Neural-Network-Based Calibration of Macroscopic Traffic Flow Models

Nkiediel Alain Akwir, Jean Chamberlain Chedjou and Kyandoghere Kyamakya

Abstract This chapter proposes a neural-network-based calibration of macroscopic traffic flow models expressed in the form of nonlinear partial differential equations (PDEs). The calibration scheme/module developed aims at improving both accuracy and stability of the nonlinear PDE models in order to make them more realistic. In essence, the macroscopic nonlinear PDE models of traffic flow (proposed in the literature) are generally inaccurate, unstable, and unlikely to describe the realistic dynamics of traffic flow. To overcome these drawbacks we exploit the artificial neural network paradigm to build a concept called NN-PDE (neural network + PDE solver), which involves the macroscopic model (in the form of a nonlinear partial differential equation) on the one hand, and the calibration scheme/module-based artificial neural network (ANN) on the other. The calibration scheme/module is used to dynamically adjust/optimize all outputs of the nonlinear "PDE" model in order to obtain a realistic set of parameters that could be used by the PDE model to describe the real/realistic dynamics of traffic flow. Overall, specific attention is devoted to some relevant issues related to the modeling concept such as accuracy, stability, and overfitting, just to name a few. These issues generally occur due to the complexity of the PDE model at stake for the sake of an accurate traffic flow model. Finally, some simulation results are shown to demonstrate the effectiveness of the NN-PDE concept developed.

N.A. Akwir (✉) · J.C. Chedjou · K. Kyamakya
Transportation Informatics Group (TIG), Institute of Smart Systems
Technologies, Alpen-Adria University of Klagenfurt, 9020 Klagenfurt, Austria
e-mail: nkiediel.akwir@aau.at

J.C. Chedjou
e-mail: jean.chedjou@aau.at

K. Kyamakya
e-mail: kyandoghere.kyamakya@aau.at

K. Kyamakya et al. (eds.), *Recent Advances in Nonlinear Dynamics and Synchronization*, Studies in Systems, Decision and Control 109,
DOI 10.1007/978-3-319-58996-1_7

1 Introduction

1.1 Background of Traffic Flow Modeling

Transportation has always been a vector of economic development of society and thereby contributes to improvement in the quality of life. However, due to the growth of population density and urbanization, traffic congestion is increasing worldwide. The transportation-related socioeconomic and environmental impacts such as increasing travel time, fuel consumption, and pollution have been given a tremendous attention by decision-makers in recent decades [1].

Developing new control and optimization strategies to better manage vehicular traffic [2–4] can be viewed as the best realistic option to deal with issues related to congestion, pollution, fuel consumption, etc., since building new roads is very expensive and could be an unrealistic solution, since the availability of land is not always ensured [5].

Different modeling concepts have been developed in order to capture traffic dynamics: (1) mathematical models (in the form of ODEs or PDEs) [6–8], (2) graphical models (graphs and Petri nets) [9–12], and (3) software tools (Synchro, Vissim, and Visum, etc.) [13–15], for macroscopic and microscopic traffic simulation.

The mathematical modeling of traffic flow has always been a challenging issue in science and engineering due to the striking and complex nature of the dynamical behavior of traffic flows (e.g., nonlinearity, hysteresis, stiffness, saturation, oversaturation, jam, shock waves, rarefaction waves, stop-and-go waves, platoons, bottleneck, chaos, just to name a few) [16–19].

The use of ordinary and/or partial differential equations for traffic flow modeling depends on the level of detail (i.e., macroscopic, microscopic, and mesoscopic levels) [20]. For instance, microscopic traffic flow models distinguish and trace the behavior of each individual vehicle (e.g., car following model, lane change model) [20]. The use of ordinary differential equations for microscopic traffic flow modeling requires a huge amount of data for the optimization of corresponding ODE parameters. Further, the use of ODE models for microscopic traffic simulation is generally time-consuming (i.e., very slow). Macroscopic models aggregate vehicles allowing a description of traffic flow as a continuum (i.e., a global view that does not distinguish vehicles). The related macroscopic models are expressed in the form of partial differential equations, in which dependent variables are generally obtained as the average of the three fundamental traffic parameters (i.e., mean flow, mean density, mean speed). Macroscopic models are generally less costly computationally than microscopic models. However, the accuracy of macroscopic models is an issue when compared with the good accuracy of microscopic models. Mesoscopic models generally encompass both microscopic and macroscopic models and usually involve human intelligence (e.g., driver behavior) and/or artificial intelligence (e.g., sensors) [21, 22].

The mathematical traffic flow models provided in the literature (see state of the art on traffic mathematical modeling) are all derived using a series of specific assumptions (e.g., see analogy with fluid dynamics [23]; see also analogy with gas kinetics [24]). However, these assumptions often do not consider all practical constraints (i.e., those faced on the road network when one is considering real traffic dynamics). Thus, the available mathematical models reveal only a partial view of the reality (i.e., the real traffic dynamics and related traffic phenomena and scenarios). This underscores and justifies the statement that most traffic models presented in the literature are not realistic enough. Further, the literature does not provide a mathematical model that can be used as a general framework for traffic modeling in considering all practical realistic constraints. Among the great number of mathematical models published so far, none of them is likely to simultaneously describe traffic phenomena (all practical realistic constraints) such as shock waves, rarefaction waves, stop-and-go waves, platoons, bottlenecks, and jams, just to name a few. The cited phenomena are generally described by different types of PDEs (e.g., linear, nonlinear, convex, concave, hyperbolic, nonhyperbolic).

A seminal mathematical model for traffic flow was proposed by Lighthill–Whitham (1955) [6] and Richards (1956) [25], the so-called LWR model, expressed in the form of a partial differential equation. This model, also known as a first-order model, is based on the continuity equation from compressible dynamics theory, which expresses the conservation of a flowing quantity from one point to another. Despite the fact that the LWR model can reproduce some phenomena, it is based on a number of assumptions that are not always realistic (e.g., constant speed, no overtaking, no ramp). To solve this issue, Payne [26], Ross (1988) [27], and Del Castillo (1993) [28] have proposed mathematical models of second order, which take into account the speed dynamics. These latter models were later improved by Zhang [8], Jiang et al. [29], and Gupta et al. [7]. Even though these PDE-based macroscopic traffic models can reproduce the spatiotemporal behavior of traffic flow on a road segment relatively well, they suffer mainly from a lack of accuracy, and they are relatively slow (although they remain significantly much faster (for traffic simulation) than their microscopic counterparts). Compared to microscopic traffic simulators, the macroscopic ones are less accurate. Essentially, in their raw forms, none of the existing macroscopic traffic models and simulation concepts fulfills the necessary requirements (e.g., robustness, accuracy, realism, ultrafast simulation) for online traffic simulation. The calibration of the PDE models for traffic flow is of importance in addressing the requirement of online traffic simulation (i.e., fast computational speed and a practical constraints aware accuracy).

1.2 Background of Model Calibration

Calibration is a process aiming at fitting a given model using a set of empirical data. This process is very important in traffic modeling, since a traffic mathematical model generally involves a number of parameters that must be adjusted for some specific scenarios. Otherwise, there is often a big gap (errors) between theoretical models and experimental data.

Several traffic flow calibration algorithms have been proposed based on the optimization process [30]. Nelder and Mead (1965) [31] proposed an algorithm for multidimensional unconstrained optimization. This algorithm was proven to be suitable for problems modeled by objective functions (or cost function) expressed in nonlinear, discontinuous, or stochastic forms. Such an optimization algorithm requires a large number of terations to achieve convergence, however, without significant improvement in some cases; further, several updates of the optimization algorithm are also needed. Genetic algorithms (GA) that have been proposed by Golberg (1989) and Holland (1992) [32, 33] are part of the largest class of evolutionary algorithms. Genetic algorithms mimic evolution in biological processes using natural selection, mutation, and crossover techniques [32, 33]. Genetic algorithms are appropriate for various optimization problems such as discontinuous objective functions and nondifferentiable, stochastic, or highly nonlinear objective functions. Despite the fact that this algorithm is flexible in searching complex solution spaces, each iteration requires as many cost function evaluations as the population size; thus the algorithm is computationally costly (i.e., slow). Another very important algorithm is the cross entropy method proposed by Rubinstein and Kroese (2004) [34] and de Boer et al. (2005) [35]. This algorithm is applicable to discontinuous, nondifferentiable, or highly nonlinear objective functions. The main drawbacks of this algorithm are the high computational cost and the slow convergence, since it requires as many cost function evaluations as the size of the population.

The few optimization algorithms underscored so far (see [22]–[27]) have been used intensively during the past decades, in classical calibration processes and/or calibration modules. Generally, despite the fact that those algorithms may be applied to complex problems (e.g., discontinuous, nondifferentiable, or highly nonlinear cost functions), they have some inherent drawbacks, which are mostly related to the computational cost (i.e., too time-consuming) and lack of convergence.

1.3 Key Contribution and Organization

A novel calibration concept for nonlinear partial differential equation (PDE) models used for macroscopic traffic modeling is developed here. The concept developed uses artificial neural networks (ANN) for the calibration of partial differential equation (PDE) models of traffic flow. The main advantage of the concept developed is the possibility of overcoming some drawbacks of classical (or traditional) calibration concepts (e.g., flexibility, universality, robustness, and accuracy). Another advantage of the calibration concept developed is the straightforward and easy applicability to traffic flow modeling and an online simulation capability. The traffic models (PDEs) and the online simulation (using PDEs) are essential in analyzing, understanding, depicting, and controlling the realistic and spatiotemporal evolution of traffic flow dynamics while considering practical operational constraints (e.g., real-time constraint, realistic nature of solutions regarding real traffic constraints, correctness of solutions, stability and robustness). In essence, the PDE model helps to obtain the spatiotemporal evolution of the behavior of the traffic flow, while the calibration helps to improve both accuracy and correctness of the related PDE models.

The remainder of this chapter is organized as follows. Section 2 is concerned with the mathematical modeling of traffic flow. A PDE model is proposed for the traffic flow scenarios under investigation. Section 3 presents the calibration concept developed. Full details of all analytical steps toward development of the calibration module are provided. The calibration module is based on artificial neural networks (ANN). Section 4 develops a novel concept to which we have assigned the acronym NN-PDE. This concept combines PDE and ANN paradigms. The PDE model considers the spatiotemporal behavior of traffic flow, while the ANN paradigm is used for calibration to improve the accuracy of the PDE model. The offline ANN training process is performed using data from quasireal traffic provided by a relevant popular simulator tool called VISSIM. VISSIM is an accurate microscopic traffic simulation tool, which is able to configure several traffic scenarios while taking into account several practical conditions. Section 5 is devoted to numerical simulation. A case study of a concrete traffic scenario modeled by PDE is considered. The calibration of the resulting PDE model is conducted, and the results of the calibration obtained are discussed. Finally, the advantage of the calibration is clearly demonstrated. This chapter ends with some concluding remarks and an outlook (Sect. 6).

2 PDE Models for Macroscopic Traffic Flow

Several traffic flow models have been proposed in the literature to express the dynamics of traffic flow on arterial roads [6, 8, 26]. These models are expressed in the form of continuous differential equations and involve the three fundamental parameters of traffic, namely (1) the flow (number of cars crossing a section of the road per unit of time), (2) the density (number of cars per unit of length), and (3) the speed (rate of variation of the position over time).

As already stated above, the past decades have witnessed the derivation of several mathematical models for the analysis of traffic flow at the macroscopic level. Some interesting mathematical models corresponding to specific scenarios are provided by Eqs. (6), (8), and (9).

Traffic Scenario 1: *Here no overtaking and no ramp (see Fig. 1).*

This scenario shows a simple case of traffic flow in a single lane. It has been shown (see [6]) that this scenario is modeled mathematically by Eq. (1). The assumption made in [36] to obtain the model in Eq. (1) is the homogeneous nature of the traffic flow (i.e., all cars are assumed to be of the same type).

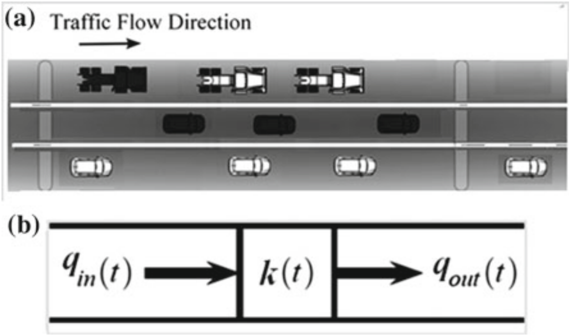

Fig. 1 General representation of the traffic flow on a road segment of finite length. **a** Illustration of the traffic flow without overtaking. **b** Synoptic representation of the traffic flow for the sake of modeling. (***source***: [37])

$$\frac{\partial k}{\partial t} + \frac{\partial q}{\partial x} = 0 \tag{1}$$

In Eq. (1), the dependent variables are represented by k and q. These two variables stand for the traffic density k and traffic flow q on a road segment. The independent variables x and t are the spatial and temporal dimensions, respectively. For the sake of solvability of Eq. (1), a second analytical expression is envisaged in Eq. (2) in order to express the relationship among the three fundamental parameters

$$q = k \cdot u. \tag{2}$$

Equations (1) and (2) cannot be used to express the spatiotemporal evolution of the three fundamental parameters of traffic flow (k, q, and u). This justifies the choice of additional relationships expressing the empirical Greenshield's model (see Eqs. (3), (4), and (5)). Equations (3), (4), and (5) are used to obtain the fundamental diagrams shown in Fig. 2. This diagram is a three-dimensional (3D) representation of the interaction between the fundamental parameters of traffic flow. Figure 2 clearly emphasizes that the speed is high at low traffic density (e.g., weak interaction between vehicles), while at high traffic density (e.g., strong interaction between vehicles), the speed is low. This observation is important, since it could be used to express the monotonicity of the speed of vehicles on roads. This is important in depicting some phenomena such as shock waves, stop-and-go waves, and rarefaction waves, just to name a few.

Many mathematical expressions have been proposed in the literature to represent the 3D relationship between the fundamental parameters of traffic flow empirically. Some classical and commonly used empirical models are Greenshield's [38], Greenberg [39], Underwood [40], Wang et al. [41]. These empirical models are generally obtained through the use of data history, which corresponds to specific measurements of the three fundamental parameters: flow, speed, and density.

The present chapter exploits Greenshield's model expressed in a 3D system of coordinates as follows:

$$u = u(k) = u_f - \frac{u_f}{k_j} k, \tag{3}$$

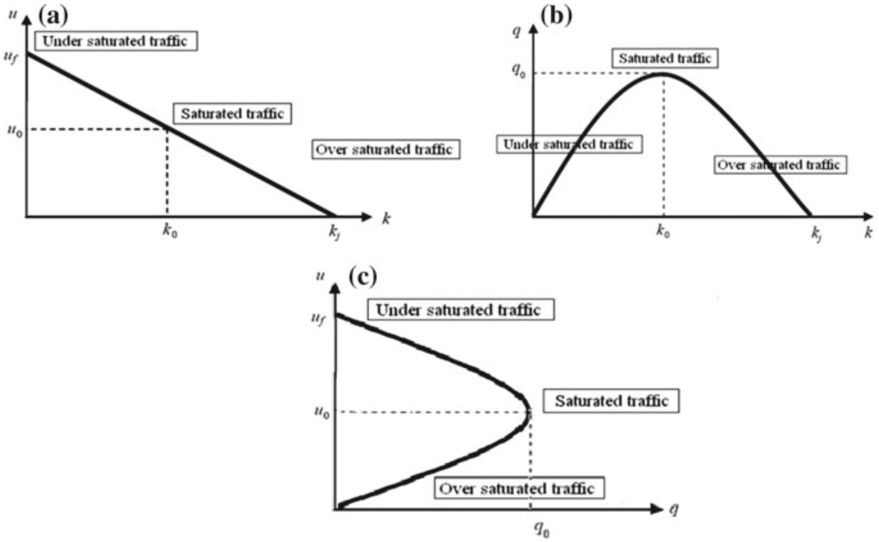

Fig. 2 Fundamental diagrams: **a** Speed versus density. **b** Flow versus density. **c** Speed versus flow

$$q = q(k) = ku_f - \frac{u_f}{k_j}k^2, \tag{4}$$

$$u = u(q) \Rightarrow u^2 = uu_f - \frac{u_f}{k_j}q. \tag{5}$$

The quantity u_f stands for the free flow speed. The free flow here expresses the possibility of driving without any interaction between vehicles. Thus, the driver could choose the speed (denoted here by u_f) at his convenience (it is the maximum allowed speed according to legal speed limit on a particular road segment). Equations (3), (4), and (5) are used to plot the fundamental diagrams of speed versus density in Fig. 2a, flow versus density in Fig. 2b, and speed versus flow in Fig. 2c. The diagrams in Fig. 2 reveal the three possible traffic states (i.e., undersaturation, saturation, and oversaturation). The quantities k_0 and u_0 are the density (critical) and speed at the capacity of the road. The quantity k_j stands for jam density, and the quantity u_f is the free flow speed, as mentioned before.

The classical and seminal model for traffic flow, the so-called LWR model, proposed by Lighthill, Whitham, and Richard [25] corresponds to Eq. (6), obtained by combining Eqs. (1) and (4):

$$\begin{cases} \frac{\partial k}{\partial t} + \frac{\partial q}{\partial x} = 0 \\ q = q(k) = ku_f - \frac{u_f}{k_j}k^2. \end{cases} \tag{6}$$

Equation (6) is a first-order quasilinear hyperbolic partial differential equation in the dependent variable $k(x, t)$. The nonconservative form of Eq. (6) is given as follows:

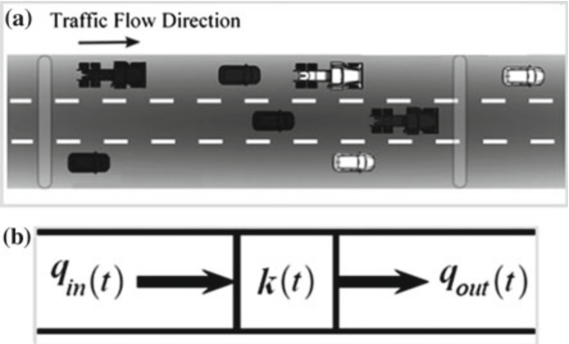

Fig. 3 General representation of the traffic flow on a road segment of finite length. **a** Illustration of the traffic flow with overtaking and without ramps and detectors. **b** Synoptic representation of the traffic flow for the sake of modeling. (***source***: [37])

$$\frac{\partial k}{\partial t} + (u_f - 2\frac{u_f}{k_j}k)\frac{\partial k}{\partial x} = 0. \tag{7}$$

Equation (7) is the traffic flow model that is considered in this chapter.

Traffic Scenario 2: *Here there is overtaking but no ramp (see Fig.* 3*).*

This scenario shows a simple case of traffic flow in multiple lanes with overtaking. It has been shown (see [42]) that this scenario is modeled mathematically by Eq. (8). Equation (8) is obtained by assuming a homogeneous traffic flow:

$$\begin{cases} \dfrac{\partial k_1}{\partial t} + \dfrac{\partial q_1}{\partial x} = \dfrac{k_2}{T_2^1} - \dfrac{k_1}{T_1^2}, \\ \dfrac{\partial k_j}{\partial t} + \dfrac{\partial q_j}{\partial x} = \dfrac{k_{j-1}}{T_{j-1}^j} - \dfrac{k_j}{T_j^{j-1}} + \dfrac{k_{j+1}}{T_{j+1}^j} - \dfrac{k_j}{T_j^{j+1}}, \\ \dfrac{\partial k_N}{\partial t} + \dfrac{\partial q_N}{\partial x} = \dfrac{k_{N-1}}{T_{N-1}^N} - \dfrac{k_N}{T_N^{N-1}}. \end{cases} \tag{8}$$

In (8), the subscripts $J = 2, \ldots, N-1$ refer to the interior lanes, and the subscripts 1 and N refer to the extreme lanes; $T_j^k = T_j^k(k_j, k_k)$ is the vehicle transition rate from two neighboring lanes (j to lane k).

Traffic Scenario 3: *Here there are both overtaking and ramps (see Fig.* 4*)* This scenario shows a simple case of traffic flow on multiple lanes with overtaking. It has been shown (see [43]) that this scenario is modeled mathematically by Eq. (9). The assumption of homogeneous traffic flow has been made to obtain Eq. (9):

$$\frac{\partial k}{\partial t} + \frac{\partial ku}{\partial x} = \pm\rho(x,t). \tag{9}$$

The quantity $\rho(x,t) \geq 0$ corresponds the rate of vehicles entering (+) the highway or leaving (−) the highway through ramps (respectively entrance and exit ramps).

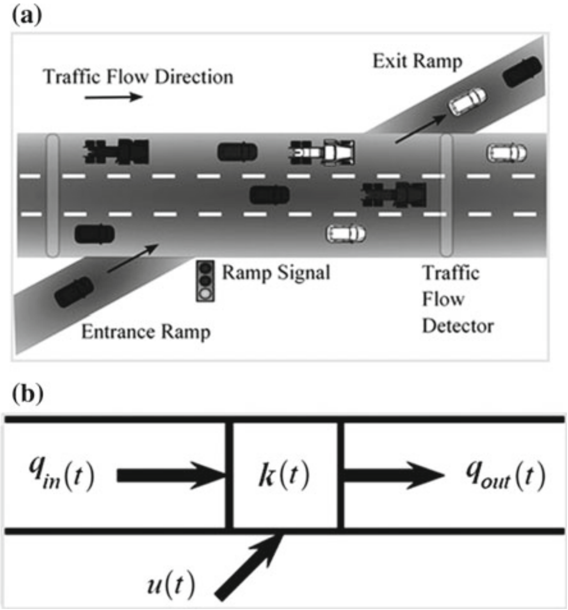

Fig. 4 General representation of traffic flow on a road segment of finite length. **a** Illustration of traffic flow and the influence of signal detectors and ramps. **b** Synoptic representation of the traffic flow for the sake of modeling. (***source***: [37])

3 Calibration Concept

3.1 General Principle

The mathematical models proposed by scientists and engineers may reflect the dynamics of systems and phenomena. However, some of these models might be inaccurate and unlikely to reflect the real dynamics of specific scenarios, especially when the model involves a large number of parameters. Therefore, without a careful prior model calibration, it would not be possible to rely on the simulated results. In order to ensure the validity of the models, they must ideally be adequately calibrated for the full range of conditions and scenarios of relevance. Several models have been developed for calibrating either macroscopic or microscopic models (see [45] and references therein).

Calibration is an optimization process involving techniques such as the genetic and memetic algorithm [44] and the cross-entropy method [45], just to name a few. The overall optimization procedure consists in estimating some parameters for minimizing a cost function that can be made, for instance, by an error (or discrepancy) between the mathematical model result and real data obtained from field measurements (or obtained from a realistic simulator like the VISSIM Simulator). This is not a trivial task, since the system of equations may be highly nonlinear in both parameters and state variables.

The parameter estimation problem can be formulated as a nonlinear least-squares output error problem that aims at minimizing the discrepancy between the model calculations and the real traffic data using the following cost function:

$$E(w) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}[k(n) - \bar{k}(n)]^2}, \tag{10}$$

where $E(w)$ is the root mean squared error (RMSE). In the calibration process, this error is minimized by finding the appropriate value of the weight w; $k(n)$ is the set of data resulting from the model, and $\bar{k}(n)$ stands for real traffic data from the field (or VISSIM Simulator).

3.2 Calibration Involving Artificial Neural Networks

An artificial neural network (ANN) [46] is a computational model implemented as a computer program aiming at emulating the key properties and operations of biological neural networks. ANNs are used to model unknown or unspecified functional relationships between the input and output of a "black box" system. In order to apply such a procedure to decision problems, a key requirement is ANN training to minimize the discrepancy between modeled and measured system output. Due to its principle or properties (e.g., training capability, usage phase, redimensioning of the network, etc.), the ANN paradigm is a good candidate for calibration tasks.

3.3 Mathematical Model of a Neural Network

A single neuron model consists of inputs, weights, an activation function, and outputs. The mathematical model of a single neuron is expressed as follows:

$$Y = f(\sum_{i} w_i x_i + b), \tag{11}$$

where the set of inputs is $X = [x_0, x_1, x_2, \ldots, x_n]$, the set of weights is $W = [w_0, w_1, w_2, w_3, \ldots, w_n]$, the output signal is $Y = (y_1, y_2, y_3, \ldots, y_n)$, the activation function is given by $f = f(W, X)$, and b is the bias.

An artificial neural network consists of neurons connected together. Several architectures of neural networks have been proposed in the literature, such as multilayer perceptron neural networks [47], recurrent neural networks [48], and Hopfield neural network [49], just to name few. In this chapter we exploit the multilayer perceptron (MLP), which consists of several layers of interconnected perceptrons. Several training methods such as the back-propagation algorithm [50] and the Levenberg–Marquardt algorithm [51, 52] are used to train this type of ANN.

Considering that we have $\bar{Y} = (\bar{y_1}, \bar{y_1}, \ldots, \bar{y_n})$ as target data (in our case they are real traffic flow data), the training is done using an optimization process iteratively in order minimize the discrepancy D between the ANN outputs and the target data:

$$minimize_w D(Y, \bar{Y}, w). \tag{12}$$

At the end of the optimization procedure, a neural network is obtained with new weight values $W^* = [w_0^*, w_1^*, w_2^*, w_3^*, \ldots, w_n^*]$ that minimize the discrepancy expressed by a suitable norm function:

$$D(Y, \bar{Y}) = \|Y - \bar{Y}\|. \tag{13}$$

Once the weight values are known, the ANN can be used to perform various tasks related to the assessment of further input data (i.e., the so-called test data).

4 The NN-PDE Concept

The NN-PDE concept is a combination of a PDE-based model of traffic flow and a calibration based on an artificial neural network. We use the PDE model, which has the capability to represent the traffic flow in the spatiotemporal domain. However, as already mentioned, the PDE models are generally inaccurate, and the calibration carried out aims at proposing a new and accurate model to which we have assigned the acronym NN-PDE. A calibration is exploited to adjust the result with reality (real traffic data), resulting in increasing the accuracy of the model. A neural network module for calibration is built offline by exploiting real traffic data (or quasireal data "from a microscopic simulator") and the data obtained from the PDE model, which takes into account some contextual information related to a specific traffic flow scenario. Let us note that real traffic data are obtained using VISSIM, which is an accurate microscopic traffic simulator that can propose and configure several scenarios. The block diagram in Fig. 5 illustrates the NN-PDE concept.

Block (a) in Fig. 5 is the PDE solver. The PDE model considered can be solved using some numerical methods such as the method of lines (MOL), finite difference methods (FDM), or CNN (cellular neural networks) [53]. The initial and boundary conditions are set up as inputs according to a given scenario. Context information (free flow speed, jam density, etc.) are also provided in order to fit with specific scenarios for a given road segment. The result obtained expresses the spatiotemporal evolution of the three fundamental parameters of traffic flow (i.e., density $k(x, t)$, speed $u(x, t)$, and flow $q(x, t)$). After solving, the models are further processed in order to fit with the input of the calibration module. Block (b) in Fig. 5 is the calibration module based on a neural network. The multilayer perceptron (MLP) architecture is used to build this module.

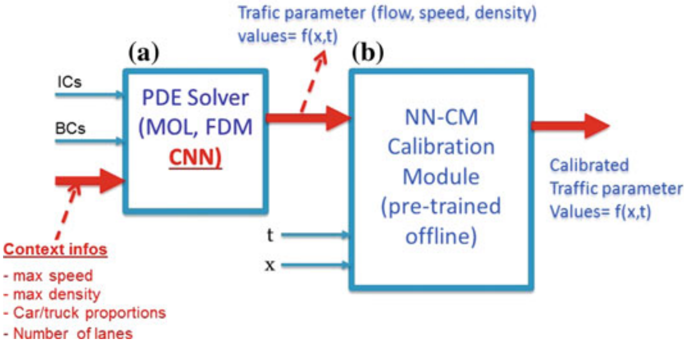

Fig. 5 The NN-PDE concept. **a** The PDE solver (ICs: initial conditions and BCs: boundary conditions). **b** The calibration module

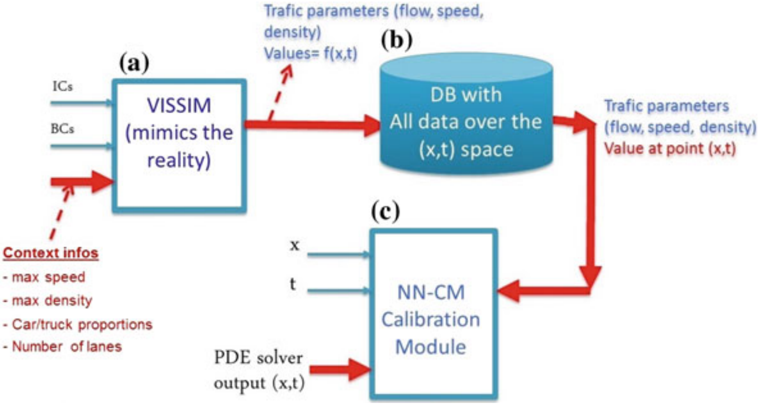

Fig. 6 Offline training of the calibration module

As mentioned before, the training of the calibration module is done offline. The block diagram in Fig. 6 shows the offline training procedure.

Block (a) in Fig. 6 is the traffic simulator tool. We use VISSIM, as mentioned before, to obtain quasireal traffic data. These data are processed in order to fit the calibration module and are stored in the database (see block (b)). The calibration module in block (c) is trained offline by considering as input the data from the PDE solver (taking into account different scenarios) and target (corresponding prestored) data from the database (real traffic data).

5 Simulation Results and Their Commenting

5.1 Numerical Schemes

We consider a scenario consisting of a single-lane model without ramp, as illustrated in Fig. 1, for numerical simulation. This scenario is modeled mathematically by Eq. (7). Applying the finite difference method (e.g., Lax–Friedrichs scheme) to Eq. (7), we obtain the following difference equation:

$$k_i^{n+1} = \left(\frac{k_{i+1}^n + k_{i-1}^n}{2}\right) - \frac{\Delta t}{2\Delta x}\left[\left(u_f - 2\frac{u_f}{k_j}\left(\frac{k_{i+1}^n + k_{i-1}^n}{2}\right)\right)\left(k_{i+1}^n - k_{i-1}^n\right)\right], \tag{14}$$

where the index i represents the road section and the index n denotes the discrete time. The stability condition of the numerical scheme is given by (see [54])

$$u_f \frac{\Delta t}{\Delta x} \leq 1; \quad k_j = p[\max(k_0(x_i))]; \quad p \geq 2. \tag{15}$$

The condition (15) is used during numerical simulation in order to guarantee the stability of solutions. Thus the parameters of Eq. (7) are chosen according to condition (15). During our various numerical simulations, several sets of parameters are envisaged, each of which encompasses the parameters Δt, Δx, u_f, k_j, and k_0 (see cases 1, 2, 3, and 4 in Sect. 5.3). In all four cases, the numerical solutions of Eq. (7) are valid if the condition (15) is satisfied. Otherwise, the solutions are not valid for the scenario in Fig. 1.

5.2 Neural Network Module for Calibration

The neural network module used for calibration encompasses the following input and output variables:

- $k(x, t)$ is the traffic flow obtained as a solution from the PDE model in Eq. (7) in the form of an MxN matrix; M corresponds to the number of iterations in the time domain, while the number of iterations in the space domain corresponds to N; $k(x, t)$ is transformed into a row vector of size $(M * N)$ and is further used as the first input of the neural network (See Fig. 7). The second input corresponds to the vector t of size M, while the size of the third input corresponding to vector x is N. An appropriate use of the neural network in (Fig. 7) is recommended by choosing the inputs 2 and 3 of equal size. This is the fundamental condition for all inputs of the neural network (inputs 1, 2, and 3) to be of the same size.
- $\bar{k}(x, t)$ is the traffic flow data obtained from the VISSIM simulator; $\bar{k}(x, t)$ is transformed into a row vector of size $(M * N)$ and is further used as the target

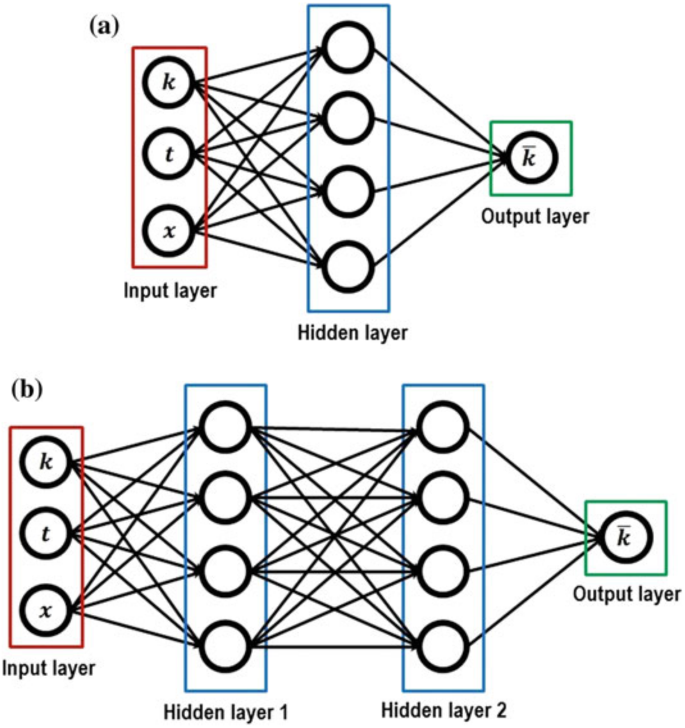

Fig. 7 The one- and two-layer neural networks used for training

solution during the training phase of the neural network. The target solution and the three inputs are of equal size.

- $K(x, t)$ is the result obtained after calibration. This corresponds to the result of the new system developed in this chapter to which we have assigned the acronym of NN-PDE. The data provided by the NN-PDE are in vector form of size $(M * N)$. This vector is of the same size as x, t, k, and $\bar{k}(x, t)$. This is an important condition to ensure that the training procedure of the neural network is effective.

Another important condition for the training procedure of neural networks is concerned with the number of layers and the number of perceptrons per layer. In Fig. 7, the number of layers and/or the number of perceptrons is monitored to increase the accuracy and avoid overfitting as well. During the training process conducted in this work, the choice of both number of layers and perceptrons is based on trial-and-error. It has been observed that increasing the number of perceptrons per layer significantly improves the accuracy of the training process. However, the main problem encountered in increasing the number of perceptrons was the memory consumption and the overfitting observed when the number of perceptrons per layer exceeds 100. It has also been observed that increasing the number of hidden layers leads to an improvement in the accuracy of the training process. However, it has been observed that the accuracy rate remains constant beyond 10 hidden layers. The next section (see Sect. 5.3) is concerned with the presentation of the results obtained using the following methods: (a) numerical solution of the original PDE in Eq. (7) to express the traffic flow dynamics corresponding to the scenario in Fig. 1; (b) solution of the same scenario provided by VISSIM; (c) solution of the same scenario provided by the new concept NN-PDE developed in this chapter. Overall, a comparison between solutions (a) and (b) has led to a significant difference. This difference can be explained by the fact that the theoretical model (Eq. (7)) does not express reality. Thus, the NN-PDE developed is a new mathematical model that is much closer to reality. This statement is justified in Sect. 5.3 through a benchmarking between NN-PDE and VISSIM.

5.3 Results and Comments

5.3.1 Original PDE Versus VISSIM

We apply the finite difference method to the original PDE in Eq. (7) to obtain the discrete form in Eq. (14). This form is further solved using MATLAB to obtain the numerical solution of the original PDE. This solution (see Fig. 8a) expresses the traffic flow dynamics in Fig. 1. We also use VISSIM to simulate the traffic flow dynamics in Fig. 1, and the solution obtained is depicted in Fig. 8b.

Using the set of parameters $\Delta x = 10\,\text{m}$; $\Delta t = 1\,\text{s}$; $u_f = 8.3\,\text{m/s}$; $k_j = 0.160$ Veh/m; and $k_0 = 0.08$ Veh/m, the solution of the original PDE is obtained in Fig. 8a. The solution in Fig. 8b shows the simulation of the same scenario in VISSIM. It can be observed from Fig. 8a, b that there is a significant divergence (see Fig. 8c) between the solutions provided by the two methods (i.e., Original PDE and VISSIM). The divergence in Fig. 8c corresponds to a normalized root mean square error (NRMSE) of 28.02%. This value of the NRMSE justifies the need of calibration in order to reduce the NRMSE. The calibration in this context consists in deriving a new model (to which we have assigned the acronym NN-PDE) that must be able to provide results similar to those obtained using VISSIM. Here VISSIM is considered the target solution, because it expresses the traffic dynamics closer to reality.

5.3.2 NN-PDE Versus VISSIM

The NN-PDE model in Fig. 5 is obtained by combining the original PDE with a neural network. The NN-PDE model is further implemented in MATLAB. Using NN-PDE, several simulation results are obtained for different configurations of neural network. These configurations are obtained by monitoring the number of hidden layers and the perceptrons involved. Further simulation results are obtained for two different values of the free flow speed (i.e., $u_f = 30$ and $u_f = 50\,\text{km/h}$).

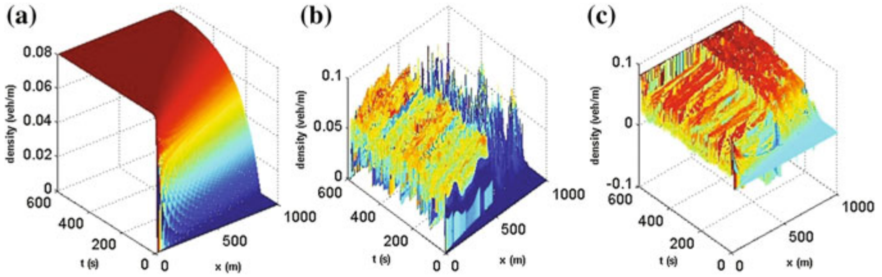

Fig. 8 Simulation results. **a** PDE results. **b** Vissim results. **c** Error (PDE-Vissim). The NRMSE corresponds to 20%

Case-1: We use the following parameters of the NN-PDE: value of the free flow speed ($\mathbf{u_f = 30\,km/h}$), neural network architecture (multilayer perceptron (MLP)), **1** hidden layer involving **50** perceptrons.

The simulation results obtained are illustrated in Fig. 9a–f. The results in Fig. 9a–c correspond to the situation without calibration shown in Fig. 8. In contrast, the results in Fig. 9d–f correspond to the situation with calibration. These results obtained using NN-PDE are compared with the results obtained using VISSIM. The outcome of comparison is depicted in Fig. 9f. Using this figure, we see that the NRMSE calculated corresponds to 6.55%. This value of the NRMSE obtained using NN-PDE shows a significant improvement when compared with the previous value of the NRMSE (28.02%) obtained without calibration (see Fig. 8).

Case-2: We use the following parameters of NN-PDE: value of the free flow speed ($\mathbf{u_f = 30\,km/h}$), neural network architecture (MLP), **10** hidden layers involving **20** perceptrons.

The simulation results obtained are illustrated in Fig. 10a–f. The results in Fig. 10a–c correspond to the situation without calibration shown in Fig. 8. In contrast, the results in Fig. 10d–f correspond to the situation with calibration. These results obtained using NN-PDE are compared with the results obtained using VISSIM. The outcome of the comparison is depicted in Fig. 10f. Using this figure, we see that the NRMSE calculated corresponds to 2.21%. This value of the NRMSE obtained using NN-PDE shows a significant improvement when compared with the previous value of the NRMSE (31.51%) obtained without calibration (see Fig. 8).

Case-3: We use the following parameters of the NN-PDE: value of the free flow speed ($\mathbf{u_f = 50\,km/h}$), neural network architecture (MLP), **1** hidden layer involving **50** perceptrons.

The simulation results obtained are illustrated in Fig. 11a–f. The results in Fig. 11a–c correspond to the situation without calibration shown in Fig. 8. In contrast, the results in Fig. 11d–f correspond to the situation with calibration. These results obtained using NN-PDE are compared with the results obtained using VISSIM. The outcome of the comparison is depicted in Fig. 11f. Using this figure, we see that the NRMSE calculated corresponds to 6.39%. This value of the NRMSE obtained using

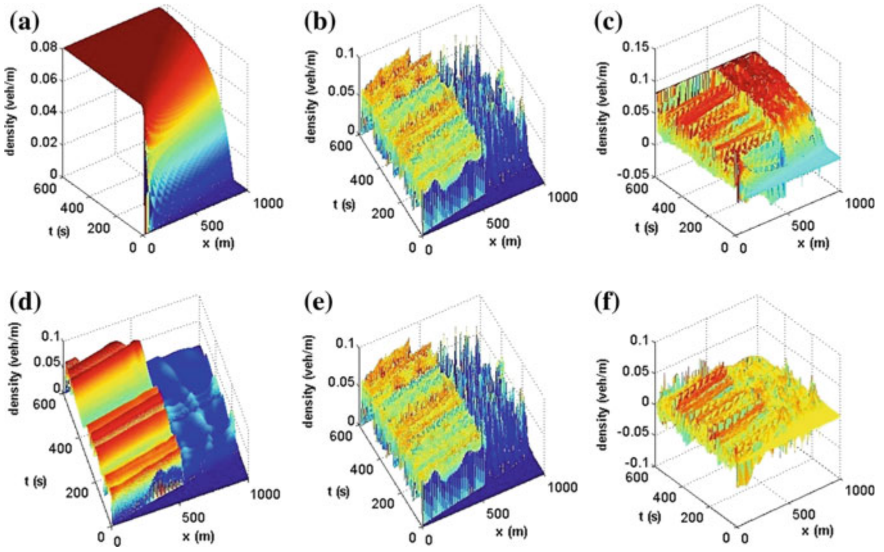

Fig. 9 Simulation results of Configuration-1: MLP, 1 layer, 50 neurons: **a** PDE model (input). **b** Vissim data (targets). **c** Initial error (PDE-Vissim). **d** Calibrated result. **e** Vissim data (targets). **f** Final error (calibrated result-Vissim). The NRMSE corresponds to 6.55%

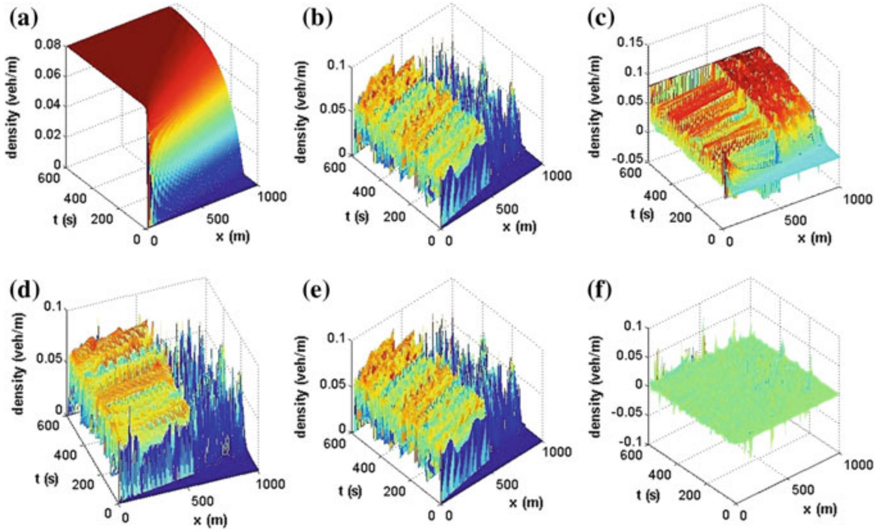

Fig. 10 Simulation results of Configuration-2: MLP, 10 layers, 20 neurons per layer: **a** PDE model (input). **b** Vissim data (targets). **c** Initial error (PDE-Vissim). **d** Calibrated result. **e** Vissim data (targets). **f** Final error (calibrated result-Vissim). The NRMSE corresponds to 2.21%

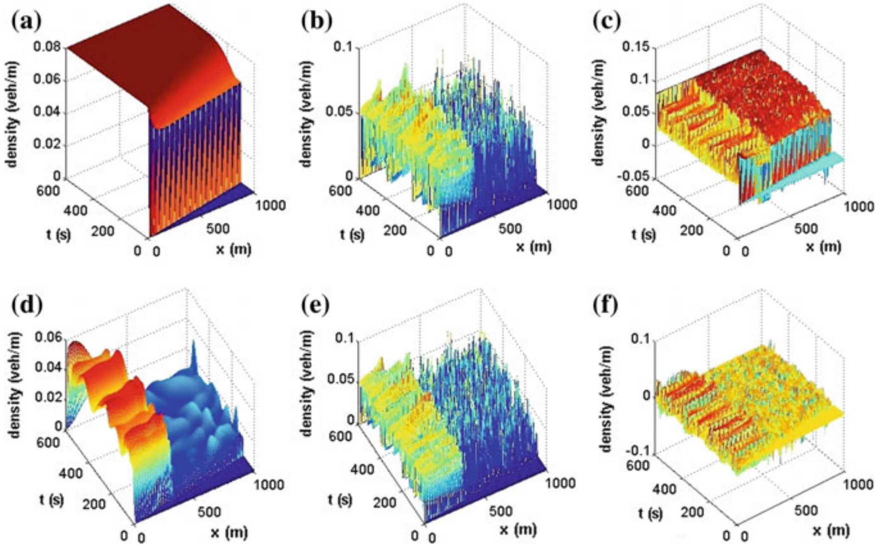

Fig. 11 Simulation results of Configuration-1: MLP, 10 layers, 20 neurons per layer: **a** PDE model (Input). **b** Vissim data (targets). **c** Initial error (PDE-Vissim). **d** Calibrated result. **e** Vissim data (targets). **f** Final error (calibrated result-Vissim). The NRMSE corresponds to 6.39%

NN-PDE shows a significant improvement when compared with the previous value of the NRMSE (28.02%) obtained without calibration (see Fig. 8).

Case-4: We use the following parameters of NN-PDE: value of the free flow speed ($\mathbf{u_f = 50\,km/h}$), neural network architecture (MLP), **10** hidden layers involving **20** perceptrons. The simulation results obtained are illustrated in Fig. 12a–f. The results in Fig. 12a-c correspond to the situation without calibration shown in Fig. 8. In contrast, the results in Fig. 12d–f correspond to the situation with calibration. These results obtained using NN-PDE are compared with the results obtained using VISSIM. The outcome of the comparison is depicted in Fig. 12f. Using this figure, we see that the NRMSE calculated corresponds to 3.5%. This value of the NRMSE obtained using NN-PDE shows a significant improvement when compared with the previous value of the NRMSE (31.51%) obtained without calibration (see Fig. 8).

5.3.3 Comment on Results

The four cases envisaged (see cases 1–4) show that the accuracy of NN-PDE significantly depends on the neural network architecture. This dependence is confirmed by the different values of the NRMSE obtained in each case. Further, it has been observed that the variation of the free flow speed also significantly affects the results obtained using NN-PDE. Our various numerical simulations have consisted in monitoring and varying both the number of perceptrons in the hidden layer and the free

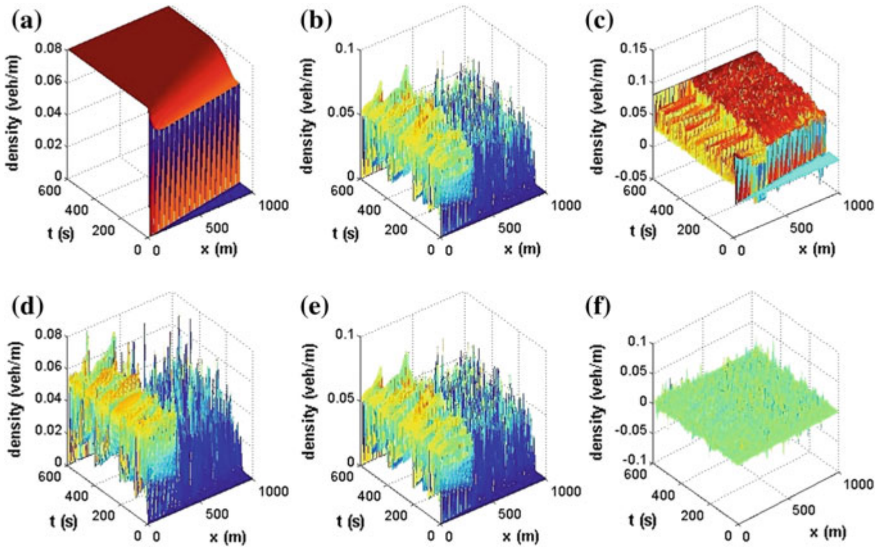

Fig. 12 Simulation results of Configuration-1: MLP, 10 layers, 20 neurons per layer: **a** PDE model (input). **b** Vissim data (targets). **c** Initial error (PDE-Vissim). **d** Calibrated result. **e** Vissim data (targets). **f** Final error (calibrated result-Vissim). The NRMSE corresponds to 3.5%

Table 1 Summary of the results of the eight configurations of NN-PDE

Network configuration	NRMSE in %	
	$\mathbf{u_f = 30\,Km/h}$	$\mathbf{u_f = 50\,Km/h}$
No calibration	28.02	31.51
1 layer, 50 perceptrons	6.55	6.39
10 hidden layers, 20 perceptrons per layer	2.21	3.5

flow speed. The results obtained by varying the two cited parameters are summarized in Table 1. This table provide the NRMSE for different combinations (each of which consists of a fix number of perceptrons in the hidden layer and a fixed value of the free flow speed). The results reported in Table 1 show the values of NRMSE for cases without calibration versus the NRMSE for cases with calibration. The outcome of this comparison clearly demonstrates that the new NN-PDE model developed provides results that are closer to those provided by VISSIM. This statement can be used to justify the realistic nature of the NN-PDE model developed in this chapter.

6 Concluding Remarks

In this chapter we have proposed a partial differential equation (PDE) based traffic flow model that is calibrated using a neural network (NN). The model developed is called NN-PDE (i.e., a combination of a PDE model with a calibration module based on NN). The calibration carried out is important, because the traditional mathematical models for traffic flow (proposed by the state of the art of traffic flow modeling) involve many parameters due to the complex dynamics undergone by traffic flow. Further, these models are based on assumptions that are generally nonrealistic, since they are unlikely to express the real traffic dynamics observed on arterial roads. Among the huge number of mathematical models for traffic flow proposed in the literature we have considered the LWR model, which is the seminal model proposed in the literature for modeling and simulating the dynamics of traffic flow at a macroscopic level of detail. The advantage of considering the LWR model is twofold. The LWR model is simple and can reproduce some relevant insights into the dynamics of traffic flow. These insights express specific phenomena (shockwaves, rarefaction waves, stop-and-go waves, etc.) that are generally observed on arterial roads. The calibration of LWR carried out in this chapter has led to a new model, called NN-PDE, which is very accurate and more realistic in describing the dynamics of traffic flow when compared to the basic LWR model. This statement has been proven in the framework of a benchmarking process conducted by considering the numerical simulation of various scenarios of traffic flow on arterial roads. Regarding the numerical simulation and benchmarking carried out in this chapter, the basic LWR model (corresponding to traffic flow in a single lane) has been simulated using the finite difference method (FDM) combined with the related stability conditions derived analytically. Both FDM scheme and stability conditions have been implemented in MATLAB to obtain numerical results. Further, in order to obtain the new NN-PDE model as a calibrated version of the basic LWR model, the MLP architecture has been used to design a neural network (NN) scheme that was trained offline using as target the data provided by VISSIM. The input data used for training the NN scheme are those provided by the LWR model. The outcome of the training has led to the NN-PDE model, which is further used for numerical simulations. Using NN-PDE, the various numerical simulations performed have revealed that the accuracy of NN-PDE can be improved by varying the configuration of the neural network module and also by monitoring the parameters (e.g., free flow speed) of the NN-PDE model. We have noticed that increasing both the number of perceptrons per layer and the number of layers involved in the NN architecture leads to a significant improvement in the accuracy of the NN-PDE model. However, we have detected a maximum number of layers and also a maximum number of perceptrons in layers above which the numerical results obtained do not lead to further significant improvement of the accuracy of NN-PDE. In contrast, a further increase of the number of perceptrons and layers may inherently lead to a waste of resources (memory), convergence issues (e.g., failure to converge), instability (loss of robustness), and low or worse computing performance. A benchmarking has been considered with the aim of comparing results of the numerical simulation of several specific traffic flow scenarios obtained using three different methods, namely the LWR model, the NN-PDE model, and VISSIM. The outcome of this comparison has revealed that the NN-PDE model developed provides results that are closer to results provided by VISSIM for all traffic flow scenarios envisaged in this chapter. For the same scenarios, the LWR model provides results that significantly diverge from those obtained using VISSIM. Hence the NN-PDE model developed shows very good accuracy and thus appears to be more appropriate (than the classical LWR) to model and simulate traffic flow scenarios on arterial roads.

References

1. Banister, D.: Sustainable urban development and transport -a Eurovision 2020. Transp. Rev.: Transnatl. Transdiscipl. J. **20**(1), 113–130 (2000)
2. Dimitrakopoulos, G., Demestichas, P.: Intelligent transportation systems. IEEE Veh. Technol. Mag. **5**(1), 77–84 (2010)
3. Deakin, E.: Frick, karen trapenberg and Skabardonis, Alexander, Intelligent Transport Systems: Linking Technology and transport Policy to help steer the future, University of california Transportation center (2009)
4. Ming-wei, L., Jun, Y.: Calculating the contribution rate of intelligent transportation system for the smooth general characteristic of urban traffic. In: 2014 International Conference on Management Science and Engineering (ICMSE), Helsinki (2014)
5. Papageorgiou, G., Maimaris, A.: Intelligent Transportation Systems. InTech, Rijeka (2012)
6. Lighthill, M.J., Whitham, G.B.: On kinematic waves: II. A theory od traffic flow on long crowed roads, in Proceeding Royal Society, London (1955)
7. Gupta, A.K., Katiyar, V.K.: Phase transition of traffic staes with on-ramp. Phys. A **371**(2), 674–682 (2006)
8. Hm, Z.: A theory of nonequilibrium traffic flow. Transp. Res. Part B **32**(7), 485–498 (1998)
9. Chera, C.M., Balino, J. L., Dauphin-Tanguy, G.: Two-equation traffic flow models framed within the bond graph theory. In: Proceedings of the 2010 Spring Simulation Multiconference in SpringSim '10, San Diego (2010)
10. Baruah, Ak: Traffic control problems using graph connectivity. Intern. J. Comput. Appl. **86**(11), 1–3 (2014)
11. Tolba, C., Lefebre, D., Thomas, P., El Moudni, A.: Continuous and timed petri ntes for the macroscopic and microscopic traffic flow modelling. Simul. Model. Pract. Theory **13**, 407–436 (2005)
12. Zhang, Y., Qiang, W., Yang, Z.: A new traffic signal control method based on hybrid colored petri net in isolated intersections. Intern. J. Intell. Transp. Syst. Res. (2016)
13. Bains, M.S., Ponnu, B., Arkatkar, S.S.: Modeling of traffic flow on Indian expressways using simulation technique in Procedia - social and behavioral sciences, Changsha (2012)
14. Trafficware: Synchro, Trafficware Group Inc. http://www.trafficware.com/synchro-studio.html (2015). Accessed 18 Oct 2016
15. PTV Visum, PTV Group. http://vision-traffic.ptvgroup.com/fr/produits/ptv-visum/ (2016). Accessed 18 Oct 2016
16. Siebel, F., Mauser, W., Moutari, S., Rascle, M.: Balanced vehicular traffic at a bottleneck. Math. Comput. Modell. **49**, 689–702 (2009)
17. Ai, W., Shi, Z., Liu, D.: Phase plane analysis method of nonlinear traffic phenomena. J. Control Sci. Eng. (2015)
18. Li, T.: Nonlinear dynamics of traffic jams. In: Second International Multi-Symposium on IMSCCS (2007)

19. Lo, S.-C., Cho, H.-J.: Chaos and control of discrete dynamic traffic model. J. Franklin Inst. **342**, 839–851 (2005)
20. Bellemans, T., De Schutter, B., De Moor, B.: Models for traffic control. J. A **43**(4), 13–22 (2002)
21. DTALite: a queue-based mesoscopic traffic simulator for fast model evaluation and calibration. Cogent Eng., *1*(1) (2014)
22. Burghout, W., Koutsopoulos, H.N.: A discrete-event mesoscopic traffic simulation model for hybrid traffic simulation. In: IEEE Intelligent Transportation Systems Conference, ITSC '06, Toronto, Ontario (2006)
23. Richards, P.: Shock waves on the highway. Oper. Res. **4**, 42–51 (1956)
24. Treiber, M., Hennecke, A., Helbing, D.: Derivation, properties, and simulation of a gas-kinetic-based, nonlocal traffic model. Phys. Rev. E, **59**(1) (1999)
25. Richards, P.I.: Shock waves on the highway. Oper. Res. **4**, 42–51 (1956)
26. HJ, P.: Models of freeway traffic control. In: Bekey, G.A. (ed.) Mathematical Models of Public System, pp. 51–61. Simulation Councils, inc., La Jolla (1971)
27. Ross, P.: Some properties of macroscopic traffic models. Transp. Res. Rec. **1194**, 129–134 (1988)
28. Del Castillo, J.M., Pintado, P., Benitez, F.G.: A formulation for the reaction time of traffic flow models. In: International Symposium on the Theory of Traffic Flow and Transportation, Berkeley (1993)
29. Jiang, R., Qing-Song, W., Zhu, Z.-J.: A new continuum model for traffic flow and numerical tests. Transp. Res. Part B: Methodol. **36**(5), 405–419 (2002)
30. Spiliopoulou, A., Papamichail, I., Papageorgiou, M., Tyrinopoulos, I., Chrysoulakis, J.: Macroscopic traffic flow model calibration using different optimization. In: 4th International Symposium of Transport Simulation (ISTS'14), Ajaccio (2014)
31. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**(4), 308–313 (1965)
32. Golberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Professional, Reading (1989)
33. Holland, J.: Adaptation in Natural and Artificial Saystems. MIT Press, Cambridge (1992)
34. Rubienstein, R.Y., Kroese, P.: The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning. Springer, New York (2004)
35. De Boer, P.-T., Kroese, D.P., Mannor, S., Rubinstein, R.Y.: A tutorial on the cross-entropy method. Ann. Oper. Res. **134**(1), 19–67 (2005)
36. Mohan, R., Ramadurai, G.: State-of-the art of macroscopic traffic flow modelling. Int. J. Adv. Eng. Sci. Appl. Math. **5**(2–3), 58–176 (2013)
37. Shlayan, N., Kachroo, P.: Feedback ramp metering using godunov method based hybrid model. J. Dyn. Sys. Meas. Control **135**(15) (2013)
38. Greenshields, B.D., Bibbins, J.R., Channing, W.S., Miller, H.H.: A study of traffic capacity. In: Proceedings of the Fourth Annual Meeting of the Highway Reseach Board, Washinton (1934)
39. Greenberg, H.: An analysis of traffic flow. Oper. Res. **7**(1), 79–85 (1959)
40. Underwood, R.T.: Speed, Volume, and Density Relationships: Quality and Theory of Traffic Flow, Yale Bureau of Highway Traffic. Yale University Press, New Haven (1961)
41. Wang, D., Ma, X., Ma, D., Jin, S.: A novel speed-density relationship model based on the energy conservation concept. IEEE Trans. Intell. Transp. Syst. **99**, 1–11 (2016)
42. Kabir, M.H., Andallah, L.S.: Numerical solution of Multilane traffic flow model. Ganit J. Bangladesh Math. Soc. **33**, 25–32 (2013)
43. Helbing, D., Treiber, M.: Numerical simulation of macroscopic traffic equations. IEEE Comput. Sci. Eng. **1**(5), 89–98 (1999)
44. Pay, A., et al.: Calibration of traffic flow models using a memetic algorithm. Transp. Res. Part C: Emerg. Technol. **55**, 432–443 (2015)
45. Spiliopoulou, A., et al.: Macroscopic traffic flow model calibration using different optimization algorithms. In: 4th International Symposium of Transport Simulation (ISTS'14), Ajaccio, France (2015)

46. Pinter, J.D.: Calibrating Artificial Neural Networks by Global Op (2010)
47. Rosenblatt, F.: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Cornell Aeronautical Laboratory, Washington DC (1961)
48. Mandic, D., Chambers, J.: Recurrent Neural Networks for Prediction: Learning Algorithms. Architectures and Stability. Wiley, Chichester (2001)
49. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA **79**(8), 2554–2558 (1982)
50. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Natl. Publ. Group **323**, 533–536 (1986)
51. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. Q. Appl. Math. **2**, 164–168 (1944)
52. Marquardt, D.: An algorithm for least-squares estimation of nonlinear parameters. SIAM J. Appl. Math. **11**(2), 431–441 (1963)
53. Chedjou, J.C., Kyamakya, K.: A universal concept based on cellular neural networks for ultrafast and flexible solving of differential equations. IEEE Trans. Neural Netw. Learn. Syst. **26**(4) (2015)
54. Ahsan, A., Andallah, L.S., Hossain, Z.: Numerical solution of a fluid dynamic traffic flow model associated with a constant rate inflow. Am. J. Comput. Appl. Math. **5**(1) (2015)