FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Estimation of Fundamental Diagram from Evacuation Experiments
Student:	Juraj Kmec
Supervisor:	Ing. Pavel Hrabák, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2021/2022

Instructions

Important characteristics of traffic or pedestrian flow are the so-called Fundamental Diagram. Its estimation from evacuation data is still a challenging task.

Instructions:
1. Get acquainted with the term Fundamental Diagram (FD) in relation to the analysis of pedestrian dynamics.
2. Perform research of methods for FD estimation from the data measured in evacuation experiments, focus on velocity-density relation.
3. Get acquainted with a suitable open-source simulator of pedestrian evacuation, e.g. JuPedSim.
4. In this tool simulate the course of the experiment described in [1].
5. Implement the chosen method for FD estimation and use it for analysis of data obtained from the simulation above.
6. Try to analyse in a similar manner the real experimental data (data from experiment [1] will be provided by the supervisor).

[1] Bukáček, M., Hrabák, P., Krbálek, M. (2018) Microscopic travel-time analysis of bottleneck experiments. Transportmetrica A: Transport Science. 14(5-6), 375-391.

Electronically approved by Ing. Karel Klouda, Ph.D. on 17 December 2020 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Bachelor's thesis

Estimation of Fundamental Diagram from Evacuation Experiments

Juraj Kmec

Department of Applied Mathematics
Supervisor: Ing. Pavel Hrabák, Ph.D.

May 13, 2021

Acknowledgements

I would like to express sincere gratitude to my supervisor, Ing. Pavel Hrabák, Ph.D., for swiftly answering every question and providing invaluable guidance through the vast field of pedestrian dynamics.

My thanks also goes to my good friends Alexandra Dzurovčinová and Michal Mikluš for great help with the typographic and technical aspect of the thesis.

And last but not least, my parents and my friends for supporting me and motivating me to do my best.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on May 13, 2021 . . . .

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Kmec, Juraj. *Estimation of Fundamental Diagram from Evacuation Experiments*. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2021.

Abstrakt

Fundamentální diagram je důležitou charakteristikou toku chodců, která umožňuje inženýrům posuzovat maximální kapacitu chodcovských zařízení. Klasické metody odhadu fundamentálního diagramu používají analytické aproximace tohoto vztahu. Nezohledňují však specifické vlastnosti jednotlivých geometrií a složení chodců. Tato bakalářská práce prezentuje datově orientovaný přístup k odhadu fundamentálního diagramu pomocí umělých neuronových sítí. To zahrnuje simulování evakuací v open-source simulátoru pohybu chodců, výpočet alternativní míry hustoty – průměrné vzdálenosti ke k nejbližším sousedům a následné trénování neuronové sítě k odhadu fundamentálního vztahu mezi hustotou a rychlostí. Naučené modely ukazují významnou schopnost extrahovat makroskopické vztahy z jednotlivých měření. Predikce je poté rozšířená o další příznaky – relativní polohy nejbližších sousedů a relativní vzdálenost k východu. To přispívá k dalšímu snížení chyby predikce a demonstruje tak použitelnost metod strojového učení v oblasti dynamiky chodců.

Klíčová slova dynamika chodců, fundamentální diagram, simulování pohybu chodců, tok v úzkém hrdle, umělá inteligence, umělé neurónové sítě

Abstract

The fundamental diagram is a crucial characteristic of pedestrian flow that enables engineers to judge the maximum capacity of pedestrian facilities. The classical methods of estimating the fundamental diagram use analytical approximations of this relationship. However, they fail to take into consideration the specific properties of each geometry and the composition of the pedestrians. This thesis presents a data-driven approach to estimating the fundamental diagram via artificial neural networks. This includes simulating evacuations in an open-source pedestrian simulator, computing an alternative measurement of density – the mean distance to k-nearest neighbors, and subsequently training a neural network to estimate the fundamental relationship between density and velocity. The trained models show significant ability to extract the macroscopic relationship from the individual measurements. The prediction is then expanded to include more explanatory variables – the relative positions of nearest neighbors and the relative distance to the exit. This further reduces the prediction error, demonstrating the applicability of machine-learning methods in the field of pedestrian dynamics.

Keywords pedestrian dynamics, fundamental diagram, simulating pedestrian movement, bottleneck flow, artificial intelligence, artificial neural networks

Contents

List of Figures

List of Tables

Introduction

As a result of massive urbanization that has taken place over the last century, mass gatherings have become increasingly popular. Every major city has its own set of venues where the public can gather and participate in activities ranging from football matches to political rallies and protests. This comes hand in hand with an increased emphasis on the design of security exits and evacuation routes. In an event attended by thousands of people, every minor decrease in evacuation time can potentially save lives. Proper understanding of people's actions inside large crowds is crucial for optimizing the design of buildings and ultimately make evacuations quicker and safer.

One of the most important quantitative characteristics of a pedestrian stream is the *fundamental diagram of pedestrian flow* – the empirical relationship between the *density*, *velocity*, and the *flow* of a pedestrian stream. It allows us to estimate quantities such as the maximum pedestrian flow at a given cross-section, a critical density, past which jams start to occur, or the maximum density when the traffic completely halts.

Another important use of the fundamental diagram is validating new pedestrian models. Seeing as conducting evacuation experiments with live participants is both expensive, and sometimes downright impossible, various simulation techniques have to be employed. In order for a model to be deemed realistic, its fundamental diagram has to qualitatively follow a certain shape – for example, the velocity should be a decreasing function of density, approaching zero at a maximum density.

Aim of the Thesis

The main goal of this thesis is to present a data-driven approach to estimating the density–velocity relation using machine learning techniques, namely artificial neural networks. This includes

- recreating a real experiment conducted at the Czech Technical University [1] in an open-source simulator – JuPedSim, and subsequently validating the output trajectories,

- training a neural network to predict velocity based purely on density, and

- working at improving the prediction by introducing extra features, similarly to the approach presented in [2].

Motivation

Pedestrian dynamics is a topic that combines elements from various fields of mathematics, statistics, and even physics. It is also directly connected to the field of traffic dynamics, which has seen an increase in popularity with the rise of self-driving and smart cars.

Neural networks are a powerful and versatile tool, capable of solving problems in a multitude of scientific disciplines. However, their use in the field of pedestrian dynamics is still somewhat limited and there is great potential in employing them to solve problems related to the behavior of people in a crowd. This work could be further extended to predict the entire velocity vector, and, in turn, be used as a base for computationally efficient pedestrian simulations.

Structure

This thesis consists of four chapters. The first two chapters provide theoretical background on the topic of pedestrian dynamics, as it is not a part of the standard FIT CTU curriculum.

In the first chapter, the main observed variables of pedestrian flow are introduced and details about their relationships and ways to measure them are provided. The second chapter provides an overview of the methods used to simulate pedestrian movement. Emphasis is put on the Generalized centrifugal-force model, which is used in the practical simulation work done in this thesis.

The remaining two chapters are the practical part of the thesis. The third chapter contains the details of the experiment that is being recreated, followed by the actual simulation used to obtain data for further processing. It concludes with the validation of said data, and the computation of the “mean k-NN distance” used in the following chapter. The fourth chapter includes a brief discussion of analytical approximations of the fundamental diagram, as well as definitions of the relevant neural network concepts. After that, the fundamental diagram estimation via neural networks takes place. The thesis concludes with the process of improving the predictions by adding new explanatory variables and demonstrating the caveats of this approach.

Chapter 1

Fundamental Diagram of Pedestrian Flow

Pedestrian dynamics is a discipline of traffic flow theory, a subject which is not taught at FIT CTU. For this reason, a basic overview of the concepts relevant to this thesis will be provided. A more comprehensive introduction can be found in [3].

Many ideas from traffic flow theory can be directly generalized to pedestrian dynamics. However, because pedestrians can generally move in two, or even three dimensions, some concepts have to be appropriately modified. A complete course on traffic flow can be found in [4]. In this thesis, only a strictly two-dimensional geometry will be considered (i.e. no vertical movement).

1.1 Observables

A number of variables can be observed while describing the movement of pedestrian crowds. They can generally be regarded as *microscopic* or *macroscopic* [4, Ch. 2].

Microscopic variables describe pedestrians on a per-agent level, e.g. individual *position* and its derivatives, *velocity* and *acceleration.* Others include *distance headway* – the distance to the “next” pedestrian, and *time headway* – the time required to travel said distance with the current speed. The concept of headways can be problematic, as it is not always clear who the “next” pedestrian is unless we limit ourselves to effectively one-dimensional geometries such as narrow corridors, as shown in [5]. However, this quantity can be somewhat generalized to the *average distance to k-nearest neighbors* and be used as a measure of density, as will be presented in this thesis.

Macroscopic variables describe the properties of pedestrian movement on an aggregated level. The most important ones are *density*, *velocity*, and *flow.*

In the following sections, these characteristics shall be described in greater detail. The notation used in this chapter is adapted from [3], where vector quantities are denoted by a "→" superscript and their respective norms are written using the same letter, but without the superscript, e.g. $\|\vec{v}\| =: v$. Quantities averaged over time will be denoted by $\langle \cdot \rangle_{\Delta t}$ and quantities averaged over space will be denoted by $\langle \cdot \rangle_{\Delta x}$.

1.1.1 Density

Definition 1 (Classical density)**.** The *density* ρ of a pedestrian flow is the number of pedestrians N present in a unit of area A at a given time, i.e.,

$$\rho = \frac{N}{|A|}. \tag{1.1}$$

This is known as *classical* pedestrian density. It has several practical flaws [6]:

- It can be unclear when a pedestrian is actually "in" said area.
- When viewed as a function of time, it is discontinuous.
- It becomes meaningless as the size of the area approaches zero.

The first problem can easily be solved by taking a fixed point on each pedestrian, for example, the tip of their nose. The remaining two problems motivate us to define a different measurement of density, the so-called *Voronoi* density, based on *Voronoi diagrams* [7].

The *Voronoi diagram* of a convex 2-D area A, given a finite set of points $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N$, is a decomposition of A into so-called *Voronoi cells* $A_1, A_2, \ldots, A_N$, such that

$$A_i = \{\vec{x} \in A : \|\vec{x} - \vec{x}_i\| \leq \|\vec{x} - \vec{x}_j\|, \forall j \neq i\}, \tag{1.2}$$

that is, A_i is the set of all points that are closer to $\vec{x}_i$ than they are to any other $\vec{x}_j$.

Any two distinct A_i, A_j are either non-overlapping, or equal to e_{ij}, the set of points equidistant from A_i and A_j. A comprehensive theory on Voronoi diagrams can be found in [8]. An example Voronoi diagram is shown in figure 1.1.

Voronoi diagrams of pedestrian flow in a given room have the property that, for each Voronoi cell, there is exactly one pedestrian occupying said cell. In a sense, every pedestrian "generates" [6] a density distribution ρ_i, which sum together into a *Voronoi distribution* ρ_A:

$$\rho_i(\vec{x}) = \begin{cases} 1 \,/\, |A_i| & \text{if } \vec{x} \in A_i, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{and} \qquad \rho_A = \sum_{i=1}^{N} \rho_i. \tag{1.3}$$

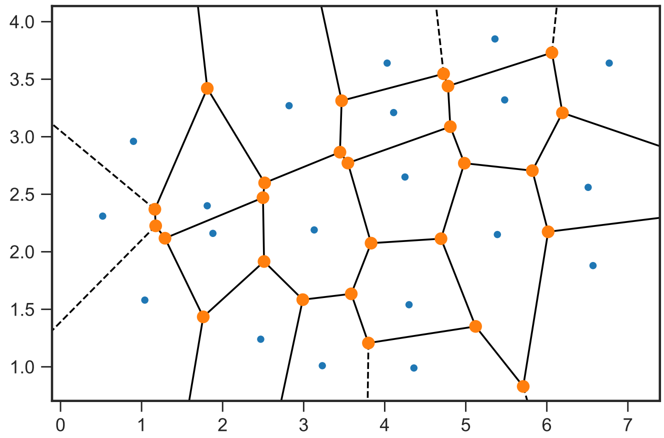


Figure 1.1: An example Voronoi diagram.

It is easy to see that

$$\int_A \rho_i(\vec{x})\,\mathrm{d}\vec{x} = \int_{A_i} \rho_i(\vec{x})\,\mathrm{d}\vec{x} = \int_{A_i} \frac{1}{|A_i|}\,\mathrm{d}\vec{x} = \frac{1}{|A_i|}\int_{A_i} 1\,\mathrm{d}\vec{x} = 1$$

and

$$\int_A \rho_A\,\mathrm{d}\vec{x} = \int_A \sum_{i=1}^{N} \rho_i\,\mathrm{d}\vec{x} = \sum_{i=1}^{N} \int_A \rho_i\,\mathrm{d}\vec{x} = \sum_{i=1}^{N} 1 = N.$$

The Voronoi density can be defined as follows:

Definition 2 (Voronoi density)**.** Let A be an area occupied by N pedestrians and ρ_A its Voronoi distribution.
The *Voronoi density* ρ, given an area $V \subseteq A$, is

$$\rho = \frac{\int_V \rho_A\,\mathrm{d}\vec{x}}{|V|}. \tag{1.4}$$

This has the benefit that the area V can be arbitrarily small and the density will not degrade to zero. When measuring over the entire area A, the Voronoi density will be equal to the classical density. It is shown in [6], that when viewed as a function of time, it does not fluctuate nearly as much as the classical density does, making it a more suitable measurement of density.

1.1.2 Velocity

It is important to distinguish the *microscopic velocity* $\vec{v}_i$ of pedestrian i and the *macroscopic velocity* v of an entire pedestrian stream.

The individual velocity $\vec{v}_i(t)$ of person i at time t is the time derivative of their position, i.e. $\vec{v}_i(t) = \frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t)$. It is usually measured as the rate of change of position in a small interval Δt around t:

$$\vec{v}_{\Delta t,i}(t) = \frac{\vec{x}(t + \Delta t/2) - \vec{x}(t - \Delta t/2)}{\Delta t}. \tag{1.5}$$

An alternative way of measuring individual velocities over a measurement area, with a given entry and exit time is:

$$\vec{v}_{\Delta x,i}(t) = \frac{\vec{x}(t_{\text{out}}) - \vec{x}(t_{\text{in}})}{t_{\text{out}} - t_{\text{in}}}, \qquad t \in [t_{\text{in}}, t_{\text{out}}]. \tag{1.6}$$

These two measurements do not necessarily correspond. In fact, the former will generally output higher instant velocities than the latter. This is due to the fact that the second method assumes that the pedestrian has been following the straight line $\vec{x}(t_{\text{out}}) - \vec{x}(t_{\text{in}})$. In reality, pedestrians do not move perfectly straight, their actual path is somewhat curved [6], meaning they effectively cover more distance in the same amount of time, resulting in higher instantaneous velocities.

Macroscopic velocity measurements involve averaging individual velocities either over time, or space. Note that velocity is a vector and speed its norm, $\|\vec{v}\| = s$, however, in the referenced literature it is quite common to refer to both the vector and its norm simply as "velocity" v, and it should be clear from the context, which is being referenced.

Definition 3 (Time mean velocity)**.** The *time mean velocity* of pedestrians passing a fixed point (e.g. a cross section) x_0, over a time period Δt, is the arithmetic mean of their individual velocities $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_N$ at x_0.

$$\langle v \rangle_{\Delta t} = \frac{1}{N} \sum_{i=1}^{N} v_i(x_0). \tag{1.7}$$

This method is more commonly used in traffic engineering, where the measurement area is too long to be monitored all at once, for example, a highway. It is more practical to set up a single measuring station along the road and measure the vehicles that pass by.

Definition 4 (Space mean velocity)**.** The *space mean velocity* of pedestrians at a fixed time t_0, over a measurement area Δx, is the arithmetic mean of their individual velocities $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_N$ at t_0.

$$\langle v \rangle_{\Delta x} = \frac{1}{N} \sum_{i=1}^{N} v_i(t_0). \tag{1.8}$$

This is, in fact, the desirable measurement to use for further analysis [9], such as the relation with flow and density. In the typical scenario of pedestrian experiments, where a camera continuously records the whole measurement area, this quantity can easily be computed, provided that we can extract the trajectories from the camera feed. This is not the case for vehicle movement, where we often only have the time mean velocity $\langle v \rangle_{\Delta t}$.

It is shown in [4, Ch. 2], that the time mean velocity and space mean velocity are connected via the *harmonic mean*: "*The space mean speed is equal to the harmonic average of the speeds collected at a cross-section x during a stationary period.*".

$$\langle v \rangle_{\Delta x} = \frac{N}{\sum_{i=1}^{N} \frac{1}{v_i(x_0)}}. \tag{1.9}$$

It follows from the fact that the harmonic mean of a data set is always less than or equal to the arithmetic mean of said data set, that the time mean velocity will be an overestimation of the space mean speed. The intuition is that slower pedestrians will spend more time in the measurement area, therefore skewing the average measurement towards lower speeds [9].

It should be noted that the concept of spatial and temporal average can also be applied to density. In this sense, the density defined in the previous section would correspond to a space mean density $\langle \rho \rangle_{\Delta x}$.

1.1.3 Flow

Definition 5 (Flow)**.** The *flow* J (also called *intensity*) of a pedestrian stream is the number of pedestrians N crossing a fixed location per unit of time T.

$$J = \frac{N}{T}. \tag{1.10}$$

Flow can also be measured by recording the individual crossing times $t_1, t_2, \ldots, t_N$ and calculating the inverse of the *mean time gap* between subsequent crossings,

$$J = \frac{1}{\langle \Delta t \rangle}, \qquad \text{where} \qquad \langle \Delta t \rangle = \frac{1}{N-1} \sum_{i=1}^{N-1} (t_{i+1} - t_i). \tag{1.11}$$

A different way to measure flow at a cross section of width b is by using the *fundamental relation of pedestrian flow*, $J = \rho v b$ [3]. In this context, the space mean velocity is used. Since both ρ and v are functions of time, this relation allows us to acquire instantaneous flow values. Strictly speaking, the flow computed using this method is a vector and the area is two-dimensional. For this reason, only the component normal to the measurement line is considered.

1.2 Fundamental Diagram

It is clear that the variables defined above are connected. For example, people will, on average, move slower in large crowds than they would on an empty street, because their movement is inhibited by the surrounding pedestrians. The relationships between the characteristics of pedestrian flow are called the *fundamental diagram of pedestrian flow (FD).* The reason why only a single diagram is mentioned, as opposed to multiple diagrams for each pair of observables, is that, furthermore, these variables are connected via the *fundamental relation of pedestrian flow*:

$$J = \rho v, \tag{1.12}$$

a concept inspired by hydrodynamics. This equation allows us to freely circle between the different representations of the fundamental diagram. From this point on, the "v" will refer to space mean velocity, unless stated otherwise.

The study of FD dates back to 1934, when Bruce D. Greenshields devised a decreasing linear relationship between density and velocity of highway traffic, validating using seven(!) data points [10, 11]. It has since been the target of extensive research.

As mentioned above, there are multiple equivalent forms of the fundamental diagram:

- density–velocity $v(\rho)$,
- density–flow $J(\rho)$,
- flow–velocity $v(J)$.

All of these diagrams have a certain qualitative shape, which I will briefly showcase. Further details on the shapes and properties of traffic flow FDs are provided in [12], although most results can be directly applied to pedestrian traffic. As Schadschneider states in [3], there is no strict consensus on the numerical values of the graphs:

> *Although several attempts to measure the fundamental diagram of pedestrian flow have been made, a lot of points are still controversial. So it is still not clear what the maximal density is which can be observed in pedestrian streams. Estimates range from about 4 P/m^2 up to more than 12 P/m^2.*

Velocity as a function of density This is the most straight-forward relation of the three. It is especially prominent in pedestrian dynamics because of its relative ease of construction from camera recordings. At very small densities, the average velocity is almost constant at a *maximum velocity* $v_{\max}$, which corresponds to a *free flow* stage. It then monotonically decreases with increasing density, until it reaches zero at a *maximum density* $\rho_{\max}$, at which point the flow completely stops. The exact slope and shape of this downward curve is still a matter of debate [13]. The Greenshields model is shown in figure 1.2.

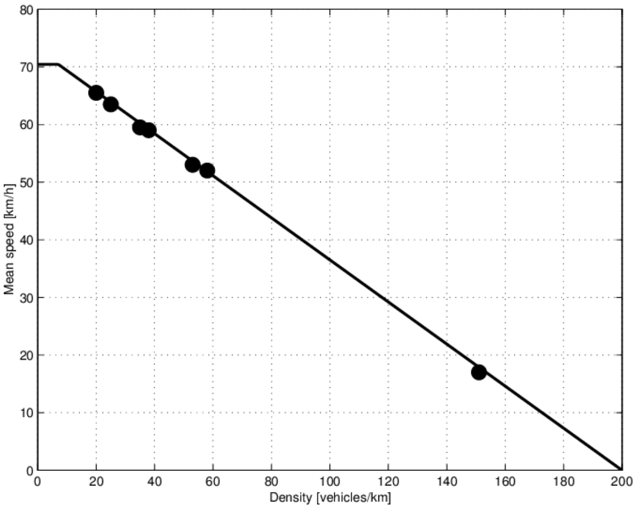


Figure 1.2: Greenshields' ρ–v diagram [12].

Flow as a function of density The $J(\rho)$ relation has a quadratic shape. It starts at the origin and increases until it hits a *capacity flow* $J_{\max}$ at a *critical density* ρ_c. It then decreases until zero is reached at $\rho_{\max}$. The density ρ_c divides the flow into the so-called *free flow branch*, where the pedestrians only marginally affect each other's movement, and the *congested branch*, where jams start to form and the overall flow decreases. An example FD is shown in figure 1.3.

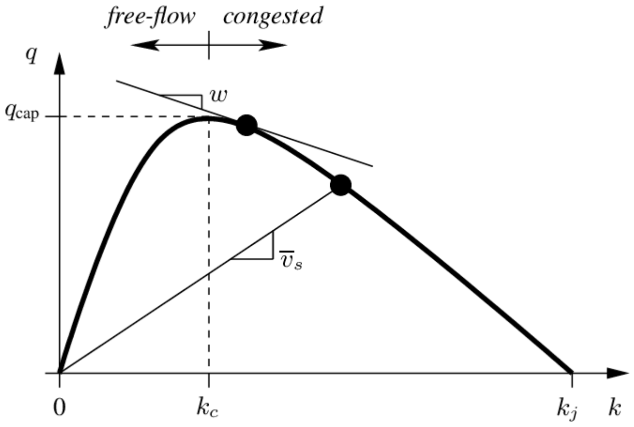


Figure 1.3: Example of a ρ–J fundamental diagram [12].

Velocity as a function of flow The $v(J)$ has a similar shape to the $J(\rho)$ curve with a free flow branch and a congestion branch. However, it does not represent a mathematical function, as two distinct average velocities can be observed at a single level of flow – one for each branch. Figure 1.4 contains a sample *J–v* diagram.

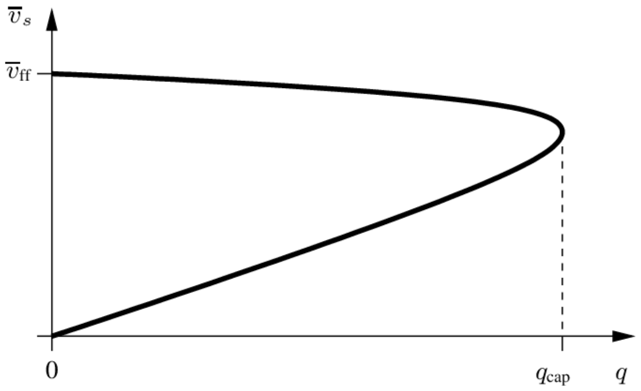


Figure 1.4: Example of a *J–v* fundamental diagram [12].

1.3 Mean k-NN Distance as a Measure of Density

The practical part of this thesis will focus on estimating the density–velocity relationship. However, instead of using the density directly, the inverse of velocity – *mean distance headway*, shall be used in its modified form of *mean distance to k-nearest neighbors.* This section provides a theoretical basis for using this quantity in the context of pedestrian movement.

In the beginning of section 1.1, the concept of *distance headways* was mentioned. In traffic flow theory, where only one-dimensional movement is considered, distance headways are a microscopic variable that measures the distance to the "vehicle in front". Assume a road of length d, starting at a and ending at b, occupied by N vehicles, positioned at $x_1, x_2, \ldots, x_N$, such that $a \leq x_1 < x_2 < \cdots < x_N \leq b$. Distance headway is the distance $d_i = (x_{i+1} - x_i)$ for $i \in \{1, \ldots, N\}$, where $X_{N+1} = b$. If we further assume that the first vehicle is at the beginning of the measurement line, $x_1 = a$, then the mean distance headway will be equal to the inverse of density:

$$\frac{1}{N}\sum_{i=1}^{N} d_i = \frac{1}{N}\sum_{i=1}^{N}(x_{i+1} - x_i) = \frac{x_{N+1} - x_1}{N} = \frac{b-a}{N} = \frac{d}{N} = \left(\frac{N}{d}\right)^{-1} = \rho^{-1}.$$

This property cannot be easily generalized to pedestrian movement, as there is no clear choice for the "next" pedestrian. However, if we can find a partitioning of the measurement area A into sub-areas $A_1, A_2 \ldots A_N$, such that $\forall i, j \in \{1, \ldots, N\}$,

- $\sum_i |A_i| = |A|$,
- $\dim(A_i \cap A_j) < \dim(A)$ (the intersection is at most a line), and

- there is exactly one pedestrian in each A_i,

the equation above will still hold:

$$\frac{1}{N}\sum_{i=1}^{N}|A_i| = \frac{|A|}{N} = \left(\frac{N}{|A|}\right)^{-1} = \rho^{-1}.$$

This is precisely the case with Voronoi cells, defined in section 1.1.1. This means that the area of a Voronoi cell can be used as a measure of local density and when aggregated over the entire area, the actual density can be computed. In fact, the third assumption can be omitted. Assuming n cells, with N_i being the number of pedestrians in i-th cell, such that $\sum_i N_i = N$, the corresponding subdensity becomes $\rho_i = N_i/|A_i|$ and the total density can be obtained via a *weighted harmonic mean* of the subdensities with n_i as weights:

$$\frac{\sum_{i=1}^{n} N_i}{\sum_{i=1}^{n}\frac{N_i}{\rho_i}} = \frac{N}{\sum_{i=1}^{n}\frac{N_i}{n_i/|A_i|}} = \frac{N}{\sum_{i=1}^{n}|A_i|} = \frac{N}{|A|} = \rho.$$

In practice, this approach faces several problems. Namely, the cells corresponding to the pedestrians at the edge of the crowd will span all the way to the edges of the measurement area, making them disproportionately large. This may be solved by introducing a cut-off radius, assigning a maximum area to the outside cells. But even then, the area of Voronoi cells shows great variance and is heavily skewed by outliers, as shown in figure 1.5.

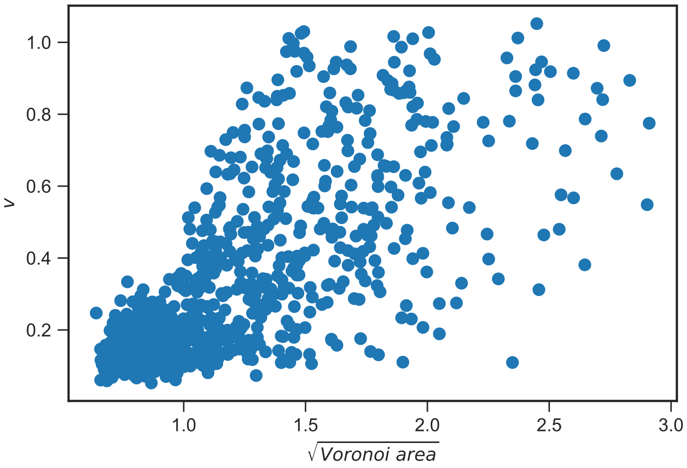


Figure 1.5: Relationship between the square root of Voronoi area and speed.

For this reason, a more robust measure of local density will be used – *mean distance to k-nearest neighbors.*

Definition 6 (Mean k-NN distance)**.** Let $\vec{x_i}$ be the position of pedestrian i, $k \geq 1$ a natural number, and $\vec{x_1}, \vec{x_2}, \ldots, \vec{x_k}$ the position of k-nearest neighbors of pedestrian i with respect to a metric $\| \cdot \|$.
The *mean k-NN distance* $\bar{d_k}$ is defined as

$$\bar{d_k} = \frac{1}{k} \sum_{j=1}^{k} \|\vec{x_i} - \vec{x_j}\|. \tag{1.13}$$

Intuitively, if the nearest pedestrians are relatively far away, there is more room to move freely, at higher velocities. On the other hand, if the nearest neighbors are very close, there is limited room to move without crashing into others. Additionally, this measure is naturally bounded by the distance to the closest pedestrian and the distance to the k-th nearest pedestrian, making it more robust. After discarding the outliers, this method shows significant correlation with the square root of the mean Voronoi cell area, and more importantly, almost a linear relationship with the mean velocity (figure 1.6).

Nevertheless, it can be skewed in situations where there are multiple distinct crowds in the measurement area and k is larger than the size of either crowd, therefore indicating low overall density, despite the local density being high. Such situations can happen during the evacuation of rooms with multiple exits – a crowd will form in front of each exit. This will not be a problem in this thesis, as only a single-exit geometry will be considered.

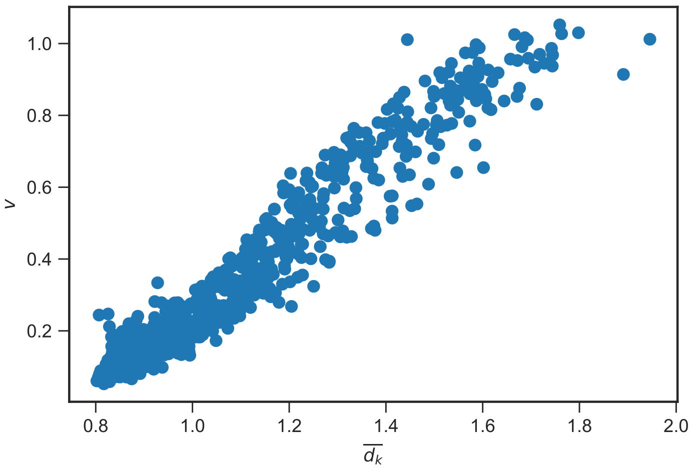


Figure 1.6: Relationship between the mean k-NN distance and speed.

Chapter 2

Simulating Pedestrian Movement

This chapter will provide a basic overview of the methods used to simulate pedestrian movement. A more detailed description of *social-force* models will follow, with emphasis on the *Generalized centrifugal-force model*, which is used in the practical part of this thesis.

2.1 Pedestrian Models

As Schadschneider notes in [14], there is a multitude of criteria for classifying pedestrian models. The following summary is on selected sections from [3] and [14].

Most pedestrian models can be thought of as *microscopic* or *macroscopic.* Macroscopic models treat groups of pedestrian as a homogenous collection of indistinguishable particles, whose movement is based on macroscopic characteristics of the crowd, such as average density and velocity. Microscopic models are more complex, in the sense that they treat each pedestrian as an individual agent that interacts with other pedestrians. More detailed behavior, like individual route planning, can take place. Microscopic models are further divided into *rule-based vs. force-based.*

Rule-based models are inspired by human psychology and base their decision-making process on a predefined set of rules. Each pedestrian has an internal state and his moves depend on his immediate neighborhood. A prominent example of rule-based models are *cellular automata (CA) models.* The main idea of CA models is to discretize the geometry into a grid of cells and model each pedestrian as a *cellular automaton*, occupying one or more cells. Their movement is realized by transitions between cells, based on probabilities designed to reflect simple rules:

1. Move closer to your goal.

2. Avoid collisions with other pedestrians.

3. Avoid collisions with terrain.

Example of CA models include *Fukui-Ishibashi model*, *Blue-Adler model*, or *Floor Field CA*.

On the other hand, the movement in force-based models is determined by outside forces acting on the pedestrian alongside an internal driving force. The resulting acceleration is obtained as a superposition of all the forces the pedestrian "feels". For example, all agents will feel an attraction force towards their goal (e.g., a door or an exit), which causes them to move towards that goal. Additionally, when close together, the agents will act on each other with a repulsive force that prevents them from crashing. Force-based models include various versions of the *social-force model*, which is used in this thesis, and a high-level overview will shortly be provided.

Force-based approach is not limited to microscopic models, macroscopic models will generally also be force-based. Other criteria include *deterministic vs. stochastic* or *discrete vs. continuous.* Of course, there are models that combine multiple approaches such as the *lattice gas model.* This model works on a triangular grid, similarly to a cellular automaton, but uses a force-based approach to compute the agents' velocities. Again, the following summary is based on selected chapters from [3].

2.2 Social-Force Models

Social-force [15] models are based on Newton's second law of motion, $\vec{F} = m\vec{a}$, with m being the mass and a acceleration. The net force $\vec{F}$ is a superposition of forces that describe

1. the pedestrian's own motivation to reach his goal,

2. the effect of nearby pedestrians,

3. the effect of the environment, and

4. miscellaneous physical forces, such as friction.

Since acceleration is the second time-derivative of position, the velocity vector can be found by numerical integration. The basic equation of motion for pedestrian i is given by

$$m_i \frac{\mathrm{d}}{\mathrm{d}t} \vec{v_i} = \vec{F}_i^{(\mathrm{pers})} + \vec{F}_i^{(\mathrm{soc})} + \vec{F}_i^{(\mathrm{env})} + \vec{F}_i^{(\mathrm{phys})}. \tag{2.1}$$

The *personal driving force* $\vec{F_i}^{(\text{pers})}$ of pedestrian i is the difference of the *desired velocity* $\vec{v_i}^{(0)}$, usually in the direction of an exit, with magnitude equal to the average free-flow speed, and the *current velocity* $\vec{v_i}$, divided by a *time factor* τ_i, which corresponds to a *reaction time.* It is also proportional to the mass of the pedestrian m_i, although this parameter is often set to unity for simplicity.

$$\vec{F_i}^{(\text{pers})} = m_i \frac{\vec{v_i}^{(0)} - \vec{v_i}}{\tau_i}. \tag{2.2}$$

The *social force* $\vec{F_i}^{(\text{soc})}$ is the sum of the forces produced by the other pedestrians and acting on i, i.e. $\vec{F_i}^{(\text{soc})} = \sum_{j \neq i} \vec{F_{ij}}^{(\text{soc})}$. Since it would be needlessly numerically expensive to compute the forces between every pair of pedestrians, only the ones in a close radius r_c are considered. Because people generally become uncomfortable when too close to strangers, this acts as a repulsive force. The exact equation varies from model to model, but in general, the individual velocities are inversely proportional to the distance between i and j, taking into consideration the shape of the pedestrian (a circle or an ellipse) and the angle between $\vec{v_i}$ and j's position (people only react to the pedestrians in front of them).

The *environment force* $\vec{F_i}^{(\text{env})}$ is modelled similarly to the social force, i.e. people do not walk too close to the walls. GCFM uses three points on each wall within a given range to compute the repulsive force, using a simplified version of the social force [16].

The *physical force* $\vec{F_i}^{(\text{phys})}$ can be used to implement concepts such as *push* or *friction*, however, in the model used in this thesis it is neglected.

2.3 Generalized Centrifugal-Force Model

The model that has been used in this thesis to simulate pedestrian movement is the *Generalized centrifugal-force model (GCFM).* It was developed by M. Chraibi et al. in 2010 [16], and is a generalization of the *Centrifugal-force model (CFM)* [17]. The force $\vec{F_{ij}}^{(\text{soc})}$ in CFM is given by the following equation:

$$\vec{F_{ij}}^{(\text{soc})} = -m_i k_{ij} \frac{v_{ij}^2}{d_{ij}} \vec{e_{ij}}, \tag{2.3}$$

where

- m_i is the mass of pedestrian i,
- k_{ij} is a coefficient that represents the field of vision of the pedestrian,
- d_{ij} is the distance between i and j,

- v_{ij} is the norm of the relative velocity between i and j, and
- $\vec{e_{ij}}$ is the unit direction vector from i to j, i.e., $\vec{d_{ij}}/\|\vec{d_{ij}}\|$.

The coefficient k_{ij} ensures that only pedestrians within a 180° field of view are considered, and the maximum force is attained for the persons directly in front of i. If θ is the angle between $\vec{v_i}$ and $\vec{e_{ij}}$, then k_{ij} is given by

$$k_{ij} = \begin{cases} \cos(\theta) & \text{if } \vec{v_i} \cdot \vec{e_{ij}} > 0 \text{ and } \vec{v_i} \neq \vec{0}, \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

Additionally, v_{ij} is defined such that slower pedestrians are not affected by faster pedestrians in front of them,

$$v_{ij} = \begin{cases} (\vec{v_i} - \vec{v_j}) \cdot \vec{e_{ij}} & \text{if } (\vec{v_i} - \vec{v_j}) \cdot \vec{e_{ij}} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

CFM also implements a *collision detection technique* that prevents excessive overlap of pedestrians.

The Generalized centrifugal-force model modifies CFM in a way so that no explicit collision detection is needed. Additional realism is achieved by assuming a dynamically expanding, elliptical shape of agents, rather than a static and circular one. The ellipse modelling a pedestrian has the semiaxes a and b, a being the major semiaxis (in the direction of movement) and b being the minor semiaxis (in the lateral direction). Both of these axes are functions of the current velocity. This is motivated by the fact that faster-moving people

(a) need more forward space to maneuver safely, and
(b) tend to sway less laterally (i.e., their trajectories are more “straight”).

This also means that it does not always hold that $a > b$, such as when standing still. The semiaxes take the forms

$$a(v_i) = a_{\min} + \tau_a v_i, \tag{2.6}$$

$$b(v_i) = b_{\max} - \frac{v_i}{v_i^{(0)}}(b_{\max} - b_{\min}), \tag{2.7}$$

where $a_{\min}$, $b_{\min}$, $b_{\max}$, $v_i^{(0)}$ can be determined empirically, and τ_a is a free parameter. The modified repulsive force is then given by

$$\vec{F_{ij}}^{(\text{soc})} = -m_i k_{ij} \frac{(\eta v_i^{(0)} + v_{ij})^2}{d'_{ij}} \vec{e_{ij}}. \tag{2.8}$$

The difference from CFM is in the extra term $\eta v_i^{(0)}$, where η is a free parameter that controls the strength of the force. It is proportional to the desire speed $v_i^{(0)}$, as faster pedestrians also need more “headroom”. This solves the problem of vanishing repulsive forces and overlap with low relative velocities,

but introduces oscillation at bottlenecks with high values of η. Therefore, the parameter η has to be carefully calibrated to get realistic results.
The second difference is in the distance d'_{ij} which is the *effective distance* between i and j, meaning that the elliptic pedestrian shape is taken into account and a distance between ellipses is computed.

The authors of [16] have shown that this model can reasonably recreate human behavior on a quantitative level (i.e., the fundamental diagram), but will not reproduce certain collective phenomena, such as changing lanes and overtaking.

The typical values of GCFM parameters, as given in [3] and [16], are shown in table 2.1.

Table 2.1: Typical parameter values for force-based models.

Parameter	**Value**
$v_i^{(0)}$	0.6–1.5 m/s
m_i	1
τ_i	0.5 s
$a_{\min}$	0.18 m
τ_a	0.5 s
$b_{\min}$	0.20 m
$b_{\max}$	0.25 m
η	0.1–0.6

Chapter 3

Practical Simulation

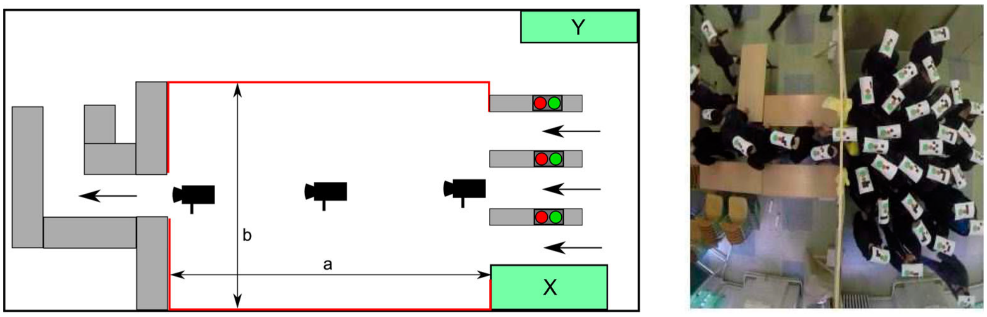
This chapter will start with a summary of the experiment regarding flow at bottlenecks, conducted at the CTU in Prague in 2014 [1]. This experiment was the basis for simulation and subsequent FD estimation done in this thesis. After that, a description of the simulation process will follow, along with validation of the simulated data using the fundamental diagram. This chapter will conclude with processing of the simulated trajectories to compute the mean k-NN distance for each row of the resulting table.

3.1 Experiment Description

The experiment in question was a part of the “Experimental Study of Phase Transition in Pedestrian Flow” study by M. Bukáček, P. Hrabák, and M. Krbálek [1]. It involved a simple artificial room with 3 entrances, each approximately 50 cm wide, and a single exit 60 cm wide. The room itself was rectangular, with the distance from the middle entrance to the exit being $a = 7.2$ m and the width being $b = 4.5$ m. A schema of the geometry is provided in figure 3.1.

The pedestrians were composed of 76 student volunteers, who would repeatedly pass through the room. The idea of the experiment is described in [18]:

> *The volunteers were instructed to enter the room, pass through it as fast as possible avoiding running, and then return to the pedestrian cluster in front of the entrance. This technique enabled to maintain constant flow through the room. The inflow rate was controlled using three independent signalling devices informing the pedestrians in the crowd to enter the room trough one of three available entrances. To simulate random inflow conditions, green light was alternated by $k\dot{\Delta}h$ seconds of red light, where k was generated from geometric distribution. $\Delta h = 0.6$ s was the minimal*



(a) (b)

Figure 3.1: (a) The diagram of the geometry used in [1] and (b) a snapshot of the experiment [18].

> *time step, to which pedestrians were able to react reliably. Each round started with empty room.*

The room was recorded by three cameras for the entire duration of the experiment. Automatic extraction of the trajectories was possible due to the students wearing special markers on their head.

Unfortunately, we were unable to obtain the actual trajectory data in time to reliably incorporate it into the model, therefore artificial data will have to suffice.

3.2 Simulation

The simulation itself was realized in the open-source framework for simulating pedestrian movement JuPedSim [19]. The framework is capable of creating complex simulations in custom geometries, with the main focus on evacuations. Apart from the simulations themselves, it is capable of visualizing the results in the form of animations as well as providing basic analysis of the trajectories by computing the selected characteristics of the pedestrian flow, such as travel times or densities in given measurement areas.

Geometry The geometry described in the previous section was modelled using three connected subrooms (named A, B, C):

- A, the main measurement room – 7.2 m × 4.5 m
- B, an initial room (a “spawn point”) – 5.0 m × 4.5 m
- C, an exit corridor – 2.0 m × 0.6 m

The floor plan of the geometry is shown in figure 3.2.

At the beginning of the simulation, all agents are randomly distributed throughout room B and their goal is to reach the far side of room C, where

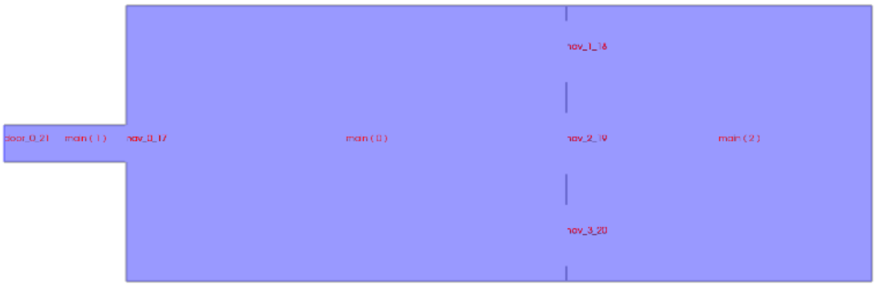


Figure 3.2: The geometry modelled in JuPedSim.

an imaginary exit is located. There are three “doors” between room B and room A in accordance with the original experiment. The random starting positions lead to random times of entering room A, simulating random inflow conditions. JuPedSim also supports pedestrian sources that create new agents continuously over a given period, however, this approach proved more difficult to calibrate and automate.

Model and agent parameters As mentioned in chapter 2, the Generalized centrifugal-force model was used as the operational model of the individual pedestrians. In addition to the force parameter η described in section 2.3, three extra parameters are required (f, r, ϵ). These are used to ensure numerical stability and do not greatly affect the quality of the simulation. The values of these parameters were determined by a process of trial-and-error, based off of the recommended values provided by the developers.

In order to partially simulate the heterogeneity in pedestrian crowds, JuPedSim allows agent parameters to be modelled as normally distributed random variables with a mean μ and variance σ^2. The values of μ were picked using the recommendations from table 2.1, with the standard deviation σ an order of magnitude lower. The concrete values are shown in table 3.1.

Number of pedestrians and batch simulation The output of a single simulation is a set of trajectories, that is, the x and y coordinates of each agent at each point in time (called a “frame”), along with additional quantities including the velocity. Since no continuous generation of new pedestrians is employed, the macroscopic characteristics of the flow are determined solely by the initial number of pedestrians.

In order to get a diverse dataset for further analysis, it is necessary to run multiple simulations with different numbers of pedestrians. However, JuPedSim only allows one simulation to be run at a time. For this reason, a script

Table 3.1: Concrete parameter values for simulation.

	Parameter	**Value**
Model param.	η_{ped}	0.375
	f_{ped}	5.0
	r_{ped}	2.0
	ϵ_{ped}	0.1
	η_{wall}	0.2
	f_{wall}	1.0
	r_{wall}	2.0
	ϵ_{wall}	0.1
Agent param.	$v^{(0)}$	$\sim \mathcal{N}(1.3, 0.2^2)$
	τ	$\sim \mathcal{N}(0.4, 0.001^2)$
	a_{min}	$\sim \mathcal{N}(0.15, 0.001^2)$
	τ_a	$\sim \mathcal{N}(0.35, 0.005^2)$
	b_{min}	$\sim \mathcal{N}(0.2, 0.002^2)$
	b_{max}	$\sim \mathcal{N}(0.25, 0.002^2)$

that automates the simulation process has been written. It allows running multiple simulations in a row, taking the number of agents via a range argument similar to Python's `range()` function. It sets up the output directory structure, modifies the configuration file of the simulation accordingly, and runs the simulation, repeating this over the given range.

The next question is what agent numbers to choose. Evacuating more people (i.e., high-density situations) is generally going to take longer, resulting in more data. Indeed, a simulation of 8 pedestrians results in, on average, approximately 1 100 data points, whereas a simulation of 50 pedestrians produces more than 24 000 data points. Therefore, it is necessary to conduct more simulations of low-density evacuations, otherwise the data would be overly biased. To keep the dataset somewhat balanced, the simulations were classified into three categories based on the number of pedestrians $p \in \mathbb{N}$ present.

- $p \in [0, 15) \Rightarrow$ low density
- $p \in [15, 30) \Rightarrow$ medium density
- $p \in [30, \infty) \Rightarrow$ high density

The endpoints of these intervals are not precise, as they were chosen solely based on observing the qualities of the pedestrian crowd in JuPedSim animations. However, these rough estimates are enough to offset the domination of high-density simulations. A total of 383 108 data points was acquired in simulations. The distributions of the individual runs are shown in table 3.2.

Table 3.2: Composition of the complete dataset.

ρ	**Agents**	**Iterations**	**Total data points**
Low	6	10	11 856
	8	10	17 699
	10	10	24 870
Med.	15	5	23 051
	20	5	36 563
	25	5	51 579
High	30	3	41 534
	40	3	73 009
	50	3	102 947

3.3 Empirical FD of Simulated Data

In order to validate the simulation, it is necessary to verify that the empirical fundamental diagram obtained from the simulated data conforms to the expected shape described in section 1.2. JuPedSim features a tool for automatic extraction of selected flow properties from trajectory files. These include the average spatial density $\langle \rho \rangle_{\Delta x}$ in both the classic and Voronoi form as well as the corresponding spatial average velocity $\langle v \rangle_{\Delta x}$ as defined in section 1.1. This tool, however, can only process a single file at a time, which is why it was necessary to create a batch script, similar to the one in the previous section. This allows running the necessary computations over the entire dataset at once.

The resulting data from which the fundamental diagram will be created has the following form:
For each time frame t of every simulation $S_1, S_2, \dots, S_N$, the mean spatial density and mean spatial velocity is returned, i.e., $t \mapsto [\langle \rho_t \rangle_{\Delta x}, \langle v_t \rangle_{\Delta x}]$. By averaging over t, we get N independent points from which the ρ–v diagram can be constructed. The resulting empirical FD is shown in figure 3.3.

The shape of the fundamental diagram in both the classical and Voronoi form matches the theoretical properties described in section 1.2, in that the average speed decreases monotonically with growing density and tends to zero. The theoretical range of densities where the average speed is constant is not present, as the bottleneck is relatively narrow (0.6 m) and the pedestrians negatively affect each other even in very low densities.

Numerically, the densities here are skewed towards lower values. The reason is that the measurement area is static and does not consider the local distribution of the pedestrians. This can be seen from the temporal plot of density and velocity in the individual runs (3.4a). The density increases as new pedestrians enter the room until it reaches a maximum value. After that,

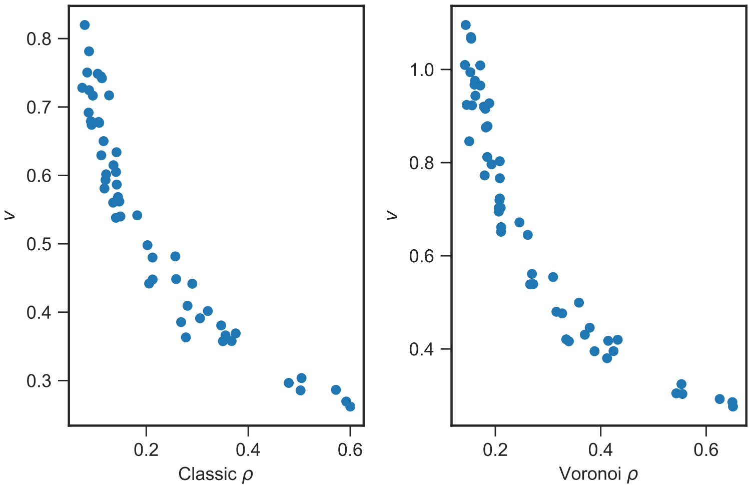
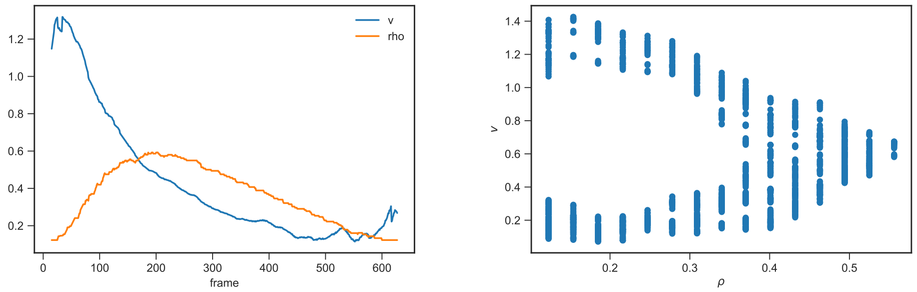


Figure 3.3: The empirical ρ–v FD obtained from simulations.

as pedestrians leave the room, the density decreases. On the other hand, the average velocity steadily decreases over the entire course of the evacuation. This means that when the (immediate) average velocity is plotted as a function of density, a single value of ρ can correspond to two values of v (3.4b), one for the "front" of the crowd and one for the "back". While the former is expected behavior, the latter is a "false positive", meaning that despite the perceived local density being high, the global density is relatively low, causing a bias towards lower densities.



(a) The temporal evolution of ρ and v.

(b) Individual FDs with different k.

Figure 3.4: Empirical FD of simulated data.

The mean k-NN distance proposed in section 1.3 does not have this problem, as it does not depend on the measurement area. The following section will discuss its computation as well as the choice of the appropriate k value.

3.4 k-NN Distance Computation and Choice of k

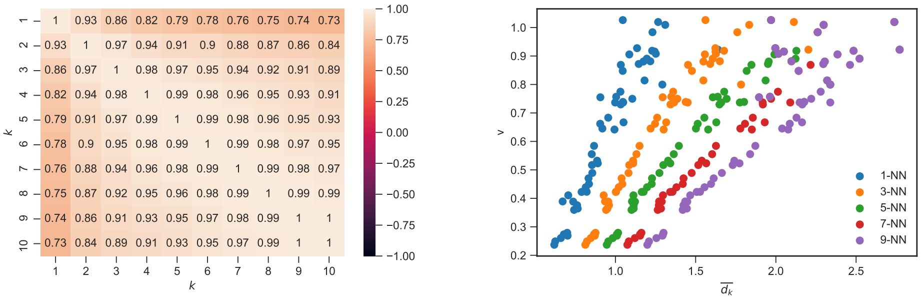
Computation The computation of the mean k-NN distance was done using the module `spatial` of the Python library SciPy[1]. This module supports fast computations of nearest-neighbor queries using the class `KDTree`, which implements a *kd-tree* (stands for *k-dimensional tree*) data structure for nearest-neighbor lookup. Details of the algorithm are provided in [20].

The output of a kd-tree query is the coordinates of the k nearest neighbors. Since the main measurement area is rectangular and therefore convex, all of the nearest neighbors found will be valid. In more complex geometries, where a nearest neighbor may be behind a wall, this may pose a problem. An upper bound for k was chosen as $k = 10$, in accordance with [2]. Since $k-1, k-2, \ldots, 1$ nearest neighbors are a subset of k nearest neighbors, it is computationally cheap to also calculate all of the lesser mean distances. In addition to the distances themselves, the *relative* positions of nearest neighbors were also stored, as they will be used in section 4.3.3 to improve the predictions of the model. In the situations where less than k pedestrians remain (e.g., at the end of the simulation), the mean distance to the remaining ones was stored and the extra relative coordinates were set to infinity.

Choice of k The choice of k can be regarded as a hyperparameter of the model. In the case of this simulation, where only a single large crowd forms, the concrete choice does not have a significant impact on the overall performance of the model. As can be seen in figure 3.5a, the individual distances correlate heavily. Figure 3.5b shows the relationship between different k-NN distances and the velocity. In this figure, the distances were discretized into intervals of even length and the mean velocity was computed for each interval. The different choices of k show similar behavior. With larger k, the distances become larger and more numerically stable.

A disadvantage of large values of k is that they result in the necessity to discard more data, as it would contain invalid values in the form of missing or infinite relative coordinates. For this reason, a value of $k = 5$ was chosen for further prediction. The empirical fundamental diagram of 5-NN-dst.–v is given in 3.6. Almost a linear relationship is shown. However, the true underlying relationship is clearly not linear, seeing as the average velocity will not grow indefinitely, but will instead stabilize at a free flow speed $v_{\max}$. This problem will be solved in the next chapter.

[1] `https://scipy.org/`



(a) Correlations of individual k-NN distances.

(b) Individual FDs with different k.

Figure 3.5: Comparison of k-NN distances by k.

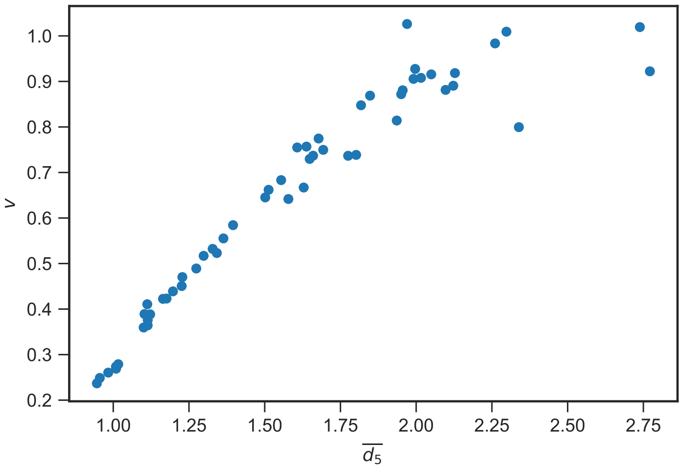


Figure 3.6: The empirical fundamental diagram of $\bar{d}_5$ and v.

Chapter **4**

Estimating the Fundamental Diagram

This chapter will start with a description of the analytical approach to estimating the fundamental diagram, and a baseline exponential fit will be provided. After that, the relevant concepts regarding neural networks will be introduced, along with a discussion of their use in the field of pedestrian dynamics. Finally, after further preprocessing of the data, practical fitting of a neural network will take place.

4.1 Analytical Forms

As was mentioned in section 1.2, there is no general consensus on the precise shape of the pedestrian FD. This can be illustrated using figure 4.1, where various shapes of the ρ–J from different experiments are shown.

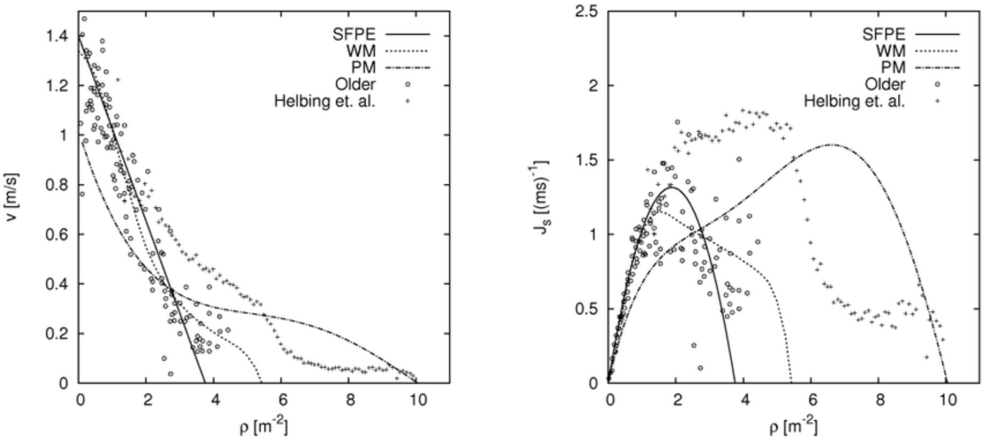


Figure 4.1: Different empirical FDs of pedestrian movement [3].

Schadschneider in [3] gives a comprehensive list of mutually disagreeing studies regarding the shape of the fundamental diagram. He concludes with the following words:

> *This brief discussion clearly shows that up until now there is no consensus on the precise form of the fundamental diagram and even on the origin of the observed discrepancies. This is not only unsatisfactory with regard to applications especially in safety planning, but also makes the validation and calibration of models rather difficult.*

These discrepancies may be caused different measurement methods, different types of flow, and even cultural differences.

An often cited analytical form of the FD was given by Weidmann in [21]. It models the ρ–v relationship as an exponential function with the parameters v_0, ρ_0, and $\rho_{\max}$.

$$v(\rho) = v_0 \left[1 - \exp\left(\frac{1/\rho_{\max} - 1/\rho}{\rho_0}\right)\right]. \tag{4.1}$$

He fitted this model with the values $v_0 = 1.34$ m/s, $\rho_0 = 1.913$ P/m^2, and $\rho_{\max} = 5.4$ P/m^2.

It was argued in section 1.3 that the mean k-NN distance $\bar{d_k}$ is a reasonable approximation of $1/\rho$. Substituting this into equation 4.1 gives us an analytical approximation of the $\bar{d_k}$–v relation.

$$v(\bar{d_k}) = v_0 \left[1 - \exp\left(\frac{1/\rho_{\max} - \bar{d_k}}{\rho_0}\right)\right]. \tag{4.2}$$

This equation, although in a different but equivalent form, was used as a benchmark for the prediction in [2] and it will serve the same purpose in this thesis. It should be noted that the ρ in equation 4.1 is a macroscopic quantity, whereas the $\bar{d_k}$ is a microscopic quantity that describes the local density. Since the total density is the harmonic mean of the subdensities (from section 1.3), the velocity obtained as the arithmetic mean of $v_i(\bar{d_k})$ will be an overestimation of the $v(\rho)$ from equation 4.1.

Fitting the benchmark model The values provided by Weidmann are not universal and should be adapted to fit our data. The module `scipy.optimize` provides a `curve_fit()`[2] function for this purpose. It uses a non-linear least squares solver to fit a given function to the data. Since our parameters are naturally non-negative, bounds can be provided for the solver. The fitted parameters are shown in table 4.1.

[2] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html`

Table 4.1: Fitted parameters of the Weidmann model.

Parameter	**Value**
v_0	1.67169 m/s
ρ_0	1.44555 P/m^2
$\rho_{\max}$	1.59436 P/m^2

The least-squares estimates of the parameters do not perfectly match their physical interpretation, e.g., an average speed of 1.67 m/s is too high. Nevertheless, the curve itself provides a decent fit for the data, as shown in figure 4.2. The per-run averages from figure 3.6 are also included. It generally holds true that the curve obtained via the local densities slightly overestimates the values obtained via per-run averages.

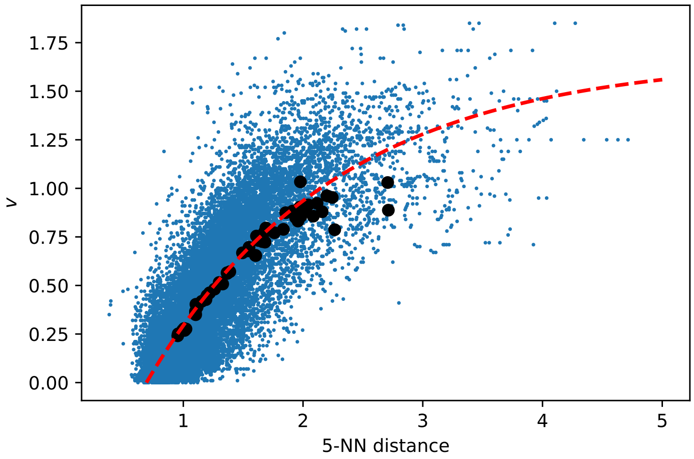


Figure 4.2: Simulated data fit by a Weidmann curve. The black points indicate the per-run averages from figure 3.6

4.2 Artifical Neural Networks

This section assumes a knowledge of machine learning and artificial neural networks (ANNs) at the level of the BI-VZD course taught at FIT CTU in the winter semester of 2019[3]. After a discussion of the use of neural networks in the field of pedestrian dynamics, the relevant concepts used in this thesis will be introduced. The notation in this section will mirror the notation used in the aforementioned course., i.e., vectors will be denoted by a bold font instead

[3] `https://courses.fit.cvut.cz/BI-VZD/@B191/`

of an arrow superscript. The theory of neural networks themselves will not be explored, as it is not the primary focus of this thesis. A comprehensive introduction to ANNs can be found in [22].

4.2.1 ANNs in Pedestrian Dynamics

With the rising popularity of ANNs in the recent years, authors have begun to test their usability in problems related to pedestrian motion. The advantage of this data-driven approach is that ANNs automatically adapt to the given geometry without the need to specifically calibrate a FD-based model. The downside is that the trained parameters of a neural network are generally uninterpretable and the entire system works as a semi-black box.

Examples of the use of ANNs in pedestrian dynamics include predicting pedestrian speed using *multilayer perceptrons (MLPs)* in [2], whose work is expanded upon in this thesis. MLPs were also used in [23] to estimate different characteristics of pedestrian flow, and in [24] to predict entire trajectories, albeit only at a straight crossing. More complex neural networks were also used, for example *Long short-term memory (LSTM)* networks for trajectory prediction in [25]. ANNs were also successful in simulating fluid physics [26], and given the analogies of fluid dynamics and pedestrian dynamics, this could be expanded upon in the future.

4.2.2 Technologies and Concepts Used

This thesis uses a standard feedforward neural network (multilayer perceptron), as defined in the BI-VZD course [27]. For practical applications, the TensorFlow [4] implementation of the Keras API [5] is used. The computations are done in the Jupyter Notebook [6] environment, using standard Python libraries for numerical computing, including NumPy [7], Pandas [8], and Matpliotlib [9].

The layers themselves are *dense* – every neuron in layer l_i is connected to every neuron in layer l_{i+1}. The entire model is *sequential* – a linear stack of layers. The weights in individual layers are initialized using the *Glorot uniform initialization* method – chosen uniformly from the interval

$$\left[-\sqrt{\frac{6}{n_{\text{in}}+n_{\text{out}}}}, \sqrt{\frac{6}{n_{\text{in}}+n_{\text{out}}}}\right], \tag{4.3}$$

where n_{in} and n_{out} is the number of input units and output units respectively. The biases are initialized with zeros.

[4] https://www.tensorflow.org/

[5] https://keras.io/

[6] https://jupyter.org/

[7] https://numpy.org/

[8] https://pandas.pydata.org/

[9] https://matplotlib.org/

Loss functions As this is a regression task, both the *mean absolute error (MAE)* and *mean squared error (MSE)* were tested. Given N samples with Y_i being the true value of i-th sample and $\hat{Y}_i$ the predicted value, the MAE and MSE are given by

$$\mathrm{MAE}(Y, \hat{Y}) = \frac{1}{N} \sum_{i=1}^{N} \left| Y_i - \hat{Y}_i \right|, \tag{4.4}$$

$$\mathrm{MSE}(Y, \hat{Y}) = \frac{1}{N} \sum_{i=1}^{N} \left(Y_i - \hat{Y}_i \right)^2. \tag{4.5}$$

Activation functions Activation functions directly influence the shape of the resulting curve. Three activation functions were considered, namely the *rectified linear unit (ReLU)*, *exponential linear unit (ELU)*, and *hyperbolic tangent (tanh)*, given by the following equations:

$$\mathrm{ReLU}(x) = \max(0, x), \tag{4.6}$$

$$\mathrm{ELU}(x) = \begin{cases} x, & \text{for } x \geq 0, \\ e^x - 1, & \text{for } x < 0, \end{cases} \tag{4.7}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{4.8}$$

The output layer consists of a single layer and uses no activation function.

Optimizers TensorFlow implements a number of gradient descent optimization algorithms. In [22], Goodfellow suggests using algorithms with *adaptive learning rates*, e.g., AdaDelta, RMSProp, or Adam. This is in agreement with Ruder in [28], who considers Adam a reasonable choice. A process of trial-and-error has shown *Adam* [29] to give the most consistent results.

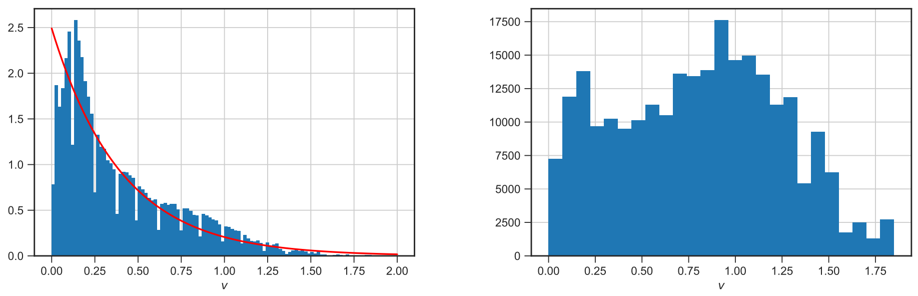
Regularization As will be shown later, the tested ANN models showed significant resistance to overfitting, with relatively fast convergence times and high stability. Therefore, no explicit weight regularization was employed.
It can be seen from the histogram (figure 4.3a) that the distribution of the velocities is heavily biased towards the lower values and the high speeds are rather sparse. In order to make the network pay more attention to higher velocities, *sample weights* were developed. Every sample's contribution towards the total loss gets scaled with its corresponding weight, i.e., the total loss J, given true values $Y_1, Y_2, \ldots, Y_N$, predictions $\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_N$, weights $w_1, w_2, \ldots, w_N$, and a loss function L is equal to

$$J = \frac{1}{N} \sum_{i=1}^{N} w_i \, L(Y, \hat{Y}). \tag{4.9}$$

As can be seen in figure 4.3a, an exponential distribution with the parameter $\lambda = 1/\bar{v}$, where $\bar{v}$ is the arithmetic mean of velocities, provides a reasonable fit. Simple weights can be obtained by finding a mapping of this distribution to a standard uniform distribution.

$$w \, \frac{1}{\bar{v}} \exp\left(\frac{-x}{\bar{v}}\right) = 1 \qquad \Longrightarrow \qquad w = \frac{\bar{v}}{\exp\left(\frac{-x}{\bar{v}}\right)}. \tag{4.10}$$

Weights obtained via equation 4.10 cause a shift in the (weighted) histogram towards higher velocities as intended (figure 4.3b). Whether or not to use these weights will be considered a hyperparameter.



(a) Normalized histogram of velocities, fit by an exponential curve.

(b) The weighted histogram of velocities.

Figure 4.3: The effect of weights.

4.3 Model Implementation

The model implemented in this section will predict the velocity based on only a single feature – the k-NN distance. The output will be a function $\hat{v}(\bar{d}_k) : \mathbb{R} \to \mathbb{R}$, the approximation of the fundamental diagram represented by the trained neural network. Before training, further preprocessing of the data will take place.

4.3.1 Preprocessing and Dataset Splitting

Data Cleanup The dataset contains points with falsely high velocity. This happens in situations directly in front of the exit as the individual pedestrians leave the room. When the way for them becomes clear, they drastically accelerate because their perceived density rapidly drops. This leads to points with low k-NN distance (high density) and high speed.

To make the learning more robust, attempts were made to discard these points. The most effective one proved to be simply discarding all the rows with k-NN distance smaller than the 0.001-quantile. This does not solve the problem on a fundamental level, but it results in discarding the obvious outliers, thus making learning more stable. The discarded points are shown in figure 4.4.

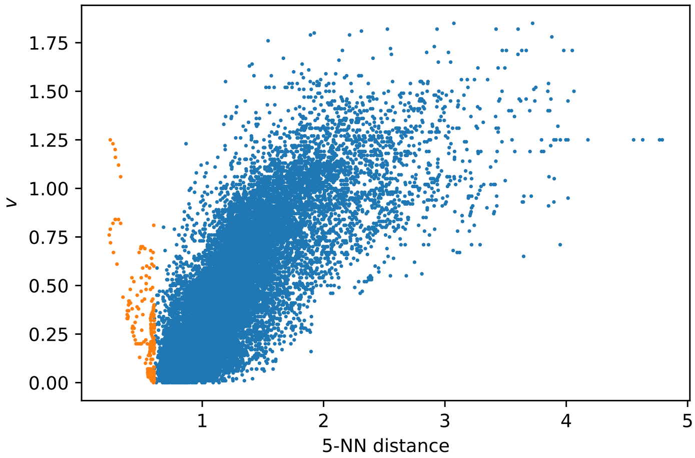


Figure 4.4: The data discarded via the 0.001-quantile (in orange) and a 10% sample of the total data (blue).

Train-Test Split The next step is to divide the dataset into a *training set*, a *validation set*, and a *test set*. An obvious way used even in some of the cited work is to randomly split the data into three disjunct subsets of given size. However, this approach is not fit for the task in this thesis. The individual data points from the same simulation run are *heavily* correlated, as they contain entire trajectories. A simple nearest-neighbors regressor reaches almost a perfect accuracy on the test set, while being clearly overfit(figure 4.5).

To keep the individual sets independent, the data was randomly split on a per-run basis, with the fraction being

$$\text{train} : \text{val} : \text{test} = 0.64 : 0.16 : 0.20$$

of the total runs, meaning that, for example, the test set contains 20% of the total runs, rounded to the nearest integer. Since the runs with more pedestrians contribute with more data points, in order to keep the fraction of the actual data points in each set roughly the same as the fraction of runs in said set (e.g., a 20% per-run split contains roughly 20% of the total data points), the runs were divided into three groups based on the number of pedestrians present, similarly to the classification in table 3.2. The intervals were slightly altered in order to keep the fractions more even. The split was then done in each group separately and the resulting datasets were obtained

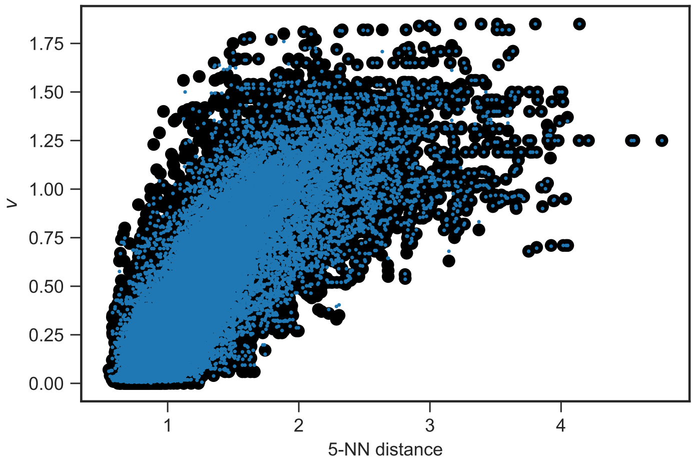


Figure 4.5: The predictions of a KNN regressor (blue) the true values (black).

via the union of the sub-splits. With the dataset properly split, we can proceed to training.

4.3.2 Training and Hyperparameter Tuning

The training is done in *epochs*. An epoch is a single pass over the entire dataset (i.e., when the network sees each sample once). After every epoch, the dataset is shuffled. TensorFlow divides the data into *minibatches* of size 32 (or less in case of the last minibatch), computes the gradient of the loss function over each minibatch and updates the parameters accordingly. The number of epochs is determined by *early stopping*, with a *patience* of 5 epochs. This means that the network monitors the *validation loss* and when this quantity has not improved for more than 5 consecutive epochs, the training is halted and the parameters corresponding to the lowest validation loss are recovered. This is to prevent overfitting. The upper limit of the number of epochs is set to 50, however, the training almost always finishes early.

With a reasonable learning rate, the training and validation errors almost do not improve in time. An acceptable solution is often found after the first epoch, and further training only causes slight improvements. This is not surprising, as the task being solved is essentially simple curve fitting of a continuous function of one variable.

Seeing that the training process is relatively fast, it is computationally acceptable to train the same network multiple times with different dataset splits in order to judge the robustness of the model with respect to the randomness introduced by random dataset splitting or the initial values of the weights.

Three different architectures were tested:

- "tiny" network – 1 hidden layer with 16 neurons
- "small" network – 2 hidden layers with 16×16 neurons
- "medium" network – 3 hidden layer with 32×32×32 neurons

Any further increase to the size of the network does not bring further improvement. Since one one explanatory variable is present, it is possible to plot the resulting curve.

The process of hyperparameter tuning was done using *grid search* – exhaustive testing of combinations from the cross product of given options. Both the MAE and MSE were tracked as a metric for every model. To sum up, the following hyperparameters were tested:

- *activation* $\in \{\text{ReLU}, \text{ELU}, \tanh\}$
- *architecture* $\in \{\text{tiny}, \text{small}, \text{medium}\}$
- *learning rate* $\in \{0.005, 0.001, 0.0005\}$
- *loss* $\in \{\text{MAE}, \text{MSE}\}$
- *weights* $\in \{\text{true}, \text{false}\}$

Since this amounts to more than 100 different models, the effect of the learning rate and weights was tested separately on a baseline model. Learning rates higher than 0.005 tend to underfit and finish with a linear model.

Other than the test loss, there are certain qualitative factors of the resulting curve that should be paid attention to:

- the behavior at low densities
- the behavior at high densities
- the variance of the curve with respect to dataset splitting

From our knowledge of the underlying model, we know that the pedestrians are modelled with $v^{(0)} \sim \mathcal{N}(1.3, 0.2^2)$. Therefore, it is desirable that the resulting function converges to the value 1.3 at high k-NN distances, as opposed to increasing indefinitely. At critically high densities, the predicted velocity should be very close to zero. This means that the model should not be influenced be the "falsely high velocity" points described in paragraph 4.3.1. Finally, since the resulting test errors generally only differ slightly, the models that show less variance with respect to data splits are preferred.

A complete set of plots containing the fitted functions can be found on the attached USB. Samples of typical behavior are shown in figure 4.6. The following observations can be made:

- The MAE and MSE achieved by the baseline Weidmann model are approximately 0.135 and 0.033 respectively.
- In the regions where most of the data points are situated, all the networks perform almost identically and in accordance with the Weidmann model.
- The output of the “tiny” networks shows significant variance as the k-NN distance grows.
- The shape of the resulting curves provided by the “small” and “medium” does not qualitatively change, suggesting that the small network is complex enough to capture the underlying relationship.
- The ELU and tanh activations perform similarly, while the “sharp edge” of the ReLU is clearly visible.
- Additionally, the ReLU-curves tend to not reach zero and instead stay parallel to the x-axis at a small value of v, suggesting that the ReLU activation is not suited for this particular task.
- MSE models consistently reaches the desired value of $v^{(0)} = 1.3$ m/s in high k-NN distances, while MAE models tend to slightly underestimate this values, especially with tanh activation.
- The lowest overall MSE is achieved with a medium network, MSE loss, and tanh activation, however, the differences are not significant enough to declare one model superior.

The effect of the learning rate and the use of weights was investigated on a small network, using the ELU activation, with both the MSE and MAE loss:

- The different learning rates do not have a major influence over the resulting curve, neither in terms of shape nor convergence speed.
- The use of weights universally worsens the performance of the network, causing them to overestimate the free flow velocity.

The next section concentrates on improving the prediction by introducing extra features.

4.3.3 Improving the Prediction

There is substantial variance in the distribution of $(v \mid \bar{d_k})$, as it is a microscopic variable. The models in the previous section were trained to extract the average macroscopic behavior, which is a function of the geometry and the pedestrian composition. But when more explanatory variables are available, better predictions are possible. For example, when processing a live camera feed, the immediate coordinates of each pedestrian can be extracted and used in the prediction.

This section will work at reducing the prediction error by introducing new features. Only the hyperparameters that proved effective in the previous section will be considered. Additionally, a "large" architecture consisting of 4 layers with $128 \times 128 \times 128 \times 128$ neurons will be tested. In summary, the full list of hyperparameters for this section is

- *activation* $\in \{\text{ELU}, \tanh\}$,
- *architecture* $\in \{\text{small}, \text{medium}, \text{large}\}$,
- *learning rate* $\in \{0.001\}$,
- *loss* $\in \{\text{MSE}\}$,
- *weights* $\in \{\text{false}\}$.

The training itself will not be modified, i.e., the effective number of epochs will be decided via early stopping.

Relative positions The first added explanatory variable is the *relative coordinates of nearest neighbors*, similarly to [2]. This could potentially give the neural network the ability differentiate the pedestrians leading the crowd. They travel at a high velocity despite their neighbors being relatively close-by. The outliers that were dropped in paragraph 4.3.1 are purposefully left in the dataset – they are in front of the crowd and their relative positions should indicate that. Figure 4.7 shows a sample of the predicted values and the true values for each hyperparameter. The complete set is on the attached USB. Both the test MAE and test MSE drop significantly with the introduction of relative positions. The small architecture fails to detect the pedestrians that sharply accelerate near the bottleneck. The medium and large network, however, detects these situations and makes the prediction accordingly. The lowest average MAE of 0.070 and MSE of 0.009 is achieved with the large network using the ELU activation, with the standard deviations an order of magnitude smaller. For comparison, the MAE and MSE achieved with the standard Weidmann model is 0.131 and 0.031 respectively (the shift from the values in the previous section is caused by the slightly larger dataset).

Relative distance to exit The second added variable is the relative distance to the exit. This enables the network to detect areas where the average velocity tends to drop. The major disadvantage of this is that outside information about the geometry is introduced into the system. Until this point, no explicit properties of the measurement area were used. This is not necessarily a bad thing. For practical purposes, specific calibration for a given geometry is desirable, as long as it improves the overall performance of the model. It may even be necessary if the room is non-convex or contains obstacles the network should know about. Figure 4.8 illustrates the predicted values similarly to previous paragraph.

Both the MAE and MSE have further dropped, suggesting that the extra information about the geometry brought new information to the model. The ELU activation outperforms tanh in every architecture.

In order to see how much of the geometry information has "leaked" into the model, one can test the prediction on a dataset that has been horizontally flipped, that is, for every point (x, y), the augmented dataset will contain the the point $(-x, y)$, with the $\bar{d_k}$ and v staying the same. A network that has been trained to recognize the relationship between density and speed should be invariant to this transformation. Figure 4.9 shows the predictions of both the extended networks. The model that only uses the relative positions of other pedestrians, while achieving worse scores, retains the general shape of the relationship. On the other hand, the second model's predictions *completely* break down and the output values are nonsensical. It is clear that the introduction of extra geometry information played a major role in shaping the predictions.

This could be combated by rotating each point (x, y) before training by a random angle θ using the *rotation matrix*

$$\mathbf{R} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix},$$

thus forcing the network to only work with the distances, however, further experiments are needed.

4.4 Experimental Data Validation

As was mentioned earlier, the trajectory data from [1] could not be obtained in time for practical use. This section will discuss the extra steps required to validate the trained models with experimental data.

Assuming trajectory files containing only the x and y coordinates of each pedestrian in every point in time, the true velocities would have to be approximated. This can be done using the methods described in section 1.1.2, namely by equation 1.6. An appropriate time window Δt would have to be selected in order to minimize excessive fluctuations in time. The speed would then be calibrated to match the selected units m/s.

With the speeds computed, the learning can proceed as with artificial data. The video recording of the experiment [10] shows that the students involved in the experiment formed a crowd denser than the one in simulations. This would likely lead to a different slope of the resulting fundamental diagram.

[10] `https://www.youtube.com/watch?v=v1jtmqjZOMO`

Attempts to recreate highly dense crowds in JuPedSim proved difficult to calibrate properly and led to excessive overlapping of pedestrians.

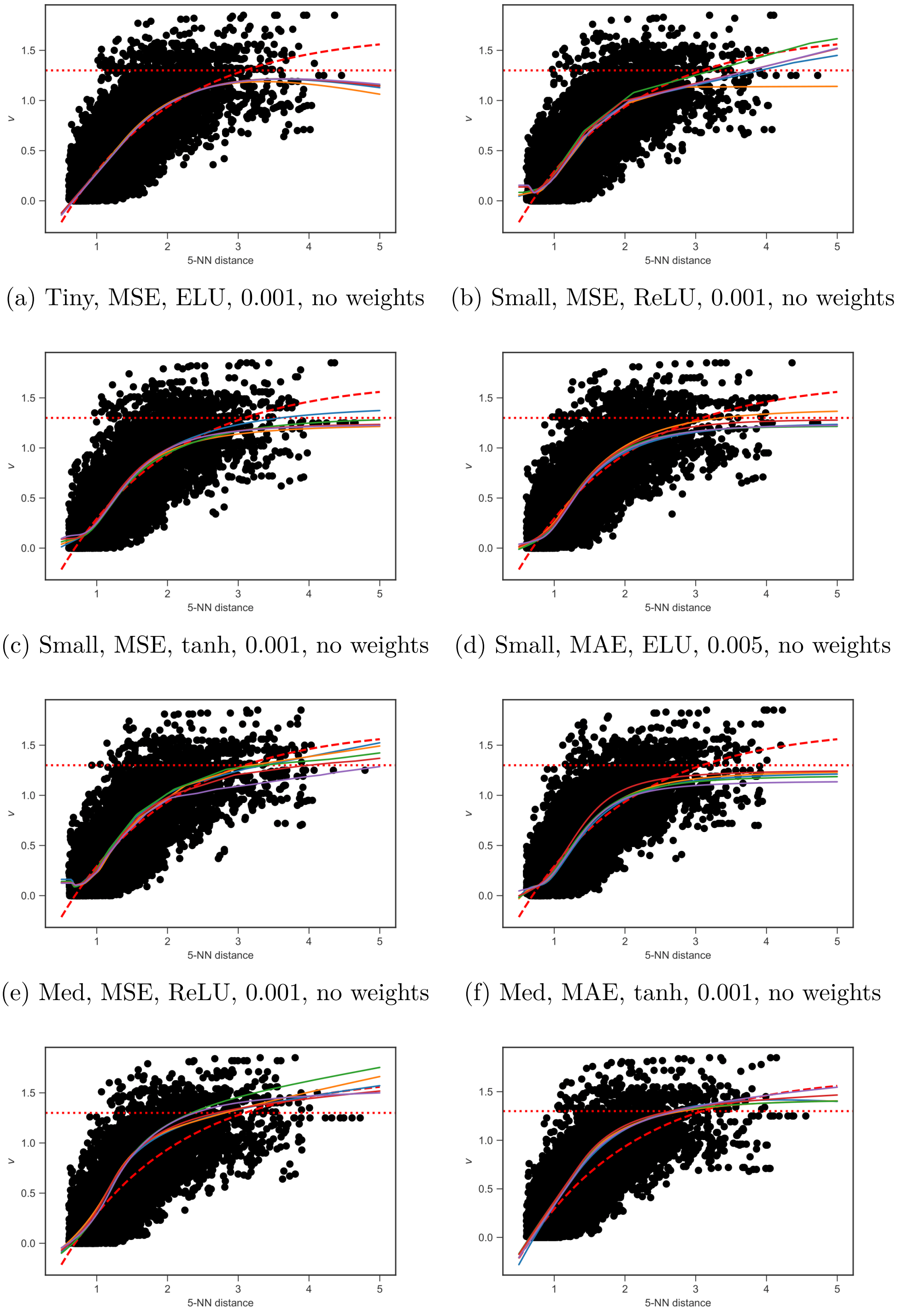
Further study of the experiment in [18] identified different "types" of pedestrians based on their strategies of navigating the crowd. This could potentially be identified by the extended neural networks from the previous section, and lead to further improvements over the FD models.

4.5 Results and Discussion

Using artificial data, several approximations of the fundamental diagram were obtained. Purely in terms of the prediction error, the improvement over the classical methods is not substantial, however, the qualitative shape of the resulting curves provides a more accurate description of the true underlying model. This can be illustrated by the fact that the Weidmann model fitted by least squares visibly overestimates the value of the free flow speed.

An immediate disadvantage of the ANN-based models is that the parameters of the fitted model are not interpretable. Classical models given by analytical equations have parameters with clear meanings and using the fundamental equation $J = \rho v$, it is easy to convert between different representations of the FD. With the neural network approach, this can only be done numerically and the results may get very inaccurate as division by small numbers takes place. It remains to be seen how this approach adapts to more complex geometries or different pedestrian compositions.

The introduction of additional features causes a drastic improvement in the prediction accuracy. Since the input trajectories can be considered time series data, specialized neural network architectures such as LSTMs may be employed to further improve the precision of the model.



(a) Tiny, MSE, ELU, 0.001, no weights

(b) Small, MSE, ReLU, 0.001, no weights

(c) Small, MSE, tanh, 0.001, no weights

(d) Small, MAE, ELU, 0.005, no weights

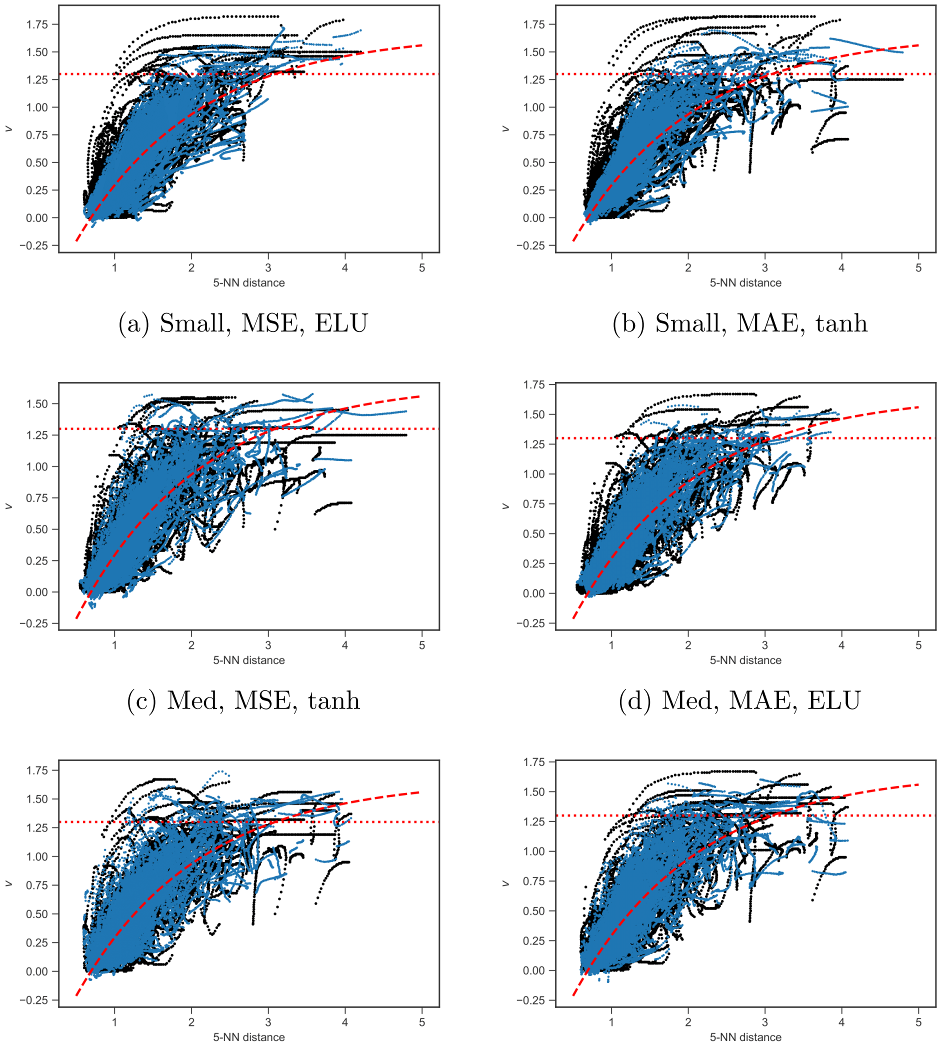
(e) Med, MSE, ReLU, 0.001, no weights

(f) Med, MAE, tanh, 0.001, no weights

(g) Tiny, MSE, ELU, 0.001, no weights

(h) Small, MSE, ELU, 0.005, weights

Figure 4.6: Typical resulting curves based on hyperparameters. Each plot contains five subsequent models trained on differently split datasets. The dashed red curve is the Weidmann model and the horizontal dashed red line is the desired free-flow velocity 1.3 m/s.



(a) Small, MSE, ELU

(b) Small, MAE, tanh

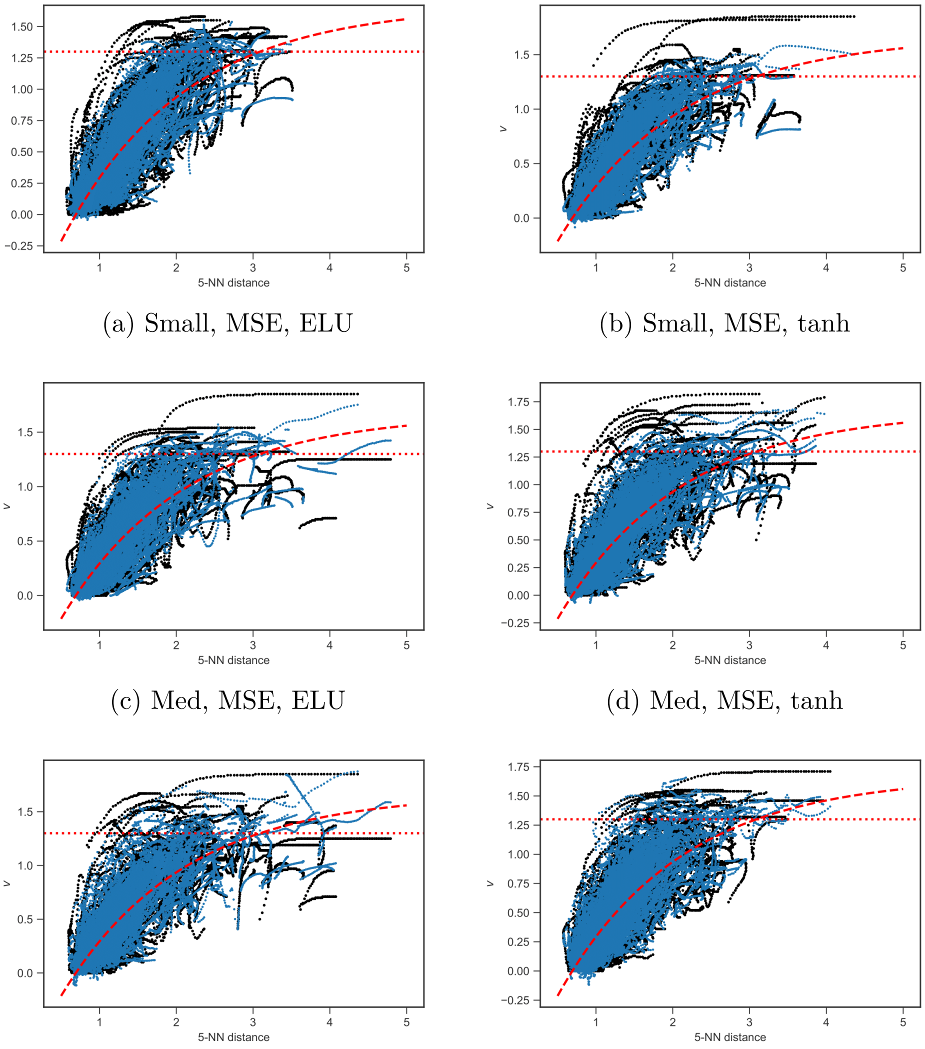
(c) Med, MSE, tanh

(d) Med, MAE, ELU

(e) Large, MSE, ELU

(f) Large, MAE, tanh

Figure 4.7: Samples of predictions by networks using relative positions of nearest neighbors in addition to k-NN distance. The dashed red curve is the Weidmann model and the horizontal dashed red line is the desired free-flow velocity 1.3 m/s.



(a) Small, MSE, ELU

(b) Small, MSE, tanh

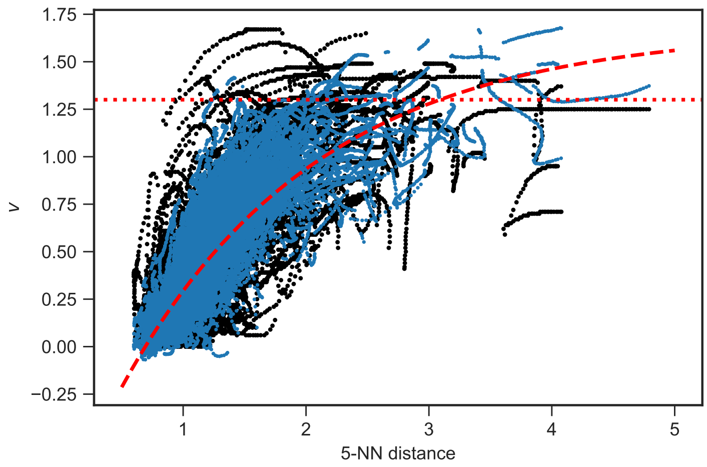
(c) Med, MSE, ELU

(d) Med, MSE, tanh

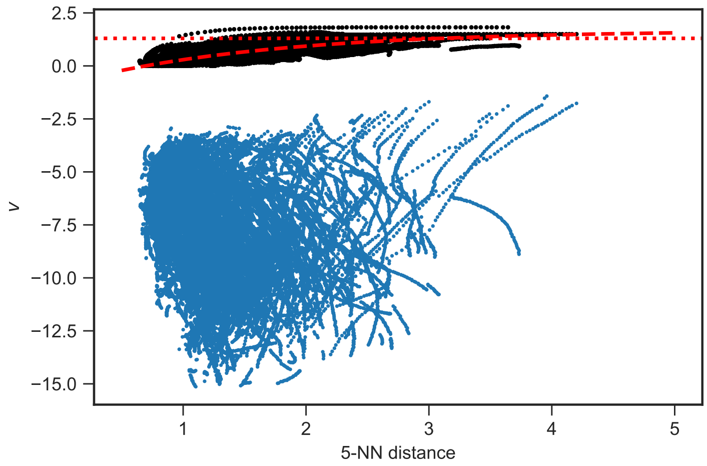
(e) Large, MSE, ELU

(f) Large, MSE, tanh

Figure 4.8: Samples of predictions by networks using relative positions of nearest neighbors as well as relative distance to exit. The dashed red curve is the Weidmann model and the horizontal dashed red line is the desired free-flow velocity 1.3 m/s.



(a) Network with relative positions only.



(b) Network that includes relative distance to the exit.

Figure 4.9: The predictions on a horizontally flipped dataset.

Conclusion

The aim of this thesis was to present a data-driven approach to estimating the relationship between the macroscopic variables of pedestrian flow. This included

- getting acquainted with the basics of the vast field of pedestrian dynamics and its parent field of traffic engineering,
- configuring a pedestrian simulation model and implementing a data pipeline that automates the simulation process,
- using a state-of-the-art machine learning framework to create functional neural network models,
- making practical predictions and interpreting the results, and
- identifying the weaknesses of this approach.

The practical part validated and expanded upon the work of researchers in [2] and proposed questions for further research. The assignment also included verifying the resulting models using real-world data. However, due to unforeseen circumstances, we were unable to obtain the experimental data. For this reason, the methodology was described on a theoretical level, with the practical validation being left an open topic for further work.

The classical methods of estimating the fundamental diagram have the clear disadvantage that they are confined to a predetermined set of curves and fail to consider the specific properties of each situations. The fundamental diagram approximations obtained via the use of neural networks adapt to the given geometry and the composition of the pedestrians automatically. However, various questions remain unanswered, laying the ground for further work. These include the generalization to different geometries, adding further complexity to the explanatory variables, or validating the models using real experimental data.

Bibliography

1. BUKÁČEK, M.; HRABÁK, P.; KRBÁLEK, M. Experimental Study of Phase Transition in Pedestrian Flow. *Transportation Research Procedia*. 2014, vol. 2, pp. 105–113. ISSN 2352-1465. Available from DOI: `https://doi.org/10.1016/j.trpro.2014.09.014`. The Conference on Pedestrian and Evacuation Dynamics 2014 (PED 2014), 22-24 October 2014, Delft, The Netherlands.
2. TORDEUX, Antoine et al. Prediction of Pedestrian Speed with Artificial Neural Networks. In: 2019, pp. 327–335. ISBN 978-3-030-11439-8. Available from DOI: `10.1007/978-3-030-11440-4_36`.
3. SCHADSCHNEIDER, Andreas; CHOWDHURY, Debashish; NISHINARI, Katsuhiro. Chapter Eleven - Pedestrian Dynamics. In: SCHADSCHNEIDER, Andreas; CHOWDHURY, Debashish; NISHINARI, Katsuhiro (eds.). *Stochastic Transport in Complex Systems*. Amsterdam: Elsevier, 2011, pp. 407–460. ISBN 978-0-444-52853-7. Available from DOI: `https://doi.org/10.1016/B978-0-444-52853-7.00011-7`.
4. HOOGENDOORN, Serge Paul; KNOOP, Victor. *Traffic Flow Theory and Simulation*. Delft University of Technology, 2016. Available also from: `https://ocw.tudelft.nl/courses/traffic-flow-theory-simulation/`.
5. SEYFRIED, Armin et al. The Fundamental Diagram of Pedestrian Movement Revisited. *Journal of Statistical Mechanics: Theory and Experiment*. 2005, vol. 10. Available from DOI: `10.1088/1742-5468/2005/10/P10002`.
6. STEFFEN, B.; SEYFRIED, A. Methods for measuring pedestrian density, flow, speed and direction with minimal scatter. *Physica A: Statistical Mechanics and its Applications*. 2010, vol. 389, no. 9, pp. 1902–1910. ISSN 0378-4371. Available from DOI: `https://doi.org/10.1016/j.physa.2009.12.015`.

7. VORONOI, Georges. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites. *Journal für die reine und angewandte Mathematik (Crelles Journal).* 1908, vol. 1908, no. 133. Available from DOI: `doi:10.1515/crll.1908.133.97`.

8. DOBRIN, Adam. A review of properties and variations of Voronoi diagrams. 2005.

9. MATHEW, Tom. *Fundamental Relations of Traffic Flow.* Indian Institute of Technology Bombay, 2019. Available also from: `https://www.civil.iitb.ac.in/tvm/nptel/512_FundRel/web/web.html`.

10. GREENSHIELDS, Bruce Douglas. The Photographic Method of studying Traffic Behaviour. In: *Proceedings of the 13th Annual Meeting of the Highway Research Board.* Highway Research Board, 1934, pp. 382–399.

11. KUHNE, Reinhart D. Greenshields' Legacy – Highway Traffic. In: *Foundations of Traffic Flow Theory I.* 2008.

12. MAERIVOET, Sven; DE MOOR, Bart. Traffic Flow Theory. *Physics.* 2005, vol. 1.

13. VANUMU, Lakhsmi; RAO, K Ramachandra; TIWARI, Geetam. Fundamental diagrams of pedestrian flow characteristics: A review. *European Transport Research Review.* 2017, vol. 9. Available from DOI: `10.1007/s12544-017-0264-6`.

14. SCHADSCHNEIDER, Andreas et al. Evacuation Dynamics: Empirical Results, Modeling and Applications. In: 2009, pp. 3142–3176. ISBN 978-0-387-75888-6. Available from DOI: `10.1007/978-0-387-30440-3_187`.

15. HELBING, Dirk; MOLNÁR, Péter. Social force model for pedestrian dynamics. *Phys. Rev. E.* 1995, vol. 51, pp. 4282–4286. Available from DOI: `10.1103/PhysRevE.51.4282`.

16. CHRAIBI, Mohcine; SEYFRIED, Armin; SCHADSCHNEIDER, Andreas. Generalized centrifugal-force model for pedestrian dynamics. *Phys. Rev. E.* 2010, vol. 82, p. 046111. Available from DOI: `10.1103/PhysRevE.82.046111`.

17. YU, W. J. et al. Centrifugal force model for pedestrian dynamics. *Phys. Rev. E.* 2005, vol. 72, p. 026112. Available from DOI: `10.1103/PhysRevE.72.026112`.

18. BUKÁČEK, M.; HRABÁK, P.; KRBÁLEK, M. Microscopic travel-time analysis of bottleneck experiments. *Transportmetrica A: Transport Science.* 2018, vol. 14, no. 5-6, pp. 375–391. Available from DOI: `10.1080/23249935.2017.1419423`.

19. *JuPedSim - Open source framework for simulating, analyzing and visualizing pedestrian dynamics.* 2021. Available also from: `https://www.jupedsim.org/`.

20. MANEEWONGVATANA, Songrit; MOUNT, David M. It's Okay to Be Skinny, If Your Friends Are Fat. In: *Center for Geometric Computing 4th Annual Workshop on Computational Geometry.* 1999.

21. WEIDMANN, Ulrich. *Transporttechnik der Fussgänger. Transporttechnische Eigenschaften des Fussgängerverkehrs, Literaturauswertung.* Zürich: Institut für Verkehrsplanung, Transporttechnik, Strassen- und Eisenbahnbau (IVT), ETH Zürich, 1992-01. Tech. rep. Available from DOI: `10.3929/ethz-a-000687810`.

22. GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

23. DAS, Pritikana; PARIDA, Manoranjan; KATIYAR, V. Analysis of interrelationship between pedestrian flow parameters using artificial neural network. *Journal of Modern Transportation.* 2015, vol. 35. Available from DOI: `10.1007/s40534-015-0088-9`.

24. MA, Yi; LEE, Eric Wai Ming; YUEN, Richard Kwok Kit. An Artificial Intelligence-Based Approach for Simulating Pedestrian Movement. *IEEE Transactions on Intelligent Transportation Systems.* 2016, vol. 17, no. 11, pp. 3159–3170. Available from DOI: `10.1109/TITS.2016.2542843`.

25. ALAHI, Alexandre et al. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016, pp. 961–971. Available from DOI: `10.1109/CVPR.2016.110`.

26. SANCHEZ-GONZALEZ, Alvaro et al. *Learning to Simulate Complex Physics with Graph Networks.* 2020. Available from arXiv: `2002.09405 [cs.LG]`.

27. VAŠATA, Daniel; MALDONADO, Pablo; KLOUDA, Karel. *Přednáška 11: Neuronové sítě.* 2019. Available also from: `https://courses.fit.cvut.cz/BI-VZD/@B191/lectures/files/BI-VZD-11-cs-handout.pdf`.

28. RUDER, Sebastian. *An overview of gradient descent optimization algorithms.* 2017. Available from arXiv: `1609.04747 [cs.LG]`.

29. KINGMA, Diederik; BA, Jimmy. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations.* 2014.

Appendix A

Acronyms

ANN artificial neural network

BI-VZD “Vytěžování znalostí z dat” – FIT CTU data mining course

CA cellular automaton

CFM Centrifugal-force model

CTU Czech Technical University

ELU exponential linear unit

FD fundamental diagram

FIT Faculty of Information Technology

GCFM Generalized centrifugal-force model

k-NN k-nearest neighbors

LSTM Long short-term memory

MAE mean absolute error

MSE mean squared error

ReLU rectified linear unit

Appendix **B**

Contents of Enclosed USB

```
│ readme.txt ....................... the file with USB contents description
├── Python............................the directory containing Python code
│   ├── Notebooks .......................................IPython notebooks
│   └── Scripts................Scripts that automate the simulation process
├── JuPedSim.................................simulation configuration files
├── LaTeX ..................the directory of LaTeX source codes of the thesis
│   └── thesis.pdf ...............................the thesis in PDF format
└── Datasets ................................... the artificial datasets used
```