Physica D 427 (2021) 133016

Contents lists available at ScienceDirect

Physica D

journal homepage: www.elsevier.com/locate/physd

One-parametric bifurcation analysis of data-driven car-following models

Paul Petersik [a,c], Debabrata Panja [b,c], Henk A. Dijkstra [a,c,*]

[a] Institute for Marine and Atmospheric Research Utrecht, Department of Physics, Utrecht University, Utrecht, The Netherlands
[b] Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
[c] Centre for Complex Systems Studies, Utrecht University, Utrecht, The Netherlands

ARTICLE INFO

Article history:
Received 19 December 2019
Received in revised form 28 June 2021
Accepted 29 June 2021
Available online 28 August 2021

Keywords:
Machine learning
Traffic modelling
Bifurcation analysis

ABSTRACT

In this study, an equation-free method is used to perform bifurcation analyses of various artificial neural network (ANN) based car-following models. The ANN models were trained on Multiple Car Following (MCF) model output data (ANN-m) and field data (ANN-r). The ANN-m model could capture the behaviour of the MCF model in quite detail. A bifurcation analysis, using the circuit length L as parameter, for the ANN-m model leads to good results if the training data set from the MCF model is sufficiently diverse, namely that it incorporates data from a wide range of vehicle densities that encompass the stable free-flow and the stable jam-flow regimes. The ANN-r model is in general able to capture the feature of traffic jams when a car takes headway and velocity of itself and of the two cars ahead as input. However, the traffic flow of the ANN-r model is more regular in comparison to the field data. It is possible to construct a partial bifurcation diagram in L for the ANN-r using the equation-free method and it is found that the flow changes stability due to a subcritical Hopf bifurcation.

1. Introduction

Many-particle systems of simple microscopic components can be found in numerous research areas. Their dynamics can have astonishing and unexpected features on a macroscopic scale. For instance, thousands of male fireflies in south-east Asia synchronize their flashing to create an impressive natural spectacle [1], pedestrians tend to walk in lines although nobody tells them to do so [2] or grains can organize in imposing patterns when they are placed on a vibrating plate [3]. A less spectacular and rather unpopular phenomenon is the appearance of congestions in traffic flow. One can characterize traffic flow broadly into two states. In a free-flow state, vehicles keep their desired velocity and perturbations to their dynamic state (position, velocity, acceleration) die out over time. In the jam-flow state those disturbances can grow in time and eventually the macroscopic feature of a congestion wave arises that can travel through the flow. A state when crashes occur in the traffic flow is sometimes considered as a third state.

A huge variety of models has been proposed to study the physics of traffic flow. On the one hand, macroscopic models simulate traffic flow based on macroscopic variables that have a low level of detail, such as vehicle density and mean velocity. A famous example from the beginnings of traffic modelling is for instance the "Lighthill–Whitham model" [4]. On the other hand, microscopic models, also called follow-the-leader or car-following models, directly simulate the dynamics of single cars and thus have a high level of detail. In addition to these, gas-kinetic traffic models can be considered as mesoscopic models that apply probability density functions of car density to simulate traffic flow. Other approaches use cellular automata where the road is divided into discrete cells. A comprehensive overview of the different classes of traffic models is given in [2] and in [5].

A popular microscopic model is the optimal velocity model (OVM) and its extension, the so-called multiple-car following (MCF) model. In the OVM the following car adjusts its velocity based on the distance to the car in front of it, called the headway [6]. An astonishing feature of the OVM is the emergence of phantom congestions much akin to what is observed in reality [7]. Phantom congestions appear without the existence of a permanent obstacles such as roadworks. Although the OVM can reproduce the feature of phantom jams, it turns out that the OVM model, if calibrated to experimental data, leads to unrealistic high accelerations and declarations [8]. As a response to this, Helbing and Tilch [8] introduced the generalized force model (GFM) which takes the velocity difference between the following car and its leading car into account when the velocity difference is positive. However, in Jiang et al. [9] it is shown that a car-following model actually is more realistic when also a negative velocity difference is considered. Therefore, Jiang et al. [9] call their model

* Corresponding author at: Institute for Marine and Atmospheric Research Utrecht, Department of Physics, Utrecht University, Utrecht, The Netherlands.
E-mail address: h.a.dijkstra@uu.nl (H.A. Dijkstra).

https://doi.org/10.1016/j.physd.2021.133016

the full velocity difference model (FVDM). Moreover, they show that relaxing the velocity of the following car to the velocity of the leading car stabilizes the free flow. As another extension of the OVM, Lenz et al. [10] introduced the multi-anticipative car-following model. Instead of considering just the headway to one car ahead, the headways to multiple cars ahead are considered. They show that the reaction to multiple cars ahead leads to a more stable free flow. Peng and Sun [11] finally combined the ideas of multi-anticipative driving and the reaction to velocity differences for their multiple-car-following (MCF) model. Again, they find that the consideration of multiple cars ahead and the velocity differences stabilizes the free flow.

The true dynamics of traffic flow are still far more complex than any existing traffic model could describe. A huge variety of factors such as psychological characteristics of drivers, individual properties of the vehicles or the influence of road conditions make the task of finding the correct underlying rules of traffic flow a daunting task. However, the recent developments of machine learning (e.g. using artificial neural networks (ANNs)) have stimulated new approaches to extract the 'underlying' laws of traffic flow from observational data [12–14]. Chong et al. [15] propose an ANN that predicts longitudinal (parallel to the lane) as well as lateral (perpendicular to the lane) movements of cars. The ANN-model was trained on observational data from the Naturalistic Truck Driving Study (NTDS) and the Naturalistic Car Driving Study (NCDS) collected by the Virginia Tech Transportation Institute. Colombaroni and Fusco [16] propose an ANN with one hidden layer that takes the headway and velocity difference to the leading car as input to predict the acceleration. Furthermore, they propose an extended version that incorporates memory by also taking headway and velocity difference from previous time steps as input. They trained their models on data that was collected during experimental rides with two or four cars in line, each equipped with a GPS device for tracking. They find that their ANN models can approximate the driving behaviour of cars in an adequate way.

Tanaka [17] examines the performance of different ANNs. All models have two hidden layers each with 5 neurons and their task is to predict the acceleration of the following car at time $t+T$ from input data at time t. Here, T can be understood as a reaction time. The simplest model takes just the headway and the velocity difference to the leading car as input. The most complex model takes, in addition to the velocity and the current acceleration of the following car, also the acceleration of the leading car as input to predict the acceleration of the following car. Interestingly, Tanaka [17] finds that velocity difference and acceleration of the following car as well acceleration of the leading car at time t are the most important inputs to predict the acceleration of the following car at a later point in time $t+T$. This is a surprising result considering that most car-following models are built based on headway and velocity.

In the OVM, traffic jams occur by a loss of stability of the free-flow through a Hopf bifurcation that can be sub- or supercritical [18]. Already before the free-flow becomes unstable, stable solutions of the congested flow can exist and hence bi-stable regimes can be present. However, full bifurcation analyses of the OVM are often limited to a small number of cars [i.e. $N < 2$ in18–20] because the integration of a microscopic car-following model is computational expensive. Equation-free analysis developed by Kevrekidis et al. [21] offers a tool to compute macroscopic behaviour in many-particle systems, i.e. a bifurcation analysis. Equation-free methods extrapolate the macroscopic evolution of a system into the future by using small bursts of the microscopic model. Marschler et al. [22] refined the equation-free method by formulating it in an implicit manner. They perform a bifurcation analysis using the equation-free method for an OVM with $N = 60$. For this, they use a pseudo-arclength continuation scheme that makes it possible to also find the unstable solutions of the dynamical system.

Up until now, the bifurcation behaviour of ANN-based car-following models is unknown and we focus on this problem here. In Section 2.2, an ANN model is constructed with an architecture similar to those in [17] and [16]. A first ANN model is trained on output from an MCF model that is reformulated by means of mesoscopic variables, namely weighted vehicle density and weighted group velocity of the platoon in front of the following car. Next, another ANN model is trained on observational data from [7]. Bifurcation analyses are performed using the equation-free method for both ANN models using the implicit equation-free method from [22]. The methodology of this study is described in Section 2 and the results are presented in Section 3. We provide a summary and discussion of the results in Section 4.

2. Methods

To test the performance of the ANN model, we train it on data from a mesoscopic multiple car-following model which is shorty described in Section 2.1. The general construction of the ANN is discussed in Section 2.2 and the equation-free bifurcation analysis in Section 2.3.

2.1. The mesoscopic multiple-car (MCF) following model

The original OVM was modified, refined and advanced by several studies to incorporate velocity differences and multi-anticipative driving [i.e. 8,10,11,23]. The formulation of the MCF in [11], their equation (5), is

$$\begin{aligned}\ddot{x}_n = a(V(\Delta x_n, \Delta x_{n+1}, \ldots, \Delta x_{n+m-1}) - \dot{x}_n) \\ +\lambda G(\Delta\dot{x}_n, \Delta\dot{x}_{n+1}, \ldots, \Delta\dot{x}_{n+m-1})\end{aligned} \tag{1}$$

for $n = 1, 2, \ldots, N$, with N the number of cars in the model and $m \ll N$ the number of the cars in the leading platoon a car considers to adjust its acceleration. Furthermore, the velocity of the nth car is indicated by $\dot{x}_n$, its acceleration by $\ddot{x}_n$ and the headway to the leading car by Δx_n. The variables a and λ are sensitivity parameters. $V(\cdot)$ is the linear weighted optimal velocity function (OVF) and $G(\cdot)$ is a monotonically increasing function whose value can be understood as a weighted velocity difference to the considered leading cars.

In this paper, we use a version of the MCF model [11] which is reformulated using mesoscopic variables. These mesoscopic variables are the weighted vehicle density, ρ_n, and the weighted group velocity, J_n, ahead of the following car. The dynamics of the mesoscopic MCF are governed by the equations

$$\ddot{x}_n = a(\mathcal{V}(\rho_n) - \dot{x}_n) + \lambda(J_n - \dot{x}_n) \tag{2}$$

with $\mathcal{V}(\rho) = V(\rho^{-1})$. Here we use

$$V(\rho^{-1}) = v_0(\tanh(\rho^{-1} - h) + \tanh h) \tag{3}$$

as OVF with the velocity magnitude v_0 and the safety distance h. We calculate the mesoscopic variables by

$$\rho_n = \frac{m}{m\sum_{i=1}^{m} p_i \Delta x_{n+i-1}} = \frac{1}{\sum_{i=1}^{m} p_i \Delta x_{n+i-1}} \tag{4}$$

$$J_n = \sum_{i=1}^{m} p_i \dot{x}_{n+i} \tag{5}$$

where exponential weighting is applied, i.e.

$$p_i = \frac{e^w(m-i)}{\sum_{i=1}^{m} e^w(m-i)}. \tag{6}$$

and w is a weight parameter.

If not stated differently, $m = 10$ in this study. If $w > 0$, the effect is included that drivers perceive the traffic as more dense when headways of cars in their vicinity are small. By using a weighting function, it is possible to describe different kinds weighting by one parameter, namely w.

We can rescale the time such that the parameter a drops out of the model. Let t^* denote the old time coordinate and t the new one and it holds $t^* = \tau t$. Then, the mesoscopic MCF using the new time coordinate, with the choice $\tau = a^{-1}$, can be written as:

$$\ddot{x}_n = \mathcal{V}(\rho_n) - \dot{x}_n + \lambda'(J_n - \dot{x}_n) \tag{7}$$

with in total five parameters, namely, $\lambda' = \frac{\lambda}{a}$, v_0, m, w and h. From here on, the dash sign for the rescaled sensitivity parameter λ' is dropped.

2.2. ANN based car-following models

ANNs consist out of an input layer, one or more hidden layer(s) and an output layer. Each layer consists of a certain number of neurons, also called nodes, that are connected by synapses to neurons in the previous layer and the subsequent layer [24]. Each synapse has a weight assigned to it. If each neuron is connected to all neurons in the previous and the subsequent layer, the neural network is called fully connected. In this study, all neural networks are fully connected. In the input layer there are as many neurons as there are input variables. The amount of neurons in the output layer is equal to the number of the variables that should be predicted. The number of neurons in the hidden layers is a matter of choice and must be at least one.

The output of a neuron from the input layer is the value of the associated input vector element. For all other layers the output of a single neuron is computed by calculating the weighted sum of its inputs using the synapse weights, adding a bias and applying an activation function to this value. In mathematical terms this reads as follows:

$$o_j = f\left(\sum_{i=0}^{n} w_{ij} o_i\right) \tag{8}$$

Here, o_j denotes the output of the considered neuron and o_i the outputs of the neurons from the preceding layer. Note, that for each layer one neuron is added such that the output is $o_0 = 1$. This neuron accounts for the added bias for the neurons in the following layer. The quantity w_{ij} is the weight of the synapse from neuron i in the preceding layer to the considered neuron j and f is the activation function. There exist several activation functions such as the identity, the sigmoid function, the hyperbolic tangents or the rectified linear units (ReLU) that reads as follows:

$$f(x) = \max(0, x) \tag{9}$$

The resulting output value of a neuron is used as one of the inputs in each neuron in the subsequent layer. The output values are fed from layer to layer until the output layer is eventually reached. The output of the neurons in the output layer is the prediction of the ANN.

ANNs are trained by a method called back-propagation [25,26] to perform a regression or classification task [24]. For this, one needs at first to define a loss function. This could be for instance the mean squared error (MSE) or the mean absolute error (MAE). Then, the ANN can be trained on a training data set that consists of features and labels. One vector of input variables is called a feature. Labels are the desired target value(s) that the ANN should predict. After the input is fed forward through the ANN, one can calculate its prediction error by using the chosen loss function. The errors can be back-propagated through the network to adjust the weights of the neural network. For each synapse weight one can calculate the gradient of loss by applying the chain rule of derivations. Setting a learning rate one can calculate the adjustments of the weights with this gradient.

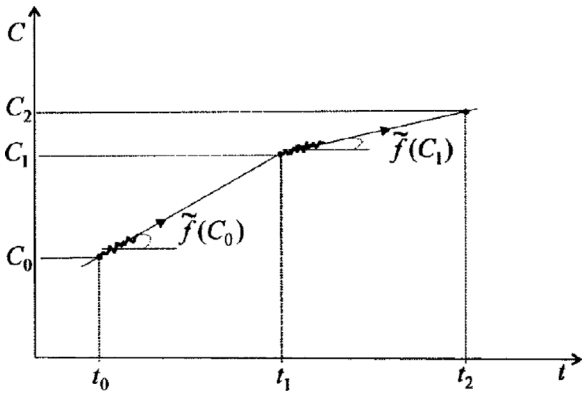


Fig. 1. Sketch to visualize a macroscopic time stepper that uses the equation-free method [from 21]. The microscopic model is initialized at t_0 with the macroscopic state C_0. A small integration of the microscopic model starting at t_0 is used to extrapolate the macroscopic development till t_1. At t_1 the microscopic model is again initialized now with the macroscopic state C_1 to repeat the described extrapolation to estimate the macroscopic state at t_2.

2.3. Equation-free bifurcation analysis

Many-particle systems that are described by microscopic evolution equations often exhibit coherent behaviour on the macroscopic scale. Sometimes it is even possible to derive macroscopic evolution equations for this system i.e. the Navier–Stokes equation in fluid dynamics can be derived from kinetic theory. However, for many other systems no macroscopic evolution equations are known. The equation-free method offers a tool to approximate the macroscopic behaviour by using small bursts of the microscopic model (cf. Fig. 1).

To apply the equation-free method, it is necessary that the microscopic system is a so-called slow-fast system. On a long time scale, the evolution of this system is determined by the evolution of the slow system which can be described by a limited number of state variables. Moreover, any perturbation away from the slow flow quickly converges back to the slow flow due to the fast dynamics of the system. A lifting operator, $\mathcal{L}$, and a restriction operator, $\mathcal{R}$, have to be defined for the equation-free method. The lifting operator maps a macroscopic state to a microscopic state. The restriction operator does the opposite. Note that all these operators depend on parameters, which we will later make explicit in the bifurcation analyses. For the microscopic traffic model, we choose as macroscopic state variable the standard deviation of the headways, indicated by $\sigma \in \mathbb{R}$. The microscopic state is described by $u = (x, \dot{x}) \in \mathbb{R}^{2N}$. The restriction operator is now defined as

$$\mathcal{R}(\Delta x) = \sigma = \sqrt{\sum_{n=1}^{N} (\Delta x_n - \langle \Delta x \rangle)^2} \tag{10}$$

where $\langle \cdot \rangle$ denotes the mean value. The lifting operator is defined as:

$$\mathcal{L}_{\tilde{u}}(\sigma) = \quad (x_{\text{new}}, \dot{x}_{\text{new}}) \in \mathbb{R}^N \times \mathbb{R}^N, \quad \text{where} \tag{11a}$$

$$\Delta x_{\text{new},n} = \quad \frac{\sigma}{\tilde{\sigma}} \left(\Delta \tilde{x}_n - \langle \Delta \tilde{x} \rangle\right) + \langle \Delta \tilde{x} \rangle, \ \text{for } n = 1, \ldots, N \tag{11b}$$

$$x_{\text{new},1} = 0,\ x_{\text{new},n} = \sum_{i=1}^{n-1} \Delta x_{\text{new},i} \quad \text{for } n = 2, \ldots, N \tag{11c}$$

$$\dot{x}_{\text{new},n} = V(\Delta x_{\text{new},n}) \text{ for } n = 1, \ldots, N \tag{11d}$$

Here, tildes relate to the reference state, $\tilde{u}$, that is computed beforehand using the microscopic model. The reference state will be further discussed in the next subsection.

When the microscopic evolution is denoted as $M(t, u)$, the macroscopic evolution, indicated by $\Phi(t, \sigma)$, is defined by the three steps: Lift, evolve and restrict. In mathematical terms this reads as follows:

$$\Phi(t, \sigma) = \mathcal{R}(M(t, \mathcal{L}(\sigma))) \tag{12}$$

This equation can be solved explicitly. For this, it is important that δ is larger than the healing time, t_{skip}, after which one can assume that the dynamics of microscopic model converged towards the slow flow.

However, as explained in section 1.1. of Marschler et al. [22], the explicit equation-free method can lead to incorrect results. They therefore introduced an implicit method of the equation-free analysis. The macroscopic time stepper is implicitly defined by solving

$$\mathcal{R}(M(t_{\text{skip}} + \delta, \mathcal{L}(\sigma))) = \mathcal{R}(M(t_{\text{skip}}, \mathcal{L}(y))) \tag{13}$$

for y. Now, the macroscopic time stepper for an integration over the time interval δ is given by $\Phi(\delta, \sigma) := y$. Hence, the explicit time stepper is defined in the image of $\mathcal{R}(M(t, \mathcal{L}(x)))$, whereas the implicit method finds its solution in the domain of $\mathcal{R}(M(t, \mathcal{L}(x)))$.

To perform a bifurcation analysis for a general mapping H, one has to follow solutions of the equations

$$H(\sigma, q) = 0, \tag{14}$$

where q indicates the parameter in which the bifurcation analysis is done. In this study, we will apply the method of pseudo-arclength continuation using a predictor–corrector scheme. In the predictor step, a first guess of a new fixed point is made by using the secant between two known fixed points:

$$\hat{w} = (\hat{w}^{(\sigma)}, \hat{w}^{(q)}) = (\sigma^1 - \sigma^0, q^1 - q^0). \tag{15}$$

The prediction of the next fixed point is then

$$(\hat{\sigma}, \hat{q}) = (\sigma^1, q^1) + s\frac{\hat{w}}{|\hat{w}|}, \tag{16}$$

where $|\cdot|$ is the euclidean norm. In the corrector step, one finds the fixed point of the system that is located on a line that goes through $(\hat{\sigma}, \hat{q})$ and is perpendicular to the secant from the predictor step:

$$H(\sigma, q) = 0 \tag{17}$$

$$\hat{w} \cdot (\sigma - \hat{\sigma}, q - \hat{q}) = 0 \tag{18}$$

To solve this equation system one can apply Newtons method:

$$(\sigma^{k+1}, q^{k+1})^T = (\sigma^k, q^k)^T + J^{-1}(H(\sigma, q), 0)^T \tag{19}$$

where J is the Jacobian:

$$J = \begin{pmatrix} H_\sigma & H_q \\ w^{(\sigma)} & w^{(q)} \end{pmatrix} \tag{20}$$

The derivatives H_σ and H_q can be approximated by

$$H_\sigma = \frac{H(\sigma + \Delta\sigma, q) - H(\sigma, q)}{\Delta\sigma} \tag{21}$$

$$H_q = \frac{H(\sigma, q + \Delta q) - H(\sigma, q)}{\Delta q}. \tag{22}$$

Here, we use the implicit method from Marschler et al. [22], with $H = F^\delta$ given by the time derivative of the macroscopic flow that can be approximated by finite differences as:

$$F^\delta(\sigma) = \frac{\Phi(\delta, \sigma) - \sigma}{\delta} \tag{23}$$

The first term in the numerator is found by the implicit equation-free method. This means that, for each corrector step, three microscopic model runs have to be performed, namely for (σ, q), $(\sigma + \Delta\sigma, q)$ and $(\sigma, q + \Delta q)$. We choose $\Delta\sigma = \Delta q = 10^{-3}$ since this leads approximately to the fastest convergence in Newtons method (estimated by comparing different values of $\Delta\sigma$ and Δq). In this study, we assume that the Newton method converged towards a fixed point when $|\sigma^{k+1} - \sigma^k| < 0.01$.

For the bifurcation analyses in one parameter, for which we take either the velocity magnitude $q = v_0$ or the circuit length $q = L$, the pseudo-arclength continuation scheme in association with the equation-free method is used. The bifurcation analysis is always done for the single jam solution. First, two initial fixed points are computed using long simulations of the microscopic model. For this, the model is initialized with perturbations to the positions of the cars that lead to a single-jam solution:

$$x_{n,\text{init}} = x_n + \mu \sin\left(\frac{2\pi n}{N}\right) \tag{24}$$

where μ is a constant. Furthermore, from the second initial fixed point the corresponding microscopic state is taken as reference state, $\tilde{u}$, for the lifting operator. By applying the pseudo-arclength continuation scheme as described further up, new fixed points can be found.

The stability of the fixed point is inferred from the last iteration in the corrector step by the sign of the derivative F^δ_σ at the fixed point, with $F^\delta_\sigma(\sigma, q) < 0$ indicating a stable fixed point and $F^\delta_\sigma(\sigma, q) > 0$ an unstable one, After five fixed points are found, a new reference state is computed, because the initial reference state might have a car distribution (shape of the traffic jam) that is far from the true macroscopic state. If the reference state is not updated, it could lead to the development of multiple-jam solutions.

3. Results

Below, we first present the bifurcation diagrams for the MCF (Section 3.1) which serves as a test case for the (MCF model based) ANN-m model bifurcation study in Section 3.2. Finally, the results for the ANN-r model are presented in section 3.3.

3.1. Bifurcations in the MCF model

The mesoscopic MCF model is integrated using the Runge–Kutta-4 (RK4) and an Euler forward method. With the RK4, the values of the velocities of the cars, $\dot{x}_n$, at the new time step are found. Then, an Euler forward step using the new velocities is applied to calculate the new positions of the cars. The analysis is restricted to $N = 60$ and $L = 60$. For this set-up it is known from Marschler et al. [22] for the OVM ($m = 1$, $\lambda = 0$) that bistable regimes solely exist if $h \gtrsim 1.1$. Since we want to keep this feature and see how it is altered by the mesoscopic MCF model, we fix $h = 1.4$.

For the bifurcation analyses of the mesoscopic MCF model, the numerical set-up in Table 1 is used. The bifurcation diagram of the mesoscopic MCF model versus v_0 with $\lambda = 0.04$ and $w = 1$ is depicted in Fig. 2. The free flow changes stability at a subcritical pitchfork bifurcation at $(v_0, \sigma) \approx (1.36, 0)$ from stable to unstable for increasing values of v_0. This indicates that higher velocities destabilize the free flow. Note that the negative branch of the

Table 1
Standard values of the parameters in the pseudo-arclength continuation scheme for the bifurcation analysis of the different models.

Model:	MCF	ANN-m	ANN-r
	parameter values	parameter values	parameter values
	$s = 0.01$	$s = 0.01$	$s = 0.02 - 2$
	$t_{\text{skip}} = 300$	$t_{\text{skip}} = 10$	$t_{\text{skip}} = 2$
	$\delta = 2000$	$\delta = 100$	$\delta = 200$
	$dt = 0.05$	$dt = 0.05$	$dt = 0.1$
	$\mu = 5$	$\mu = 0.5$	$\mu = 5$

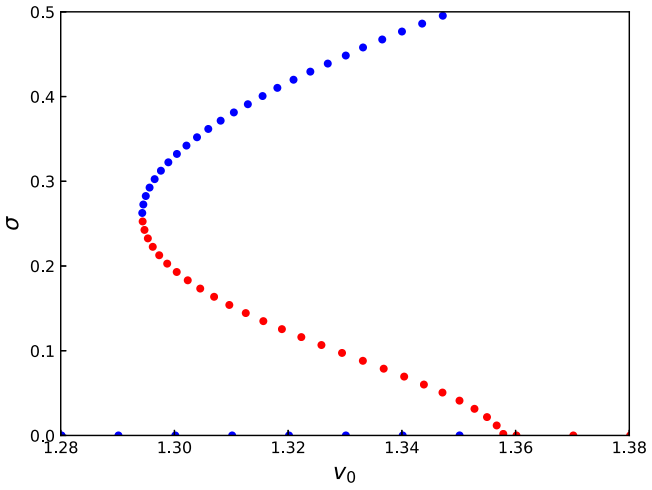


Fig. 2. Bifurcation diagram in parameter v_0 for $\lambda = 0.04$ and $w = 1$. Blue circles are stable fixed points, red circles unstable fixed points. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

pitchfork is not present because by definition $\sigma \geq 0$. Since in the microscopic model this change of stability occurs due to a subcritical Hopf bifurcation that gives rise to a limit cycle in the headway-velocity space this bifurcation point is called a Hopf point. Furthermore, note that a fixed point in the macroscopic model corresponds to a limit cycle in the headway-velocity space of the microscopic model. Furthermore, the jam flow changes stability at a fold point at $(v_0, \sigma) \approx (1.29, 0.25)$, which is a cyclic fold in the microscopic model, but referred to below as a fold point. Hence, a bi-stable regime is present in the microscopic system between the Hopf and the fold. In this bi-stable regime, the favoured flow, jam or free flow, depends on the initial condition.

For increasing values of λ the bifurcation diagram (Fig. 3a) is shifted to higher values of v_0. This means, stronger relaxation of the cars velocity towards the group velocity of the platoon in front of the car stabilizes the free-flow. Fig. 3b summarizes this in a regime diagram. It shows that increasing values of λ linearly shift the bifurcation diagram towards higher values of v_0. In addition, the bi-stable region does not significantly change its size. When w is increased the regimes are shifted towards lower values of v_0 and would eventually converge towards the solution for $m = 1$. Interestingly, the slope of the Hopf and the fold point curve becomes flatter with decreasing w. This means that the stabilization effect of λ on the free flow becomes stronger when more cars ahead are considered.

In Fig. 4a the bifurcation diagram in parameter v_0 for various values of w is depicted. Higher values of w imply that the weighting in the MCF model is in favour of cars that are close to the following car. Hence, for a very high value of w the following car basically just considers the car in front of it. One can see that decreasing values of w shift the bifurcation diagrams towards higher values of v_0. Therefore, considering more cars ahead stabilizes the free-flow. The corresponding regime diagram in Fig. 4b shows that decreasing values of w shift the regimes towards higher values of v_0. Interestingly, the subcritical Hopf point turns into a supercritical Hopf point at $w \approx 0.3$ for decreasing w. Hence, the fold disappears and no unstable limit cycle is found. This hints at the existence of a (co-dimension 2) Bautin bifurcation (see e.g. section 8.3 of [27]) at small values of w, which are estimated for the different cases by the crosses in Fig. 4b. For very high values of w an asymptote appears. Here, the regimes approach the solution if just one car ahead would have been considered. Therefore, for $\lambda = 0.0$ the regimes converge towards the solution for a normal OVM ($m = 1$, $\lambda = 0$) as indicated by the vertical black lines.

3.2. Bifurcation analysis of the ANN-m model

To investigate how an ANN captures the dynamics of the MCF model, an ANN with one hidden layer is trained on data from the MCF model with $N = 10$, $\lambda = 0.1$, $m = 3$ and $w = 2.0$ is considered. Because the ANN model is trained on model data output, it will be referred to as the ANN-m model. The task of the ANN is to predict the acceleration of the following car from headway and velocity data from all cars. Hence, the ANN has 20 neurons in the input layer and one neuron in the output layer. Headway and velocity are chosen as input because they are the variables used in the MCF. The input data is normalized by subtracting the mean and dividing this value by the standard deviation. Furthermore, the hidden layer is constructed with 20 neurons. As activation function the model uses a ReLU in the hidden layer and the identity in the output layer. The advantage of using a ReLU over a sigmoid or tangent hyperbolic as activation function is that the ReLU does not have the problem of vanishing gradients that can lead to a very slow convergence during training. Using the identity in the output layer is a necessary constrained to perform a regression since the ANN should be able to predict any real value of acceleration.

For the training data set, the MCF model is run multiple times for $L \in \{9, 10, 10.5, 11, 12, 14, 16, 17, 17.5, 18, 19\}$ with $t_{\text{max}} = 10^4$ and $dt = 0.1$. This gives $11 \cdot 10^5$ time steps in total. Furthermore, noise is applied to the velocity when it is used as input in the MCF. To do so, a random number from a uniform distribution between -0.2 and 0.2 is added at each time step to the velocity. Introducing noise makes the training data set more diverse and the ANN-model more robust. As optimizer in the back-propagation process, the Adam algorithm is used to update the weights [28] based on the MSE. Before the training can start, 20% of the training data set is removed from it to use it as validation data set. On the validation data set, the ANN is not trained but its performance is measured using the MSE. Furthermore, the ANN is trained on so-called batches of data that are subsets of the training data set. All features from a batch are propagated through the ANN without updating the weights. However, the corresponding loss gradients are saved. After a batch is fed through the network, the weights get updated based on the average gradients. When all batches are propagated through the network, a so-called epoch has passed. For the training, an adaptive learning rate is chosen [24]. This means, if after two subsequent epochs the MSE of the performance of the ANN on the validation data set did not decrease, the learning rate is divided by 5. If it does not decrease for another two epochs the training is stopped. After each epoch the training data set is shuffled. This method helps to avoid local minima in the trainings process.

Two ANN-m models are compared. One model (ANN-m-1) was trained on the full range of the trainings data set and another (ANN-m-2) was only trained on data coming from simulations with $L \in \{11, 12, 14, 16, 17\}$. Due to this choice, data around the bifurcation points are intentionally excluded from the training data

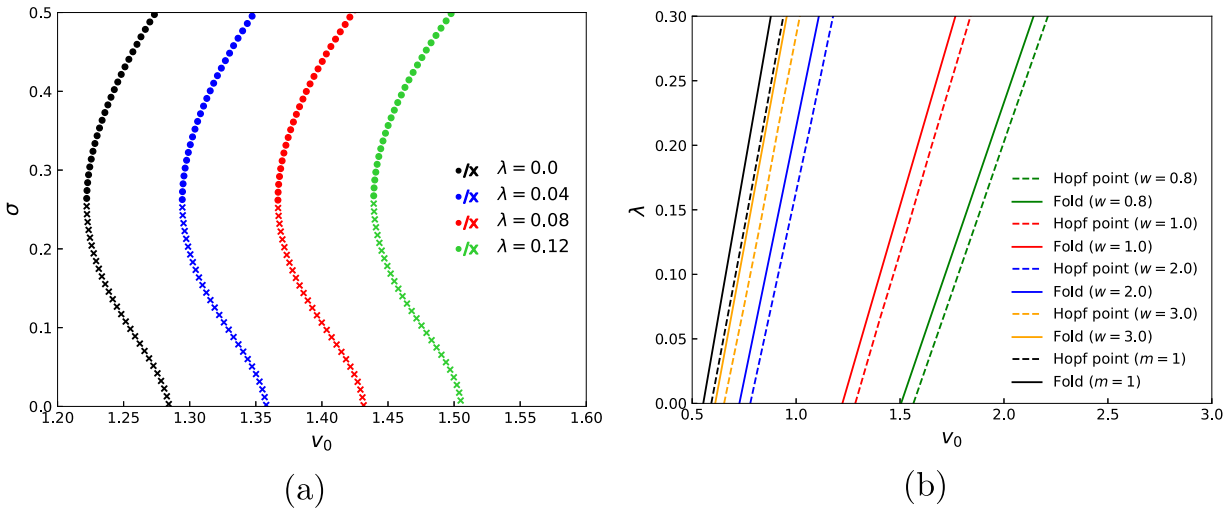


Fig. 3. (a) Bifurcation diagram in parameter v_0 for $\lambda \in \{0, 0.04, 0.08, 0.12\}$ with $w = 1$. Stable fixed points are indicated by circles and unstable fixed points by crosses. (b) Regime diagram in the parameter space (v_0, λ) for $w \in \{0.8, 1, 2, 3\}$. The positions of the Hopf points are indicated by dashed and the fold points by solid lines. In addition, the positions of the fold and Hopf points for a MCF that just considers one car ahead ($m = 1$) are depicted in black.

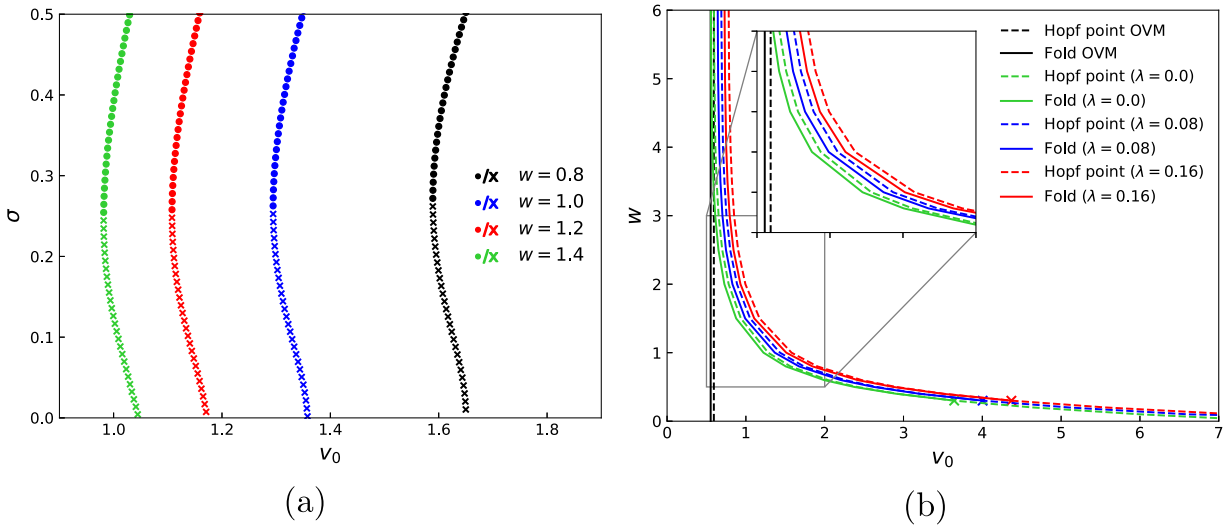


Fig. 4. (a) Bifurcation diagram in parameter v_0 for various values of $w \in \{0.8, 1, 1.2, 1.4\}$ with $\lambda = 0.04$. Stable fixed points are indicated by circles and unstable fixed points by crosses. (b) Regime diagram in the parameter space (v_0, w) for $\lambda \in \{0, 0.08, 0.16\}$. The positions of the Hopf points are indicated by dashed and the fold points by solid lines. In addition, the positions of the fold and Hopf points for a simple OVM ($\lambda = 0.0, m = 1$) are depicted in black as straight vertical lines. Crosses indicate the estimated positions of the possible Bautin bifurcation points.

set. The ANN models were built and trained using the scikit-learn python package (https://scikit-learn.org/stable/). The weights of synapses between the input and hidden layer for the ANN-m-1 model (after convergence was reached) are depicted in Fig. 5a. One can clearly see that the absolute values of weights from synapses connecting neurons number 0 and 10 of the input layer with neurons of the hidden layer are enhanced. These neurons correspond to the headway of the following car and its velocity. In addition, one can see a slight enhancement of the weights for input neurons 1, 2, 8 and 9, corresponding to the two leading cars and the two cars behind the following car. In Fig. 5b the mean value of normalized absolute weights are depicted. The mean was taken over 10 ANN-m-1 models that where trained with a different random initialization of the weights at the beginning of the learning process. Again, a clear picture arises that input values from nodes 0, 1, 2, 8, 9 and 10 are enhanced. This underlines the robustness of the described feature.

The enhanced weights of nodes 0, 1, 2 and 10 show the relationship of acceleration with headways and velocities that was prescribed to the MCF that generated the training data set. Here, a box of three cars ahead ($m = 3$) influenced the acceleration of the following car with a rather weak influence of group velocity ($\lambda = 0.1$) and a strong weighting of nearby leading cars ($w = 2.0$). However, also a weak feature of the ANN-m-1 arises, namely the enhanced weights of nodes 8 and 9 that are the headways of the two cars behind the following car. This is most likely due to the influence of the following car on the car behind it. Hence, the ANN finds correlations in the data, but this is a case of "correlation does not imply causation", because from the formulation of the MCF it is known that cars in the back do not influence the acceleration of following car. These findings motivated a more restricted choice for the input data in case the ANN-r model.

Although these shortcomings arise, the ANN-m-1 model is still able to reproduce the dynamics of the MCF model in an appropriate manner. In Fig. 6 the development of limit cycles for car number 0 is depicted for different circuit length, L, using the MCF and the ANN-m-1 model ($N = 10$). The models are initialized with a sinusoidal disturbance in the position (with

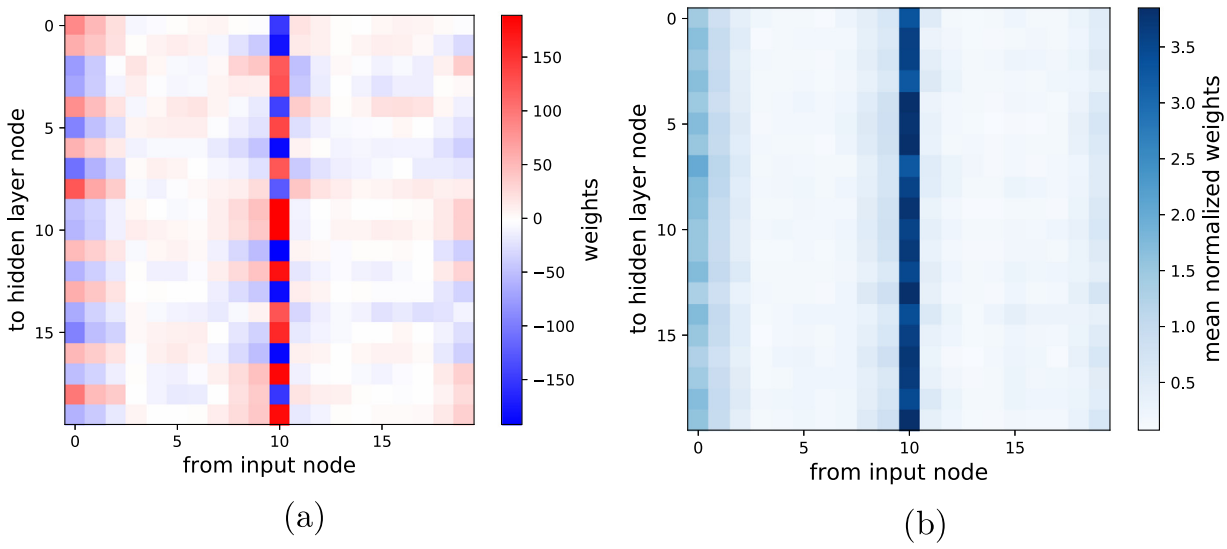


Fig. 5. (a) Weights of synapses from the input layer to the hidden layer for the first random state. (b) Mean over 10 ANN-m-1 models of the normalized weights from the synapses from the input layer to the hidden layer.

$\mu = 1$ in Eq. (24)). Differences between the ANN-m-1 and the MCF model are very small. In general, the amplitudes of the limit cycles of the ANN-m-1 model are slightly smaller.

In contrast to the MCF model that had various parameters for which one can perform a bifurcation analysis, there are just two parameters in the constructed ANN models, namely the circuit length L and the number of cars N. However, note that both variables relate to the mean density $\overline{\rho} = N/L$. We choose to make the bifurcation analysis in L because it is a continuous parameter in contrast to N that is discrete. The equation-free method from [22] again offers a suitable tool to perform the bifurcation analysis. In comparison to the bifurcation analysis in Section 2, the microscopic evolution, $\mathcal{M}$, in Eq. (13) is done by the ANN model instead of the MCF model.

For a bifurcation analysis in the parameter L for the MCF and the ANN-m models, a new lifting operator has to be defined, such that the length of the circuit can change. We therefore replace ((11b)) by

$$\Delta x_{\text{new},n} = \frac{\sigma}{\tilde{\sigma}}\left(\Delta\tilde{x}_n - \langle\Delta\tilde{x}\rangle\right) + \frac{L}{\tilde{L}}\langle\Delta\tilde{x}\rangle \quad \text{for } n = 1,\ldots,N \tag{25}$$

Furthermore, a new computational setting (see Table 1) of the bifurcation analysis is used to account for the different dynamics of MCF and the trained ANN models. The main difference to the setting in Section 2 is that t_{skip} and δ are smaller than in the previous case. This is because the model converges on a faster time scale towards a stable solution and large t_{skip} and δ would lead to small gradients.

The free-flow state of the MCF model (green in Fig. 7) loses stability at a supercritical Hopf point at $L \approx 10$ when L is increased and at $L \approx 18$ for decreasing L. The ANN-m-1 model (blue) can fairly well reproduce the bifurcation diagram of the MCF model with the Hopf points close to those of the MCF model. The standard deviations σ of the fixed points are slightly smaller for the ANN-m-1 model than for the MCF model. This corresponds to the smaller limit cycle seen before (Fig. 6). Furthermore, two stable free-flow fixed points are found around $L = 14$. We suspect that this occurs because very little training data came from this region in parameter space (L around 14 and relatively low σ) because the MCF model that generated the training data set converged fast towards the jam flow (relatively high σ) for L around 14. The ANN-m-2 (red) that was trained with less data still adequately captures the bifurcation diagram of the MCF model for $11 < L < 17$. However, it exhibits abnormal behaviour close to the Hopf point of the MCF model at $L \approx 10$. It actually predicts the left Hopf point to be situated at $L \approx 6.4$. Next to this malfunction, an unstable jam-flow fixed point is found at $L \approx 17.5$ and as for the ANN-m-1 the algorithm finds a stable free-flow fixed points close to $L = 14$. The identified failures of the ANN-m models highlight the need of training data sets that adequately cover important regions in the phase space to estimate the bifurcation diagram of an ANN car-following model.

3.3. Bifurcation analysis for the ANN-r model

Next, field data from the case 1 experiment from [7] is used to train an ANN model. The trained model will be referred to as the ANN-r model. In the experiment from [7] a phantom jam occurred when 22 cars drove on a circuit of a circumference of 230 m. Cars were supposed to drive with a velocity of about $30\,\text{km}\,\text{h}^{-1} \approx 8.33\,\text{m}\,\text{s}^{-1}$ and position data of the cars was recorded with a frequency of 3 Hz. In total 751 data points were collected per car. From this position data, velocities and acceleration are calculated by centred differences in time.

The task of the ANN-r model is to predict the acceleration of the following car. Therefore, the output layer has one neuron. In contrast to the ANN-m model that was trained on MCF model output data, we choose as input values just the normalized headways and normalized velocities of two cars ahead of the following car (and not all cars). Hence, the input layer has 6 neurons (2 variables from 3 cars). We make this choice to prevent the ANN to find any non-physical correlations. Note, that the model is constructed with velocity as input and not velocity difference as proposed by Chong et al. [15] and Colombaroni and Fusco [16]. As proposed by Tanaka [17], the ANN-r was constructed with two hidden layers that have each 5 neurons. The ReLU is chosen as activation function for neurons in the hidden layers and the identity for the output neuron. Again, the weights are updated using the Adam algorithm with the MSE as loss function. The construction and training of the ANN-r model is performed using the Keras library (https://keras.io/).

When an ANN-r model is used as a traffic model ($L = 230\,\text{m}$, $N = 22$) to predict the acceleration of the following car, the prediction accuracy on the validation data set is better when the target acceleration comes from one time step later in time (1/3 s shift). This can be understood due to the presence of a reaction

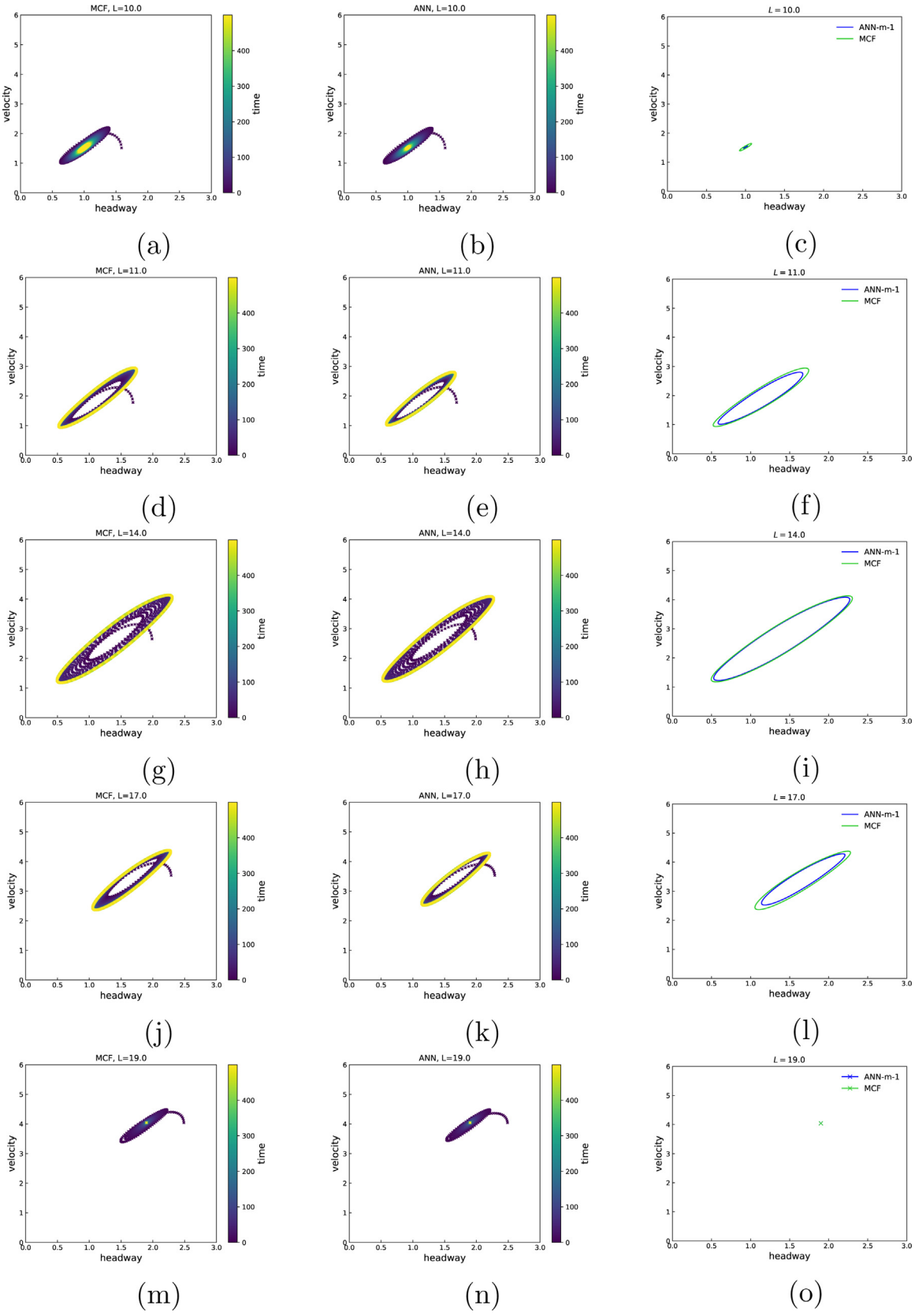


Fig. 6. Phase diagrams in the headway and velocity space, for car number 0 for the MCF model (left column) and the ANN-m-1 model (middle column) for $L \in \{10, 11, 14, 17, 19\}$. The limit cycles that develop are shown for both models in the right column.

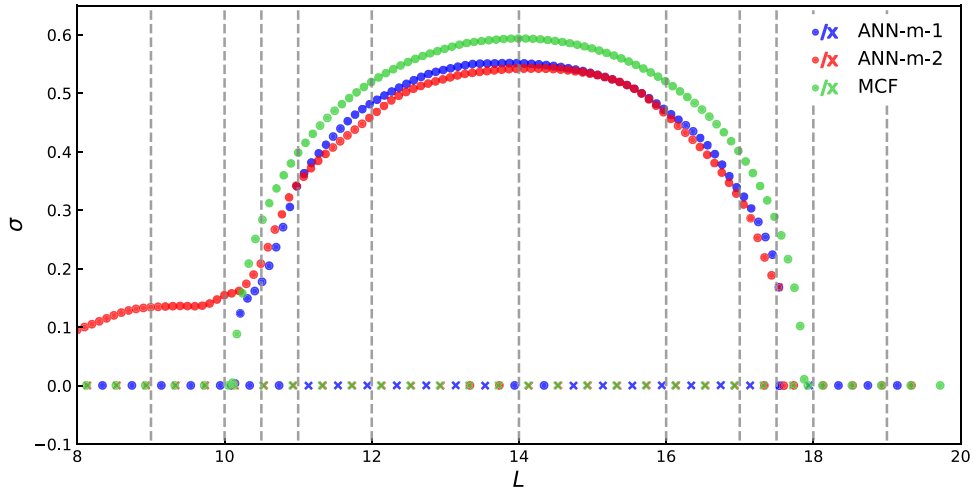


Fig. 7. Bifurcation diagram in L for the MCF model (green) and two ANN-m (blue, red) models. Dots indicate stable fixed points and crosses unstable fixed points. The ANN-m models were trained with a data set from the MCF simulations. For the ANN-m-1 model (blue) the training data set comes from MCF simulations with $L \in \{9, 10, 10.5, 11, 12, 14, 16, 17, 17.5, 18, 19\}$ (indicated with the grey dashed lines). For the ANN-m-2 model (red) the training data set comes from MCF simulations with $L \in \{11, 12, 14, 16, 17\}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

time. Moreover, the problem arose that ANN-r models with a different initialization of the weights showed differing dynamics. Some models produced crashes, some did produce traffic-flow without crashes but were not able to generate congestions and others eventually generated a crash-free flow with traffic jams. The chosen ANN architecture (2 hidden layers with each 5 neurons and ReLU as activation) and the limitation of the training data set to just one car was found to work best after a qualitative comparison with other model architectures and choices of the training data set. Eventually, a trained ANN-r with the described architecture that could produce traffic jams without generating crashes was chosen for the subsequent investigations. A thorough statistical analysis would be needed to better justify the choice of the ANN architecture, the restriction to the training data set and the choice of the crash-free ANN that produced congestions out of the trained ANN models. However, this is unfortunately beyond the scope of this paper and therefore the following results should only be considered as "proof of concept" for a bifurcation analysis of ANN car-following models.

The trajectories and velocities of the ANN-r model with $L = 230\,\mathrm{m}$ and $N = 22$ are shown in Fig. 8 in comparison to the data from [7] that was used to train the ANN. In contrast to the observations, the ANN-model trajectories are smooth. This can be attributed to the fact that the ANN-r was actually just trained on data for car number 0. Therefore, driving behaviour of the other cars is missing in the ANN-model. Furthermore and probably more important, the ANN can be considered as a "best fit" to the trainings data and is not able to capture every specific driving behaviour. Despite these shortcomings, it is remarkable that the ANN-r model still captures the feature of backwards-moving congestion waves. In the ANN-r model the congestion wave moved backwards with about $4\,\mathrm{m\,s^{-1}}$, whereas it moved backwards with about $5.5\,\mathrm{m\,s^{-1}}$ in the real experiment.

In Fig. 9 the macroscopic variables, i.e., the standard deviation of the headways (a, b) and that of the velocities (c, d), are compared with observational data. Again, the model data appear smooth in comparison to the observations. Whereas both standard deviations could capture the developing traffic jam for the ANN-r model, the standard deviation of the headway cannot be used as macroscopic measure for traffic jams in case of the real data. This is because in the real case each driver has a different headway that she/he considers as save. This implies in turn, although no traffic jam appears and hence all cars are driving with the same speed (no standard deviation of $\dot{x}$), there could still be a standard deviation of the headway different to 0. For the bifurcation analysis of the ANN-r below, the standard deviation of the headways will still be used as macroscopic measure for the traffic jam, because in case of the ANN-r it still indicates the existence of a traffic jam.

For the ANN-r model, some more changes need to be made for the bifurcation analysis. First, one cannot use the OVM function $V(\cdot)$ anymore to initialize the model. Therefore, in case of the real data experiment the last line of the lifting operator reads as follows:

$$\dot{x}_{\mathrm{new},n} = \frac{\mathrm{std}(\dot{x}_{\mathrm{old}})}{\mathrm{std}(\tilde{\dot{x}})}\left(\tilde{\dot{x}}_n - \langle\tilde{\dot{x}}\rangle\right) + \langle\tilde{\dot{x}}\rangle \text{ for } n = 1, \ldots, N \tag{26}$$

where $\mathrm{std}(\dot{x}_{\mathrm{old}})$ indicates the standard deviation of the velocities from the previous fixed point. The velocities come from the last time step of the microscopic simulation in the last corrector step of the previous fixed point. Second, due to computational limitations, the explicit equation-free method is used instead of the implicit one. Third, the settings are adjusted to the dynamics of the real traffic (see Table 1). Here, the parameter s is not fixed any more but a function of the secant $\hat{w}$ in the predictor step:

$$s = 0.02\sin(\alpha) + 2\cos(\alpha)\alpha = \arctan\left(100\frac{\hat{w}^{(\sigma)}}{\hat{w}^{(L)}}\right) \tag{27}$$

This is done to account for the different scales of σ ($\mathcal{O}(1)$) and L ($\mathcal{O}(100)$).

When L is varied for the ANN-r, bifurcation characteristics can be observed. Fig. 10 shows the time evolution in a phase diagram for car 0 in phase space $(\Delta x, \dot{x})$ for different L. One can see that for a value of L equal to 215 m, 230 m and 245 m a limit cycle develops whereas for L of 200 m and 260 m the dynamic state of the model collapses towards a point. Hence, a congestion wave is present for 215 m, 230 m and 245 m and the free flow is stable for 200 m and 260 m. This bifurcation characteristics are similar to the one observed for the MCF model (see Fig. 7): First, an increase of L (decreasing density) leads to the development of a stable jam flow solution. A further increase eventually stabilizes the free flow again.

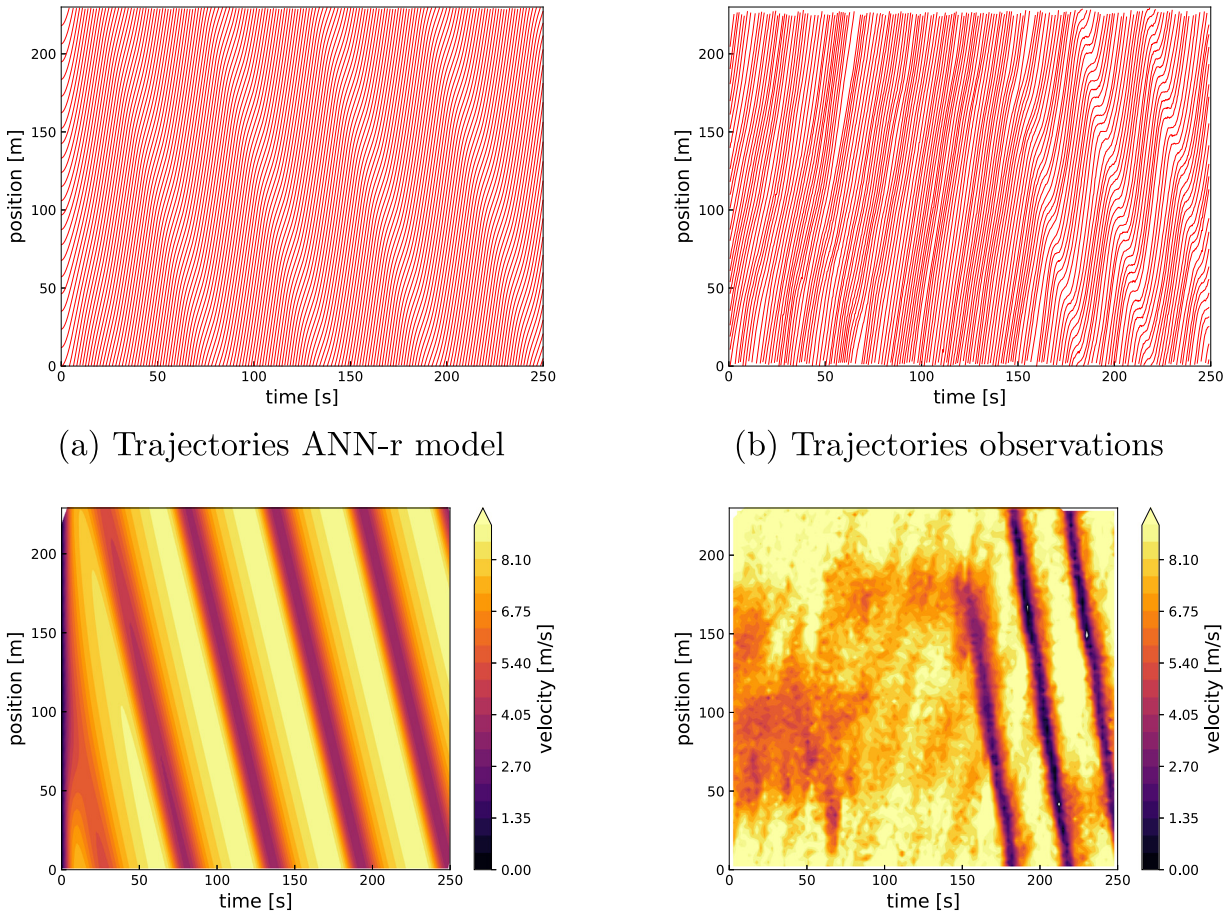


Fig. 8. Trajectories of cars in the ANN-r model with $L = 230\,\text{m}$ and $N = 22$ (a) and in the real case experiment (b). The corresponding velocities are shown in plot (c) for the ANN-r model and plot (d) for the observations.

In Fig. 11 the bifurcation diagram for the ANN-r model is shown. Following the free-flow branch ($\sigma = 0$) towards higher values of L, the free flow loses stability at around $L = 215\,\text{m}$ at a subcritical Hopf point. This corresponds to a vehicle density of

$$\rho = \frac{N}{L} = \frac{22\,\text{cars}}{215\,\text{m}} \approx 10\,\text{cars per } 100\text{ m}. \tag{28}$$

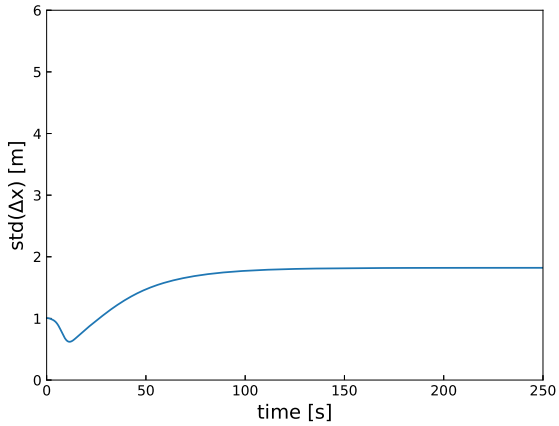
The free flow becomes stable again at another subcritical Hopf point for $L \approx 235\,\text{m}$ ($\rho \approx 9\,\text{cars}/100\,\text{m}$). For the jam flow, fold points are found for $L \approx 196\,\text{m}$ ($\rho \approx 11\,\text{cars}/100\,\text{m}$) and $L \approx 256\,\text{m}$ ($\rho \approx 8.5\,\text{cars}/100\,\text{m}$). Hence, the ANN-r model has two bi-stable regions where both, free flow and jam flow, can be stable. The unstable branches connecting the fold points and the Hopf points are difficult to find. A reason for this could be that the ANN-r produces indistinct derivatives at the fold points that make it impossible for the pseudo-arclength continuation to pass the fold. One could maybe find unstable fixed points by applying backwards integration using the backwards extrapolation of the equation-free method as described in Marschler et al. [22]. However, this is beyond the scope of this paper.

The bifurcation diagram in Fig. 11 certainly does not fully correspond to the bifurcation diagram of a real traffic situation. For this, the training data set is too small and not diverse enough. However, it is remarkable that it was even possible to construct a bifurcation diagram based on an ANN model that was trained on real data. A more diverse data set that captures a wider range of vehicle densities would probably lead to a more realistic bifurcation diagram.

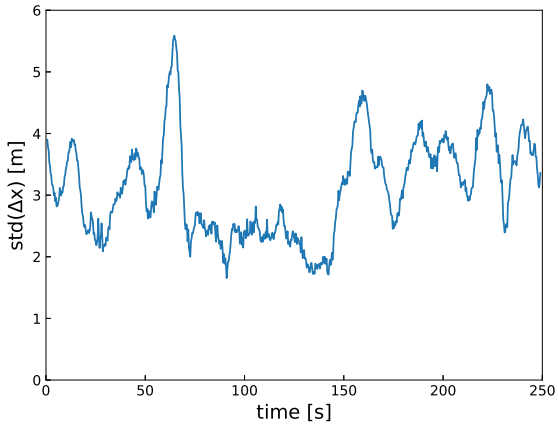
4. Summary and discussion

In this study, the equation-free bifurcation analysis technique was applied to a MCF model and two different ANN models. Due to the equation-free method it was possible to compute partial bifurcation diagrams of the MCF with a rather large number of cars ($N = 60$). Furthermore, regime analyses were done by finding the (cyclic) fold and Hopf point of the bifurcation diagram. It was found that the (cyclic) fold and the Hopf point shift towards higher values of the velocity scale, v_0, if w is decreased and λ increased. This means, that considering more cars ahead and relaxing the velocity of the following car to the group velocity of the leading cars makes a stable free-flow possible for higher velocities. In more practical terms this means, one could avoid traffic jams if car drivers would be able to react to multiple cars ahead. This is an interesting feature that was known from previous studies [i.e. 11]. However, our study better quantifies this effect by introducing a weighting function with parameter w and by performing bifurcation and regime analyses for w and λ where the existence of a bi-stable regime was found that was not mentioned by Peng and Sun [11].

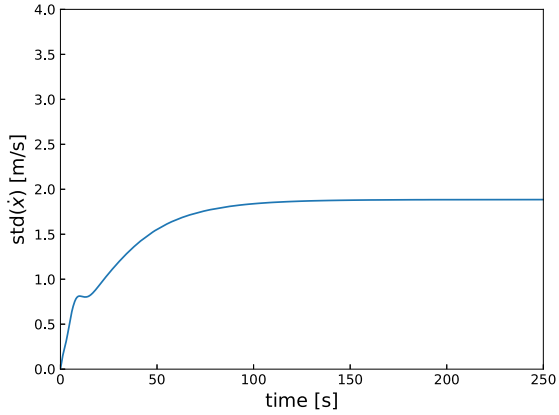
Interesting future work would be to study the bifurcation diagram in more detail in the two-parameter plane (v_0, h). As already mentioned in Section 3.1, the bifurcation behaviour found hints to the existence of a (co-dimension 2) Bautin bifurcation (cf. section 8.3 of Kuznetsov [27]). In particular, the existence of two limit cycles and of a cyclic fold, and the change from a supercritical to subcritical Hopf bifurcation are characteristic properties of a



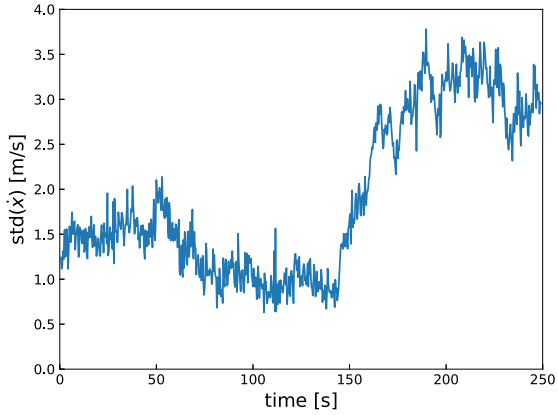
(a) Standard deviation of headway ANN-model



(b) Standard deviation of headway observations



(c) Standard deviation of velocity ANN-model



(d) Standard deviation of velocity observations

Fig. 9. Time series of different macroscopic variables from the ANN-r model and from observational data. In the upper plots the standard deviation of the headways is depicted for the ANN-r model (a) and the observations (b). In the lower plots the standard deviation of the velocity is shown for the ANN-r model (c) and the observations (d).

Bautin bifurcation. In this case, one has to demonstrate that the first Lyapunov coefficient of the microscopic model is zero and that the second Lyapunov is non-zero.

ANN models were trained on MCF model data output (ANN-m-1/2) and real data (ANN-r). The ANN-m-1 model was trained on MCF model data with a wide variety of vehicle densities that incorporated the free-flow regime and the jam-flow regime. The weights of the synapses connecting the first layer neurons with the hidden layer neurons clearly showed the relationship that was prescribed to the MCF model between the following car and the leading cars. However, a malfunction of the ANN-m-1 model became clear since it rated information from cars behind the following car as relatively important. The trained ANN-m-1 model could fairly well reproduce the bifurcation characteristics in variable L. The ANN-m-2 model that was trained only on data from the jam-flow regime failed to correctly reproduce the bifurcation diagram of the MCF model for densities lower or higher than the ones in the training data set. These findings highlight that the training data set has to come from a wide variety of vehicle densities to find the correct bifurcation characteristics.

Another ANN model (ANN-r) was trained on real data coming from the case 1 experiment of Sugiyama and Yamada [29]. The input data of the following car was limited just to cars ahead to avoid over-fitting, avoid correlations with the car behind the following car and to keep the computation fast. It was shown that an ANN-r model is in general able to produce a crash-free traffic-flow with congestions. However, the traffic flow was considerably more regular than in the real experiment. This occurred most likely because the ANN-r was able to predict only the main characteristics and not every driving behaviour based on the limited input. It was possible to construct the bifurcation diagram of the ANN-r model using again the equation-free method. Unfortunately, unstable jam-flow branches were not detected by the algorithm. However, the results show that it is possible to construct a bifurcation diagram from an ANN model that is trained on real data. This indicates that it is possible to construct bifurcation diagrams of real traffic situations if the architecture of the ANN model is improved and the training data sets are more diverse.

Such knowledge of the bifurcation characteristics of real traffic could for instance help traffic control to better avoid the traffic-jam regime. Furthermore, if bi-stable regimes are found in real traffic, the opportunity arises to change the traffic regime from stable jam flow to stable free flow by artificially introduced

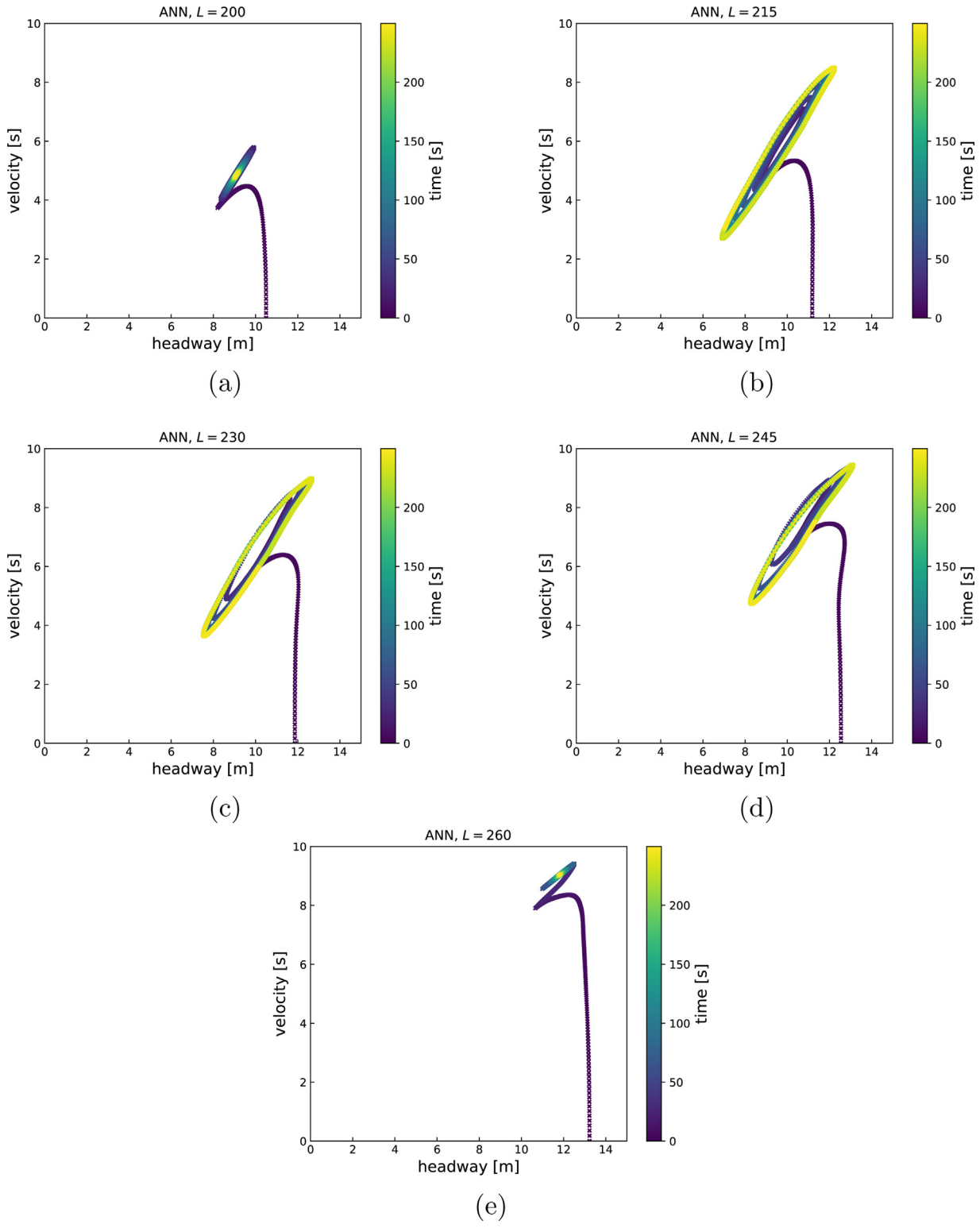


Fig. 10. Phase diagrams for car number 0 in the velocity–headway phase space for the ANN-r model simulations using different circuit length, L.

perturbations. In addition, the characteristics of the bifurcation diagram of the ANN-r correspond well with the characteristics that are found for car-following traffic models, supporting the notion that car-following models such as the MCF model capture essential physics of traffic flow.

CRediT authorship contribution statement

Paul Petersik: Conceived the idea of this study, Writing of the paper, Carried out the work. **Debabrata Panja:** Conceived the idea of this study, Writing of the paper, Gave guidance for carrying

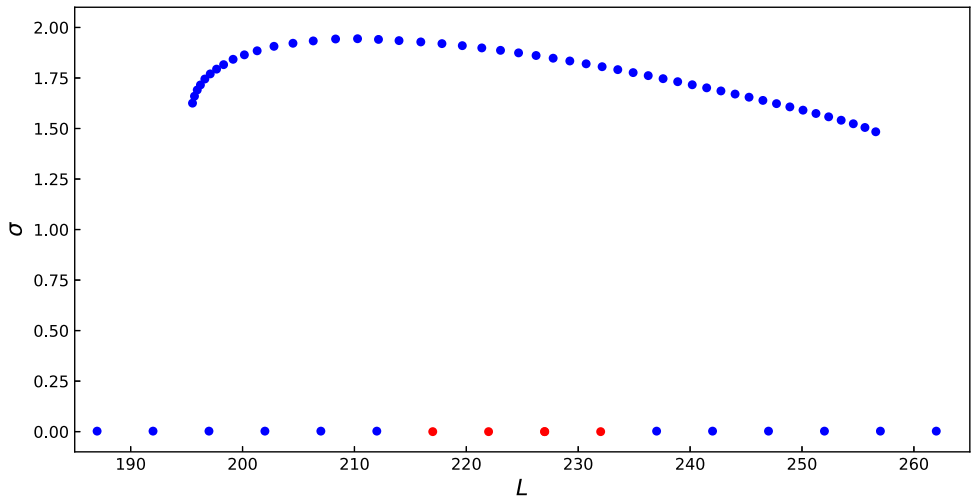


Fig. 11. Bifurcation diagram of the ANN-r model. Blue dots are stable fixed points and red dots are unstable fixed points. The blue dots that are shown with less opacity are most likely occurring due to a malfunction of the ANN-r model. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

out the work. **Henk A. Dijkstra:** Conceived the idea of this study, Writing of the paper, Gave guidance for carrying out the work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] B. Blasius, A. Huppert, L. Stone, Complex dynamics and phase synchronization in spatially extended ecological systems, Nature 399 (6734) (1999) 354–359.
[2] D. Helbing, Traffic and related self-driven many-particle systems, Rev. Modern Phys. 73 (4) (2001) 1067.
[3] I.S. Aranson, L.S. Tsimring, Patterns and collective behavior in granular media: Theoretical concepts, Rev. Modern Phys. 78 (2) (2006) 641–692.
[4] M.J. Lighthill, G.B. Whitham, On kinematic waves. I. Flood movement in long rivers, Proc. R. Soc. A 229 (1178) (1955).
[5] G. Orosz, R.E. Wilson, G. Stépán, Traffic jams: dynamics and control, Phil. Trans. R. Soc. A 368 (1928) (2010) 4455–4479.
[6] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, Y. Sugiyama, Dynamical model of traffic congestion and numerical simulation, Phys. Rev. E 51 (2) (1995) 1035.
[7] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S.-i. Tadaki, S. Yukawa, Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam, New J. Phys. 10 (3) (2008) 033001.
[8] D. Helbing, B. Tilch, Generalized force model of traffic dynamics, Phys. Rev. E 58 (1) (1998) 133–138.
[9] R. Jiang, Q. Wu, Z. Zhu, Full velocity difference model for a car-following theory, Phys. Rev. E 64 (1) (2001).
[10] H. Lenz, C. Wagner, R. Sollacher, Multi-anticipative car-following model, Eur. Phys. J. B 7 (2) (1999) 331–335.
[11] G.H. Peng, D.H. Sun, A dynamical model of car-following with the consideration of the multiple information of preceding cars, Phys. Lett. A 374 (15) (2010) 1694–1698.
[12] J. Zheng, K. Suzuki, M. Fujita, Car-following behavior with instantaneous driver–vehicle reaction delay: A neural-network-based methodology, Transp. Res. C 36 (2013) 339–351.
[13] V. Papathanasopoulou, C. Antoniou, Towards data-driven car-following models, Transp. Res. C 55 (2015) 496–509.
[14] X. Wang, R. Jiang, L. Li, Y. Lin, X. Zheng, F.Y. Wang, Capturing car-following behaviors by deep learning, IEEE Trans. Intell. Transp. Syst. 19 (3) (2018) 910–920.
[15] L. Chong, M.M. Abbas, A. Medina Flintsch, B. Higgs, A rule-based neural network approach to model driver naturalistic behavior in traffic, Transp. Res. C 32 (2013) 207–223.
[16] C. Colombaroni, G. Fusco, Artificial neural network models for car following: Experimental analysis and calibration issues, J. Intell. Transp. Syst. 18 (1) (2014) 5–16.
[17] M. Tanaka, Development of various artificial neural network car-following models with converted data sets by a self-organization neural network, J. East. Asia Soc. Transp. Stud. 10 (2013) 1614–1630.
[18] I. Gasser, G. Sirito, B. Werner, Bifurcation analysis of a class of 'car following' traffic models, Physica D 197 (3) (2004) 222–241.
[19] G. Orosz, R.E. Wilson, B. Krauskopf, Global bifurcation investigation of an optimal velocity traffic model with driver reaction time, Phys. Rev. E 70 (2) (2004).
[20] G. Orosz, B. Krauskopf, R.E. Wilson, Bifurcations and multiple traffic jams in a car-following model with reaction-time delay, Physica D 211 (3) (2005) 277–293.
[21] I.G. Kevrekidis, C.W. Gear, G. Hummer, Equation-free: The computer-aided analysis of complex multiscale systems, AIChE J. 50 (7) (2004) 1346–1355.
[22] C. Marschler, J. Sieber, R. Berkemer, A. Kawamoto, J. Starke, Implicit methods for equation-free analysis: Convergence results and analysis of emergent waves in microscopic traffic models, SIAM J. Appl. Dyn. Syst. 13 (3) (2014) 1202–1238.
[23] G.H. Peng, X.H. Cai, C.Q. Liu, B.F. Cao, M.X. Tuo, Optimal velocity difference model for a car-following theory, Phys. Lett. A 375 (45) (2011) 3973–3977.
[24] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[25] S. Dreyfus, The numerical solution of variational problems, J. Math. Anal. Appl. 5 (1) (1962) 30–45.
[26] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Pearson Higher Education & Professional Group, 2016, Google-Books-ID: XS9CjwEACAAJ.
[27] Y.A. Kuznetsov, Elements of Applied Bifurcation Theory, Springer Verlag, New York, U.S.A., 1995.
[28] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv:1412.6980, arXiv:1412.6980 [cs].
[29] Y.-b. Sugiyama, H. Yamada, Simple and exactly solvable model for queue dynamics, Phys. Rev. E 55 (6) (1997) 7749.