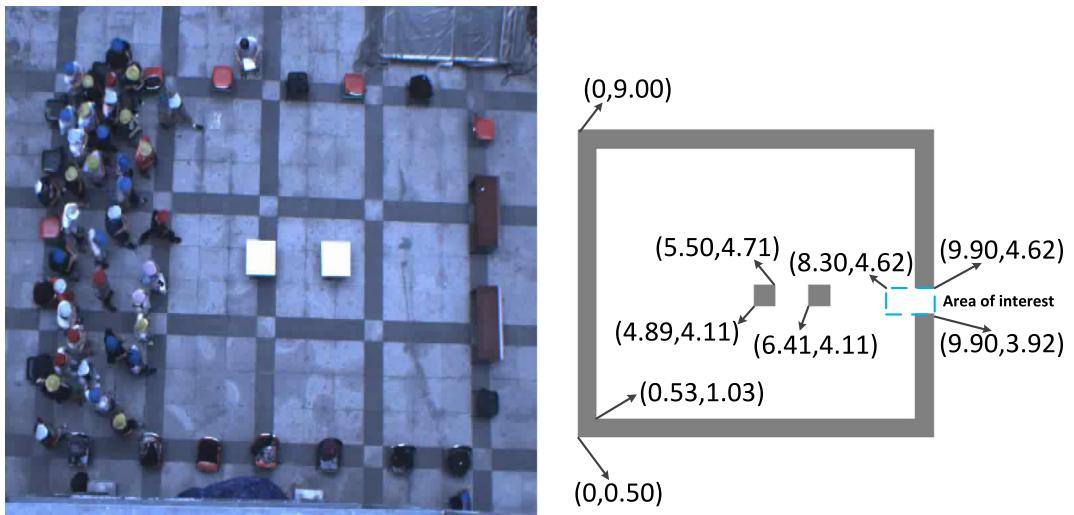


Fig. 13. Scenario of another real-life evacuation. Left is the scenario. Note that there are two obstacles and there are 35 students in this experiment. Right shows the coordinate values of this scenario.

$$\vec{F}_i^b = \sum_b \vec{F}_{ib} = k_1 g(d_{ij}) \vec{n}_{ij} + k_2 g(d_{ij}) \Delta v_{ji}^t \vec{t}_{ij} \quad (9)$$

In SFM, the self-driven force is denoted by \vec{F}_i^d in Eq. (6). It means each pedestrian moves with a certain desired speed v_i^0 in a certain direction e_i^0 , which can often be simulated with path planning algorithms such as A^* , Dijkstra etc. Meantime, the person can change his actual velocity \vec{v}_i with a certain characteristic time τ_i . In this paper, the shoulder width $2r = 0.36\text{--}0.44$ m, $m_i = 40\text{--}70$ kg, $\tau_i = 0.5$ s, $v_0 = 1.5\text{--}2$ m/s.

The repulsive psychological force is denoted by \vec{F}_i^r and \vec{F}_i^b in Eqs. (7)–(8). To avoid collisions, the psychological force exists among pedestrians and obstacles. \vec{F}_i^r denotes the repulsive force between pedestrians when they are not touched. $A_i = 2000$ N, $B_i = 0.08$ m. d_{ij} is the actual distance between the center of pedestrian i and j . \vec{n}_{ij} is the unit vector in the normal direction.

\vec{F}_i^b denotes the body force between pedestrians when they touch each other. The parameters $k_1 = 1.2 \times 10^5$ kg s $^{-2}$ and $k_2 = 2.4 \times 10^5$ kg m $^{-1}$ s $^{-1}$ determine obstruction effects in case of physical contact. The function $g(x)$ is zero if the pedestrians do not touch each other ($d_{ij} < 0$), and otherwise equal to the argument x . $\vec{t}_{ij} = (-n_{ij}^2, n_{ij}^1)$ is the tangential direction and $\Delta v_{ji}^t = (\vec{v}_j - \vec{v}_i) \cdot \vec{t}_{ij}$ is the tangential velocity difference.

5.1. Test scenario 1: Counterflow

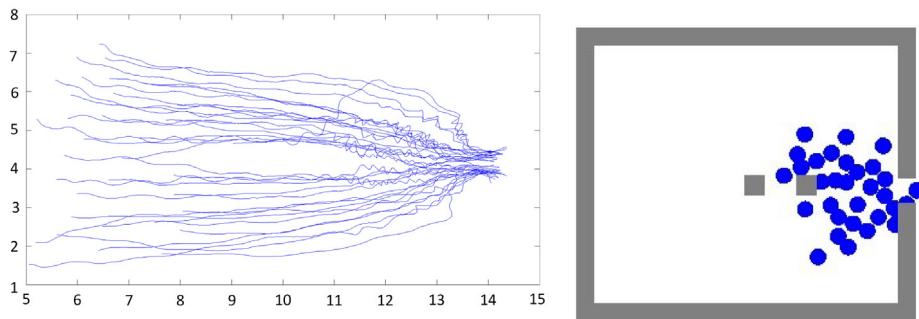
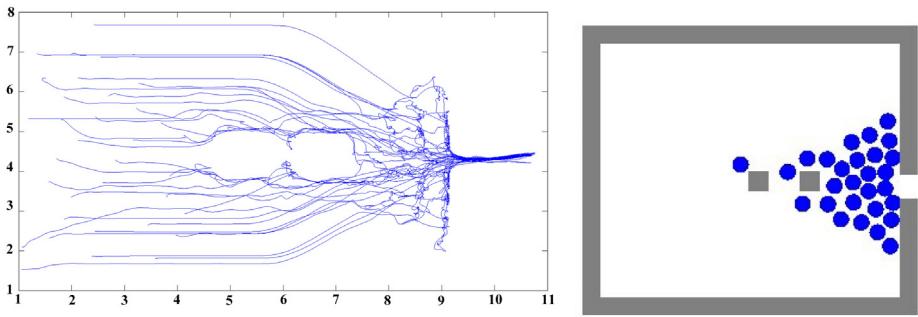
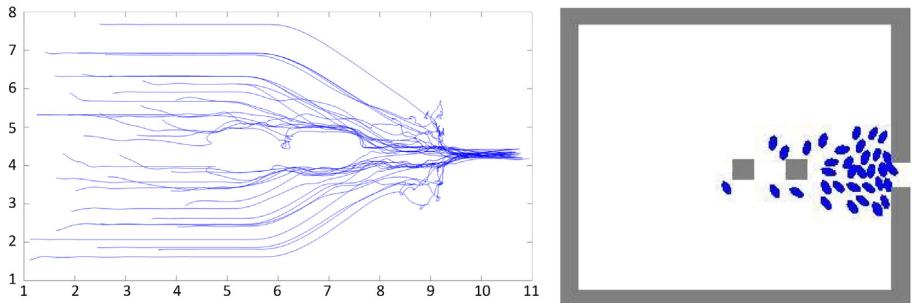
This scenario of counterflow is similar to the crowd on a zebra crosswalk. Considering in most cases road-crossing pedestrians walk non-competitively, we only test the non-competitive situation.

Its length is 15 m and width is 5 m. The recorded movement is about 10 s. Both the training and test scenario have 26 persons but the initial positions of volunteers and their trajectories are different. As it is difficult to mimic all figures' distinct walking characteristics, our validation is mainly focused on the crowd's total characteristics such as their distributions and trajectories, as shown in Fig. 8.

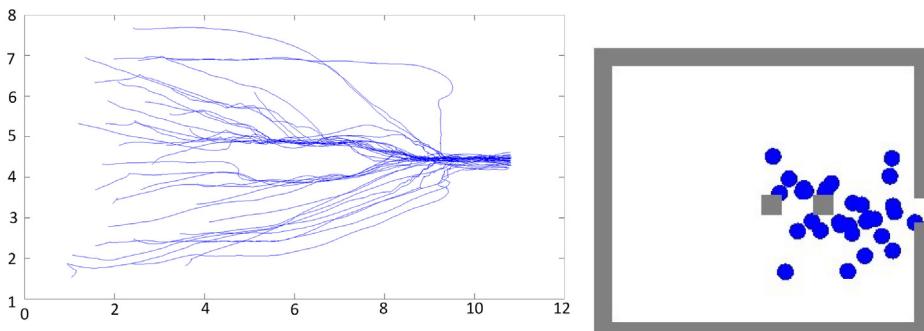
Fig. 8 shows that all the three models demonstrate self-organization phenomena. But SFM shows unrealistic trajectory vibrations due to its repulsive forces, as shown in Fig. 8b. Comparably, ANN learned the random repulsive forces among the crowd and shows a smoother trajectory.

5.2. Test scenario 2: Evacuation from a room with 26 pedestrians and one obstacle

Fig. 9 shows the test evacuation scenario and again the three models are tested. Compared with SFM, ANN possesses smaller relative distances among the pedestrians and less unrealistic trajectory fluctuations. Meanwhile, we tried to follow the method of reference [40] 'A Learning based Approach for Social Force Model Parameter Estimation' to improve the parameter values of SFM. But the result was not good because we found we can only compare several people's trajectories with real data to modify the parameter of repulsive force in SFM. The parameter B in Eq. (8) was proved to be 0.1 instead of 0.08 in the original SFM. But when we used the calibrated B in the experiment, the trajectories of pedestrians were almost the same as the original model, which means this method is not effective in model promotion. As such, we tried to make an improvement of SFM by designing an elliptical model (based on Refs. [23] and [41]) that can simulate the rotations of

(a) Real-life model. Similar behaviors are observed compared with **Figure 10**.(b) Social force model. Compared with **Figure 10**, similar relative distances and fluctuations near the exit are observed.

(c) Improved social force elliptical model. Pedestrians show smoother trajectories near the bottleneck.



(d) Artificial neural network model. People learned to evacuate from the room even smoother than real-life model.

Fig. 14. Evacuation scenario. Left figures are pedestrian trajectories; right are the position distributions at $t = 5$ s. For SFM, the desired speed $v_i^0 = 1.7$ m/s.

pedestrians. The results in Fig. 10c show that the trajectories and density distribution of the improved elliptical SFM is better. However, ANN still shows more realistic trajectories than this improved elliptical SFM.

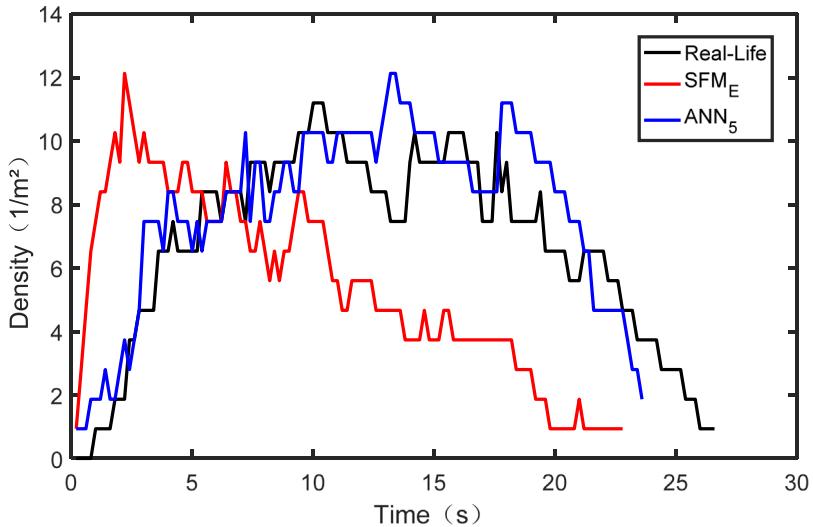


Fig. 15. Comparisons of density change with time. Various density values are observed in real-life and ANN models, but relatively less value is observed for SFM.

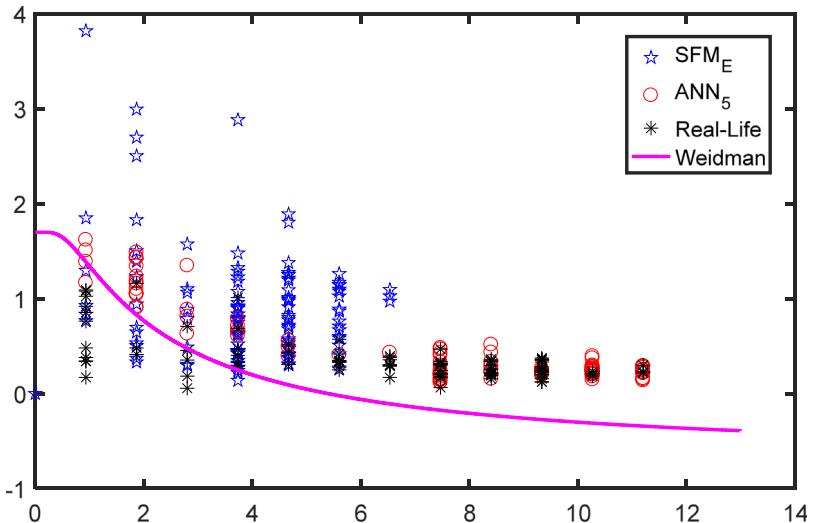


Fig. 16. Comparisons of speed-density diagram with real-life, SFM, ANN and Weidmann [39] models.

Moreover, the comparisons of speed and density diagrams are made in Figs. 11 and 12. Note that it takes 15 s for real-life, 16.1 s for SFM and 15.4 s for ANN to evacuate all the persons.

As many evacuation studies focus on shorter evacuation time near the crowded area, the area of interest (AOI) is selected as the exit door area of $0.53 \text{ m} \times 1.0 \text{ m}$ (the wall width is 0.53 m and the door width is 1.0 m), which is the dotted rectangle shown in Fig. 9.

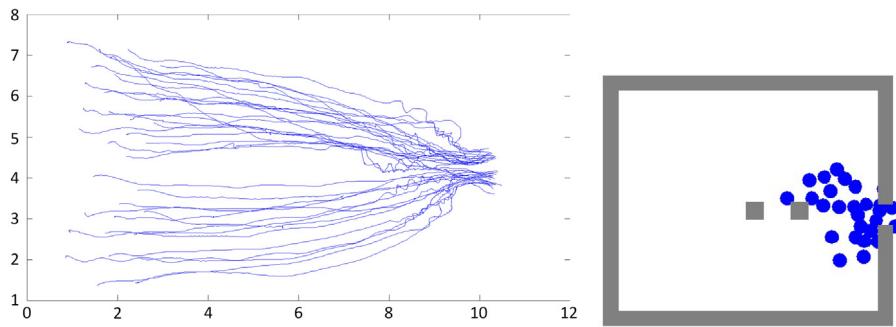
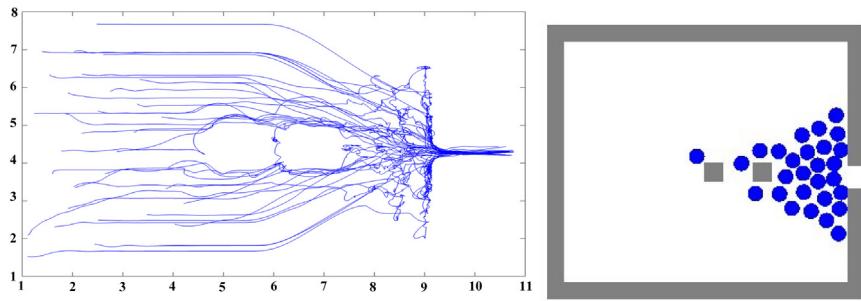
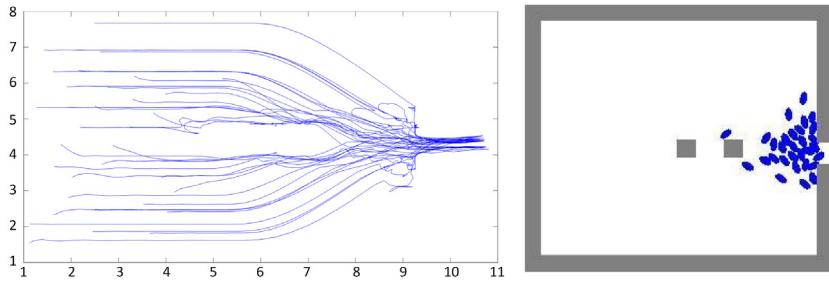
It is found that SFM only has fewer density values, while real-life and SFM have more various speed and density values. The speed of Weidmann is slower because the university students have a higher speed than normal crowd.

5.3. Test scenario 3: Evacuation from a room with 35 pedestrians and two obstacles

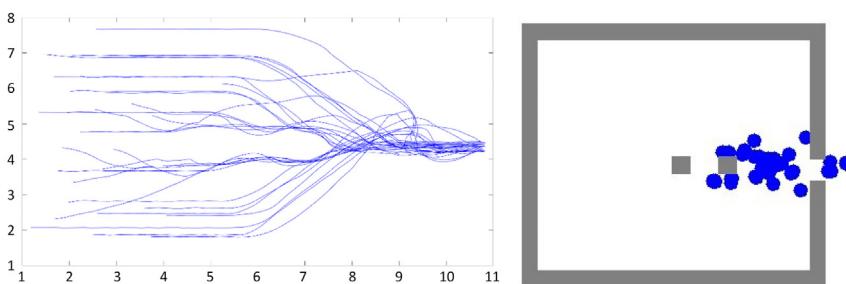
To validate our ANN model with various scenarios, the trajectory data of 35 students is extracted from another real-life experiment which is also used as a test scenario shown in Fig. 13.

5.4. Test scenario 4: Competitive Evacuation from a room with 35 pedestrians and two obstacles

For the previous two scenarios, pedestrians evacuated with normal speed which is often no greater than 2.5 m/s . Correspondingly, ANN trained with such data generates normal speed output.

a) Real-life model. Similar behaviors are observed compared with **Figure 10**.b) Social force model. Compared with **Figure 10**, similar relative distances and fluctuations near the exit are observed.

c) Improved social force elliptical model. Pedestrians show smoother trajectories near the bottleneck but this model still shows some unrealistic repulsive forces.



d) Artificial neural network model. People keep relative small distance with each other. There are few unrealistic position fluctuations in the trajectories.

Fig. 17. Evacuation scenario. Left figures are pedestrian trajectories; right are the position distributions at $t = 5$ s. For SFM, the desired speed $v_i^0 = 3.0$ m/s.

But if we want to simulate a competitive evacuation, although the network structure is the same as in Fig. 3, we must train ANN with the competitive data. Note that there is no desirable speed as network input, i.e., the network trained with non-competitive data can only be applied to low speed cases.

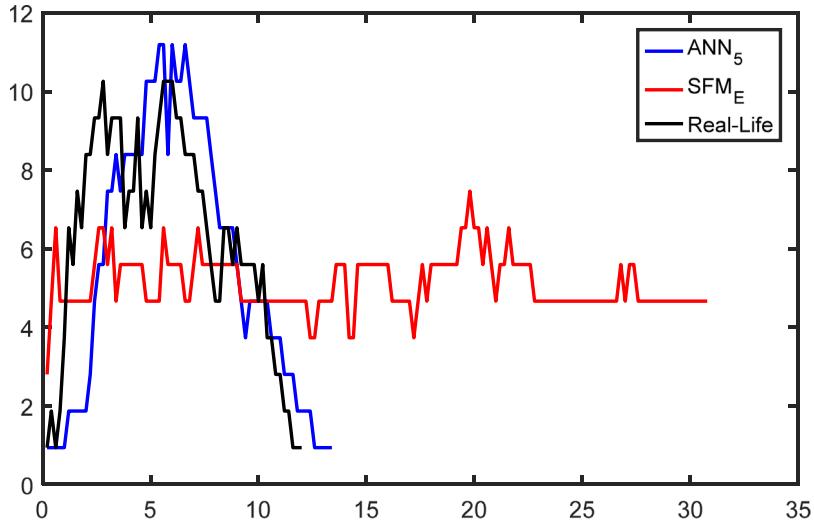


Fig. 18. Comparisons of density change with time. Various density values are observed in real-life and ANN models, but relatively less value is observed for SFM.

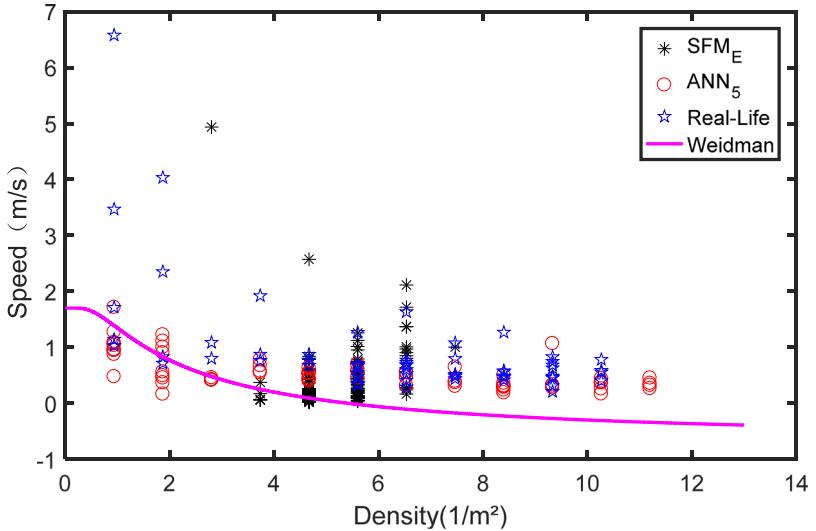


Fig. 19. Comparisons of speed-density diagram with real-life, SFM, ANN and Weidmann [39] models.

As such, a competitive network was trained using the competitive data. We found it much more difficult to extract the data because the volunteers ran fast in this case. Deep neural network techniques including MDnet [36] were employed to extract the data.

After data acquisition, ANN is then trained with the data and we found that the MSE error is around 0.07 when 4000 epochs were implemented. This error is a little higher than 0.05 in the non-competitive case. This shows that it is more difficult to train ANN with competitive data because we found some students move slower when they learned they will not receive the bonus. Also, a wider range of speed raises the difficulty of ANN learning.

Anyway, the trained ANN is used to simulate the scenario including 35 pedestrians and the results are shown in Fig. 17.

Fig. 17 shows that the first part trajectories generated by competitive ANN are similar to that of SFM, which are not smooth as real data. For the latter part near exit bottleneck, ANN again avoids the unrealistic fluctuations generated by SFM. This means that competitive ANN shows a little worse performance than its non-competitive counterpart, but still is better than SFM.

Moreover, the comparison of density against time is made in Fig. 18. Note that it takes about 14 s for real life, 22 s for SFM and 12 s for ANN to evacuate all the persons. This shows that SFM generates unrealistic repulsive forces near exit and makes the evacuation time longer than real life. And ANN shows the phenomenon of over-fitting as it shows less repulsive force than real life and generates less evacuation time (see Figs. 14–16).

Meanwhile, we can observe that ANN has some higher density red points than SFM and real life experiment, as shown in Fig. 19. It demonstrates that the competitive situation is more difficult for ANN to learn because the experiments are more difficult to implement as some volunteers might be reluctant to run, and the speed ranges much larger than non-competitive situation.

6. Conclusion

In this paper, a novel neural network based approach simulating pedestrian movement is proposed. Compared with traditional social force model, it is data-driven and encapsulates multi-scenario pedestrian behavior into a four layer neural network, which then can generate pedestrian position and velocity. The simulation results shows that our ANN outperforms SFM by showing less unrealistic position fluctuations and better speed-density curve.

The shortcoming of recent neural networks modeling pedestrian movement is that they are mostly focused on learning only one scenario. However, our ANN learned manifold scenario data by normalization of relative positions among pedestrians, transferring velocity vector to scalar and incorporating more path planning information, and thus is more adaptive to various scenarios.

Although the proposed ANN shows more realistic simulation results than SFM, it still has some difference compared with real-life data. Also, it is more capable of learning non-competitive scenarios than competitive situations. Future works should consider more typical scenario data, improved network and generate better result.

Acknowledgments

This research was supported by grant 61473013 from National Natural Science Foundation of China and the Fundamental Research Funds for the Central Universities, China. Authors thank reviews for their valuable comments.

References

- [1] Jeongin Koo, Yong Seog Kim, Byung-In Kim, Keith M. Christensen, A comparative study of evacuation strategies for people with disabilities in high-rise building evacuation, *Expert Syst. Appl.* 40 (2013) 408–417.
- [2] Lu Tan, Mingyuan Hu, Hui Lin, Agent-based simulation of building evacuation: Combining human behavior with predictable spatial accessibility in a fire emergency, *Inform. Sci.* 295 (2015) 53–66.
- [3] Kashif Zia, Katayoun Farrahi, Andreas Rieger, Alois Ferscha, An agent-based parallel geo-simulation of urban mobility during city-scale evacuation, *Simulation: Trans. Soc. Model. Simul. Int.* 89 (10) (2013) 1184–1214.
- [4] Marco D’Orazio, Luca Spalazzi, Enrico Quagliarini, Gabriele Bernardini, Agent-based model for earthquake pedestrians’ evacuation in urban outdoor scenarios: Behavioral patterns definition and evacuation paths choice, *Saf. Sci.* 62 (2014) 450–465.
- [5] Joao Ribeiro, Joao Emilio Almeida, Rosaldo J.F. Rossetti, Antonio Coelho, Antonio Leca Coelho, Using serious games to train evacuation behavior, *Inf. Syst. Technol. (CISTI)* 7 (2012) 1–6.
- [6] Hendrik Vermuyten, Jeroen Belien, Liesje De Boeck, Genserik Reniers, Tony Wauters, A review of optimisation models for pedestrian evacuation and design problems, *Saf. Sci.* 87 (2016) 167–178.
- [7] A. Schadschneider, W. Klingsch, H. Kluepfel, T. Kretz, C. Roggisch, A. Seyfried, Evacuation dynamics: empirical results, modeling and applications, in: Encyclopedia of Complexity and System Science, 2008, pp. 517–550, Available from: 0802.1620, http://dx.doi.org/10.1007/978-0-387-30440-3_187.
- [8] Y. Ma, E.W.M. Lee, R.K.K. Yuen, An artificial intelligence-based approach for simulating pedestrian movement, *IEEE Trans. Intell. Transp. Syst.* 17 (11) (2016) 3159–3170.
- [9] Mingbi Zhao, Stephen John Turner, Wentong Cai, A data-driven crowd simulation model based on clustering and classification, in: Proceedings of 17th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications, 2013, pp. 125–134.
- [10] Xiao Song, Jinghan Sun, Hongnan Xie, et al., Characteristic time based social force model improvement and exit assignment strategy for pedestrian evacuation, *Physica A* 505 (9) (2018) 530–548.
- [11] D. Helbing, I. Farkas, T. Vicsek, Simulating dynamical features of escape panic, *Nature* 407 (9) (2000) 487–491.
- [12] Claudio Castellano, Santo Fortunato, Vittorio Loreto, Statistical physics of social dynamics, *Rev. Modern Phys.* 81 (2) (2009) 591–646.
- [13] J. Van Den Berg, S. Guy, M. Lin, et al., Reciprocal n-body collision avoidance, *Robot. Res.* (2011) 3–19.
- [14] S. Kim, S.J. Guy, W. Liu, et al., BRVO: Predicting pedestrian trajectories using velocity space reasoning, *Int. J. Robot. Res.* 34 (2) (2015) 201–217.
- [15] John Nash, John nash the bargaining problem, *Econometrica* 18 (1950) 155–162.
- [16] Xiao Song, Liang Ma, Yaofei Ma, Chen Yang, Hang Ji, Selfishness- and selflessness-based models of pedestrian room evacuation, *Physica A* 447 (4) (2016) 455–466.
- [17] P. Shao, A more realistic simulation of pedestrian based on cellular automata, in: IEEE International Workshop on Open-Source Software for Scientific Computation, IEEE, 2009, pp. 24–29.
- [18] F. Farina, D. Fontanelli, A. Garulli, A. Giannitrapani, D. Prattichizzo, Walking ahead: The headed social force model, *PLoS One* 12 (1) (2017) e0169734, 1–23.
- [19] Xiao Song, Ziping Xie, Yan Xu, Gary Tan, Supporting real-world network-oriented mesoscopic traffic simulation on GPU, *Simul. Model. Pract. Theory* 74 (3) (2017) 46–63.
- [20] Yulin Wu, Xiao Song, Guanghong Gong, Real-time load balancing scheduling algorithm for periodic simulation models, *Simul. Model. Pract. Theory* 52 (1) (2015) 123–134.
- [21] Dirk Helbing, Lubos Buzna, Anders Johansson, Torsten Werner, Self-organized pedestrian crowd dynamics: experiments, simulations, and design solutions, *Transp. Sci.* 39 (1) (2005) 1–24.
- [22] Fernando Alonso-Marroquin, Jonathan Busch, Coraline Chiew, Simulation of counterflow pedestrian dynamics using spheropolgons, *Phys. Rev. E* 90 (2014) 063305.
- [23] Paul A. Langston, Robert Masling, Basel N. Asmar, Crowd dynamics discrete element multi-circle model, *Saf. Sci.* 44 (2006) 395–417.
- [24] M. Chraibi, A.K.U. Seyfried, A. Schadschneider, Force-based models of pedestrian dynamics, *Netw. Heterog. Media* 6 (3) (2013) 425–442.
- [25] A. Alahi, K. Goel, V. Ramanathan, et al., Social-LSTM: Human trajectory prediction in crowded spaces, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2016, pp. 961–971.

- [26] Z. Zainuddin, L.E. Aik, Intelligent exit-selection behaviors during a room evacuation, *Chin. Phys. Lett.* 29 (29) (2012) 18901–18904.
- [27] P.T. Szemes, H. Hashimoto, P. Korondi, Pedestrian-behavior-based mobile agent control in intelligent space, *IEEE Trans. Instrum. Meas.* 54 (6) (2005) 2250–2257.
- [28] J.L. Castro, M. Delgado, J. Medina, et al., An expert fuzzy system for predicting object collisions. Its application for avoiding pedestrian accidents, *Expert Syst. Appl.* 38 (1) (2011) 486–494.
- [29] L. Zhao, C.E. Thorpe, Stereo- and neural network-based pedestrian detection, *IEEE Trans. Intell. Transp. Syst.* 1 (3) (2002) 148–154.
- [30] G.K.H. Pang, K. Takabashi, T. Yokota, et al., Adaptive route selection for dynamic route guidance system based on fuzzy-neural approaches, *IEEE Trans. Veh. Technol.* 48 (6) (1999) 2028–2041.
- [31] X. Cheng, C.X. Wang, B. Ai, et al., Envelope level crossing rate and average fade duration of nonisotropic vehicle-to-vehicle Ricean fading channels, *IEEE Trans. Intell. Transp. Syst.* 15 (1) (2014) 62–72.
- [32] E. Gelenbe, F.-J. Wu, Future research on cyber-physical emergency management systems, *Future Internet* 5 (3) (2013) 336–354.
- [33] O.J. Akinwande, H. Bi, E. Gelenbe, Managing crowds in hazards with dynamic grouping, *IEEE Access* 6 (1) (2016) 1060–1070.
- [34] Xiao Song, Liang Ma, Yaofei Ma, Chen Yang, Hang Ji, Effect of authority figures for pedestrian evacuation at metro stations, *Physica A* 447 (1) (2016) 455–466.
- [35] William Bialeka, Andrea Cavagna, et al., Statistical mechanics for natural flocks of birds, *Proc. Natl. Acad. Sci. USA* 109 (13) (2012) 4786–4791.
- [36] H. Nam, B. Han, Learning multi-domain convolutional neural networks for visual tracking, *Comput. Vis. Pattern Recognit.* (2016) 4293–4302.
- [37] Andrews Sobral, Antoine Vacavantb, A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos, *Comput. Vis. Image Underst.* 122 (5) (2014) 4–21.
- [38] P. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences* (Ph.D. dissertation), Department of Mathematics, Harvard Univ, Cambridge, MA, USA, 1974.
- [39] U. Weidmann, *Transporttechnik der fußgänger*, Transporttechnische Eigenschaften des Fussgangerverkehrs, second ed., in: Schriftenreihe des IVT, vol. 90, ETH Zurich, Zurich, Switzerland, 1993.
- [40] Zhiqiang Wan, Xuemin Hu, Haibo He, Yi Guo, A learning based approach for social force model parameter estimation, in: IEEE International Joint Conference on Neural Network, 2017, pp. 4058–4064.
- [41] Fernando Alonso-Marroquin, Jonathan Busch, Coraline Chiew, Simulation of counterflow pedestrian dynamics using spheropoligons, *Phys. Rev. E* 90 (2014) 063305.