Ingenieurfakultät Bau Geo Umwelt
Lehrstuhl für Wasserbau und Wasserwirtschaft

Development of Data Driven Models for Hydromorphology and Sediment Transport in Rivers

Keivan Kaveh

Vollständiger Abdruck der von der Ingenieurfakultät für Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Kai-Uwe Bletzinger

Prüfer der Dissertation:

1. Prof. Dr. sc. techn. Peter Rutschmann
2. Prof. Dr. -Ing. Dr.h.c. mult. Franz Nestman
3. Prof. Dr-Ing. Jürgen Stamm

Die Dissertation wurde am 08.08.2018 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 08.01.2019 angenommen.

Most people say that it is the intellect which makes a great scientist. They are wrong: it is character.

(Albert Einstein, 1879-1955)

Abstract

One of the weakest aspects of hydromorphological models is the use of empirical formulae for calculating sediment transport rates, which are of limited generality. In many cases, unreasonable morphological changes are predicted and the results of the different formulae often strongly vary. The reasons arise from the complexity of the interaction between flow and sediment transport, and in the limitations of the non-linear regression applied in these methods. In contrast to most traditional empirical methods, which need prior knowledge about the nature of the relationships within the data, the data-driven models (e.g. Artificial Neural Networks) learn from the data examples presented to them in order to capture the subtle functional relationships between the data. This can be determined even if the underlying relationships are unknown or the physical meaning is difficult to explain.

The main objective of this study is to develop optimal Artificial Neural Network models (ANNs), able to be integrated into a hydromorphological model system and adequately predict the morphological changes in alluvial channels under different flow conditions. To achieve this aim, the open source finite element system, TELEMAC-MASCARET, was applied to simulate different complex hydrodynamic and morphodynamic situations. The calibrated results were then used as input-data in ANN to obtain ANN-based approximators for the new proposed schemes of hydromorphodynamic-model system. In the first scheme, 2 or 3 ANN models might be utilized for hydrodynamic calculations. At each time step, the hydrodynamic variables including velocity field and water depth were transferred into the morphodynamic model (ANN-based approximator), which then sent back the updated bed elevation to the hydrodynamic model.

In the second scheme, only one ANN based approximator model was integrated into the sediment transport model SISYPHE while the TELEMAC-2D/3D remained unchanged for hydrodynamic calculations. The time-step of morphodynamic part (ANN-based approximator) was much larger than the time step of TELEMAC-2D/3D and during the flow computation, the bed level was assumed to stay constant.

The novelty of the proposed schemes is that they reduce the computation costs significantly in the prediction of both hydrodynamic and morphodynamic variables. To evaluate the prediction qualities of the proposed models, a comparative study evaluating the errors associated with the model was carried out. A statistical parameter included in this study was the measurement of goodness-of-fit between the estimated bed change and TELEMAC simulation.

Zusammenfassung

Die Verwendung von nur begrenzt allgemein gültigen, empirischen Formeln zur mathematischen Beschreibung von Sedimenttransport stellt eine Unsicherheit und Limitierung von hydromorphologischen Modellen dar. Häufig prognostizieren diese Modelle unrealistische morphologische Veränderungen und die Ergebnisse verschiedener Formeln für ein identisches Gebiet variieren stark. Ursache dieser Limitierung ist es, dass die Wechselwirkung zwischen Strömung und Sedimenttransport sehr komplex ist und nur begrenzt durch nichtlineare Regression, was diesen Methoden zu Grunde liegt, abbildbar ist.

Im Gegensatz zu den meisten traditionellen empirischen Methoden, die umfassende Kenntnis über die Beziehung zwischen Daten benötigt, können datengetriebene Modelle (z. B. künstliche neuronale Netze) von diesen Daten lernen und detaillierte funktionale Beziehungen ermitteln. Das geschieht sogar, wenn die grundlegende Beziehung zwischen den Daten unbekannt ist oder ihre physikalische Bedeutung sich schwer beschreiben lässt. Das Hauptziel dieser Arbeit ist es, geeignete Architekturen künstlicher neuronaler Netze zu entwickeln, die in ein hydromorphologisches Modellsystem integriert werden können und in der Lage sind, die morphologischen Veränderungen in alluvialen Gerinnen unter verschiedenen Strömungsbedingungen vorherzusagen.

Für die Simulation verschiedener hydrodynamischer und morphodynamischer Situationen wurde das Open-Source-System TELEMAC-MASCARET angewandt. Die kalibrierten TELEMAC-MASCARET-Ergebnisse wurden als Eingangsdaten verwendet, um ANN-basierte Approximatoren für die neu vorgeschlagenen Schemata des hydromorphodynamischen Modellsystems zu erhalten. Im ersten Schema können wahlweise 2 oder 3 ANN-Modelle verwendet werden, um die hydrodynamische Berechnungen durchzuführen. Die ermittelten Variablen, wie Geschwindigkeitsprofil und Wassertiefe, werden in jedem Zeitschritt in das neu entwickelte morphodynamische Modell (ANN-basierter Approximator) übertragen, welches die aktualisierte Flusssohle an das hydrodynamische Modell zurückspielt.

In dem zweiten Schema wird nur ein ANN-basiertes Modell in das morphodynamische Modell integriert, während für die Hydrodynamik auf das konventionelle TELEMAC-2D/3D zurückgegriffen wird. Diese Entwicklung erlaubt einen deutlich vergrößerten Zeitschritt für die morphologische Berechnung, als der Zeitschritt von TELEMAC2D/3D.

Die Neuheit der vorgeschlagenen Systeme besteht darin, dass sie den Berechnungsaufwand bei der Vorhersage von hydrodynamischen und morphodynamischen Größen deutlich reduzieren. Um die Vorhersagequalität der entwickelten Modelle zu bewerten, wurde eine Vergleichsstudie durchgeführt. Maßgebendes Kriterium war dabei die Genauigkeit, mit der die hier vorgestellten Modelle die Ergebnisse der TELEMAC-MASCARET Simulationen reproduzieren können.

Acknowledgments

First and foremost, I want to thank my advisors Prof. Dr. Peter Rutschmann, and Dr-Ing. Minh Duc Bui for providing me with the opportunity to complete my PhD thesis at the Lehrstuhl für Wasserbau und Wasserwirtschaft of the Technische Universität München. It has been a great honor to work under their supervision. I appreciate all their contributions of time, ideas, and continuous support to make my Ph.D. experience productive and stimulating. Their guidance helped me in all the time of research and writing of this dissertation. Besides my advisors, I would like to thank the rest of my thesis committee for their insightful comments and encouragement, but also for the questions, which incented me to widen my research from various perspectives. My sincere thanks also go to Dr-Ing. Franz Zunic and Tobias Liepert for taking time out of their busy schedule to help me in different situations. I would also like to express my gratitude to my colleagues and friends in Munich and Obernach for the help and support they have offered me. Without their help, I doubt I would have been able to complete the project in such a proficient and timely manner. I consider myself very fortunate for having a chance to work in a very friendly environment. Last but not the least, I would like to express my special appreciation and thanks to my family: my wife, my parents and to my brothers for supporting me spiritually throughout writing this thesis and my life in general.

Contents

Chapter 1

Introduction

1.1 Historical background

Fluid flowing over the bed of a river, channel, or estuary acts to deform the shape of the bed by transporting sediment. This process can have a detrimental impact on the coastal infrastructure and environment. For example, dredged navigational channels and coastal inlets can be rendered useless by the accumulation of transported sediment producing significant cost to return these structures to operational status or to maintain them. As another example, the structural integrity of bridges and piers may be compromised due to excessive scour of the bed around abutments. Besides these infrastructure issues, transport of pollutant with or as sediments can cause serious environmental damage. Therefore, accurate prediction of sediment transport and morphological bed level changes is necessary to manage these costly problems.

Determining the evolution of a given bed configuration due to the motion of the fluid and the resulting sediment transport was first examined in a theoretical context by Exner (1925). Exner's work in this area can be considered as a classical treatment of the problem and appears in many texts (e.g., Graf 1971; Leliavsky 1955; Raudkivi 1967; Sleath 1984; Yang 1996). It states the conservation of sediment mass and in the literature is often referred to the Exner equation which is the foundation of estuarine and river morphodynamic. From equations for the conservation of fluid and sediment mass, and through a number of simplifying assumptions, Exner derives a simplified bed evolution model that takes the form of a nonlinear hyperbolic scalar equation. Despite the relative simplicity of this model, the results obtained are, to a limited extent, in good agreement with what is observed in nature. The analytical solution provided by Exner for his model is the so-called classical or genuine solution of the initial-value problem, which is valid while the solution is continuous (Kubatko, 2007).

Numerical morphological models also involve coupling between a hydrodynamic model and equation for bed level change. The sediment conservation equation is physically a nonlinear equation for the bed level like other mass conservation equations. A common feature of these conservation laws are shock waves. It means that discontinuities of the respective physical quantities will develop when particle velocity approaches celerity (Long et al. 2008). Several decades of research effort have been devoted to the development of numerical solution techniques for obtaining accurate and stable simulation of shock behavior with invention of shock-capturing methods. The methods can be classified into two general categories: classical and modern methods. The well-known classical shock-capturing methods include the MacCormack method, Lax-Wendroff method, and Beam-Warming method. Examples of modern shock-capturing schemes include higher-order total variation diminishing (TVD) schemes, flux-corrected transport scheme, Monotonic Upstream-centered Schemes for Conservation Laws (MUSCL) based on Godunov approach (1959), and the piecewise parabolic method (PPM). Another important class of high-resolution schemes belongs to the approximate Riemann solvers by Roe (1981). Recent examples of such models are the work of Nicholson et al. (1997), Alam (1998), Kassem and Chaudhry (2002), Johnson and Zyserman (2002), and Hudson et al. (2005).

Nicholson et al. (1997) reviewed some state-of-art morphodynamic models in which many of them used classical shock capturing schemes for bed level simulation. Alam (1998) applied MacCormack scheme to the study of aggradation-degradation in alluvial channels. Kassem and Chaudhry (2002) developed a two-dimensional numerical model to predict the time variation of bed deformation in alluvial channel bends. A variety of numerical schemes, including versions of Lax-Friedrichs, Lax-Wendroff, MacCormack and Roe schemes based on shallow water equation for hydrodynamics and simple power law for sediment transport rate are discussed in Hudson et al. (2005). He showed that a flux-limited version of the Roe scheme was much more stable than Lax-Friedrichs and Lax-Wendroff schemes. However, Roe scheme suffers from calculation of eigenvectors for the Roe averaged Jacobian matrix of the entire hydrodynamics and morphology system. This calculation is possible for simple problems, like system of shallow water and simple power law of sediment transport rate, but it becomes very complex for more comprehensive hydrodynamic and sediment transport models.

In recent years, different comprehensive morphodynamic modeling systems are developed such as ECOMSed (Hydroqual, 2002), Mike-21 (Warren and Bach, 1992), Delft3D (Lesser et al., 2004), ROMS (Warner et al., 2008), FAST3D (developed at the Institute of Hydromechanics, University of Karlsruhe,Germany), and TELEMAC-

MASCARET. These modeling systems generally consist of different flow modules (from 1D to 3D): a wave propagation model, and a sand transport model encompassing bed-load and suspended load (Papanicolaou et al., 2008, Amoudry, 2008). Most existing morphodynamic modeling systems rely on finite difference methods and are therefore constrained by the use of boundary (orthogonal curvilinear horizontal coordinate systems, sigma stretched vertical coordinates), which are only suitable for simplified geometry. Moreover, filtering methods such as the lengthening of the tide or the use of the so-called morphodynamic factor (Latteux, 1995), which have been extensively applied to reduce computational costs for long term applications, introduce an additional source of uncertainty (cf. van der Wegen and Roelvink, 2008). On the other hand, most existing morphodynamic modeling systems rely on numerical methods, therefore, having high computational costs for long-term applications.

1.2 Application of data-driven methods

Due to either the physical complexity of a natural phenomenon or the time-consuming process of analyzing of a system, modeling a system is always challenging in the field of water resource engineering. Data-driven methods have been found as powerful tools to help overcome those difficulties by building the subtle functional relationships from data examples presented to them and accelerating the response of decision-makers in facing real-world problems. The popularity of these methods arises from their ability to derive relationship between complex data without need for deep understanding of their physical meanings. Data-driven methods encompass mainly two different categories, statistical and soft computing (artificial intelligent) methods. Some important advantages of these methods are that they are inexpensive, precise, and flexible. They can be used to deal with the problems, which are considered too complicated by our knowledge of mathematical equations. In system modelling, as the complexity of a system increases the efficiency of offered models by data-driven methods rises. In general, some problems of interest for which data-driven methods can be used are listed below:

- Data classification and clustering
- Function approximation
- Prediction
- Optimization

- Data generation
- General simulation

For the sake of the previously mentioned advantages and capabilities, there has been a growing enthusiasm on data-driven methods in the field of water resources and environmental engineering during the recent decade.

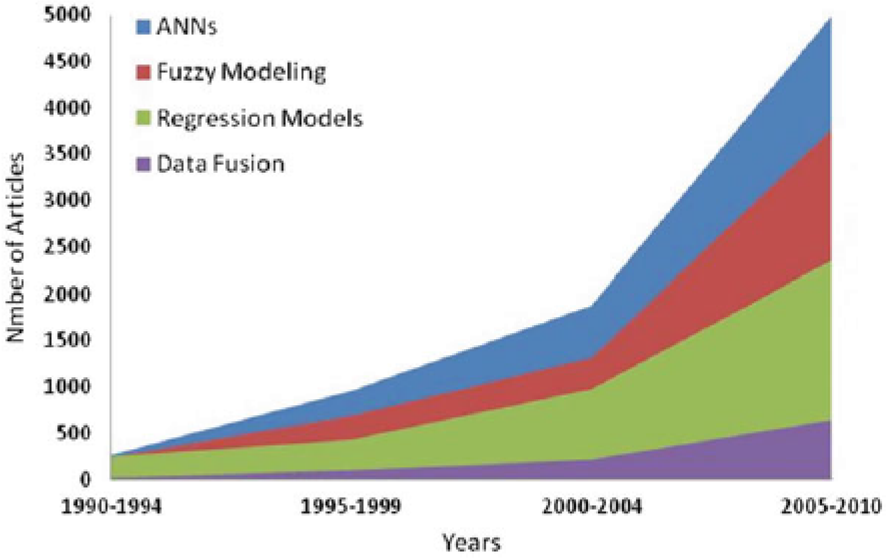


Figure 1.1: Number of articles published in the selected data-driven techniques (Google Scholar)

The trend of the number of published papers that use data-driven methods including artificial neural networks (ANNs), fuzzy modeling, regression models, and data fusion in the field of water resources and environmental engineering is illustrated in Fig 1.1 for a period of 20-year. However, the presented statistic might not represent the actual number of researches in those fields; their relative changes demonstrate the fact of an increasing demand on the application of those models. The figure shows a considerable ascending trend in all classic and modern techniques (Araghinejad, 2013).

1.3 Objectives of the research

The main objective of this research is to develop new schemes for hydromorphological model system based on data-driven methods to predict the morphological changes with lowest computational costs. To achieve this, the research has the following steps to pursue:

1. Test and analyze the performance of data-driven methods in the context of prediction and function approximation, and improve their efficiency in the field of water engineering if possible.

2. Develop analytical and conventional numerical hydro-morphological solutions, for bed level calculations to generate input data for training and testing of the networks.

3. Test different networks to obtain information on the optimum network structures and parameters.

4. Integrate the optimum networks into new proposed schemes of hydrodynamic-morphologic-model system.

5. Apply both proposed integration models on different test cases and validate their capacity in reproducing the morphodynamic modeling.

1.4 Outline of the thesis

In **Chapter 2**, the fundamentals of data-driven methods are discussed in preparation for their application in Chapters 3 and 5. The first part of this chapter deals with Artificial Neural Networks. A description of a simple artificial neuron and its analogy with biological neuron is provided. Activation functions, various type of neural network architectures and methods of training are described, followed by a discussion of generalization capability of neural networks. The second part of this chapter consists of a discussion of Adaptive Neuro Fuzzy Inference System. A description of fuzzy logic and Fuzzy Inference Systems are described extensively. The common learning rules applied to ANFIS is also provided. This chapter ends with development of some new proposed learning rules for ANFIS.

Chapter 3 essentially focuses on the basics of the morphological bed level calculations. An introduction to the Navier-Stokes equations, its simplified form including Reynolds-Averaged-Navier-Stokes and shallow water theory are detailed in this part. The modeling of morphodynamic component is provided by description of the Exner equation. The concept of incipient motion of sediment particles and sediment transport are outlined therein. This chapter also encompasses the principles of numerical schemes, focusing mainly on Finite Difference Method, for solving a system of partial differential equations, which model morphological evolution of the bed.

Chapter 4 gives an overall presentation of the TELEMAC-MASCARET system which is a comprehensive morphodynamic modeling system.

Chapter 5 is concerned with probably the most crucial objective of this research. It describes two new integration concepts for the hydro-morphodynamic modeling system using ANN models to predict the morphological changes. It also explains the motivation behind this research and novelty of the proposed concepts.

Chapter 6 discusses the criteria on which the proposed models are evaluated. This includes the examples and statistical indices that are used for assessment of model performances.

Chapter 7 tries to validate the ability of data-driven methods in the field of water engineering and sediment transport. Two test cases are considered to achieve our objective. The first one consists of the development of an ANN and ANFIS models for contraction scour estimation. The data collection and input parameters are described in details. A description of the chosen networks architectures and the reason for this choice are described extensively. This involved evaluating the optimal number of hidden layers and neurons as well as the activation function for ANN model. A same procedure was performed to evaluate the optimal parameters for ANFIS such as the number and type of membership functions. The Levenberg-Marquardt algorithm is the selected method of training for both ANN and ANFIS methods. The networks were then validated using unseen data set and the results were compared to some well-known empirical equations. This chapter also conducts a comparative study of three different learning algorithms applied to ANFIS for predicting daily-suspended sediment concentration using the second test case. The comparison is made between the two common learning rules implemented on ANFIS using the MATLAB software package and one proposed rule applied using a FORTRAN-based computer code. A comprehensive explanation of data collection is provided therein. A description of the chosen input combination and the reason for this selection is discussed. The networks were tested using new data and finally the best performance of each were compared.

Chapter 8 focuses on the development of 1D ANN for modeling dynamic channels. This section describes the development of an ANN model that can predict the morphological changes in a straight alluvial channel under steady flow discharge and uniform bed material, where the bed level changes are calculated directly from the defined flow without calculation of the bed load. This chapter clarifies extensively how the required data are selected for ANN training and explains the way in which the ANN model is designed.

Chapter 9 analyses the performance of our first proposed concept, which applies 2D/3D ANN models for hydro-morphodynamic processes. To achieve this, two examples including flow and bed change in a straight and curved channel are considered. In contrast to the 1D model where an analytical approximation is used for data generation, the open-source finite-element TELEMAC-MASCARET system will be used for 2D/3D models to simulate the morphodynamic evolution and generate input data set for ANN training process.

Chapter 10 discusses how the second proposed concept, which requires modification of some subroutines TELEMAC-MASCARET system, is implemented. A description of the chosen input parameters and the reasons for this choice are described. The similar examples of chapter 9 are used here to assess the effectiveness of the model.

Chapter 11 is a discussion of the results and conclusions with recommendations of further research.

Chapter 2

Data-driven methods

The term “method” points out a wide range of tools and programs, which can be used to imitate a real-world system. These methods are usually divided into physical and mathematical categories. A framework of different types of methods is depicted in Fig.2.1a. As can be seen from this figure, the mathematical method is broken down into three types of methods: data-driven, conceptual and analytical. Different types of mathematical methods could be classified by the complexity of a system as well as our mathematical knowledge, as shown in Fig.2.1b. Data-driven methods are split into the two general forms of soft computing methods, which is the major focus of this research, and statistical methods. Some examples of the soft computing methods are fuzzy logic, neuro-computing and genetic algorithms. Each one of these methods have been developed to deal with a specific type of data. For instance, methods such as probabilistic neural networks usually use discrete data to classify a set of input variables, where fuzzy inference systems can consider descriptive data.

In this research, Artificial Neural Networks (ANN) and Adaptive Neuro Fuzzy Inference Systems (ANFIS) are considered as the two most popular soft computing methods. These methods offer numerous advantages, such as requiring less formal statistical training, ability to implicitly detect complex nonlinear relationships between

Parts of this chapter were published as:
Kaveh, K.; Bui, M. D.; Rutschmann, P. (2015a): Improvement of ANFIS model by developing of novel hybrid learning algorithms for contraction scour modeling. Mathematics in Engineering, Science & Aerospace (MESA), 6(4).
Kaveh, K., Bui, M. D., & Rutschmann, P. (2015): New hybrid learning algorithms in adaptive neuro fuzzy inference systems for contraction scour modeling. In Proc. of the 14th International Conference on Environmental Science and Technology Rhodes, Greece.
Kaveh, K.; Bui, M. D.; Rutschmann, P. (2017): A comparative study of three different learning algorithms applied to ANFIS for predicting daily suspended sediment concentration. International Journal of Sediment Research, 32(3), 340-350.

dependent and independent variables, ability to detect all possible interactions between predictor variables, and the availability of multiple training algorithms. Disadvantages include its “black box” nature, greater computational burden, proneness to overfitting, and the empirical nature of model development (Tu, 1996). The following represents an extensive description of these two types of methods.

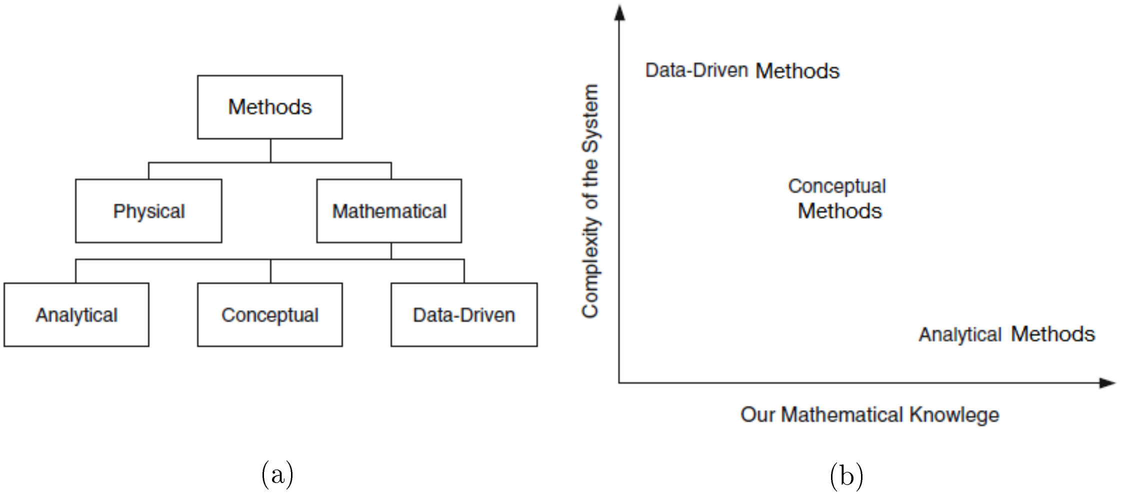


Figure 2.1: (a) A general classification of different types of methods; (b) Status of three types of mathematical methods based on two characteristics of complexity and mathematical knowledge (Modified from Araghinejad, 2013)

2.1 Artificial Neural Networks

Artificial Neural Networks are inspired by the way biological nervous systems in the human brain process information, and are used for complex problems of pattern recognition, prediction, function approximation, etc. The way computation is done by a human brain is entirely different from that of a conventional computer. Computers normally operate linearly, whereas the brain is a highly complex, nonlinear, and parallel system (Haykin, 2005). The brain gains experience during the lifetime and builds its own rules. The nervous system adapts to its surrounding environment by plasticity. ANNs mimic the way a human brain is working by using a massive interconnection of simple computing cells, known as “neurons” or “processing units”. It has been proven that ANNs are universal function approximator that can map any complicated nonlinear function.

2.1.1 Human brain

The human nervous system can be viewed as a three-stage system, as depicted in the block diagram of Fig.2.2. The central part of the system represented by the neural net is the brain, which continually receives information, perceives it, and makes appropriate decisions.

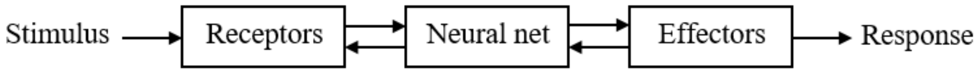


Figure 2.2: Block diagram representation of nervous system (Haykin, 2005)

In this figure, the arrows pointing from left to right indicate the forward transmission of information signals through the system while the arrows from right to left signify the presence of feedback in the system. The receptors convert stimuli from the human body or the external environment into electrical impulses that convey information to the neural net. The effectors convert electrical impulses generated by the neural net into discernible responses as system outputs (Haykin, 2005). The term neuron is introduced to explain the structural constituents of the brain. A biological neuron consists of four major parts, namely, soma, dendrite, axon, and synapse as shown in Fig.2.3. The cell body of the neuron (soma) can store small electrical charges, similarly to a battery. This storage is loaded by incoming electrical impulses from other neurons and through dendrites, which are receptive zones (Black and Ertel, 2011). It should be noted that neural networks work much faster on specific problems than conventional systems because of the massive number of interconnections (synapses) between the different neurons.

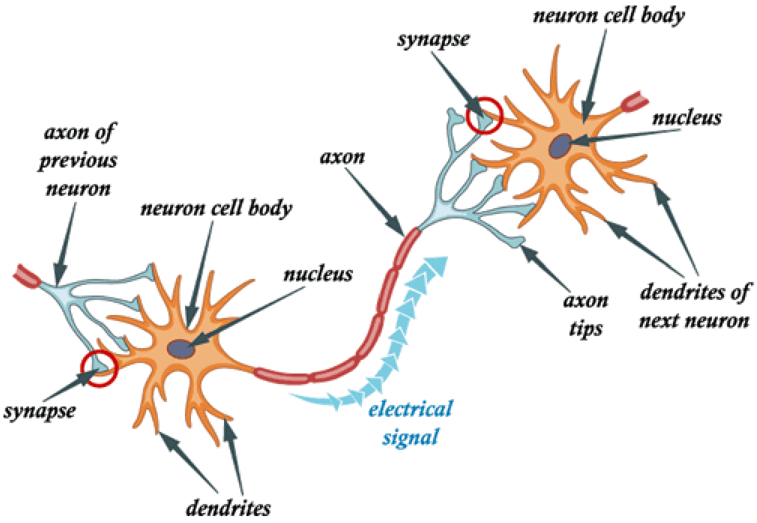


Figure 2.3: Schematic of a biological neuron

Plasticity in an adult brain might be described by two mechanisms: the creation of new synaptic connections between neurons and modification of existing neurons.

2.1.2 Model of an artificial neuron

The idea of developing an artificial neuron uses the same process. Let us considers the neuron as a unit that functions in combining the inputs that come in (X) and comparing the combined inputs with a specific activation function (θ) to determine an appropriate output, as shown in Fig.2.4.

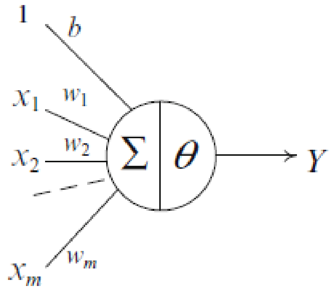


Figure 2.4: Schematic of an artificial neuron (Araghinejad, 2013)

Like synapses that control the magnitude of each single input, the inputs to an artificial neuron could be weighted by a weight matrix. For a mathematical reason, which is described in the next section, an artificial neuron usually benefits from an additional unit input with a weight known as bias. The mathematical relation of the functional process of an artificial neuron is defined as:

$$I = \mathbf{w} \times \mathbf{x} + b \tag{2.1}$$

$$Y = \begin{cases} 1 & if\, I \geq \theta \\ 0 & if\, I < \theta \end{cases} \tag{2.2}$$

where $\mathbf{x}$ is input vector, $\mathbf{w}$ is weight matrix, b is bias, I is sum of the weighted inputs, θ is threshold, and finally Y is output. The crude analogy between an artificial and biological neuron is that the connections between nodes represent the axons and dendrites, the connection weights represent the synapses, and the threshold approximates the activity in the soma (Jain et al., 1996).

2.1.3 Activation functions

The threshold value in the second half of a neuron could be replaced by a mathematical function, namely, the activation function to limit the amplitude of the output of a neuron. The activation function is also referred to as a squashing function in that it squashes the permissible amplitude range of the output signal to some finite value. Typically, the normalized amplitude range of the output of a neuron is written as the closed unit interval [0,1] or alternatively [-1,1] (Haykin, 2005). Here we identify the most common types of activation functions as illustrated in Fig.2.5. The Logistic Sigmoid function is by far the most common form of activation function. It takes the input I of any value and squashes the output a into the range of 0 to 1. A linear Transfer Function is normally used in the final (or output) layer.

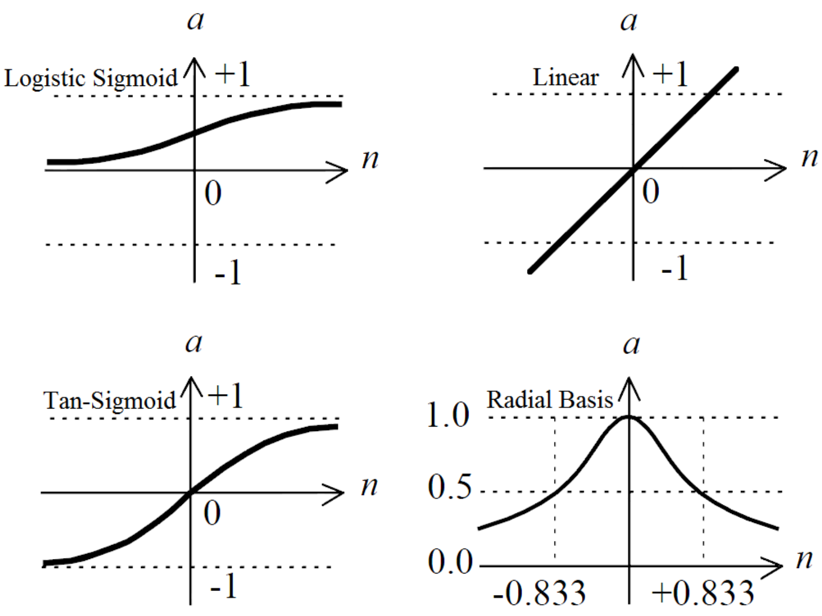


Figure 2.5: Four different forms of transfer functions

2.1.4 Network architectures

The way the neurons of a neural network are structured is closely linked with the learning algorithm used to train the network. In general, the three fundamentally different classes of network architectures may be identified as single layer feedforward, multilayer feedforward, and recurrent networks. The following provides a short description of each class of architecture.

2.1.4.1 Single-layer feedforward networks

In a layered neural network the neurons are organized in the form of layers. In the simplest form of layered network, we have an input layer of source nodes that projects onto an output layer of neurons (computation nodes), but not vice versa. In other words, this network is strictly a feedforward or acyclic type. It is illustrated in Fig.2.6 for the case of m nodes in the input layer and S_1 nodes in the output layer. Such a network is called a single-layer network, with the designation "single-layer" referring to the output layer of computation nodes (neurons). We do not count the input layer of source nodes because no computation is performed there (Haykin, 2005).

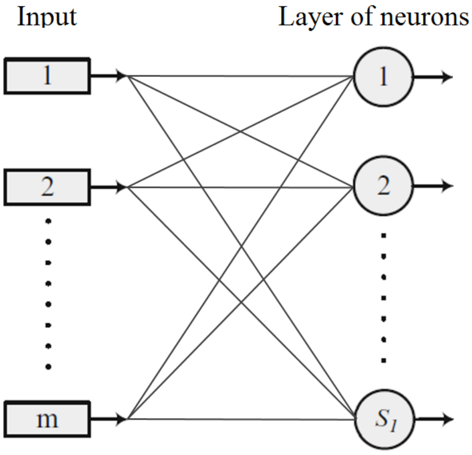


Figure 2.6: Feedforward or acyclic network with a single layer of neurons

2.1.4.2 Multilayer feedforward networks

The second class of a feedforward neural network distinguishes itself by the presence of one or more hidden layers, whose computation nodes are correspondingly called hidden neurons or hidden units. The function of hidden neurons is to intervene between the external input and the network output in some useful manner. By adding one or more hidden layers, the network is enabled to extract higher-order statistics. The ability of hidden neurons to extract higher-order statistics is particularly valuable when the size of the input layer is large (Haykin, 2005).

The architectural graph in Fig.2.7 illustrates the layout of a multilayer feedforward neural network for the case of a single hidden layer. The neural network in Fig.2.7 is said to be fully connected when every node in each layer of the network is connected to every other node in the adjacent forward layer (Haykin, 2005).

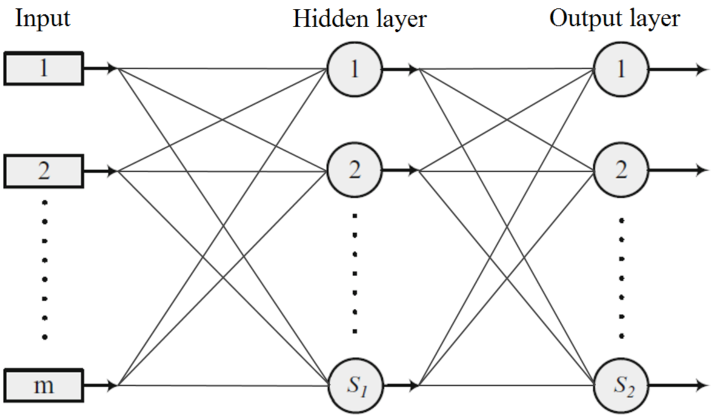


Figure 2.7: Multilayer feedforward artificial network with one hidden layer

2.1.4.3 Recurrent networks

A recurrent neural network can be distinguished from a feedforward neural network in that it has at least one feedback loop. There are different classes of recurrent neural networks. For example, a recurrent network may consist of a single layer of neurons with each neuron feeding its output signal back to the inputs of all neurons. Fig.2.8 illustrates a class of recurrent network called layer-recurrent network, which contains a feedback loop, with a single delay, around each layer of the network except for the last layer.

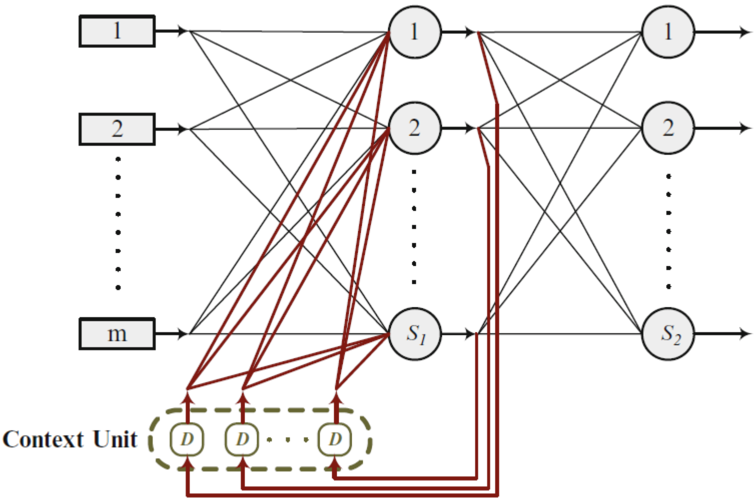


Figure 2.8: Schematic of a layer-recurrent network (Araghinejad, 2013)

The first version of this network was introduced by Elman (1990). The presence of feedback loops, whether in the recurrent structure of Fig.2.8 or that of other classes, has profound impact on the learning capability and performance of the network. Moreover, the feedback loops involve the use of particular branches composed of unit-delay elements, which result in a nonlinear dynamical behavior, assuming that the neural network contains nonlinear units (Haykin, 2005).

2.1.5 Learning rules of neural networks

To make statements over a specific task or environment, the configured ANN needs to have some knowledge of the task or environment. For generation of knowledge a so-called learning or training process needs to be passed (Fischler and Firschein, 1987). The weights and biases are parameters of a network that should be modified and assigned during this process, so that the network is able to evolve an appropriate behavior towards its application. This could be obtained by either supervised or unsupervised approaches. Training is an expression that is typically termed to the supervised approach for determining weights and biases of a network. In supervised learning, the learning rule is provided with a set of examples (the training set) of proper network behavior while in unsupervised learning, the weights and biases are modified in response to network inputs only. There are no target outputs available. The problem of learning an input-output mapping from a set of examples can also be understood as the minimization of a suitably defined error function. Although different definitions of the error have been used, for concreteness we consider the sum-of-square-differences error function defined as:

$$E(\mathbf{x}, \mathbf{w}) = \frac{1}{2} \sum_{p=1}^{p} \sum_{n=1}^{n} e_{pn}^2 = \frac{1}{2} \sum_{p=1}^{p} \sum_{n=1}^{n} [t_{pn} - o_{pn}(\mathbf{x}, \mathbf{w})]^2 \tag{2.3}$$

where $\mathbf{x}$ is input vector, $\mathbf{w}$ is weight matrix, t_{pn} and o_{pn} are the target and the correct output values for pattern p, respectively, and n is the number of output units. Training can be implemented in two different ways: incremental and batch mode. In the incremental mode, the weights and biases are updated after each input is applied to the network. In batch mode, this is done after all inputs have been presented to the network.

2.1.5.1 Back-Propagation learning rule

The Back-Propagation (BP) algorithm, also called the generalized delta rule, provides a way to calculate the gradient of the error function efficiently using the chain rule of differentiation (Bose and Liang 1996). In this algorithm, for each iteration the network weights are moved along the negative of the gradient of the performance function in the steepest descent direction. Normally, gradient $\mathbf{g}$ is defined as the first order derivative of total error function:

$$\mathbf{g} = \frac{\partial E(\mathbf{x}, \mathbf{w})}{\partial \mathbf{w}} \tag{2.4}$$

With definition of gradient $\mathbf{g}$, the update rule of the Back-Propagation algorithm could be written as:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta\, \mathbf{g}_k \tag{2.5}$$

where η is the learning constant and k is iteration.

2.1.5.2 Levenberg-Marquardt learning rule

The Levenberg-Marquardt algorithm (LM), which was developed by Kenneth Levenberg (1994) and Donald Marquardt (1963) is fast and has stable convergence. In the artificial neural networks field, this algorithm is suitable for training small and medium sized problems. This algorithm blends the Back-Propagation method and the Gauss-Newton algorithm. Although the LM algorithm tends to be a bit slower than Gauss-Newton algorithm, it converges much faster than the Back-Propagation method. In the LM method, a second order convergence speed can be achieved without computing the expensive Hessian Matrix.

$$\mathbf{H} = \mathbf{J}^T\mathbf{J} \tag{2.6}$$

The Jacobian matrix $\mathbf{J}$ contains the first derivatives of the network errors in respect to the weights and biases (Penz, 2013). The Jacobian matrix can be computed through a standard Back-Propagation technique. Compared to the computation of the Hessian matrix, a lot of computing time can be saved. To ensure the Hessian matrix is invertible, which is required for the applied Newton's method, a further approximation is added:

$$\mathbf{H} \approx \mathbf{J}^T\mathbf{J} + \mu\,\mathbf{I} \tag{2.7}$$

where μ is always positive, called combination coefficient and $\mathbf{I}$ is the identity matrix. Finally, the following formula can be applied to update the weights:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - (\mathbf{J}^T\mathbf{J} + \mu\,\mathbf{I})^{-1}\mathbf{J}^T\mathbf{e} \tag{2.8}$$

where $\mathbf{e}$ is error vector.

As the combination of the Back-Propagation and Gauss-Newton algorithms,the LM algorithm switches between the two algorithms during the training process. When the combination μ is very small, the equation approaches to the Gauss-Newton method. When combination coefficient μ is very large, the equation approximates to equal the Back-Propagation method (Yu & Wilamowski, 2011).

2.1.6 Design of ANNs

Fig.2.9 (Penz, 2013) illustrates the scheme for a general ANN. Five different parts can be identified as follows:

1. Input Pattern: presents the unprocessed data (set) to the network.
2. Input Layer: the input data are normally preprocessed in this section.
3. Hidden Layer: contains at least one neuron, but normally has more. This part is a black box, which receives inputs and after data processing provides an output signal.
4. Output Layer: counterpart of the Input Layer, where the data received from the hidden layer are post-processed.
5. Output Pattern: providing data to the environment.

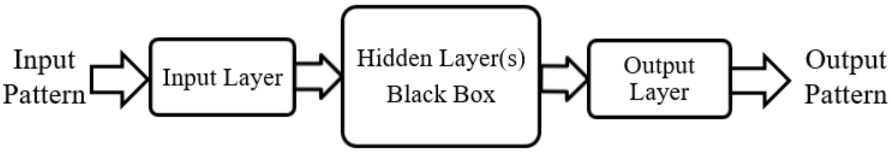


Figure 2.9: Scheme for a general ANN (Penz, 2013)

2.1.7 Generalization capability of ANNs

Generalization with respect to ANN is defined as the ability of the network to handle unseen patterns. However, ANNs may suffer from underfitting and overfitting during the training procedure, which tend to decrease the capability of the network in the generalization performance. There might be different reasons causing these two factors. For instance, a reason for overfitting might be a high number of parameters in the network compared to the samples in the training set (Lawrence & Giles, 2000). Good generalization can be obtained by going for a smaller network, which can fit the data well (Lawrence et al. 1998). Nevertheless, it is not easy to determine the smallest network, which best fits the data. The counterpart of overfitting is underfitting. This means that the network is not able to model a certain issue in a satisfactory way. It might be due to the low number of neurons for the present problem (Huber, 2014). Fig.2.10 is briefly showing how under-fitting, properly fitting, and over-fitting models fare on the training compared to the test sets. In order to find the perfect architecture for a problem, a trial and error process is required, although several methods are available to get an idea of the number of neurons.

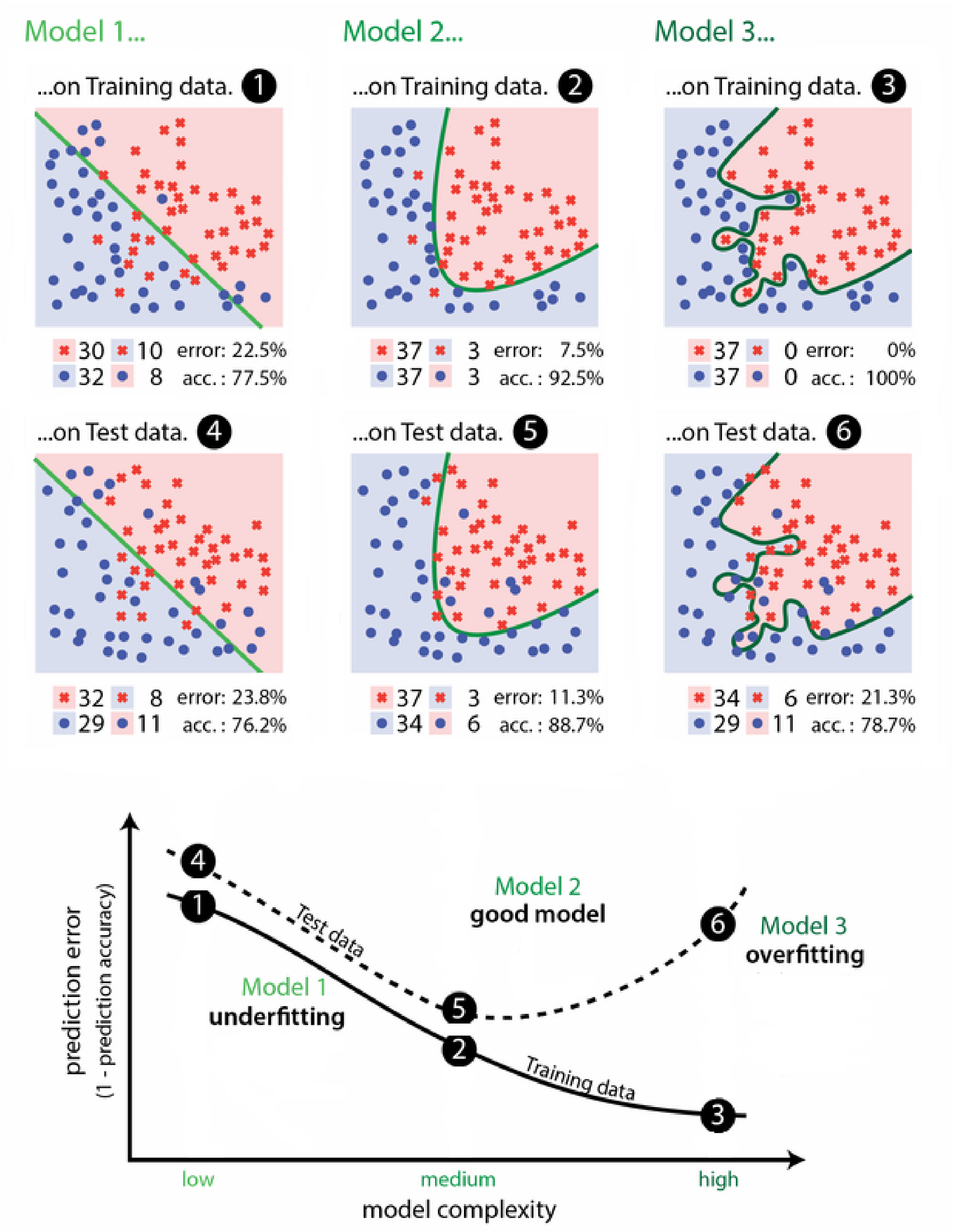


Figure 2.10: Overfitting, underfitting, and generalization

Another reason for underfitting and overfitting can be inappropriate number of epochs or iterations. Increasing the number of epochs in the training procedure results in decreasing the underfitting of the network, but if the number of epochs is greater than a specific number, overfitting may occur. In this case, the technique of early stopping improves the generalization of the network. It implements a feature that stops the training at an early stage to avoid overfitting. To achieve this, the data set is divided into three subsets, the training- , validation-, and the test data set. The one mentioned first is used to adjust the weights and biases, which get fixed in the validation data set. Based on this, two curves with the number of epochs versus the observed error of the data sets are plotted, as shown in Fig.2.11. The two curves decrease until the point, where the validation curve starts to increase is reached. At this point the training stops. The weights and biases are now returned to the values of the epoch with the minimum validation error because the network essentially just learned the noise contained in the training data (Penz, 2013).

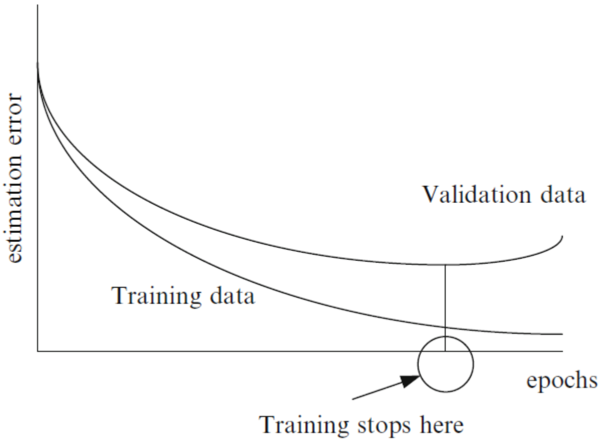


Figure 2.11: Selection of optimum epoch based on the network performance in data training and testing (Araghinejad, 2013)

The training curve usually keeps decreasing with an increasing number of iterations. The utilized method is a so-called cross validation technique. The training stops if the number of validation checks, representing the number of successive iterations that the validation performance fails to decrease, reaches a certain value (Beale et al., 2013).

2.2 Adaptive Neuro-Fuzzy Inference System

System modelling based on conventional mathematical tools is not well suited for dealing with ill-defined and uncertain systems. By contrast, a fuzzy inference system employing fuzzy if-then rules can model the qualitative aspects of human knowledge and reasoning processes without employing precise quantitative analyses. This general idea is very suitable for engineering applications where a precise representation of the real world is desired. The idea of fuzzy modelling, first explored systematically by Takagi and Sugeno (1985), has found numerous practical applications in control (Pedrycz, 1993), prediction and inference (Kandel, 1991), and its application in the engineering field has significantly increased in recent years.

Adaptive Neuro-Fuzzy Inference System, or simply ANFIS, can serve as a basis for constructing a set of fuzzy if-then rules with appropriate membership functions able to generate the stipulated input-output pairs.

2.2.1 Fuzzy logic

Lotfi Zadeh introduced the Fuzzy Logic tool in 1965. It is a mathematical tool for dealing with uncertainty and offers to a soft computing partnership the important concept of computing with words. It implements a technique to deal with imprecision and information granularity by providing a mechanism for representing linguistic constructs, such as “many”, “low”, “medium”, “often”, “few”. In general, the fuzzy logic provides an inference structure that helps human reasoning. While, the traditional binary set theory describes crisp events, events that either do or do not occur, the fuzzy logic uses probability theory to explain if an event will occur and to measure the chance of its occurrence. The theory of fuzzy logic starts with the concept of a fuzzy set and is based upon the notion of relative graded membership. (Sivanandam, 2007).

2.2.2 Classical sets and fuzzy sets

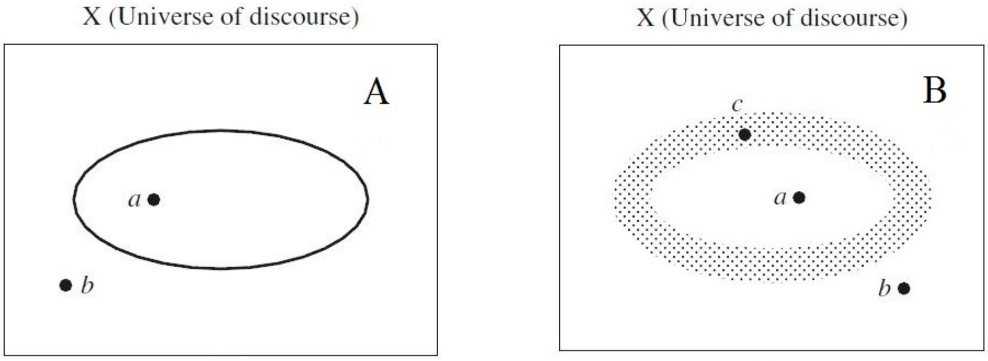
Traditional system modeling and analysis techniques are too precise for real world problems, whose complexity involves a degree of uncertainty. In a classical set, the membership value is “1” if it belongs to the set or “0” if it is not a member of the set. Thus, membership in a set is found to be binary i.e., the element is a member of a set or not. It can be indicated as:

$$X_A(x) = \begin{cases} 1 & , x \in A \\ 0 & , x \notin A \end{cases} \tag{2.9}$$

where $X_A(x)$ is the membership of element x in set A and A is the entire set on the universe.

Zadeh extended this membership to possess various “degrees of membership” on the real continuous interval [0,1]. He formed fuzzy sets as the sets on the universe X that can accommodate "degrees of membership". In other words, the concept of fuzzy set contrasts with a classical concept of a crisp set, whose boundary is required to be precise (Sivanandam, 2007).

Fig.2.12 helps to explain this idea, but from a two-dimensional perspective. Point *a* in Fig.2.12a is clearly a member of crisp set A; point *b* is unambiguously not a member of set A. Fig.2.12b shows the vague, ambiguous boundary of a fuzzy set B on the same universe X: the shaded boundary represents the boundary region of B. In the central (unshaded) region of the fuzzy set, point *a* is clearly a full member of the set. Outside the boundary region of the fuzzy set, point *b* is clearly not a member of the fuzzy set.



(a) (b)

Figure 2.12: (Diagrams for (a) crisp set boundary and (b) fuzzy set boundary (Ross, 2009)

However, the membership of point *c*, which is on the boundary region, is ambiguous. If complete membership in a set is represented by the number 1, and no-membership in a set is represented by 0, then point *c* in Fig.2.12b must have some intermediate value of membership (partial membership in fuzzy set B) on the interval [0,1]; say about 0.7 (Ross, 2009). A key difference between crisp and fuzzy sets is their membership function; a crisp set has a unique membership function, whereas a fuzzy set can have an infinite number of membership functions to represent it (Ross, 2004). A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1 (The Math Works, 2013). Fig.2.13 illustrates most common membership functions used in neuro-fuzzy systems.

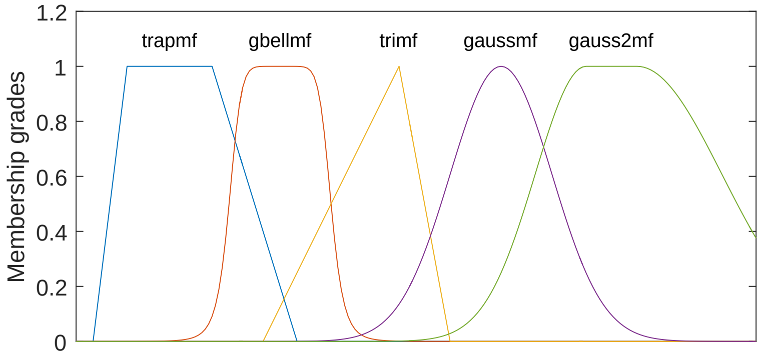


Figure 2.13: Membership functions

2.2.3 Logical operations

To combine two fuzzy numbers, AND, OR, and NOT operators exist in fuzzy logic, usually defined as the minimum, maximum, and complement. It means that the membership function of the output variable, in the case of combining two fuzzy numbers would be the maximum/minimum membership function of them in case of using AND/OR operators. In case of using the NOT operator, the membership function of the output variable would be the complement of the single input variable (Araghinejad, 2013). Fig.2.14 shows an example of combining two fuzzy variables by different operators.

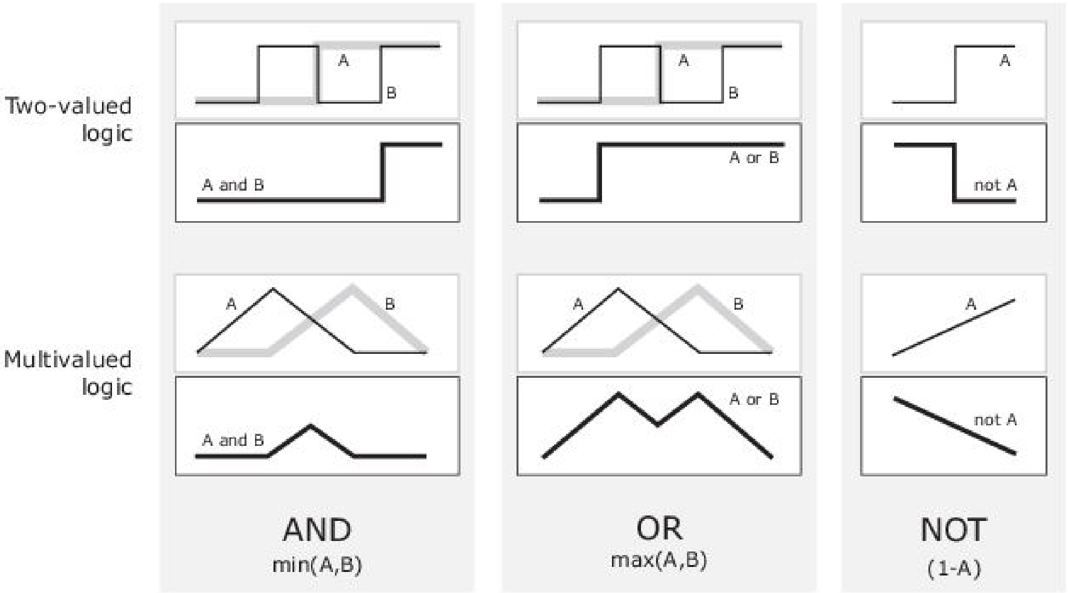


Figure 2.14: Logical operations (The Math Works, 2013)

2.2.4 Fuzzy if-then rules

Fuzzy if-then rules are expressions of the form ***IF A THEN B***, where ***A*** and ***B*** are labels of fuzzy sets characterized by appropriate membership functions. Due to their concise form, fuzzy if-then rules are often employed to capture the imprecise modes of reasoning that play an essential role in the human ability to make decisions in an environment of uncertainty and imprecision. A simple example is:

If pressure is high, then volume is small

where *pressure* and *volume* are linguistic variables, *high* and *small* are linguistic values or labels that are characterized by membership functions.
Another form of fuzzy if-then rule, proposed by Takagi and Sugeno, has fuzzy sets involved only in the premise part. By using Takagi and Sugeno's fuzzy if-then rule, we can describe the resistant force on a moving object as follows:

If velocity is high, then force $= k{*}(velocity)^2$

where, again, *high* in the premise part is a linguistic label characterized by an appropriate membership function. However, the consequent part is described by a non-fuzzy equation of the input variable, *velocity*(Jang, 1993).

2.2.5 Fuzzy Inference System

Fuzzy inference system (FIS) can deal with both linguistic and quantitative variables in the process of modeling. In contrast to the conventional data-driven methods, which try to find a logical relationship between input and output variables from the observed data, FIS benefits from both the concept of the problem and the information within the observed data. According to Jang (1993), a fuzzy inference system is composed of five functional blocks (see Fig.2.15):

1. a rule base containing a number of fuzzy if-then rules;

2. a database defining the membership functions of the fuzzy sets used in the fuzzy rules;

3. a decision-making unit performing the inference operations on the rules;

4. a fuzzification interface transforming the crisp inputs into degrees of match with linguistic values;

5. a defuzzification interface transforming the fuzzy results of the inference into a crisp output;

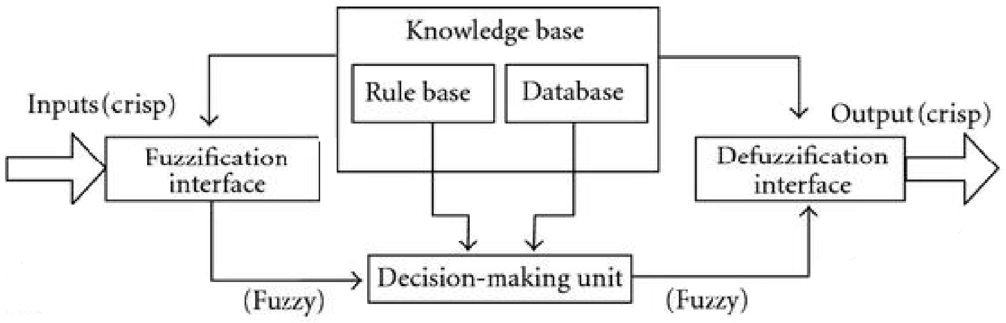


Figure 2.15: Fuzzy inference system (Jang, 1993)

Several types of fuzzy reasoning have been proposed in the literature. Depending on the types of fuzzy reasoning and fuzzy if-then rules employed, most fuzzy inference systems can be classified into three types (Fig.2.16):

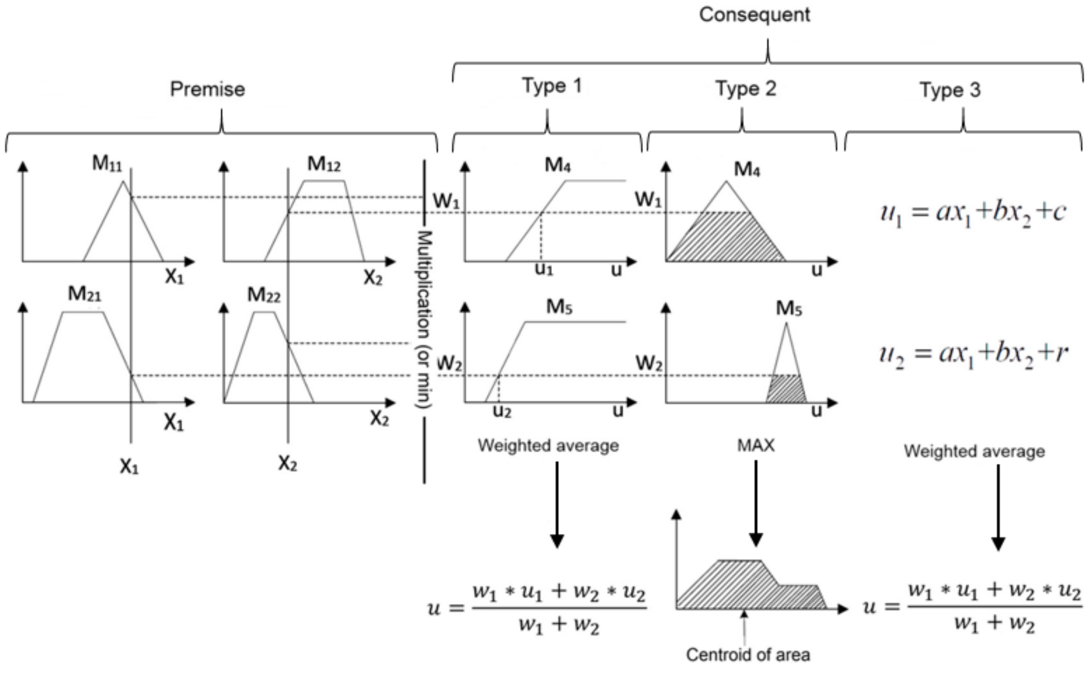


Figure 2.16: Commonly used fuzzy if-then rules and fuzzy reasoning mechanisms (modified from Jang, 1993)

Type 1: The overall output is the weighted average of each rule's crisp output induced by the rule's firing strength (the product or minimum of the degrees of match with the premise part) and output membership functions.

Type 2: The overall fuzzy output is derived by applying the "max" operation to the qualified fuzzy outputs (each of which is equal to the minimum of firing strength and the output membership function of each rule). Various schemes have been proposed to choose the final crisp output based on the overall fuzzy output. Some of them are centroid of area, bisector of area, mean of maxima, maximum criterion, etc.

Type 3: Takagi and Sugeno's fuzzy if-then rules are used. The output of each rule is a linear combination of input variables plus a constant term, and the final output is the weighted average of each rule's output.

2.2.6 ANFIS architecture

ANFIS is a kind of neural network that is based on Sugeno fuzzy inference system. Since it integrates both neural networks and fuzzy logic principles, it has potential to capture the benefits of both in a single framework. Its inference system corresponds to a set of fuzzy if-then rules that have learning capabilities to approximate nonlinear functions. Its network structure consists of a number of nodes connected through directional links. Each node is characterized by a node function with fixed or adjustable parameters. A basic ANFIS introduced by Jang (1993) is illustrated in Fig.2.17.

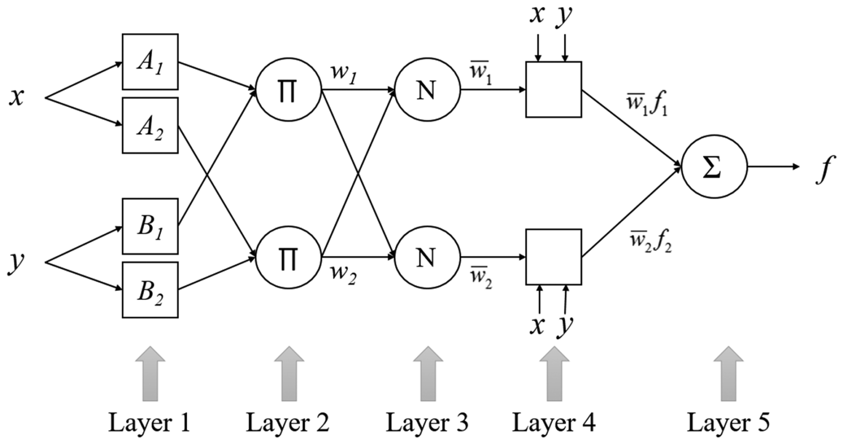


Figure 2.17: ANFIS architecture (modified from Jang, 1993)

For simplicity, we consider the fuzzy inference system having two inputs x and y and one output f. For a first-order Sugeno fuzzy model, a typical rule set with two fuzzy if-then rules can be expressed as:

Rule 1: If x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$

Rule 2: If x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$

Each nodes of the same layer have similar functions, as described below:

Layer 1: Each node in this layer produces membership grades of an input variable. The output of the i-th node in layer 1 is denoted as O_i^1. Assuming a generalized bell function as the membership function, the output O_i^1 can be computed as:

$$O_i^1 = \mu_{Ai}\left(x\right) = \frac{1}{1+\left(\left(x-c_i\right)/a_i\right)^{2N_i}} = \begin{cases} \mu_{Ai}\left(x\right)\ ; & i=1,2 \\ \mu_{Bi-2}\left(x\right)\ ; & i=3,4 \end{cases} \tag{2.10}$$

where x is input to node i and $\{a_i,\, c_i,\, N_i\}$are adaptable variables known as premise parameters. The outputs of this layer are the membership values of the premise part.

Layer 2: Every node in this layer multiplies the incoming signals:

$$O_i^2 = w_i = \mu_{Ai}\left(x\right) \times \mu_{Bi}\left(y\right), \qquad i = 1\,,\,2 \tag{2.11}$$

Layer 3: The i-th node of this layer calculates the normalized firing strengths as:

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1+w_2}, \qquad i = 1\,,\,2 \tag{2.12}$$

Layer 4: Node i in this layer calculates the contribution of the ith rule towards the model output, with the following node function:

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i\left(p_i x + q_i y + r_i\right) \tag{2.13}$$

where $\bar{w}$ is the output of layer 3, and $\{p_i,\, q_i,\, r_i\}$are the parameter set. Parameters of this layer are referred to as consequence parameters.

Layer 5: The single node in this layer calculates the overall output of the ANFIS as (Jang & Sun, 1995):

$$O_1^5 = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} = f \tag{2.14}$$

2.2.7 Learning rules of ANFIS

A nonlinear neuro-fuzzy model can be generally expressed as:

$$f = g\left(\mathbf{z}, \theta\right) \tag{2.15}$$

where $\mathbf{z}$ is the input vector, $\theta = [\theta_1, \ldots, \theta_n]$ is the parameter vector, and f is the model's (scalar) output. Given a set of training data $\{\left(\mathbf{z}_p; t_p\right), p = 1, \ldots, m\}$, a squared error measure takes the form:

$$E(\theta) = \sum_{p=1}^{m} \left[t_p - g\left(\mathbf{z}_p, \theta\right)\right]^2 \tag{2.16}$$

which is the objective function to be minimized.
From the ANFIS architecture (see Fig.2.17), it is observed that given the values of premise parameters, the overall output can be expressed as a linear combinations of the consequent parameters. More precisely, the output f in Fig.2.17 can be written as:

$$\begin{aligned} f &= \tfrac{w_1}{w_1+w_2} f_1 + \tfrac{w_2}{w_1+w_2} f_2 = \bar{w}_1 f_1 + \bar{w}_2 f_2 \\ &= (\bar{w}_1 x)\, p_1 + (\bar{w}_1 y)\, q_1 + (\bar{w}_1)\, r_1 + (\bar{w}_2 x)\, p_2 + (\bar{w}_2 y)\, q_2 + (\bar{w}_2)\, r_2 \end{aligned} \tag{2.17}$$

which is linear in the consequent parameters (p_1, q_1, r_1, p_2, q_2, and r_2). As a result, we have $\boldsymbol{S}$ = set of total parameters, $\boldsymbol{S_1}$ = set of premise parameters, and $\boldsymbol{S_2}$ = set of consequent parameters.
For the ANFIS architecture, the task of the learning algorithm is to adjust all the adaptable parameters $\boldsymbol{S_1}$ and $\boldsymbol{S}\ _{\mathbf{2}}$. There are two well-known methods to update the parameters: gradient descent or Back-Propagation and hybrid-learning rule.

2.2.7.1 Basic Back-Propagation learning rule

The actual output of the model is computed directly from Eq.(2.14) when using the basic Backpropagation learning algorithm, in the forward pass, for a given input pattern and initial values of all parameter set $\boldsymbol{S}$. In the backward pass, the error signal resulting from the difference between the actual output and the desired output of the model is propagated backward where these parameters are then adjusted using the error correction rule. Training of network, i.e. error correction, is stopped when the value of the error function E has become sufficiently small below a certain threshold. One of the most important problems in learning is the prevention of over fitting. This issue can be addressed by observing the error index of testing data during the learning iterations. The learning algorithm will be terminated when the error index of the testing data starts to increase on average. Prevention of over fitting is the most common way to provide high generalization.

The error function describes the error when approximating or classifying the training data as a function of the parameters of the activation functions. To find the minimum of this function, we are using the Gradient descent method, where the parameters are updated on pattern by pattern basis, until the complete set of training data was utilized for the training of the network. The parameter update is equal to the slope of the error function with an opposite sign of the gradient of error. The update rate is further scaled by a learning rate η, which controls the speed at which we do the error correction or decides for the rate at the network learns. The parameter θ of the set of total parameters $\boldsymbol{S}$ is updated using the following equation:

$$\theta^{k+1} = \theta^{k} + \eta\,(\Delta\theta)^{k}; \quad (\Delta\theta)^{k} = -\left(\frac{\partial E}{\partial \theta}\right)^{k} \tag{2.18}$$

As the basic Back-Propagation algorithm depends on the gradient of the error function, the adjustments for flat spots in the error surface may be very small. Therefore, this results in slow and time consuming progression in error minimization. The speed of convergence can be improved by increasing the learning rate η. However, with a large value of η, the learning algorithm can overshoot the error minimum leading to oscillations of the weight values between relatively poor solutions.

2.2.7.2 Basic hybrid learning rule

The most well-known learning algorithm for ANFIS is hybrid algorithm, where:

- In the forward pass, node outputs values go forward until layer 4 and the consequent parameters belonging to the subset $\boldsymbol{S_2}$ are identified by the least squares method.

- In the backward pass, the output errors are propagated backward and the premise parameters are updated by gradient descent method mentioned above, with θ being the parameter of the set of premise parameters $\boldsymbol{S_1}$. Table.2.1 summarizes the activities in each pass.

Using this method, the consequent parameters identified are optimal under the condition that the premise parameters are fixed. Generally, the hybrid approach converges faster than the basic Back-Propagation approach, since it reduces the dimension of the search space of the original Back-Propagation method.

Table 2.1: Two passes in the hybrid learning procedure for ANFIS (Jang, 1993)

	Forward pass	Backward pass
Premise parameters	Fixed	Gradient descent
Consequent parameters	Least Square Estimate	Fixed

2.2.8 Alternative learning rules

According to Jang and Mizutani (1996), ANFIS is a network architecture that allows systematic calculations of gradient vectors (derivatives of output error with respect to modifiable parameters), so the analysis is not limited to Back-Propagation or Hybrid learning algorithms only. In fact, any gradient-based techniques in nonlinear regression and optimization can be applied, such as the Gauss-Newton method, the Levenberg-Marquardt algorithm, and the extended Kalman filter algorithm.

On the other hand, the MATLAB toolbox only provides a graphical user interface for ANFIS models trained with the Hybrid and Back-Propagation (BP) algorithms. This fact hold significance because these models have been widely used by researchers in the field of water resources and environmental engineering. Therefore, the current research analyzes the application of alternative learning algorithms for ANFIS applied to hydrological issues. To achieve this aim, the FORTRAN programming language is applied to build an ANFIS model using alternative learning rules. The following sections clarify these algorithms and their training process, which are applied to adjust the premise and consequence parameters of ANFIS networks.

2.2.8.1 Levenberg-Marquardt learning rule

Before introducing the Levenberg-Marquardt algorithm for minimizing Eq.(2.16), the closely related the Gauss-Newton method is reviewed. The Gauss-Newton method, also known as the linearization method, uses a Taylor series expansion to obtain a linear model that approximates the original nonlinear model and then applies the ordinary least-squares method to estimate the parameters. Specifically, let the current parameters be denoted by θ^k; then the nonlinear model in Eq.(2.15) can be expanded in a Taylor series around $\theta = \theta^k$and only the linear terms are retained:

$$f = g\left(\mathbf{z}, \theta^k\right) + \sum_{i=1}^{n} \left(\frac{\partial g\left(\mathbf{z}, \theta\right)}{\partial \theta_i} |_{\theta=\theta^k} \right) \left(\theta_i - \theta_i^k\right) \tag{2.19}$$

Inspection of Eq.(2.19) reveals that the translated output $f - g\left(\mathbf{z}, \theta^k\right)$ is linear function of the translated parameters, $\theta_i - \theta_i^k$. Therefore, a better estimator, θ^{k+1}, can be obtained by means of the well-known pseudo-inverse formula:

$$\theta^{k+1} = \theta^k + \left(\mathbf{J}^T\mathbf{J}\right)^{-1} \mathbf{J}^T\mathbf{e} = \theta^k + \Delta\theta \tag{2.20}$$

where $\mathbf{e}$ is the error vector of which the pth element is equal to $t_p - g\left(\mathbf{z}_p, \theta^k\right)$, $\Delta\theta$ is, $\left(\mathbf{J}^T\mathbf{J}\right)^{-1} \mathbf{J}^T\mathbf{e}$ and the element at row p and column j of matrix $\mathbf{J}$ is $\frac{\partial g\left(\mathbf{z}_p, \theta\right)}{\partial \theta_j} |_{\theta=\theta^k}$.
A potential problem with the Gauss-Newton method is that $\left(\mathbf{J}^T\mathbf{J}\right)^{-1}$might not always exist, rendering this method practically unusable. Such a situation is handled by the

Levenberg-Marquardt procedure, which defines $\Delta\theta$ as follows:

$$\Delta\theta = (\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})^{-1}\mathbf{J}^T\mathbf{e} \tag{2.21}$$

where **I**, is the identity matrix and λ is usually small positive constant. Depending on the magnitude of λ, the algorithm transits smoothly between two extremes: the Gauss-Newton method ($\lambda \to 0$) and gradient descent method ($\lambda \to \infty$). Usually the Gauss-Newton method is more efficient but less stable; the gradient descent method is more stable but less efficient. By properly setting the value of λ, the Levenberg-Marquardt algorithm can be efficient as well as stable.
Fig. 2.18 shows the flowchart of the ANFIS program used in this research. As can be seen, three sub sets of data are generated to build the model.

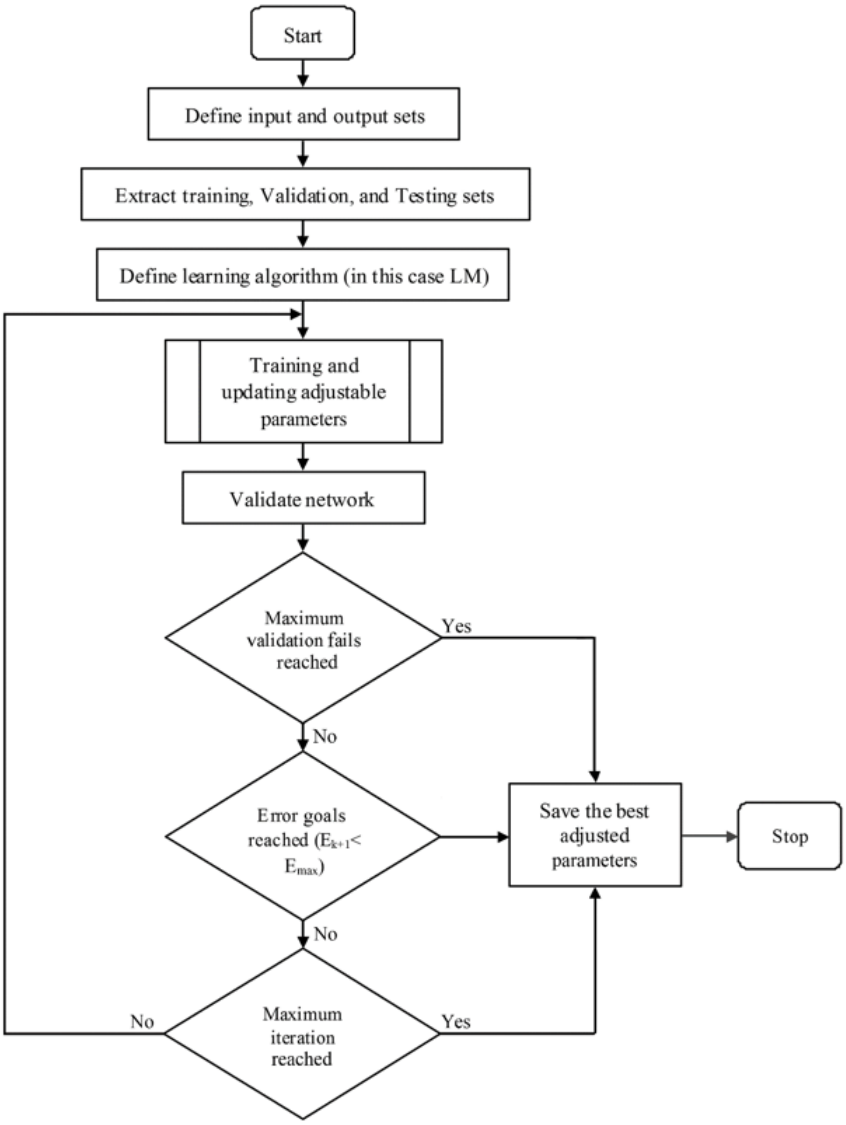

Start
Define input and output sets
Extract training, Validation, and Testing sets
Define learning algorithm (in this case LM)
Training and updating adjustable parameters
Validate network
Maximum validation fails reached
Yes
No
Error goals reached (E_{k+1}< E_{max})
No
Save the best adjusted parameters
Stop
No
Maximum iteration reached
Yes

Figure 2.18: Flowchart of the ANFIS program (Kaveh et al. 2017)

The first one is the training set, used to update the adjustable parameters. With the update rule of the Levenberg-Marquardt algorithm (Eq. 2.21) and the computation of matrix $\mathbf{J}$, the next step is to organize the training process. The aim of the training (learning) process is to achieve the best possible performance of the ANFIS network by adjusting the premise and consequence parameters.

The diagram of the training subroutine is shown in Fig.2.19. As can be observed from this diagram, if the error reduces, which means it is smaller than the error for the previous iteration, the quadratic approximation of the total error function is working and the combination coefficient, λ, could be reduced to decrease the influence of the gradient descent part (ready to speed up). On the other hand, if the error increases, which means it's greater than the error for the previous iteration, it's necessary to follow the gradient more closely to look for a proper curvature for quadratic approximation and the combination coefficient λ is increased.

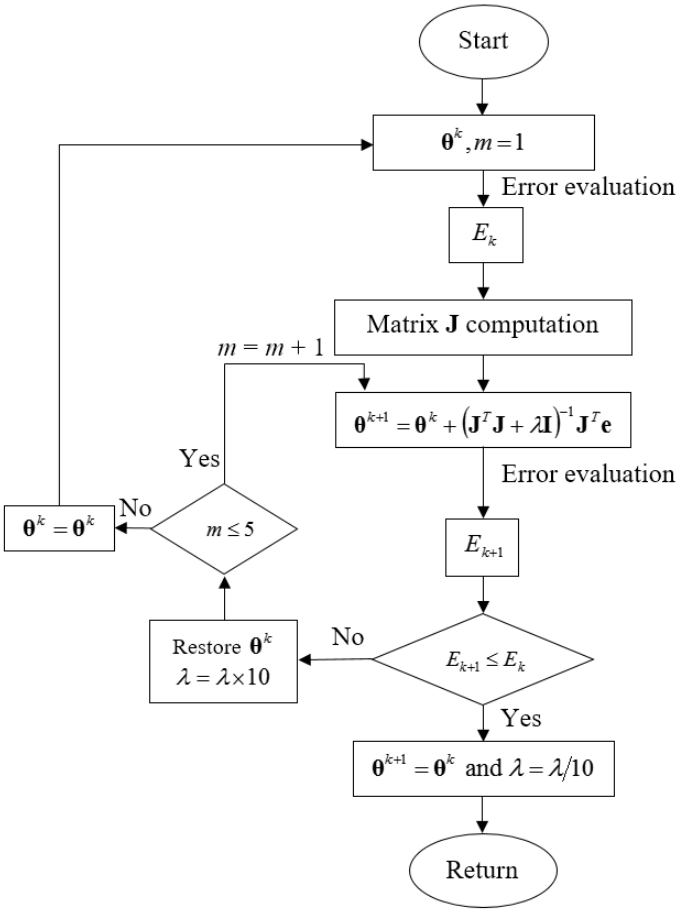

Start
$\boldsymbol{\theta}^k, m = 1$
Error evaluation
E_k
Matrix $\mathbf{J}$ computation
$m = m + 1$
$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + (\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})^{-1}\mathbf{J}^T\mathbf{e}$
Yes
Error evaluation
No
$\boldsymbol{\theta}^k = \boldsymbol{\theta}^k$
$m \leq 5$
E_{k+1}
Restore $\boldsymbol{\theta}^k$
$\lambda = \lambda \times 10$
No
$E_{k+1} \leq E_k$
Yes
$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k$ and $\lambda = \lambda/10$
Return

Figure 2.19: Diagram for training using the Levenberg-Marquardt algorithm: E_{k+1} is the current total error, and E_k is the previous total error (modified from Kaveh et al. 2017)

The training process using the Levenberg-Marquardt algorithm could be designed as follows:

1. With the initial parameters, evaluate the total error (SSE).

2. Update the parameters as directed by Eq.(2.21).

3. With the new parameters, evaluate the total error (SSE).

4. If the current total error increased as a result of the update, then retract the step and increase combination coefficient, λ, by a factor of 10 or by some other factor. Then go to the second step and try parameter update again.

5. If the current total error decreased because of the update, then accept the step and decrease the combination coefficient, λ, by a factor of 10 or by the same factor as step 4.

6. Go to the second step, with the new parameters until the current total error is smaller than the required value (E_{max}).

During the training process, the error of the validation set is monitored and normally decreases during the initial phase of training. In contrast, the error of the validation set increases when the network begins to over-fit the data. At the minimum of the validation set error the premise and consequence parameters of the ANFIS are selected and saved. The last subset is used to test the developed ANFIS. Besides the stopping criteria (called early stopping criteria), several other conditions to stop the training are implemented. In this paper, the maximum number of epochs is set to 1000, while the minimum performance goal (or maximum accepted error, E_{max}) is set to zero. The maximum combination coefficient, λ, of the Levenberg-Marquardt algorithm is set to 1×10^{10}. The training stops when one of these values is reached (Kaveh et al. 2017).

2.2.8.2 Back-Propagation with momentum learning rule

Another way of improving the rate of convergence is the inclusion of a momentum to the gradient expression. Therefore, a part of the previous parameter change is added to the current parameter change. Due to the momentum, the Back-Propagation can respond to recent trends in the error surface. This helps to smoothen the gradient descent path by ignoring local anomalies in the error surface. Hence, prevention against extreme changes in the gradient is added to the algorithm. The parameter θ is now updated using the following equation:

$$\theta^{k+1} = \theta^{k} + \eta\,(\Delta\theta)^{k} + \mu\,(\Delta\theta)^{k-1} \tag{2.22}$$

where μ is the momentum coefficient. If the direction of the gradient remains constant, the algorithm will take increasingly large steps. This improves the convergence rate. Also in a flat plateau, the momentum can decelerate the decrease (due to the flat gradient) in weight adjustment. Another aspect is that the momentum can help to escape from local minima of the error function. This is useful when the learning rate η alone is not able to generate large enough steps to overcome the ridge of the minima “valley”. Of course, if the learning starts in deep local minima (due to the random weight initialization) the momentum does not help to find the way out. However, in general the momentum increases the chances of the Back-Propagation in finding the global minima of the error surface (Bui et al. 2017).

2.2.8.3 New proposed hybrid learning rules

Differently from the common hybrid learning rule which combines the gradient method and the least squares estimate (LSE) to identify parameters, the two new proposed hybrid rules are combination of LM and LSE methods (called New Hybrid 1), as well as of LM and BP methods (called New Hybrid 2). Similarly, each epoch of these hybrid learning procedures is composed of a forward and backward pass.
For the Hybrid 1 rule, we supply input data and functional signals go forward until the forth layer, and the parameters in $\boldsymbol{S_2}$ are identified by the sequential least squares estimate. After identifying parameters in $\boldsymbol{S_2}$, the functional signals keep going forward until the error measure is calculated. In the backward pass, error rate propagate from the output end toward the input end, and the parameters in $\boldsymbol{S_1}$ are updated by the LM method. Fig.2.20 shows the flowchart of this new proposed hybrid rule combining the LSE and LM methods.

The main difference between the second proposed hybrid rule and the common hybrid one is that both premise and consequent parameters are updated in the backward pass. In the forward pass, we supply initial premise and consequent parameters, input data and functional signals go forward to calculate output in layer five. In the backward pass, error rate propagates from the output end toward the input end, and the parameters in $\boldsymbol{S_2}$ and $\boldsymbol{S_1}$ are updated by the gradient descent Back-Propagation and the LM method, respectively. The advantage of this proposed method compared

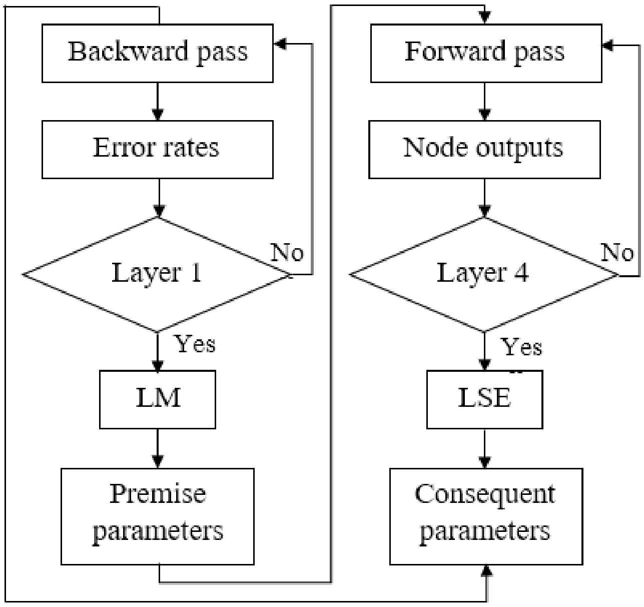


Figure 2.20: Flowchart of the first new hybrid learning procedure of ANFIS (Kaveh et al. 2015a; 2015b)

to LM is that it can significantly decrease the CPU run time, since the calculation of huge Jacobian matrix for consequent parameters is not required (Kaveh et al. 2015a; 2015b).

Chapter 3

Conventional hydro-morphodynamic modeling

The transport of sediment as bed load is an important process that occurs in rivers, estuaries, and coastal regions. In many situations, this process and the resulting morphological changes of the bed can have a detrimental impact on the infrastructure and environment. Clearly, the processes of sediment transport and morphological evolution of the bed are determined by the properties of the fluid flow, which in turn are affected by the changes in the morphology of the bed that they induce. Thus, the motion of the fluid and the motion of the bed form an interdependent two-phase phenomenon that must be analyzed using a model system made up of two distinct but interdependent model components: (1) a hydrodynamic component defining the evolution of the flow; and (2) sediment transport/morphological component defining the evolution of the bed. Such a modeling system is often referred to as a hydromorphological model. The conventional method for performing hydromorphological simulations in rivers is to decouple the hydrodynamic and the morphodynamic systems. The decoupling approach is based on the rationale that the channel bed reacts at a much slower timescale than the flow. At implementation level, these modules communicate through a quasi-steady morphodynamic time-stepping mechanism: during flow computation, the bed level is assumed to be constant and during computation of the bed level, the flow and sediment transport quantities are assumed invariant to the bed level changes. The modules are linked together at the programming level. The hydrodynamic system is usually modeled by Navier-Stokes equations (NSE). The morphodynamic component is modeled by Exner equation.

3.1 The Navier-Stokes equations

The foundation of nearly all hydrodynamic calculations are a set of equations called Navier-Stokes Equations. To describe the governing laws of hydrodynamics the development of the main equations is given in the following.

***Equation of continuity*:** The equation of continuity is based on the law of conservation of mass, which states that mass can neither be created nor destroyed. The continuity equation can be written in either differential or integral form. In differential form, consider the infinitesimal control volume in Fig.3.1.

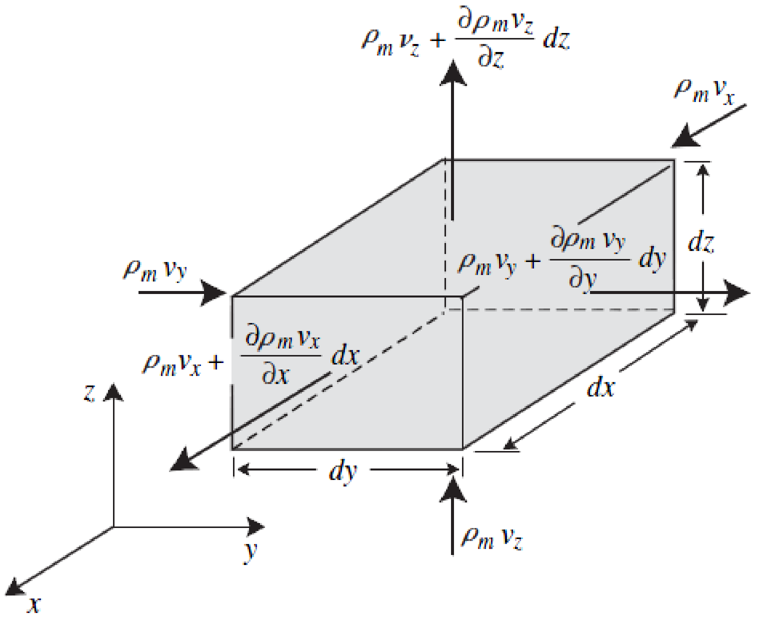

$\rho_m v_z + \frac{\partial \rho_m v_z}{\partial z} dz$
$\rho_m v_x$
$\rho_m v_y$
$\rho_m v_y + \frac{\partial \rho_m v_y}{\partial y} dy$
dz
$\rho_m v_x + \frac{\partial \rho_m v_x}{\partial x} dx$
z
dx
dy
$\rho_m v_z$
y
x

Figure 3.1: Infinitesimal element of fluid

The difference between the mass fluxes entering and leaving the differential control volume equals the rate of increase of internal mass. The assumption of a continuous fluid medium yields the following differential relationship:

$$\frac{\partial \rho_m}{\partial t} + \frac{\partial}{\partial x}(\rho_m v_x) + \frac{\partial}{\partial y}(\rho_m v_y) + \frac{\partial}{\partial z}(\rho_m v_z) = 0 \qquad (3.1)$$

For homogeneous incompressible fluid, the mass density is independent of space and time (ρ_m = constant); consequently, $\partial \rho_m / \partial t = 0$ and the divergence of the velocity vector must be zero:

$$\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} = 0 \qquad (3.2)$$

***Momentum equations*:** Using the law of conservation of momentum, which considers the forces acting on a domain, the momentum equations can be written in its non-conservative form as:

$$\rho_m \frac{\partial v_i}{\partial t} + \rho_m v_j \frac{\partial v_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + f_i \tag{3.3}$$

where v is three-dimensional flow velocity vector, p is pressure, τ_{ij} is three dimensional shear stress vector, and f_i is forces.
Given an inviscid, Newtonian fluid the constitutional form of the shear stress τ_{ij} is described in Eq.(3.4) with the dynamic viscosity μ (kg/sm^2), (Laurien & Oertel jr. 2013).

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{3.4}$$

Finally combining both equations results in the momentum equation for incompressible fluids:

$$\rho_m \frac{\partial u_i}{\partial t} + \rho_m u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \mu \frac{\partial^2 u_i}{\partial x_j^2} + f_i \tag{3.5}$$

***Conservation of energy*:** Conservation of energy is neglected, because typical no relevant temperature gradients in free surface application occurs. The interested reader is referred to (Hervouet 2007), (Chanson 2004) and (Laurien & Oertel jr. 2013). To sum up, the equations of continuity Eq.(3.2) and momentum Eq.(3.5) states the Navier-Stokes equations for incompressible fluids in three dimensions. An analytical solution of this equation set is only possible in certain well-defined cases. So far only numerical solutions exist, which are quite time consuming. In a so-called direct numerical simulation (DNS) the complete temporal and spatial spectrum of the flow is solved, but requires enormous computational effort. Practical it is not necessary to resolve the complete spectrum to gain a satisfactory result. Rather the influence of turbulence is treated separately from the flow, (Laurien & Oertel jr. 2013).

3.1.1 The Reynolds-Averaged-Navier-Stokes equations

As mentioned the full resolution of the Navier-Stokes equation is very complex and only for well-defined cases, like fully developed, laminar flows, a direct solution can be found. In turbulent flows, the field properties become random functions of space and time. Hence, the field variables v_i and p must be expressed as the sum of time-averaged and fluctuating parts as (Fig.3.2):

$$v_i = \bar{v}_i + v_i^+ \ , \ p = \bar{p} + p^+ \tag{3.6}$$

The time-averaged values at a fixed point in space are given by:

$$\bar{v}_i = \frac{1}{t_1} \int_{t_0}^{t_0+t_1} v_i \, dt \tag{3.7}$$

Taking the mean values over a sufficiently long time interval t_1, the time-averaged values of the fluctuations equal zero; thus, $\overline{v_i^+} = \overline{p^+} = 0$.

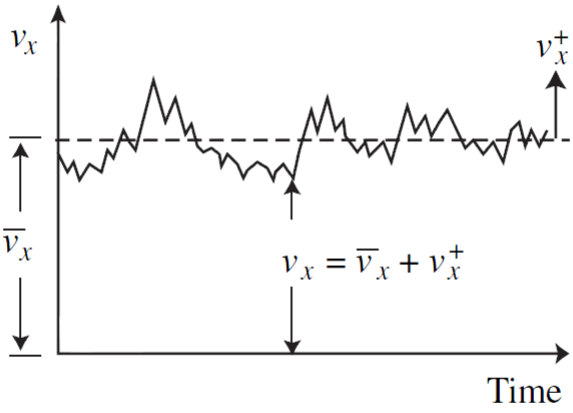


Figure 3.2: Velocity measurements versus time (Julien, 2010)

Inserting Eq.(3.6) into the Navier-Stokes equations gives the Reynolds-Averaged Navier-Stokes equations or shortly RANS equations. Here only the final three-dimensional RANS equations are listed, with Eqs.(3.8) and (3.9) for continuity and momentum, respectively.

$$\frac{\partial \bar{v}_i}{\partial x_i} = 0 \tag{3.8}$$

$$\rho_m \frac{\partial \bar{v}_i}{\partial t} + \rho_m \bar{v}_j \frac{\partial \bar{v}_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\mu \left(\frac{\partial \bar{v}_i}{\partial x_j} + \frac{\partial \bar{v}_j}{\partial x_i} \right) - \rho_m \overline{v_i^+ v_j^+} \right) + f_i \tag{3.9}$$

The last noted stress term is called Reynolds stress tensor $\tau^{Re} = -\rho_m \overline{v_i^+ v_j^+}$ and has to be provided external by a turbulence model.

3.1.2 The shallow water equations

A numerical solution of the three dimensional Navier-Stokes equations is quite complex and needs high computational effort, even with the separated treatment of turbulence as in RANS equations. The shallow water equations or shortly SWE, which are derived from the Navier-Stokes equations under the assumption that the horizontal length scale is much larger than the vertical on, describe of a hydrostatic homogeneous incompressible fluid in response to gravitational and rotational accelerations. The combination of the mentioned assumptions and their application to the Navier-Stokes equations, leads to the depth averaged shallow-water equations for continuity and momentum, (Hervouet 2007):

$$\frac{\partial h}{\partial t} + \frac{\partial hU_x}{\partial x} + \frac{\partial hU_y}{\partial y} = 0 \tag{3.10}$$

where U_x and U_y are the depth-averaged quantities of local velocities u_x and u_y.

$$\rho_m \frac{\partial U_i}{\partial t} + \rho_m U_j \frac{\partial U_i}{\partial x_j} = -\rho_m g \frac{\partial h}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\mu \frac{\partial U_i}{\partial x_j} - \rho_m U_i^+ U_j^+\right) + f_i \tag{3.11}$$

The term $\rho_m U_i^+ U_j^+$ is the disturbed part of the averaged diffusion. This is similar to the τ^{Re} in the RANS equations and estimated by a turbulence model.

3.2 Sediment transport models

Non-cohesive bed particles enter motion as soon as the shear stress applied on the bed material exceeds the critical shear stress. Generally, silt and clay particles enter suspension, and sand and gravel particles roll and slide in a thin layer near the bed called the bed layer. The bed layer thickness is typically less than 1 mm in sand-bed channels, up to tens of centimeters in gravel-bed streams. The total sediment transport rate q_t is defined as the flux of sediment transported per unit width and time, and can be split up in bed load and suspended load.

$$q_t = q_b + q_s \tag{3.12}$$

3.2.1 Theory of incipient motion

According to Shields theory, the movement begins when the shear stress exceeds a critical shear value. Above this value, the current is able to transport the granular sediment. The dominating forces acting on a sediment particle at the river bottom are: the bed shear stress τ_0, sediment density ρ_s, fluid density ρ_m, grain diameter d_s, gravity g and the dynamic viscosity μ, (Chanson 2004). Applying dimensionless analysis on these parameters give two characteristic values as:

$$Re_* = \frac{u_* d_s}{\upsilon_m} \tag{3.13}$$

$$Fr_* = \frac{(u_*)^2}{(s-1)gd_s} \tag{3.14}$$

where Re_* is grain Reynolds number, Fr_* is grain Froude number (Shields parameter τ_*), $s = \rho_{s/}\rho_m$ is relative density, d_s is grain diameter and $u_* = (\tau_{0/}\rho_m)^{1/2}$ is friction velocity. Applying friction velocity relation into Eq.(3.14) gives the relation between the Shields parameter and the bed shear stress, as:

$$\tau_* = \frac{\tau_0}{(\rho_s - \rho_m)\, gd_s} \tag{3.15}$$

The incipient motion condition is expected as $\tau_* > \tau_{*c}$ where:

$$\tau_{*c} = \frac{\tau_c}{(\rho_s - \rho_m)\, gd_s} \tag{3.16}$$

From the analysis of fully developed two-dimensional flows in a laboratory channel and the evaluation of the beginning of sediment motion, a relation between both quantities is derived. Fig.3.3 illustrates the dependence between Re_* and critical τ_c (Julien, 2010).

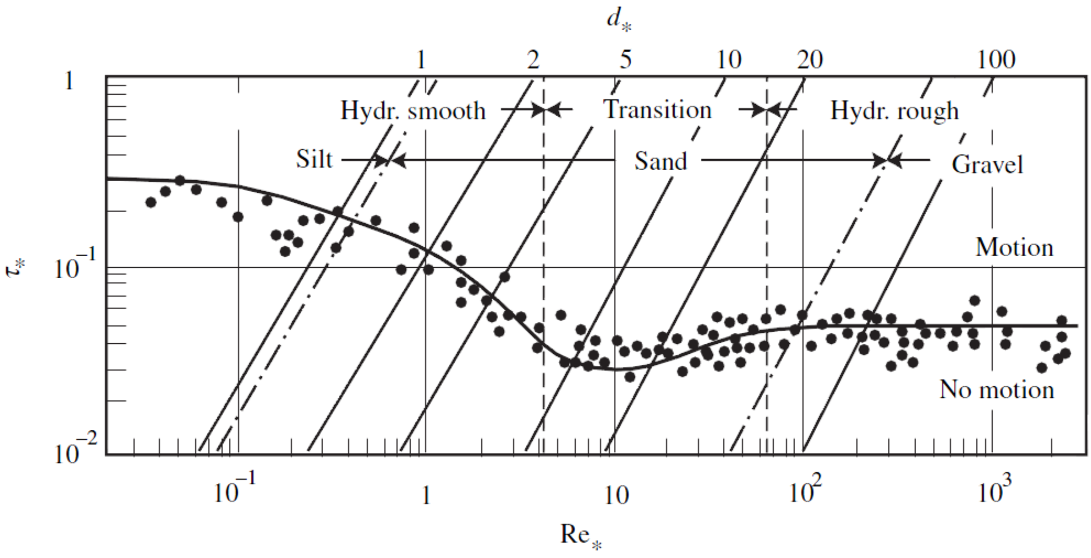


Figure 3.3: Shields diagram for granular material (modified from Julien, 2010)

3.2.2 Bedload

As sketched in Fig.3.4, the bed layer thickness a is a few grain diameters thick, and $a = 2d_s$ has been commonly used. Bed-load, or contact load, refers to the transport of sediment particles which frequently maintain contact with the bed. Bedload transport can be treated either as a deterministic or a probabilistic problem. Deterministic methods have been proposed by DuBoys and Meyer-Peter Müller; probabilistic methods were developed by Kalinske and Einstein.

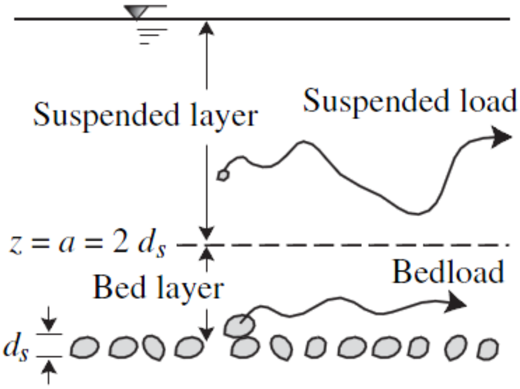


Figure 3.4: Definition sketch of bedload and suspended load (Julien, 2010)

DuBoys' equation: The pioneering contribution of DuBoys (1879) is based on the concept that sediment moves in thin layers along the bed. The applied bed shear stress τ_0 must exceed the critical shear stress τ_c to initiate motion. The volume of gravel material in motion per unit width and time q_b in ft^2/s is calculated from:

$$q_b = \frac{0.173}{d_s^{3/4}} \tau_0 \left(\tau_0 - 0.0125 - 0.019 d_s\right) \tag{3.17}$$

where d_s is the particle size and τ_0 is the boundary grain shear stress.

Meyer-Peter Müller's equation: Among numerous available semitheoritical bedload transport equations, the MPM is probably the most widely applied equation in both basic research and engineering applications. Meyer-Peter and Müller (1948) developed a complex bedload formula for gravels based on the mean sediment size d_{50} of the surface layer of the bed material. Chien (1956) demonstrated that the elaborate original formulation can be reduced in the following simple form:

$$\frac{q_b}{\sqrt{(s-1)\,g d_s^3}} = 8\left(\theta - 0.047\right)^{3/2} \tag{3.18}$$

This formulation is most appropriate for channels with large width-depth ratios and for grain diameters in the range of 0.4 mm $< d_{50} <$29 mm.

Einstein-Brown's equation: H.A. Einstein (1941) made the seminal contribution to bedload sediment transport. He introduced the idea that grains move in steps proportional to their size. He defined the bed layer thickness as twice the particle diameter. He extensively used probability concepts to formulate a relationship for contact sediment discharge.

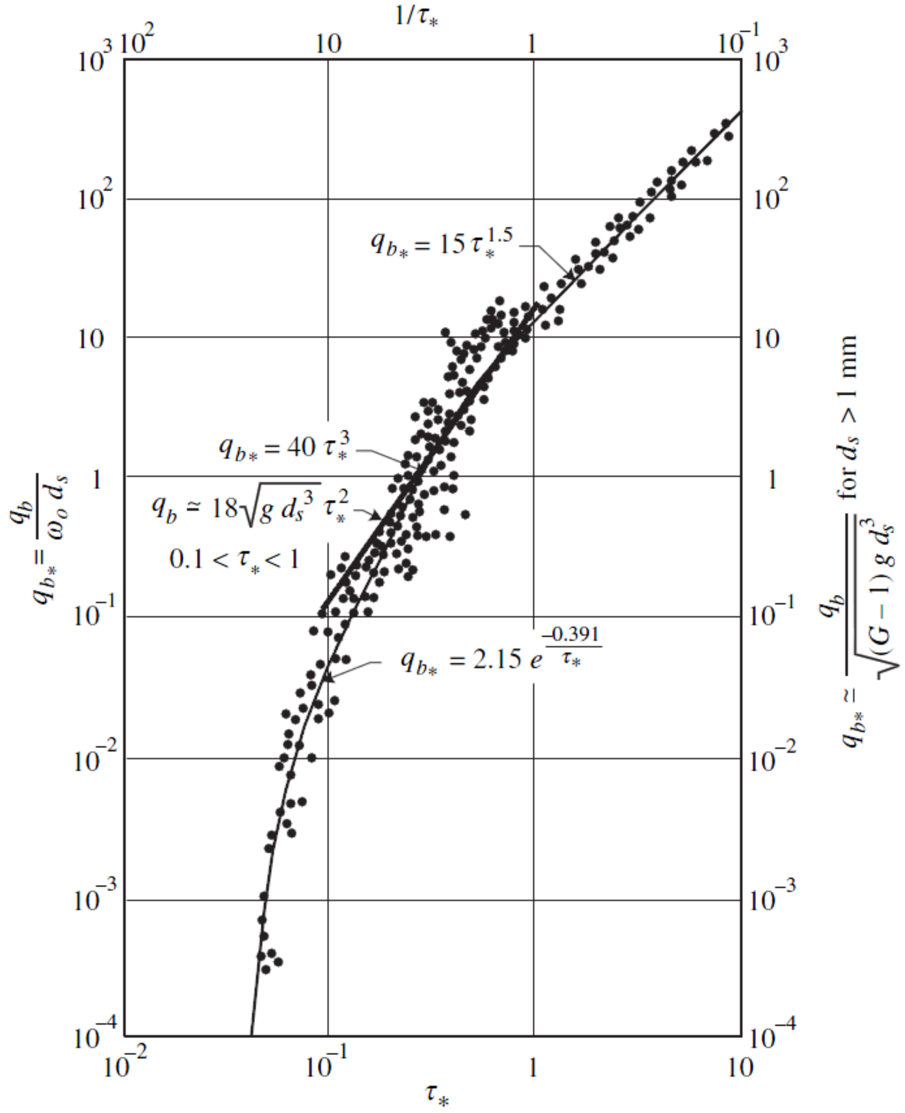


Figure 3.5: Dimensionless sediment discharge q_{b_*} versus Shields parameter τ_*(Julien, 2010)

The gravel sediment discharge q_b in volume of sediment per unit width and time (q_b in L^2/T) is transformed, using Rubey's clear-water fall velocity ω_0, into a dimensionless volumetric unit sediment discharge q_{b*} as:

$$q_{b*} = \frac{q_b}{\omega_0 d_s} = q_b \left(\sqrt{(s-1)\, g d_s^3} \left\{ \sqrt{\frac{2}{3} + \frac{36 \upsilon_m^2}{(s-1)\, g d_s^3}} - \sqrt{\frac{36 \upsilon_m^2}{(s-1)\, g d_s^3}} \right\} \right)^{-1} \tag{3.19}$$

The dimensionless rate of sediment transport q_{b_*} is shown in Fig.3.5 as a function of the Shields parameter $\tau_* = \tau_0/(\rho_s\text{-}\rho_m)\mathrm{g}d_s$, with measurements from Gilbert (1914), Bogardi (1974) and Wilson (1966). Brown (1950) suggested the following two relationships:

$$\begin{cases} q_{b*} = 2.15 e^{\frac{-0.319}{\tau_*}} & \text{when} \quad \tau_* < 0.18 \\ q_{b*} = 40 \tau_*^3 & \text{when} \quad 0.52 > \tau_* > 0.18 \end{cases} \tag{3.20}$$

Considering sediment transport data at high shear rates τ_* >0.52 one obtains:

$$q_{b*} = 15 \tau_*^{1.5} \quad \text{when} \quad \tau_* > 0.52 \tag{3.21}$$

At such high shear rates, this third approximation is not very accurate, however, because large quantities of sediment will move in suspension as discussed in next section.

3.2.3 Suspended load

As the hydraulic forces exerted on sediment particles exceed the threshold condition for beginning of motion, coarse sediment particles move in contact with the bed surface as described. Finer particles are brought into suspension when turbulent velocity fluctuations are sufficiently large to maintain the particles within the mass of fluid without frequent bed contact. The unit suspended sediment discharge q_s in natural streams and canals is computed from the depth-integrated advective flux of sediment Cv_x above the bed layer $z{>}a$:

$$q_s = \int_a^h C v_x dz \tag{3.22}$$

where C is volumetric sediment concentration, and v_x is logarithmic velocity profile. The comparison of suspended load to bedload delineates which mode of sediment transport is dominant. The suspended unit sediment discharge q_s can be calculated from Eq.(3.22) after substituting C and logarithmic velocity v_x:

$$q_s = \int_a^h C_a \frac{u_*}{k} \left[\left(\frac{h-z}{z} \right) \left(\frac{a}{h-a} \right) \right]^{\frac{\omega_0}{ku_*}} \ln \frac{z}{z_0} dz \tag{3.23}$$

in which C_a represents the reference sediment concentration at a reference elevation a, k is equal 0.4, ω_0 is fall velocity, and z_0 is a distance from the flat boundary at which the logarithmic velocity v_x hypothetically equals zero (Julien, 2010).

3.3 The Exner equation

Mathematically, the morphological evolution of the bed is defined by the so-called sediment continuity or Exner equation. This equation simply states that the time rate of change of the bed elevation is equal to the divergence of the sediment flux, which can be expressed in terms of the local flow properties through the use of an empirical sediment transport formulae (Kubatko et al., 2006).

In the case of mobile bed, it is necessary to describe the movement of the granular sediment with an appropriate equation. The solid concentration is defined as:

$$c = \frac{V_s}{V} \tag{3.24}$$

where V_s is the solid volume and V the total volume. The rates of bed level changes are calculated from the equation of conservation of sediment mass. In two dimensions, this is written as:

$$\frac{\partial z}{\partial t} + \frac{1}{1-p_0} \left(\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} \right) = \frac{\partial (c\rho_s)}{\partial t} \tag{3.25}$$

where z is bed level above a fixed datum, x and y are horizontal space coordinates, t is time, p_0 is porosity and, q_x and q_y are sediment transport rates in x and y directions. The term on the right side of the equation defines the variation of the solid materials concentration in the control volume, where ρ_s is the sediment density. Assuming that the solid concentration is constant, the system becomes:

$$\frac{\partial z}{\partial t} + \frac{1}{1-p_0} \left(\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} \right) = 0 \tag{3.26}$$

which is known as the Exner equation. The transport rates q_x and q_y are functions of several parameters, namely, currents waves water depth and sediment properties. Within morphological time step (i.e. time step used in solving Eq.(3.26), we assume the following at each grid point: (a) sediment properties are fixed, (b) currents and waves locally vary with water depth and (c) water level is at a fixed level above datum. Under these assumptions, the transport rate vary only with the bed level and Eq.(3.26) can be written as an advection equation as:

$$\frac{\partial z}{\partial t} + \frac{1}{1-p_0}\left(\frac{\partial q_x}{\partial z}\frac{\partial z}{\partial x} + \frac{\partial q_y}{\partial z}\frac{\partial z}{\partial y}\right) = 0 \tag{3.27}$$

or

$$\frac{\partial z}{\partial t} + C_x(z)\frac{\partial z}{\partial x} + C_y(z)\frac{\partial z}{\partial y} = 0 \tag{3.28}$$

where $C_x(z)$ and $C_y(z)$ are, respectively the x and y components of the bed celerity, which also depend on the bed level. Eq.(3.28) shows that morphological evolution occurs as non-linear propagation of the bed level deformations in the direction of the transport. The Exner equation needs a closure model for the sediment transport q, a matter studied by Shields (1936), who developed the theory of incipient motion.

3.4 Numerical methods

The mentioned equations for hydrodynamic (i.e. RANS or SWE) and morphodynamic (i.e. Exner equation) systems represent nonlinear hyperbolic partial differential equations and analytical solutions of these equations are not possible except for a few simplified cases. Therefore, they are often solved by numerical schemes where the continuous description of the equations are transferred into a set of discrete expressions in time and space.

3.4.1 Spatial discretization

There are three main methods for discretizing the flow field in numerical simulations. A brief description of these methods is provided below.

3.4.1.1 Finite Difference Method (FDM)

The finite difference method uses Taylor series expansions to derive difference quotient expressions for the derivatives at discrete grid points, expressing them through variable values at neighboring grid points. This is explained here by reference to a one-dimensional variable distribution as shown in Fig.3.6.

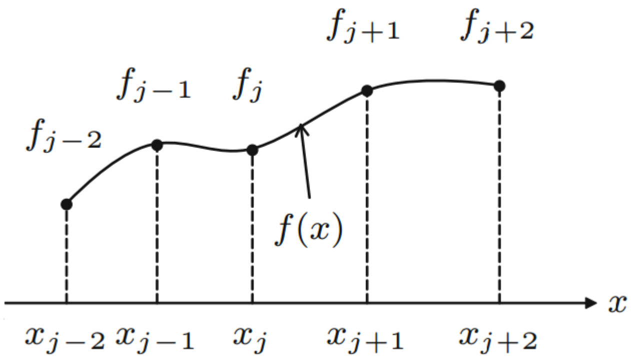


Figure 3.6: Finite difference computational stencils for 1D problems

As can be seen from this figure, a continuous function $f(x)$ can be represented as a series of discrete values f_j at discrete points (j). The value f_{j+1} at point $(j+1)$ can be expressed in terms of a Taylor series expanded about point (i) as:

$$f_{j+1} = f_j + \frac{\partial f}{\partial x}|_j\,(\Delta x)^1 + \frac{\partial^2 f}{\partial x^2}|_j\,\frac{(\Delta x)^2}{2} + \frac{\partial^3 f}{\partial x^3}|_j\,\frac{(\Delta x)^3}{6} + \cdots \tag{3.29}$$

Similar expansions can be made for other points. Expression (3.29) is exact if an infinite number of terms on the right hand side is retained and/or if $\Delta x \to 0$. The accuracy of Eq.(3.29) depends on which terms are neglected. With Eq.(3.29) and similar Taylor series expansions for other neighboring points, we can derive the following approximations:

$$1^{st} order : \frac{\partial f}{\partial x}|_j = \frac{f_{j+1} - f_j}{\Delta x} + \Gamma_1 \tag{3.30a}$$

$$2^{nd} order : \frac{\partial f}{\partial x}|_j = \frac{f_{j+1} - f_{j-1}}{2\Delta x} + \Gamma_2 \tag{3.30b}$$

The above represent the first order forward or upwind difference (3.30a) and the second order central difference approximations (3.30b). The truncation term Γ_m represents the higher order terms not accounted for in the difference approximations and is the difference between the exact solution of the derivative and its discrete approximations.

An advantage of this method is the high accuracy and direct implementation to numerical solvers. However, when we encounter irregular geometries or an unusual specification of boundary conditions, we find that finite difference techniques become hard to use.

3.4.1.2 Finite Element Method (FEM)

The powerfulness of FEM arises when more complex geometries are needed, and FDM becomes harder and harder to implement and its demands of computational power increase excessively. The FEM subdivides a model into very small but finite-sized elements of geometrically simple shapes (e.g. triangles), which can represent the real geometry very accurate. The collection of all these simple shapes constitutes the so-called finite-element mesh. As an example of how a finite difference model and a finite element model might be used to represent a complex geometrical shape, consider Fig.3.7 where an exterior shape, makes it a non-simple geometry.
A uniform finite difference mesh would reasonably cover the blade (the solution region), but the boundaries must be approximated by a series of horizontal and vertical lines (or "stair steps"). On the other hand, the finite element model (using the simplest two-dimensional element; the triangle) gives a better approximation to the region (Pletcher et al., 2012).

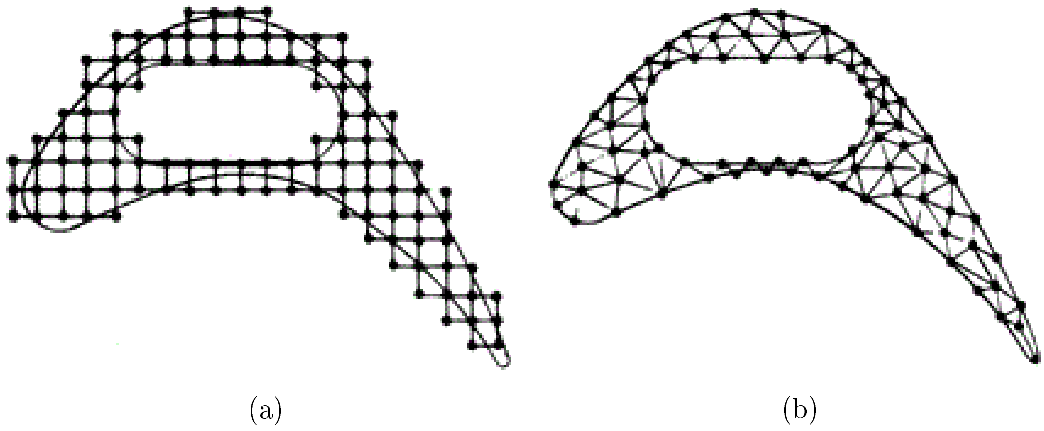


Figure 3.7: (a) Finite difference and (b) finite element discretization of geometry (Pletcher et al., 2012)

Classical finite element methods assume continuous or weakly continuous approximation spaces and ask for volumetric integrals of the weak form to be satisfied. The order of accuracy is increased by raising the approximation order within elements. The methods are not exactly conservative, thus often struggle with stability for discontinuous processes. In contrast, finite volume methods use piecewise constant approximation spaces and ask for integrals against piecewise constant test functions to be satisfied. This yields exact conservation statements.

3.4.1.3 Finite Volume Method (FVM)

The FVM offers an alternative approach for deriving the discretized equations. This method is based on the principle that the divergence term, that frequently occurs in differential equations governing various interesting scientific phenomena, can be rewritten as a surface integral using the divergence theorem. The problem then simplifies to evaluating fluxes normal to the cell walls. Since this becomes a vector problem, the cell walls can take any shape and can be arbitrarily oriented. All that is required is that the they enclose a closed volume. Since the method is based on evaluating fluxes, the FVM is conservative. Outflow from one cell becomes inflow into another. This makes the FVM stable and flexible, and yet relatively easy to implement. This is why the Finite Volume Method is commonly implemented in commercial computational fluid dynamics (CFD) solvers (Pletcher et al., 2012).

3.4.2 Time discretization

The discretization of the time derivative using finite differences is very similar to the discretization in space, and an approximation analogous to the first-order expression (3.30a) can, for instance, be derived from a Taylor series as:

$$\frac{\partial f}{\partial t} = \frac{f^{n+1} - f^{n}}{\Delta t} + \Gamma_1 \tag{3.31}$$

where Δt is time step, f^n is the value of f at time t_n, f^{n+1} is the yet unknown value of f at time $t_{n+1}$and Γ_1 is the truncation error.

The simplest time discretization schemes are explicit and implicit Euler methods, in which variable f^{n+1} is calculated from:

$$\frac{f^{n+1} - f^{n}}{\Delta t} = rhs^{n} \tag{3.32}$$

$$\frac{f^{n+1} - f^{n}}{\Delta t} = rhs^{n+1} \tag{3.33}$$

Euler methods can be considered as the analogues of forward and backward differencing in space and are first order accurate in time. Euler methods are called two-point methods, because value of f at two instances in time are involved. A second order accurate two-point method can be constructed by applying the trapezoidal rule to approximate *rhs*, which yields the semi-implicit Crank-Nicolson method:

$$\frac{f^{n+1} - f^{n}}{\Delta t} = \frac{1}{2}\left(rhs^{n} + rhs^{n+1}\right) \tag{3.34}$$

The application of the explicit Euler method provides very fast computational results, but suffer from instability for inappropriate time steps or grid resolutions. Therefore, explicit time discretization methods are subject to rigorous stability conditions, which are generally known as the CFL condition (Courant-Friedrichs-Levy conditions):

$$CFL = \frac{|u|\,\Delta t}{\Delta x} < 1 \tag{3.35}$$

where u is characteristic velocity, and Δx is grid resolution.

3.5 Numerical solutions

The numerical solution of the system constituting two hydrodynamic and one sediment equation is non-trivial. Cunge et al. (1980) discussed two approaches that can be used for any sediment transport flux that is a function of flow velocity only. However, both approaches can be adapted for any sediment transport flux with varying degree of difficulty (Hudson et al. 2005).

3.5.1 Decoupled approach

The decoupled approach assumes that the water motions are steady with respect to changes in the bed level. In other words, the timescale over which the bed changes is so much larger than those associated with the hydrodynamic motions that these individual motions do not impact on bed changes: only the mean hydrodynamics effects and responds to bed changes.

Another way of stating this is that the speed of the propagation of seabed waveforms (like ripples or tidal sand waves) is considerably smaller in magnitude than the wave speeds of the water flow. These assumptions allow the water flow to be discretized separately from the bed. Moreover, the approach takes advantages of the slow evolution of the bed by iterating the water flow,

$$\begin{bmatrix} h \\ uh \end{bmatrix}_t + \begin{bmatrix} uh \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix}_x = \begin{bmatrix} 0 \\ -ghz_x \end{bmatrix} \tag{3.36}$$

to an equilibrium state each time the bed is updated. In other words, the quasi-stationary assumption is implemented numerically by iterating to a state where the time derivatives are zero. This results in the computation time being significantly reduced for test cases where the bed evolves slowly (Hudson et al. 2005).

3.5.2 Coupled approach

For the coupled approach, no assumptions are made and the water flow and bed are calculated simultaneously. With this approach, the water motions can either be steady or unsteady and changes in the bed update are considered to be significant. Here the wave speed of the sediment continuity equation can be of a similar magnitude to the wave speeds of the water flow. For this approach, the system is discretized simultaneously (Hudson et al. 2005).

$$\begin{bmatrix} h \\ uh \\ z \end{bmatrix}_t + \begin{bmatrix} uh \\ hu^2 + \frac{1}{2}gh^2 + ghz \\ \frac{1}{1-p_0}q \end{bmatrix}_x = \begin{bmatrix} 0 \\ gzh_x \\ 0 \end{bmatrix} \tag{3.37}$$

Both approaches discussed above can be written in the general form:

$$\frac{\partial \vec{\mathbf{w}}}{\partial t} + \frac{\partial \vec{\mathbf{F}}(\vec{\mathbf{w}})}{\partial x} = \vec{\mathbf{R}} \tag{3.38}$$

where $\vec{\mathbf{F}}(\vec{\mathbf{w}})$ is the numerical flux and $\vec{\mathbf{R}}$ contains the inhomogeneous term. In order to solve the system (3.38) the continuous equations must be discretized to reflect the finite number of grid points. Using the upwind scheme, this discretization takes the form:

$$\vec{\mathbf{w}}_i^{t+1} = \vec{\mathbf{w}}_i^t - \frac{\Delta t}{\Delta x}(\vec{\mathbf{F}}_{i+1}^t - \vec{\mathbf{F}}_i^t) + \frac{\Delta t}{\Delta x}\vec{\mathbf{R}}_i \tag{3.39}$$

The upwind scheme is an example of an explicit scheme, that is a scheme where the solution at the new time-level $t+1$ can be calculated explicitly from the quantities that are already known at the previous time-level t. This is to be contrasted with an implicit scheme in which the finite difference representations of the differential equation has, on the right hand side, terms at the new time level $t+1$.

The points $x = x_0$ and $x = x_I$ are the spatial boundaries and we require numerical boundary conditions at these points. The spatial step size, Δx, is fixed and we use a variable time step,

$$\Delta t = \frac{v\Delta x}{\max_i(|\lambda_k|)} \tag{3.40}$$

where $\max(|\lambda_k|)$ is the maximum wave speed, λ_k are the eigen values of the Jacobian matrix associated with the system,

$$\mathbf{A}(\vec{\mathbf{w}}) = \frac{\partial \vec{\mathbf{F}}}{\partial \vec{\mathbf{w}}} \tag{3.41}$$

where k is the k-th component of the system and v is the required Courant number. Unless stated otherwise, the scheme discussed here is stable for Courant number of less than one. The values λ_k are the eigen values of the Jacobian, and represent the speeds at which information travels in the general system (3.38).
The condition (3.40) amounts to ensuring that time steps are small enough to ensure that, consistent with the numerical scheme (3.39), no wave can propagate further than its adjacent cell in one integration. And this is one of the main reasons why most morphodynamic codes take the decoupled approach, because the hydrodynamical wave speeds are usually so much greater than the morphodynamical ones, resulting in time steps that are orders of magnitude smaller and codes that are correspondingly slower. The decoupled approach also has the advantage that any sediment transport flux, or even a black box approach, can easily be implemented, whereas the coupled approach can experience difficulties when including more complex sediment transport fluxes (Hudson et al. 2005). More detailed information about numerical methods for morphodynamic modelling and their solution can be found in Hudson (2001) and Rezzolla (2011).

Chapter 4

TELEMAC-MASCARET system for hydro-morphodynamic modeling

The open source TELEMAC-MASCARET was developed originally by the National Hydraulics and Environment Laboratory (LNHE) of the research and development directorate of the French Electricity Board (EDF) as a hydro-informatics system for free surface flows (Hervouet 2007). All modules of the system are based on unstructured grids and finite element or finite volume algorithms. One important feature is parallelism with domain decomposition. The implicit algorithms have led to a partitioning without overlapping, with matching interface points, and linear systems are solved on the whole domain. The programs are written in Fortran 90 and can be run on Unix, Linux and Windows systems, they are compatible with any Fortran 90compiler. The model system includes 2D and 3D hydrodynamic modules (TELEMAC-2D and -3D), and a spectral wave propagation model (TOMAWAC). The environment is extended by the two dimensional morphologic module SISYPHE (developed with contribution of Dr. Minh Duc Bui at Technical University of Munich) for bed load and depth averaged suspended load and SEDI-3D for three dimensional suspended load. More detailed information about the system can be found on the website: http://www.opentelemac.org/.

4.1 Hydrodynamic module

The TELEMAC-2D flow module solves the shallow water equations with several options for the horizontal dispersion terms (e.g. depth-averaged k-ε model, Elder model, and constant eddy viscosity models) and source terms (e.g. atmospheric pressure gradients, Coriolis force, etc.). The numerical discretization includes a choice of classical methods for the advection terms (e.g. characteristics, SUPG, distributive schemes...). The use of implicit schemes enables relaxation of the limitation on time steps (typically, values of a CFL-numbers up to 10 or 50 are acceptable). Recently, ideas stemming from finite volume techniques have been coupled with these implicit schemes to ensure monotonicity of depth and sediment concentrations, as well as mass conservation at machine accuracy.

TELEMAC-3D solves the Reynolds-Averaged Navier-Stokes (RANS) equations in unstructured meshes obtained by a super- imposition of 2D meshes of triangles. The movement of the mesh can be taken into account in the advection step by transformation. The superimposed layers may not be evenly spaced. This allows a more accurate representation of the flow field by a refinement near the bed, enabling better accuracy of the turbulence models (mixing-length model, k-ε model) and leading to a better estimate of the bed shear stress. The 3D model can be applied to capture the effect of vertical recirculation cells as well as stratification effects, assuming a hydrostatic or non-hydrostatic pressure distribution (Villaret et al., 2013).

4.2 Wave propagation model

TOMAWAC is a third generation spectral wave propagation model, which accounts for the effect of wave generation by wind, white-capping, nonlinear wave-wave interaction, refraction, shoaling and dissipation (cf. Benoit etal.,1996). Internal coupling between TELEMAC-2D and TOMAWAC accounts for the effect of the waves on the mean circulation (e.g. wave-induced currents, leading to littoral drift) as well as the modulation of waves by the tides. The interaction between waves, tidal flows and storm-surges has also been studied at a regional scale by Nicolle et al.(2009).

4.3 Sediment transport and morphodynamic module

Sediment transport and bed change modules (SISYPHE) can be used to model complex morphodynamic processes for different flow conditions, sediment size classes and sediment transport modes. In SISYPHE, sediment transport processes are grouped as bed-load, suspended-load or total-load, with an extensive library of bed-load transport relations. A choice of different sediment transport formulae for bed-load or total-load is implemented.

SISYPHE is applicable to non-cohesive sediments that can be uniform (single-sized) or non-uniform (multiple-sized), cohesive sediments (multi-layer consolidation models), as well as sand-mud mixtures. A number of physically-based processes are incorporated into SISYPHE, such as the influence of secondary currents to precisely capture the complex flow field induced by channel curvature, the effect of bed slope associated with the influence of gravity, bed roughness predictors, and areas of non-erodible bed, among others. SISYPHE can be coupled to the depth-averaged shallow water module TELEMAC-2D or to the three-dimensional Reynolds-averaged Navier-Stokes module TELEMAC-3D.

The sediment transport model relies on a complete description of the flow field, through internal coupling with the flow module. At each time step, the hydrodynamic model (TELEMAC) calculates the flow field and sends to the SISYPHE model the spatial distribution of the main hydrodynamic variables: water depth, flow velocity components, and bed shear stress. These sediment transport rates are calculated bed level change is used to account the effects of sediment transport on flow. The structure of such a coupled system is shown in Fig.4.1.

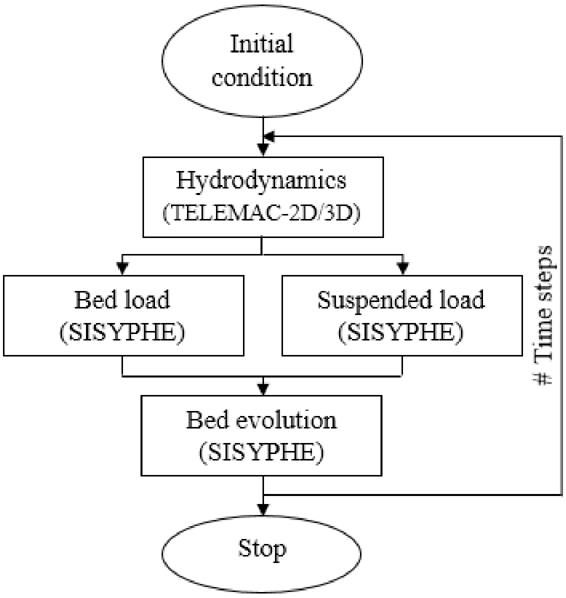


Figure 4.1: Scheme of a hydro-morphodynamic-model system

Sediment transport rates in the modeling system are calculated with classical semi-empirical concepts, which involve the decomposition of sediment transport rates into bed-load and suspended load. The resulting bed evolution is then computed by solving the Exner's equation. The model is mainly applicable to non-cohesive sediment, composed of either uniform grains or multi-grains, characterized by their mean size and density.

The conventional method for performing hydro-morphodynamic simulations in rivers is to decouple the hydrodynamic and the morphodynamic modules. The decoupling approach is based on the rationale that the channel bed reacts at a much slower timescale than the flow. At implementation level, these modules communicate through a quasi-steady morphodynamic time-stepping mechanism: during flow computation, the bed level is assumed to be constant and during computation of the bed level, the flow and sediment transport quantities are assumed invariant to the bed level changes. The modules are linked together at the programming level.

4.4 Input files for TELEMAC-MASCARET

TELEMAC demands different input files to create a simulation. Following provides a short description of these files as they are used in this thesis:

- **Geometry file (.slf)**: This is a mandatory file in Telemac software which contains the mesh information. This file can include more information like friction values as well.

- **Steering file (.cas)**: This is a mandatory text file containing all the configuration of the computation. In a way it presents the control panel of the computation. It includes a number of keywords to which values are assigned. For coupling of modules several steering files are necessary.

- **Boundary conditions file (.cli)**: This is a formatted file generated automatically by other softwares such as BLUE KENUE but it can be modified by a standard text editor. Each line of this file is dedicated to one point of the mesh boundary. This is also a mandatory file for running software. The BLUEKENUE program can be used as a pre- and post-processor of TELEMAC.

- **Liquid boundary file (.txt)**: The user is able to determine values for time-dependent boundary using this text file.

- **Fortran file (.f)**: The file contains all the Telemac subroutines modified by the user and those that have been specially developed for the computation. The software allows the user to change and modify some subroutines.

- **Previous computational file (.slf)**: If this file is included, the software uses an existing simulation to initialize the next simulation. Here it is important that the number of nodes and elements does not change.

Chapter 5

New concepts for integrating ANNs into hydro-morphodynamic models

The transport of sediment as bed load is an important process that occurs in rivers, estuaries, and coastal regions. In many situations, this process and the resulting morphological changes of the bed can have a detrimental impact on the infrastructure and environment. Clearly, the processes of sediment transport and morphological evolution of the bed are determined by the properties of the fluid flow, which in turn are affected by the changes in the morphology of the bed that they induce. Thus, the motion of the fluid and the motion of the bed form an interdependent two-phase phenomenon that must be analyzed using a model system made up of two distinct but interdependent model components: (1) a hydrodynamic component defining the evolution of the flow; and (2) sediment transport/morphological component defining the evolution of the bed. Such a modeling system is often referred to as a hydro-morphodynamic model. The conventional method for performing morphodynamic simulations in rivers is to decouple the hydrodynamic and the morphodynamic systems. At the implementation level, these modules communicate through a quasi-steady morphodynamic time-stepping mechanism: during flow computation, the bed level is assumed to be constant and during computation of the bed level, the flow and sediment transport quantities are assumed invariant to the bed level changes. The modules are linked together at the programming level. In the conventional hydro-morphodynamic models the bed level changes are governed by the equation for conservation of sediment mass (Exner's equation). The bedload transport rate (q_b) in Exner equation is a complex function of various hydrodynamic quantities such as currents and water depth as well as quantities associated with sediment properties such as sediment density and grain size. Many empirical functions are available to calculate bedload transport. Most of the formulas available in the literature have been developed by laboratory and field data analysis using statistical methods such as the regression method. It is understood that such formulas have drastic differences between them. No uniformly valid formulation for q_b currently exists. That being said, we are unable to select the most accurate formula for a particular problem, which results in challenges with interpreting the accuracy of computational sediment transport models. Therefore, in many cases, unreasonable morphological changes are predicted and the results of the different formulae often strongly vary. The reasons are assumed in the complexity of the interaction between flow and sediment transport and in limitations of the nonlinear regression applied in these methods. In contrast to most traditional empirical methods, which need prior knowledge about the nature of the relationships among the data, data-driven models learn from data examples presented to them. Through learning the input data, these models can capture the subtle functional relationships among the data even if the underlying relationships are unknown or the physical meaning is difficult to explain. Additionally these models have proven a high tolerance against data sample errors. These attributes make the utilization of data-driven methods for sediment transport predictions very promising. Thus, the main objective of this research is to develop new schemes for hydromorphological model system based on data-driven methods to predict the morphological changes. The focus of this research is especially on ANN as it has lower computational cost than ANFIS.

5.1 Motivations

The following summarizes the motivations behind the use of ANN for new proposed hydro-morphodynamic model systems:

1. To overcome imprecise and limited existing sediment transport formulas.

2. To fix the numerical instability of conventional methods in some cases.

3. No prior knowledge of data relationship and physical characteristic are required in ANN application.

4. To handle non-linearity and automatically adjusts to new information, while generally requiring little computational effort.

5. To solve difficulties in bed-load measurement, as it only needs bed-level-change observation, which is easier.

6. Its high tolerance against data sample errors.

5.2 Two new concepts

To overcome the weaknesses of the conventional models, two new concepts are proposed that integrate ANNs into hydro-morphodynamic models. In the first proposed concept, up to 3 ANN models can be used for hydrodynamic calculations depending on the complexity and dimension of the problem. Since the characteristic time scale of bed-evolution and bed load transport processes is normally much greater than that of fluid flow, it can be assumed that changes in the bed elevation during one computational time step do not significantly influence the flow field. This assumption leads to the computationally attractive possibility of coupling flow and sediment computations in an iterative manner. Hereby, the flow and sediment-transport modules communicate through a quasi-steady morphodynamic time-stepping mechanism: during the flow computation, the bed level is assumed constant and during the computation of the bed level the flow and sediment transport are assumed invariant to the bed level changes. At each time step, the hydrodynamics variables (velocity field, and water depth) are transferred into the morphodynamic model which in this case is another ANN-based approximator. The bed elevation is updated and then sent back to the hydrodynamic model. The procedure continues until the last time step is reached. This proposed coupling concept is shown in Fig.5.1.

According to the proposed concept, the configured ANNs need to obtain some knowledge of the task presented to them. The knowledge can be generated by providing the required data and passing the training process. Normally, a data set is divided randomly into three subsets, whereby the biggest amount of data is added randomly to the training subset. The remaining data set samples are used for validating and testing the networks. Finally, the weights and biases get modified and fixed during the training process, so that the networks can evolve an appropriate behavior towards their application.

In the second proposed concept, the framework is the TELEMAC-MASCARET open source. According to this model, an ANN-based approximator replaces the sediment transport model SISYPHE, which is coupled with either the 2D or 3D flow models. This requires modification of some subroutines in TELEMAC.

In fact, the hydrodynamic calculations are done by TELEMAC while the bed change is being calculated by the knowledge of ANN. Unlike the first proposed concept where both structured and unstructured meshes can be applied, here we are obliged to use

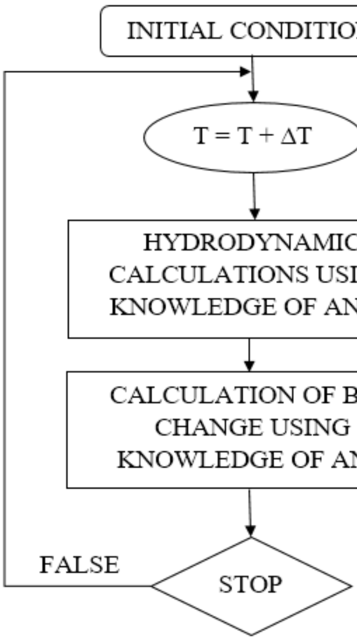

INITIAL CONDITION
T = T + ΔT
HYDRODYNAMIC CALCULATIONS USING KNOWLEDGE OF ANNS
CALCULATION OF BED CHANGE USING KNOWLEDGE OF ANN
FALSE
STOP

Figure 5.1: The first proposed hydro-morphodynamic-model system

only unstructured mesh. The reason is that all modules of the TELEMAC system are based on unstructured grids so that it must receive the calculated bed evolution at different times for every unstructured grid. Such a coupling concept is illustrated in Fig.5.2. As can be seen from the figure, the time step of morphodynamic part (ANN model) can be much larger than the time step used in TELEMAC-2D/3D. For coupling TELEMAC-ANN, some subroutines in the TELEMAC-MASCARET system must be modified.

One of the advantages of these proposed models is that the calculation of sediment transport rate and bed shear stress are not required and the prediction of bed level evolution is only based on the bed elevation and hydrodynamic characteristics. This reduces the computational costs of simulation.

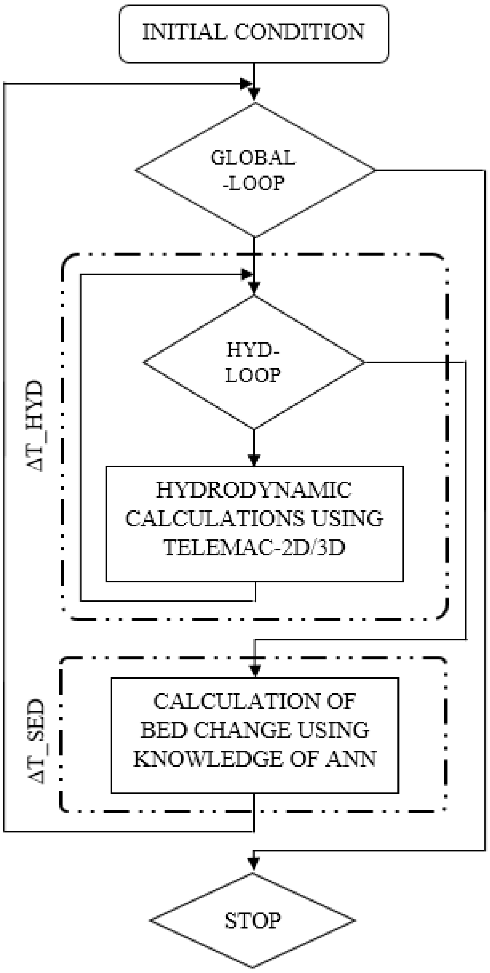
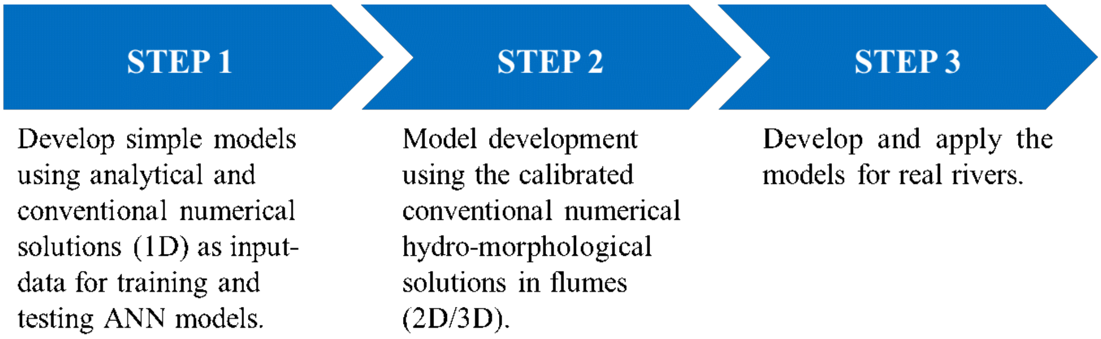

INITIAL CONDITION
GLOBAL -LOOP
ΔT_HYD
HYD- LOOP
HYDRODYNAMIC CALCULATIONS USING TELEMAC-2D/3D
ΔT_SED
CALCULATION OF BED CHANGE USING KNOWLEDGE OF ANN
STOP

Figure 5.2: The proposed hydro-morphodynamic-model system

Chapter 6

Evaluation of the model performance

To develop and evaluate the new proposed concepts for integrating ANNs into hydro-morphodynamic models, some steps have to be pursued. This research achieve this by applying very simple one-dimensional models and later more complicated two and three-dimensional models. Fig.6.1 shows a brief description of the steps which are carried out.



Develop simple models using analytical and conventional numerical solutions (1D) as input-data for training and testing ANN models.

Model development using the calibrated conventional numerical hydro-morphological solutions in flumes (2D/3D).

Develop and apply the models for real rivers.

Figure 6.1: Followed steps for model development

As can be seen from this figure, a very simple 1D model will be developed in the first step. We consider a straight channel with a finite amplitude perturbation of the bed level under steady flow discharge and uniform bed material. Under the assumption of constant water surface elevation, an analytical approximation based on the equation of conservation of sediment can be applied to generate data used for training and testing different ANN models. This step may become more complex by considering models with changing water surface elevation.

In the second step, two or three-dimensional models will be developed using the calibrated conventional numerical hydro-morphological solutions in flumes. The TELEMAC-MASCARET hydro-informatics system, which can couple Exner equation with SWE or RANS equations to simulate morphological bed level changes is used to generate the required data. Two different test cases will be considered and modeled in TELEMAC-MASCARET for this part. The first test case which is set up by the user, includes a straight channel in TELEMAC-3D modeling system with a finite amplitude perturbation of the bed level close to the center of the domain. The second test case models the evolution of the bed in a 180°channel bend under unsteady-flow conditions with uniform sediment (Yen and Lee, 1995). This is an available example in the software package. Chapter 9 provides more detailed information about each test case.

The obtained data then will be used to build the proposed hydro-morphodynamic models. After simulating each test case, the prediction qualities of the designed models are studied by evaluating several statistical parameters that describe the errors associated with the model. The statistical measure of goodness-of-fit between the estimated bed changes and those obtained from different numerical models was used to quantify this error.

The correlation coefficient (R) alone is unsuitable for evaluation of model prediction (e.g., Legates & McCabe, 1999). Legates and McCabe (1999) proposed that a perfect evaluation of the model performance should include at least one goodness-of-fit or relative error measure (e.g., coefficient of determination: R^2) and at least one absolute error measure (e.g. root mean square error: RMSE or mean absolute error: MAE). The mathematical formulas for calculating these parameters are as follows:

$$\mathrm{R}=\frac{n\sum x_i y_i-\left(\sum x_i\right)\left(\sum y_i\right)}{\sqrt{n\left(\sum x_i^2\right)-\left(\sum x_i\right)^2}\sqrt{n\left(\sum y_i^2\right)-\left(\sum y_i\right)^2}} \tag{6.1}$$

$$\mathrm{RMSE}=\sqrt{\frac{\sum\left(x_i-y_i\right)^2}{n}} \tag{6.2}$$

$$\mathrm{MAE}=\frac{\sum\left|x_i-y_i\right|}{n} \tag{6.3}$$

where n is number of pairs of data, x_i and y_i are i-th predicted and measured values, respectively. Here, the models' performances are evaluated based on the values of R^2 (or R), RMSE, and MAE. The R^2 expresses a degree of similarity between predicted and actual values and measures how well considered independent variables account for the variance of the measured dependent variable. Higher values correlate with greater model predictive capability, with R^2 values close to 1 indicating predicted and actual values to be very similar. The RMSE computes the square error of the prediction compared to actual values and computes the square root of the summation value. The RMSE is, thus, the average distance of a data point from the fitted line measured along a vertical line. In contrast to the RMSE, the mean absolute error, MAE, is a quantity used to measure how close predictions are to the measured outputs. The MAE computes average magnitude of error between predicted and actual values with no distinction between error direction. Low RMSE and MAE values indicate high confidence in the model-predicted values.

For more precise comparing the performance of different models, the present study additionally used two statistical indexes. The first index, mean absolute percentage error (MAPE) can be defined as follows:

$$\text{MAPE} = \frac{1}{n}\sum \frac{|x_i - y_i|}{|x_i|} \times 100\% \tag{6.4}$$

The MAPE is a statistical measure of predictive accuracy expressed as a percentage. The MAPE is useful for evaluating the performance of predictive models due to its relative values. MAPE effectively reflects relative differences between models because it is unaffected by the size or unit of actual and predicted values. To measure the generalization capability of the different learning rules, the non-dimensional error index (NDEI) is used as the second index, which can be defined as the RMSE divided by standard deviation of the target series (Jang, 1993).

If the first and second steps are successfully achieved, the models for real rivers could be developed and applied, however, this thesis does not deal with step 3, as the required data from real rivers are not available. Chapters 8, 9 and 10 essentially focus on development and evaluation of these new proposed concepts. However, efficiency and effectiveness of data-driven methods on river engineering issues will be tested first for the much simpler cases in chapter 7.

Chapter 7

Application of data-driven models on river engineering issues

This chapter validates the ability of data driven methods in the field of water engineering and sediment transport. To achieve this aim, two different validation test cases, are considered.

In the first test case, the accuracy of ANN and ANFIS's estimation of the maximum equilibrium depth of the contraction scour are analyzed. The developed networks are trained using the data set gathered by different investigators for long contractions under clear-water conditions. The designed ANN includes one hidden layer and seven nodes within that layer. Its hidden neurons use a hyperbolic tangent sigmoidal transfer function. The ANN model was implemented using the MATLAB software package. The importance of the individual input parameters was tested with a sensitivity analysis. This revealed the contraction ratio to be the most sensitive parameter, followed by the effect of armor layer formation for non-uniform sediments. For the designed ANN network, the training was based on the Levenberg-Marquardt algorithm in batch mode. The designed ANFIS was trained on the zero-order Takagi-Sugeno model with four bell-shaped membership functions for each input, and then the Levenberg-Marquardt algorithm was applied for network training. The ANFIS model was implemented using a FORTRAN-based computer code.

In the second test case, an ANFIS model trained with the Levenberg-Marquardt learn-

This chapter was published as:
Bui, M. D.; Kaveh, K., Penz, P.; Rutschmann, P. (2015a): Contraction scour estimation using data-driven methods. Journal of Applied Water Engineering and Research, 3(2), 143-156.
Kaveh, K.; Bui, M. D.; Rutschmann, P. (2017): A comparative study of three different learning algorithms applied to ANFIS for predicting daily suspended sediment concentration. International Journal of Sediment Research, 32(3), 340-350.

ing algorithm is considered for time series modeling of suspended sediment concentration in a river. The model is trained and validated using daily river discharge and suspended sediment concentration data from the Schuylkill River in the United States. The results of the proposed method are evaluated and compared with similar networks trained with the common Hybrid and Back-Propagation algorithms, which are widely used in the literature for prediction of suspended sediment concentration.

7.1 Contraction scour estimation

Scouring occurs due to several different reasons. One is the so called contraction scour which is often encountered in natural rivers due to channel contraction or river restoration structures. When the flow area is reduced by a natural contraction or bridge opening, the velocity and bed shear stress will increase as required by continuity and momentum considerations. The higher velocity results in an increased erosive force so that more bed material is removed from the contracted reach. As a consequence of which, the bed elevation is lowered and a scour hole develops over the general bridge cross section. Contraction scour is classified as either clear-water or live-bed. In the clear-water case, no sediment transport occurs upstream of the contraction, while in live-bed case, sediment is transported from upstream through the contraction scour area. Further, two different contraction types can be specified: the short one and the long one, according to the ratio of the length of the contraction to the width of the approaching flow. Fig.7.1 (Dey & Raikar, 2005) shows the schematic of a rectangular contraction, where d_s is equilibrium scour depth (m), L is length of contraction (m), h_1 is approaching flow depth, h_2 is flow depth in contracted depth (m), b_1 is approaching channel width (m), and b_2 is contracted channel width (m). In the literature different statements for the threshold of the ratio L/b_1, by which the contraction is designated as long or short, can be found. For example, Komura (1966) terms a contraction as long when values of $L/b_1>1$ are predominant, whereas Webby (1984) sees values of $L/b_1>2$ as relevant (Bui et al. 2015a).

7.1.1 Data collection

In the context of rivers and sediment transport, armoring describes a process by which an erosion-resistant surface layer is formed. This occurs on a river bed or a scour hole when forces of the flowing water remove finer particles and leave relatively large particles behind. The coarser bed material builds up a natural riprap like armor layer over

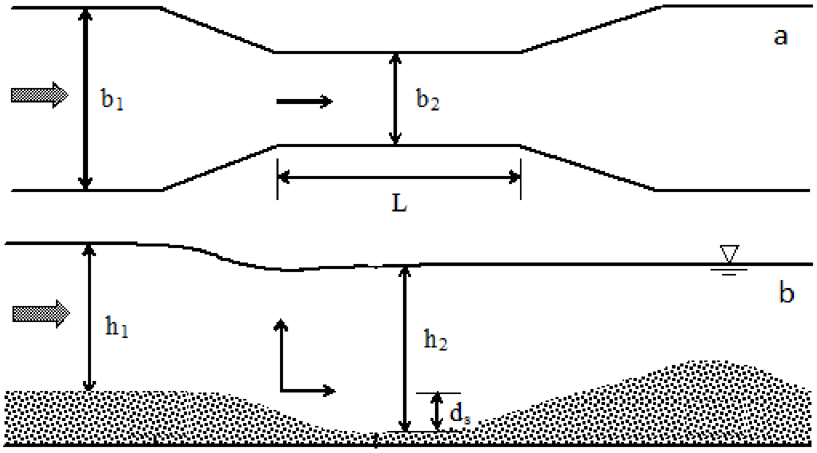


Figure 7.1: Schematic of a long rectangular channel contraction at equilibrium scour conditions; (a) top view, (b) side view

a heterogeneous mixture of sediment with a wide range of diameter. This armoring limits further scour development for a particular flow rate. In some cases, the formed armor layer can be destroyed again by higher levels of shear stress, leading to further development of the scour until a new armor layer is formed or the maximum scour is reached. Whether an armor layer is formed or not depends on the values of the geometric standard deviation (σ_g) of bed materials. According to Gessler (1971), armoring will arise for σ_g >2. Later Raudkivi and Ettema (1982) reported that the bed material will not armor for values of σ_g <1.5. An alluvial open channel is considered with rectangular cross sections and along contraction, which means the ratio of the length of the contraction to the width of the approaching flow is larger than one. The physical parameters influencing the equilibrium scour depth d_s (m) in a long contraction are the approaching flow velocity v_1 (m/s), the approaching flow depth h_1 (m), the density of water ρ (kg/m^3), the density of sediment ρ_s (kg/m^3), the acceleration of gravity g (m/s^2), the kinematic viscosity of water ν (m^2/s), the median sediment particle size d_m (m), the approaching channel width b_1 (m), contracted channel width b_2 (m), and geometric standard deviation of the grain-sized distribution σ_g (-) (Dey & Raikar, 2005).

The independent variables g, ρ, and ρ_s are presented as one combined parameter Δg. By using the Buckingham theorem for the dimensional analysis with repeating variables approaching flow velocity v_1 and width b_1, I can rewrite the relation between the 10 physical variables of the dimensional contraction scour form into the non-dimensional functional relation with only six dimensionless variables:

$$\begin{aligned}
&\overline{d_S} = \tfrac{d_S}{b_1};\ \overline{d} = \tfrac{d_m}{b_1};\ \overline{h} = \tfrac{h_1}{b_1};\ \overline{b} = \tfrac{b_2}{b_1};\ \overline{Fr} = \tfrac{v_1}{\sqrt{\Delta g d_m}};\ \Delta g = \left(\tfrac{\rho_S-\rho}{\rho}\right)\\
&\overline{d_S} = F\left(\overline{d}, \overline{Fr}, \overline{h}, \overline{b}, \sigma_g\right)
\end{aligned} \tag{7.1}$$

The channel opening ratio $\overline{b}$ shows the influence of geometric contraction on the degree of contraction scour. $\overline{d}$ represents the impact of sediment size on scour depth. $\overline{h}$ refers to the importance of approaching flow depth in scour depth. σ_g indicates the role of sediment gradation in scour depth and accounts for armoring in well-graded sediments. The densimetric Froude number $\overline{Fr}$ considers the effect of the mobility of submerged sediment particles on scour depth. These five parameters are considered as model inputs. The dimensionless equilibrium scour depth is considered as the only output parameter. More details about the derivation of Eq.(7.1) can be found in Dey and Raikar (2005).

Dey and Raikar (2005) have studied long contraction scour in a rectangular tilting flume in the Hydraulic and Water Resources Engineering Laboratory, Indian Institute of Technology. The channel was 12 m long, 0.6 m wide, and 0.7 m deep with a uniform contraction length of 1 m. The approaching channel width b_1 remains constant over the whole experiment (0.6 m), whereas the contracted channel width b_2 has four different sizes (0.24, 0.30, 0.36, and 0.42 m). The experimental data were differentiated according to the sediment type: sand or gravel. An overall number of 131 samples with relevant parameters for contraction scour were provided; of those, 95 samples were for gravel. To achieve a fully developed turbulent flow, the flume inlet was located 6 m away from the contraction entrance. After a slow initial filling of the flume, the tests ran until the equilibrium scour depth was reached. Equilibrium is attained when the time-averaged transport of bed materials into the scour hole equals what is removed from it. During all experiments, the average approaching flow velocities were kept at certain values, so that Clearwater conditions were guaranteed for all conducted tests. In the present paper, the experimental data conducted by Dey and Raikar (2005) and the data of other investigators (Komura 1966; Gill 1981; Webby 1984; Lim 1993) are used. The whole data set consists of 182 samples with the relevant parameters for contraction scour. In Table.7.1 the ranges of the different parameters are listed. They also represent the applicable domain of the data-driven models.

Additionally, since the geometric standard deviation has a range between 1.0 and 3.60, in some cases an armor layer could form during the scouring event. The data set has been divided randomly into three subsets, whereby the biggest amount of data (70%) is added randomly to the training subset. The remaining data set samples are

Table 7.1: Range of parameters

Limit	$d_m(mm)$	$v_1(m/s)$	$h_1(m)$	$b_1(m)$	$b_2(m)$	σ_g	$\overline{Fr}$	$d_S(m)$
Lower	0.350	0.1932	0.0240	0.400	0.100	1.00	1.1434	0.0100
Upper	14.25	0.9290	0.1366	1.586	0.524	3.60	3.2882	0.1626

used for testing (15%) and validating the networks (15%). The training subset is used to design the weights. The validation subset is used to test the accuracy of training while training is ongoing. After each epoch, the validation subset acts as a barometer for determining when the accuracy of the network is acceptable. After the network is considered to be optimally trained, the test subset is used to verify its performance. To achieve a better performance and faster training of the network, all data were normalized. The range after the data preprocessing is chosen to be between minus one (ra_{min} = -1) and one (ra_{max} = 1). To ensure this, Eq.(7.2) is utilized. For a simpler and more understandable comparison of the computed outputs with the targets, post-processing is also included. Here the outputs and targets are de-normalized again, thus set back to the ranges before preprocessing. For this purpose, Eq.(7.3) is applied.

$$x_{\text{pr}} = \frac{\left(ra_{\text{max}} - ra_{\text{min}}\right)\left(x - x_{\text{min}}\right)}{\left(x_{\text{max}} - x_{\text{min}}\right)} + ra_{\text{min}} \tag{7.2}$$

$$x_{\text{po}} = \frac{\left(x_{\text{pr}} - ra_{\text{min}}\right)\left(x_{\text{max}} - x_{\text{min}}\right)}{\left(ra_{\text{max}} - ra_{\text{min}}\right)} + x_{\text{min}} \tag{7.3}$$

where x_{pr} is the preprocessed variable, x_{po} the post-processed variable, x the original variable, and (ra_{min},ra_{max}) is the range (Bui et al. 2015a).

7.1.2 Network design

For many problems of hydraulic engineering, a feedforward ANN is considered as an effective tool which is also applied in the present paper. These networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons or nodes. The reason for this is that multiple layers of neurons with nonlinear activation functions allow the network to learn nonlinear relationships between input and output vectors. This type of configuration does not constrain the output result to a fixed range of values, but instead grants the network the freedom to represent any possible outcome. Learning or training algorithms decide how the weights used in configured ANN architectures are adjusted to minimize the output errors of a particular data set. This is measured by the generalization ability of the input-output mapping computed by the ANN. At present, there are numerous learning algorithms available for different network configurations and applications. Miscellaneous training algorithms can propose different equations that modify some of the weights of processing elements in response to input and output values. One of the most widely used algorithms for training a feedforward ANN is the Levenberg-Marquardt algorithm with the mean squared error (MSE) being used as an error function. The MSE computes the similarity of the prediction compared to actual values. This tool is efficient at assessing undesirably large differences. The Levenberg-Marquardt algorithm is employed in the scope of this work. Besides this, an early stopping algorithm and the Nguyen-Widrow initialization for weights were utilized. More details about ANN approaches with these methods can be found in Penz (2013).

Since the appropriate number of hidden layers and dependent nodes for the models is not known, a trial and error method was used to find the best network's configuration. An optimal architecture was determined by minimizing the difference between neural network predicted values and desired outputs for various numbers of hidden layers and neurons. The training of the neural network models was stopped when either the acceptable level of error was achieved, or the number of iterations exceeded a prescribed value. After several runs of this analysis, the neural network model configuration that minimized the MAE and RMSE and optimized the R was selected as optimal. The ANN architecture was modified by changing the number of hidden layers and its neurons, the initial weights, as well as the type of input and output functions. Each modification was tested with one hundred trials, which served as the basis for performance assessment of mean values. To compare these mean values, different modified architectures were evaluated. After extensive trial and error processes, an optimal ANN for maximum equilibrium contraction scour modeling under clear-water conditions was found with the simplest multilayer feedforward network resulting in the one hidden layer performing clearly better than the alternatives. Further, for this single hidden layer, the optimal number of neurons was determined to be seven. The weight function was set as multiplication, whereas the net input function was chosen according to trial-and error. A combination of the hyperbolic tangent sigmoid function in the hidden neurons with the linear transfer function for the output neuron was determined as the best alternative.

By applying the designed ANN model, the values of the weights and biases have been specified after a successful learning process. They represent the stored knowledge of the ANN for contraction scour depth modeling. The best designed network utilizes all 50 defined parameters, which are collected in one input weight matrix $\mathbf{IW_{1,1}}$, one hidden layer weight matrix $\mathbf{LW_{2,1}}$, one bias vector $\vec{\mathbf{b_1}}$, and one bias value b_2 with the following values:

$$\mathbf{IW_{1,1}} = \begin{bmatrix} -61.7540 & 0.7590 & 16.0804 & -2.2736 & -1.3754 \\ -36.1590 & -1.9702 & -18.4141 & 3.2689 & 3.0961 \\ 12.5222 & -.0.0331 & -14.5885 & 8.0144 & -0.2061 \\ 112.9620 & 1.4127 & 2.7952 & -5.7359 & -1.0236 \\ -26.4800 & 0.0029 & 13.6386 & 8.1508 & -0.0300 \\ 128.5200 & -1.6282 & 8.8843 & -0.1999 & 0.2204 \\ -65.1840 & -1.1853 & -12.4488 & -0.9877 & -0.6407 \end{bmatrix} \tag{7.4a}$$

$$\begin{gathered} \mathbf{IW_{2,1}} = \Big[\ 0.0276 \quad -0.0258 \quad -0.0726 \\ 0.0241 \quad -0.0312 \quad 0.2775 \quad -0.0295] \end{gathered} \tag{7.4b}$$

$$\vec{\mathbf{b_1}} = \begin{bmatrix} 2.1017 \\ 1.7659 \\ -0.0878 \\ 0.1017 \\ -7.2220 \\ 4.7620 \\ 3.9530 \end{bmatrix} \tag{7.4c}$$

$$b_2 = [-0.1876] \tag{7.4d}$$

Using the designed network, the following equation is received for the normalized equilibrium scour depth:

$$(\overline{d_s})_{pr} = \mathbf{LW}_{2,1} \times tansig\left(\mathbf{IW}_{1,1} \times \begin{bmatrix} \overline{d} \\ \overline{Fr} \\ \overline{h} \\ \overline{b} \\ \sigma_g \end{bmatrix}_{pr} + \vec{\mathbf{b_1}}\right) + b_2 \tag{7.5}$$

The dimensionless value of the scour depth $\overline{d_S}$ can be then calculated using Eq.(7.5). Based on this value and the approaching channel width $\boldsymbol{b_1}$, the absolute scour depth d_s can be defined.

For the ANFIS model, the same dimensionless inputs and output as well as the data sets are applied. A two-step process is used for faster training and to adjust the network parameters. When following a hybrid learning procedure, the premise parameters are kept unchanged in the first step, and the information is propagated forward in the network to layer 4, where a least-squares estimator identifies the consequent parameters. In the second step, the backward pass, the chosen parameters are held fixed while the error is propagated. In MATLAB, the method of either back-propagation or a combination of least-squares estimation and Back-Propagation could be applied for estimating and modifying membership function parameters. Different ANFIS models were established to estimate maximum equilibrium contraction scour. Each model had different numbers of membership functions. The triangle, Gaussian, and generalized bell functions were applied to each model. After an extensive trial-and-error search for various networks, the optimal zero-order TS ANFIS model, which has four bell-shaped membership functions for each input, was found. The training method is based on the Levenberg-Marquardt algorithm with a combination coefficient of 10^{-4} (Bui et al. 2015a).

7.1.3 Results and discussion

Although there are different parameters (input variables) that affect the contraction scour, only some are of primary significance. To determine the effect of each individual variable, a sensitivity analysis was performed. For this purpose, the ANN structure with the best validation performance was used. Table.7.2 provides a comparison between different ANN models having one of the independent variables removed in each case. It can be seen the contraction ratio has the most effect on the contraction scour. This is in accordance with the contraction scour formula of Lim and Cheng (1998) which proposes that the equilibrium scour depth is only a function of the ratio of contracted to uncontracted channel width. Another sensitive parameter, the standard deviation of the size distribution of the bed materials, stood out. On the contrary, the median sediment particle size seems to have the least influence on the equilibrium scour depth.

Table 7.2: Sensitivity analysis of input variables

Performance	Omission of variables				
	d_m/b_1	$\overline{Fr}$	h_1/b_1	σ_g	b_2/b_1
R	0.958	0.955	0.939	0.906	0.581
RMSE (m)	0.017	0.018	0.019	0.024	0.046
MAE (m)	0.012	0.013	0.014	0.016	0.037

To ensure physical plausibility of the designed ANN model, a sensitivity analysis was carried out, where the contraction ratio and the standard deviation of the size distribution of the bed materials in the test data set were altered and the effect on the modeled scour depth was assessed. The following four scenarios were considered:

1. Contraction ratio increased by 15%, other inputs unchanged,

2. Contraction ratio decreased by 15%, other inputs unchanged,

3. Standard deviation of bed materials increased by 15%, other inputs unchanged,

4. Standard deviation of bed materials decreased by 15%, other inputs unchanged.

As expected, by increasing the contraction ratio, the long contraction reduces its equilibrium scour depth, and decreasing the standard deviation of the bed materials increases the equilibrium scour depth. The nonlinearity of the response curves also provides further information; for example, the sensitivity of the equilibrium scour depth to the contraction ratio is much greater than that to the standard deviation of the bed materials. The results of the sensitivity analysis ensured physical plausibility of the designed ANN model.

Fig.7.2 shows the response curves pertaining to the effect that changing the contraction ratio (b) by -15% and +15% and changing the standard deviation of the bed materials (σ_g) in the same percentage range, while other input variables were unchanged, has on scour depth.

The curve in Fig.7.3 shows the network's response for a continuous variation of contraction ratios $\overline{b}$ in the range of the whole data (between 0.4 and 0.7). In this case, mean values for the remaining parameters were used: $\overline{d} = 0.0126$, $\overline{h} = 0.171$, $\sigma_g = 2.3$, and $\overline{Fr} = 2.2158$. The calculated results repeatedly confirmed the physical plausibility of the designed feed-forward model: by decreasing the contraction ratio, the long contraction increases the equilibrium scour depth.

Further, it should be emphasized again that the previous empirical equations have been developed principally based on the simplified one-dimensional theory of Straub (1934) and limited experimental data. Laursen (1963) assumed that the shear stress in the contracted section reaches its critical value at the end of the scouring process in a long contraction under clear-water scour condition. Using Manning's equation for the approach flow and contracted flow combined with the continuity equation, he obtained the clear-water contraction scour formula. Modifying this concept, Richardson and Davis (2001) proposed the clear-water contraction scour equations for homogeneous

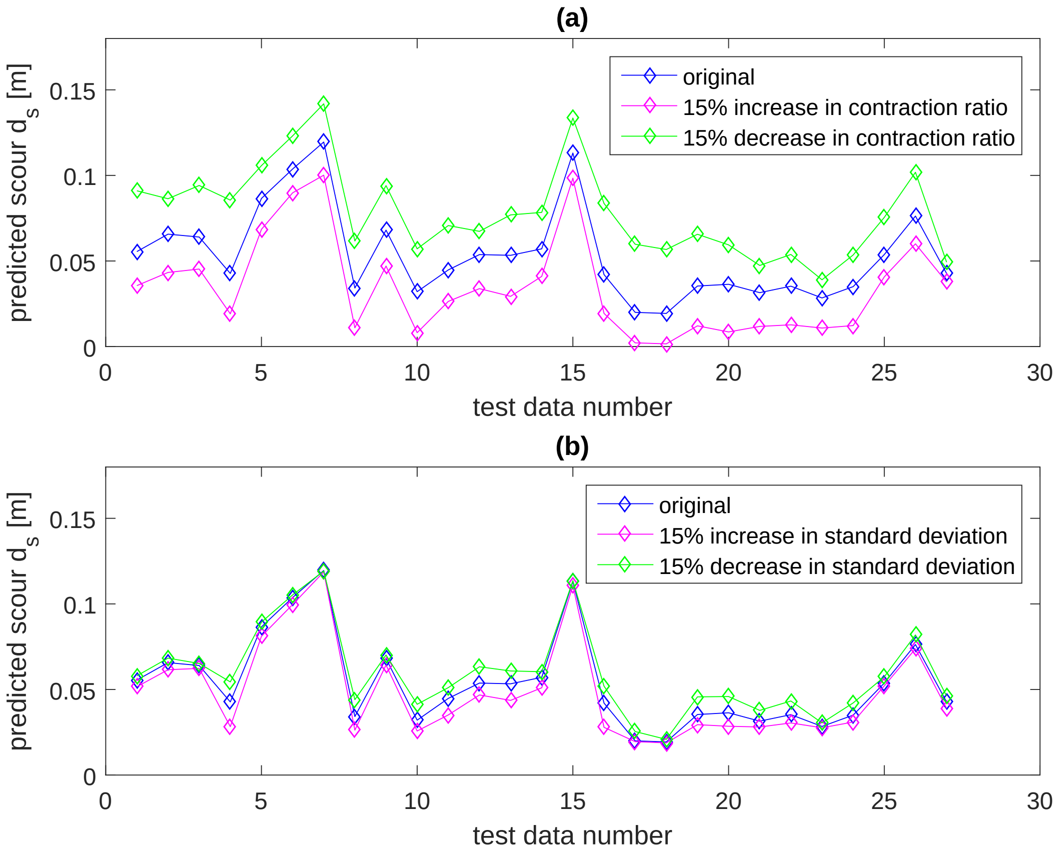


Figure 7.2: Response curves for the effect of the contraction ratio (a), and the standard deviation of the size distribution of the bed materials (b) on the modeled contraction scour depth (Bui et al. 2015a)

bed materials where the scour depth depends on the flow discharge, the contraction width, the upstream water depth, and the effective mean bed material size. Komura (1966) emphasized the influence of armoring on live-bed scour depth by arguing that the ratio of the sediment sizes in the approach flow section and contracted section influences the contraction scour depth for large contraction ratios and geometric standard deviation of sediment size distribution.

Through dimensional analysis of several laboratory experiments involving live-bed and clear-water contraction scour in a long contraction, he proposed a dimensionless scour depth formula that depends on the approach flow Froude number, the ratio of channel widths, and geometric standard deviation of sediment size distribution. Lim and Cheng (1998) derived a long contraction scour formula for live-bed scour along the same lines as that of Gill (1981) using a simple bed-load formula that assumes the sediment transport rate is proportional to excess shear stress. Eventually, they showed

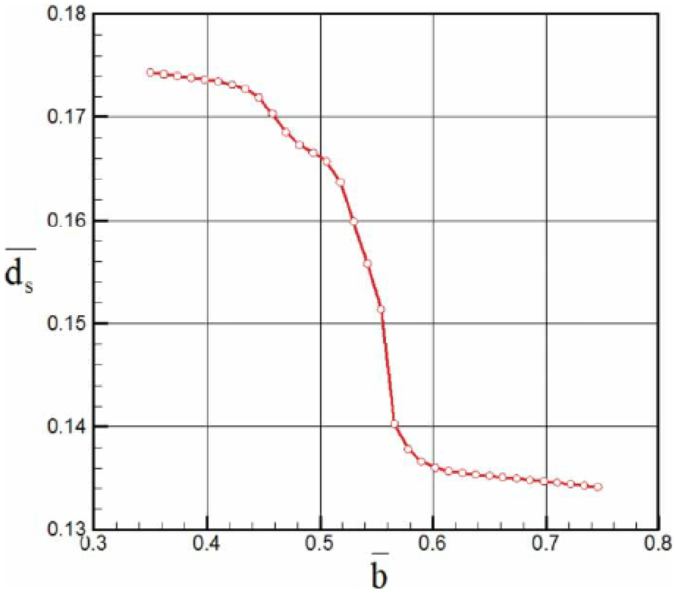


Figure 7.3: Response of the designed feedforward network for a continuous variation of contraction ratios (Bui et al. 2015a)

that the only solution of the equation was one in which the contraction scour depth depends on the ratio of channel widths alone. In the work of Dey and Raikar (2005), a detailed parametric investigation on scour depth in long contractions for uniform and non-uniform sediments under clear-water scour has been carried out. These own experimental data and the data conducted by other authors have been used to determine a new equation of maximum equilibrium clear-water scour depth. They defined that the characteristic parameters affecting the maximum equilibrium non-dimensional scour depth (scour depth - approaching flow depth ratio) are the Froude number of the excessive approaching flow, sediment size - approaching flow depth ratio, and channel width ratio. Their results showed a significant effect of sediment with a minimum value of scour depth due to armoring given as 25% of the value for uniform sediment. From the aforementioned studies, it is revealed that apart from the work conducted by Dey and Raikar (2005) a detailed investigation describing the effects of various parameters on scour depth in long contractions is inadequate. Also, little attention is paid to study the scour depth in long contractions with non-uniform sediment.

Mohammad and Al-Saffar (2010) applied a feed-forward ANN model with six inputs including the median particle diameter, the approaching flow velocity, ratio of the approaching flow velocity to the critical velocity for sediments, the approaching flow depth, the approaching channel width, and the contracted channel width. The output is the equilibrium scour depth. However, the effect of sediment gradation has not been considered in their model. It is not clear whether only uniform sediment data

have been used for the development of their model. Their designed ANN model then consists of one hidden layer with 10 neurons. Logistic activation functions are applied for hidden and output layers. Since the information about the designed ANN model and final equation for scour depth is not sufficiently detailed in their paper, comparing the performance of their model to that of our model is not possible.
To evaluate the performance of the new data-driven methods in predicting equilibrium scour depth, a comparison between the new models and four of the existing formulae, which have been proposed by Komura (1966), Lim and Cheng (1998), Richardson and Davis (2001), and Dey and Raikar (2005), was undertaken using the same 182 cases' observed data set. For this purpose, observed equilibrium scour depth values are plotted against the predicted one. Fig.7.4 and Table.7.3 illustrate the correlation for the different contraction scour formulae and the chosen networks with the statistical performance indices between predicted and observed data.

Table 7.3: Performance indices of various approaches for the whole data set

	ANFIS	ANN	Dey and Raikar	Richardson and Davis	Komura	Lim and Cheng
R	0.976	0.965	0.836	0.853	0.790	0.748
RMSE (m)	0.013	0.015	0.035	0.054	0.170	0.048
MAE (m)	0.005	0.006	0.013	0.028	0.084	0.016

Comparing Fig.7.4(e) with (f) and looking at Table.7.3 again confirm the statement of Legates and McCabe (1999) that the correlation coefficient R alone is unsuitable for model evaluation. The formula of Komura (1966) provided a better correlation coefficient R = 0.788, but a worse RMSE = 0.170 m. The corresponding correlation plot in Fig.7.4(e) shows a principal overestimation up to 200% of the Komura scour formula, whereas for small contraction scour depths the formula underestimates. The correlation plotted for the formula developed by Richardson and Davis (2001) is shown in Fig.7.4(c) (Bui et al. 2015a).

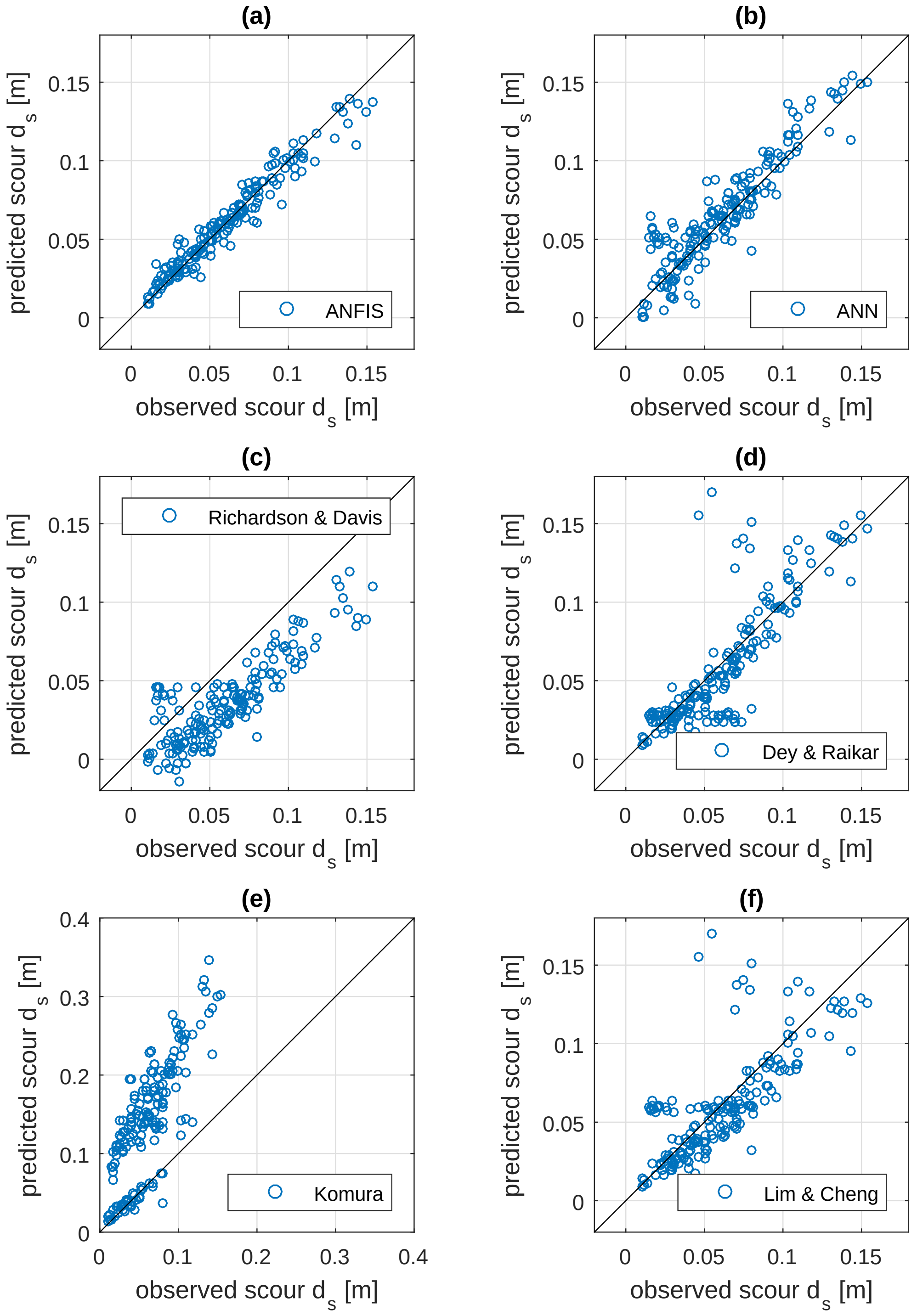


Figure 7.4: Comparison of the equilibrium scour depths computed using different methods with the experimental data for the whole data set (Bui et al. 2015a)

It is apparent that of all the compared equation Richardson and Davis (2001) is the most underestimating. Almost all points in the plot lay below the line of perfect agreement in combination with a RMSE value (RMSE = 0.054 m). Despite its simplicity that the scour depth is only dependent on the contraction ratio of the channel, the formula of Lim and Cheng (1998) has the lower RMSE = 0.048 m. From the corresponding correlation plot in Fig.7.4(f), again the tendency for underestimating is evident.

Furthermore, some outliers can be identified, which may result from neglecting a variable that would represent armoring for non-uniform sediments. Comparing all existing formulae based on the statistical values R, MAE, and RMSE proves the formula of Dey and Raikar (2005) shows a better performance. Comparing the performance qualities of the contraction scour formulas with those of the chosen ANN and ANFIS reveals that the recommended networks have the highest R and lowest RMSE values. The new models offer improved predictions of scour depth. When applying the whole data set for the best existing method (Dey & Raikar 2005), the RMSE was 0.035 m, compared to the RMSE of 0.008 (m) for the new ANFIS model. Corresponding values of the correlation coefficient (R) are 0.836 and 0.976.

Fig.7.5 and Table.7.4 present a graphical and statistical comparison between different methods based on the test data. Again, it is clear that ANFIS and ANN models provide better predictions than conventional methods. A more detailed description of the application of the newly proposed hybrid and Back-Propagation with momentum algorithms on ANFIS for contraction scour prediction can be found in (Bui et al. 2017) and (Kaveh et al. 2015a; 2015b).

Table 7.4: Performance indices of various approaches for the test data set

	ANFIS	ANN	Dey and Raikar	Richardson and Davis	Komura	Lim and Cheng
R	0.982	0.966	0.838	0.770	0.784	0.552
RMSE (m)	0.006	0.009	0.017	0.029	0.098	0.030
MAE (m)	0.004	0.007	0.011	0.027	0.085	0.020

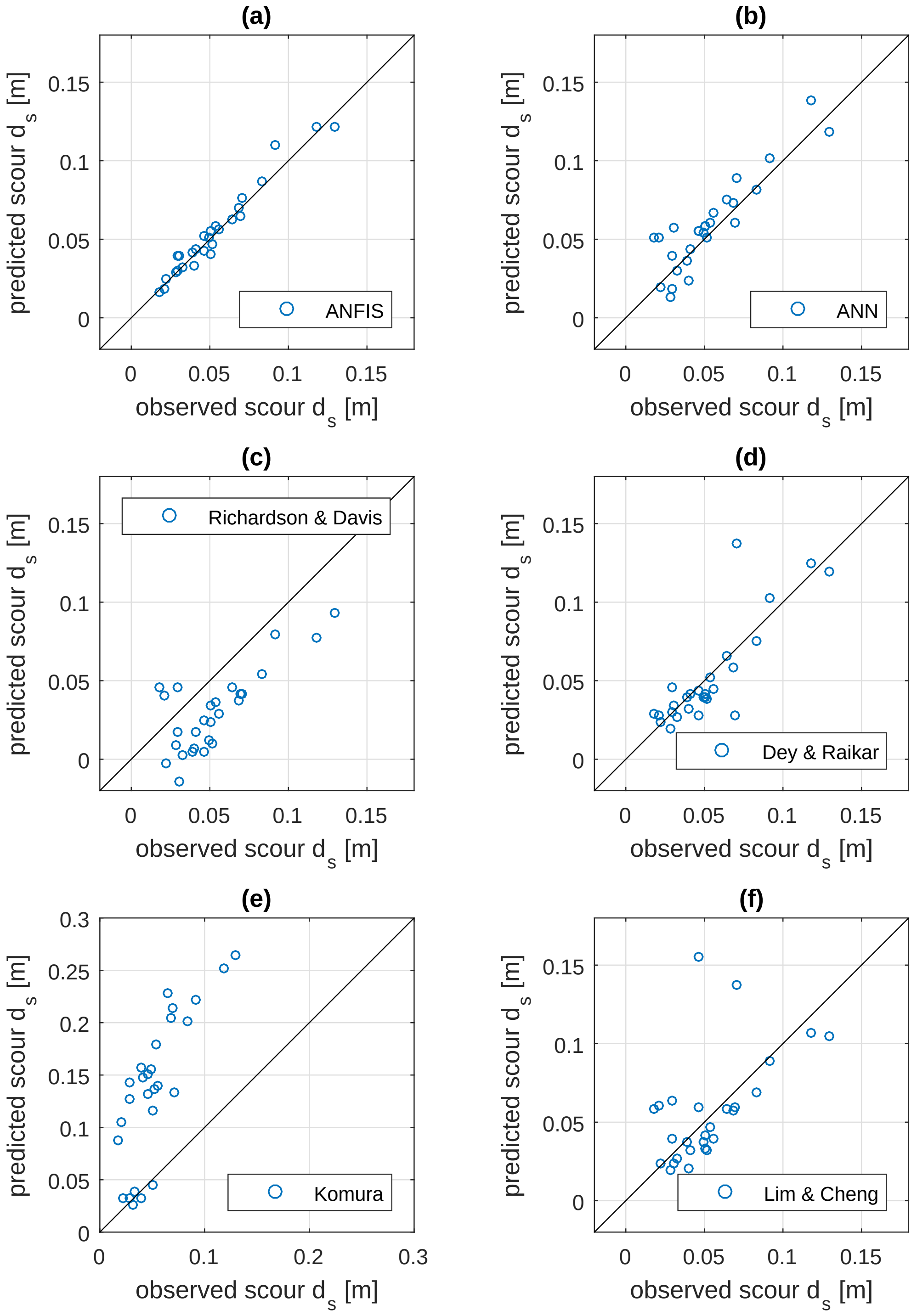


Figure 7.5: Comparison of the equilibrium scour depths computed using different methods with the experimental data for the test data set (Bui et al. 2015a)

7.2 Predicting of daily suspended sediment concentration

Suspended load refers to the sediment that is lifted by the upward components of turbulent currents and remain suspended for a considerable length of time. In recent decades, numerous studies have been done on the modeling of sediment transport processes. However, due to the large number of difficulties to measure parameters involved in this phenomenon, the theoretical governing equations may not be of much advantage in gaining knowledge on the overall process. Studies have been done to reduce the complexities of the problem in terms of developing practical techniques that do not require dwell on algorithms and/or theory. Among such techniques, classical time series models, such as Multiple Linear Regression (MLR: an approach for modeling the relation between a scalar dependent variable and one or more explanatory variables) and Auto Regressive Integrated Moving Average (ARIMA), are widely used for hydrological time series forecasting (Salas et al. 1980). However, these models are basically linear models that assume data is stationary, and prove to have a limited ability in capturing nonlinearities in hydrological and environmental data. In recent years, the use of artificial intelligence approaches is increasing due to their capability. Artificial Neural Networks (ANN) has been successfully applied in several diverse fields including water resources. Nagy et al. (2002) developed an ANN model to estimate Suspended Sediment Concentration (SSC) in rivers, achieved by training the ANN model to extrapolate several streams data collected from reliable sources. The network was set up using several parameters, such as Froude number, stream width ratio, mobility number, and Reynolds number, as the input pattern and the SSC as the output pattern. Raghuwanshi et al. (2006) proposed an ANN model for runoff and sediment yield modeling in the Nagwan watershed in India. A five-year data set was used for training and a two-year data set was considered for testing the model. Linear regression based daily and weekly runoff and sediment yield prediction models were also developed using the previously mentioned data set and tested using the testing data set. The ANN model performed better than the linear regression model in predicting both runoff and sediment yield on daily and weekly simulation scales. Zhu et al. (2007) proposed an ANN model for simulating the monthly suspended sediment flux in the Longchuanjiang River in China, whereby suspended sediment flux was related to the average rainfall, temperature, rainfall intensity, and flow discharge. Results illustrated that the ANN model is capable of simulating monthly suspended sediment flux with fairly good accuracy when considering proper variables and their correlation to the previous month suspended sediment flux.

Despite the suitable flexibility of ANN in modeling hydrologic time series, sometimes it may be difficult to train an ANN when signal fluctuations are highly non-stationary, and the physical hydrologic process operates under a large range of time scales varying from one day to several decades. In such an uncertain situation, the Fuzzy Inference System (FIS) may be applied in the estimation of uncertainties in the real situations. Developing Hybrids of an ANN and a FIS, called Neuro-Fuzzy (NF) systems, is a current research focus, that makes use of the advantages of both ANN and FIS. NF systems can capture the benefits of both these techniques in a single framework and have been applied to a number of problems in water resources and environmental engineering, including ecological status simulation in surface waters (Ocampo-Duque et al., 2007), as well as river flow modeling (Zounemat-Kermani and Teshnehlab, 2008). There are few studies of the application of fuzzy logic and NF algorithms in prediction of sediment transport. Kisi et al. (2009) studied the accuracy of an adaptive neuro-fuzzy computing technique for monthly suspended sediment prediction at the Kuylus and Salur Koprusu stations in the Kizilirmak Basin in Turkey. The results of their study illustrated that the NF algorithm provided better performance than the ANN and Sediment Rating Curve (SRC) models. Rajaee et al. (2009) studied ANN, NF, MLR, and SRC models for daily simulation of SSC for two hydrometry stations. The models were trained using daily river discharge and SSC data for the Little Black River and Salt River gauging stations in the USA. Comparison of the models' results indicated that the NF model better predicts SSC in comparison to other methods.

Several papers have been published that use the Levenberg-Marquardt algorithm to train an ANN for SSC prediction (Kisi, 2004; Rajaee et al., 2009). However, to the knowledge of the authors, no work has been reported in the literature that applies the Levenberg-Marquardt (LM) learning algorithm (Jang & Mizutani, 1996) to an Adaptive Neuro-Fuzzy Inference System (ANFIS) model for prediction of daily SSC. Therefore, the main objective of this paper is to investigate capability and accuracy of ANFIS model trained with the LM algorithm in estimating SSC. Since the MATLAB program supports only the common Hybrid and Back-Propagation algorithms for ANFIS, a computer program system written in the FORTRAN language has been developed for the LM algorithm (Kaveh et al. 2017).

7.2.1 Data collection

Data obtained from the Schuylkill River gauging station at Manayunk, Philadelphia, PA (Station No: 01473800, Basin Area = 4740 km^2, Latitude: 40^o 01'41" N and Longitude: 075^o 13'44" W) operated by the U.S. Geological Survey (USGS), was used to train and test all the models developed in this study. For this station, daily time series of river discharge (Q$_t$) and suspended sediment concentration (SSC$_t$) were downloaded from the web server of the USGS (http://co.water.usgs.gov/sediment/seddatabase.cfm).

Three subsets of the data -for training, validation and testing- are required to build the model. To achieve this, the data from January 1, 1949 to December 31, 1953 (83% of total data) were used for training and validation, and the data from January 1, 1954 to December 31, 1954 (17% of total data) were used as the testing set. Fig.7.6 shows the time series of data related to daily flow discharge and suspended sediment concentration.

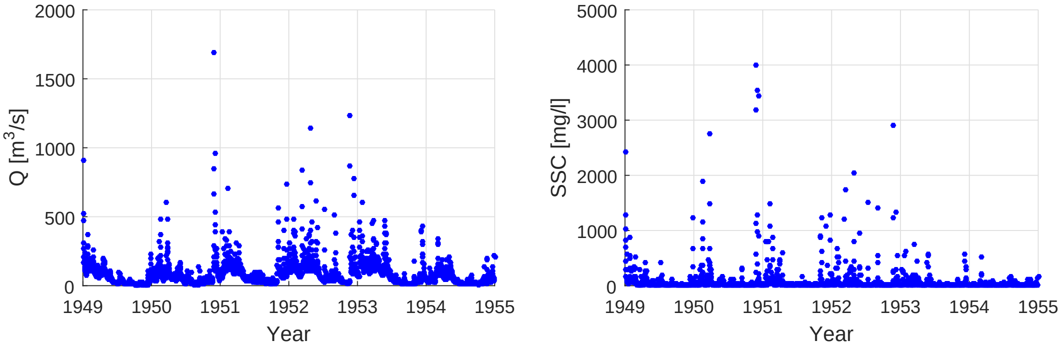


Figure 7.6: Discharge and SSC time series for the Schuylkill River Manayunk, Philadelphia, PA (Kaveh et al. 2017)

The statistical analysis for training, validation, testing, and all data is listed in Table.7.5 which includes the minimum, maximum, mean, standard deviation (S_d), skewness coefficient (Cs), and autocorrelations from 1 day lag to 3 day lag (R_1, R_2, and R_3). It should be noted that data-driven methods (e.g., ANN or ANFIS) perform best when they do not extrapolate beyond the range of data used for model training and the extreme values of the available data must be included in the training set.

It can be seen from Table.7.5 that the extreme values of Q and SSC are in the range of the training set. Skewness coefficients are low for all data sets. This is appropriate for modeling, because a high skewness coefficient has a considerable negative effect on model performance (Altun et al., 2007). At this river station, flow discharge (Q) is highly autocorrelated while the autocorrelation coefficient for SSC is lower.

Table 7.5: Statistical analysis for training, validation, testing, and all data sets

Statistical parameters	Training set		Validation set		Testing set		All data	
	Q (m^3/s)	SSC (mg/l)	Q (m^3/s)	SSC (mg/l)	Q (m^3/s)	SSC (mg/l)	Q (m^3/s)	SSC (mg/l)
Min	5.38	1.0	14.5	3.0	9.85	2.0	5.38	1.0
Max	1690	4010	603	750	337	508	1690	4010
Mean	95.96	92.44	107.38	43.99	52.31	21.45	90.62	72.55
S_d	120.88	292.74	100.89	94.45	47.73	37.13	110.08	244.29
C_s	4.847	7.569	1.623	4.557	2.362	7.695	4.656	8.949
R_1	0.733	0.535	0.836	0.522	0.841	0.570	0.758	0.542
R_2	0.474	0.236	0.676	0.214	0.678	0.366	0.522	0.247
R_3	0.376	0.196	0.624	0.213	0.576	0.167	0.433	0.208

In general, this table shows relatively similar statistical characteristics between the data sets, especially the validation and testing sets in terms of autocorrelation coefficients. The correlation coefficients between observed suspended sediment concentration and river discharge time series are calculated to choose suitable input patterns for ANFIS models. The results are listed in Table.7.6. It is obvious that SSC_t has relatively higher linear correlations with Q_t, Q_{t-1}, Q_{t-2}, SSC_{t-1}, and SSC_{t-2}.

The original data sets consist of different variables with different physical meanings and units, and, thus, their ranges are highly variable. To ensure that each variable is treated equally in a model and to also make them dimensionless, all variables are preprocessed by scaling them between 0 and 1. The following simple linear mapping of the variables is most commonly used for this purpose. Input and output variables for the present study are rescaled using this formula.

$$x_{\mathrm{pr}} = \frac{(x - x_{\mathrm{min}})}{(x_{\mathrm{max}} - x_{\mathrm{min}})} \tag{7.6}$$

where x_{pr} is the rescaled value of the original variable x, and x_{min} and x_{max} are the minimum and maximum of variable x, respectively (Kaveh et al. 2017).

Table 7.6: The correlation coefficients between observed SSC_t and Q_t for all data

Parameter	Q_t	Q_{t-1}	Q_{t-2}	Q_{t-3}	Q_{t-4}	SSC_{t-1}	SSC_{t-2}	SSC_{t-3}	SSC_{t-4}	SSC_t
Q_t	1.0	0.7578	0.5224	0.4334	0.3852	0.6081	0.3355	0.2507	0.2150	0.7929
Q_{t-1}		1.0	0.7577	0.5229	0.4336	0.7930	0.6082	0.3357	0.2510	0.4373
Q_{t-2}			1.0	0.7584	0.5235	0.4375	0.7932	0.6082	0.3364	0.2219
Q_{t-3}				1.0	0.7584	0.2225	0.4382	0.7929	0.6081	0.1721
Q_{t-4}					1.0	0.1732	0.2243	0.4416	0.7941	0.1134
SSC_{t-1}						1.0	0.5375	0.2452	0.2082	0.5372
SSC_{t-2}							1.0	0.5381	0.2472	0.2443
SSC_{t-3}								1.0	0.5420	0.2072
SSC_{t-4}									1.0	0.1322
SSC_t										1.0

7.2.2 Network design

ANN and NF models are applied as effective approaches to handle nonlinear and noisy data, especially in situations where the relations among physical processes are not fully understood. They are also particularly well suited for modeling complex systems on real time basis (Rajaee et al., 2009). The aim of this research is to investigate the efficiency of the ANFIS model trained with the LM learning algorithm for predicting SSC one day ahead. With respect to the statistical analysis presented in Table.7.6, the following combinations with differing numbers of input values (Q and SSC) were considered to predict the unique SSC value one day ahead of time t. Since the variables at times t-3 and t-4 show lower correlation, these variables are ignored in the following combinations:

- Combination 1: Q_t, SSC_{t-1}
- Combination 2: Q_t, Q_{t-1}, Q_{t-2}, SSC_{t-1}, SSC_{t-2}
- Combination 3: Q_{t-1}, SSC_{t-1}
- Combination 4: Q_{t-1}, Q_{t-2}, SSC_{t-1}, SSC_{t-2}
- Combination 5: Q_t, Q_{t-1}, SSC_{t-1}, SSC_{t-2}
- Combination 6: Q_t, SSC_{t-1}, SSC_{t-2}

The applied ANFIS models use a fuzzy inference model of Sugeno type (Jang et al., 1997), in which the membership function parameters are tuned to fit a given input-output set by optimization algorithms. In the ANFIS model each rule contains several parameters of membership functions (MFs) and each variable may have several values (in terms of rules). For example, if each variable has two rules and each rule includes three parameters, then there are [n (variables) × 2 (rules) × 3 (parameters)] = 6n parameters for the determination in layer 1. The ANFIS model trains these MFs according to the training data set. In layer 2, these rules generate 2n neurons, and furthermore, there are 2n × (n + 1) parameters undetermined within the defuzzification process in layer 4. Choosing the number of MFs for each input reflects the complexity of selecting parameters for the ANFIS model. In this research, three different ANFIS models are considered for predicting SSC values in the river. In all these models, three Gaussian membership functions for each input are found to provide the best model performance. The differences between the models lie in their training algorithms, namely the common Hybrid, Back-Propagation, and Levenberg-Marquardt algorithms (Kaveh et al. 2017).

7.2.3 Results and discussion

The results of the model performance are listed in Table.7.7. In the testing phase, many combinations had the LM algorithm yield better performance than the Hybrid algorithm, while for some others the Hybrid algorithm performs better. However, for all combinations, the Hybrid and LM learning algorithms have better performance in comparison to the BP algorithm.

According to Table.7.7, in the Schuylkill River, the ANFIS model trained with the LM algorithm provides the best efficiency using combination 2 with the highest value of R^2 = 0.7513 and the lowest RMSE = 25.955 mg/l and MAE = 11.859 mg/l. For ANFIS model trained with the Hybrid algorithm the results are a little more complex, from R^2, RMSE, and MAE viewpoints, it shows the best performance for combinations 5, 6, and 2, respectively. Using the LM algorithm, the values of R^2, RMSE, and MAE vary in the ranges of 0.3103 to 0.7513, 21.467 to 47.709 mg/l, and 10.938 to 24.826 mg/l, respectively. Using the Hybrid algorithm they are in the ranges of 0.3273 to 0.7186, 26.639 to 41.945 mg/l, and 14.575 to 25.271 mg/l, respectively. In the training phase, except for only one combination, all RMSE values for the LM algorithm are lower than those for the Hybrid and BP algorithms. Similar to the testing phase, the LM algorithm provides better R^2 values for 4 combinations.

Table 7.7: Performance of the algorithms for SSC estimation in the Schuylkill River, at Manayunk, Philadelphia, PA

Input	ANFIS-Hybrid			ANFIS-BP			ANFIS-LM		
combination	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE
	Test data								
1	0.5981	27.900	17.446	0.5759	43.167	35.264	0.6917	25.955	11.859
2	0.6603	38.133	14.575	0.5788	44.908	34.690	0.7513	21.467	10.938
3	0.3530	36.555	20.925	0.2934	53.810	45.841	0.3415	36.488	22.004
4	0.3273	41.945	25.271	0.2515	58.159	46.255	0.3103	47.709	24.826
5	0.7186	41.407	16.146	0.5814	44.416	34.634	0.7510	32.685	14.713
6	0.5975	26.639	17.211	0.5776	42.776	34.387	0.6400	33.691	24.041
	Validation data								
1	0.6243	25.123	16.523	0.6419	41.541	32.645	0.7213	23.185	11.078
2	0.9187	17.328	12.868	0.6848	39.578	30.085	0.9107	14.842	9.1580
3	0.3041	39.813	23.715	0.2984	54.012	47.132	0.3534	35.861	22.176
4	0.4251	34.675	20.312	0.2816	57.182	44.274	0.3864	48.193	26.826
5	0.8827	33.198	14.185	0.5756	43.437	34.032	0.8845	27.345	13.864
6	0.8075	24.689	16.197	0.5798	45.781	36.842	0.8108	28.345	21.175
	Training data								
1	0.6908	228.559	56.588	0.7326	157.622	70.566	0.7821	115.741	63.342
2	0.9686	51.368	19.488	0.7559	145.136	60.743	0.9856	46.873	16.524
3	0.3449	234.617	69.068	0.3275	238.986	89.222	0.3404	221.891	64.573
4	0.4482	215.326	63.871	0.3483	235.733	86.945	0.4298	218.452	62.124
5	0.9626	56.044	22.023	0.7608	143.865	60.506	0.9804	47.548	18.371
6	0.8532	111.060	39.824	0.7634	143.173	60.010	0.9088	102.745	31.985
	All data								
1	0.6797	202.440	55.565	0.6773	155.591	72.846	0.7512	112.032	60.347
2	0.9459	60.787	20.576	0.7091	136.008	60.822	0.9768	57.428	18.486
3	0.3540	195.307	56.704	0.3339	200.456	78.355	0.3519	201.435	61.072
4	0.4361	181.988	55.507	0.3433	198.428	77.196	0.3966	198.468	57.841
5	0.9385	60.5830	22.986	0.7154	132.900	59.999	0.9623	58.887	19.103
6	0.8335	99.5159	36.903	0.7239	131.043	59.089	0.8756	90.961	29.941

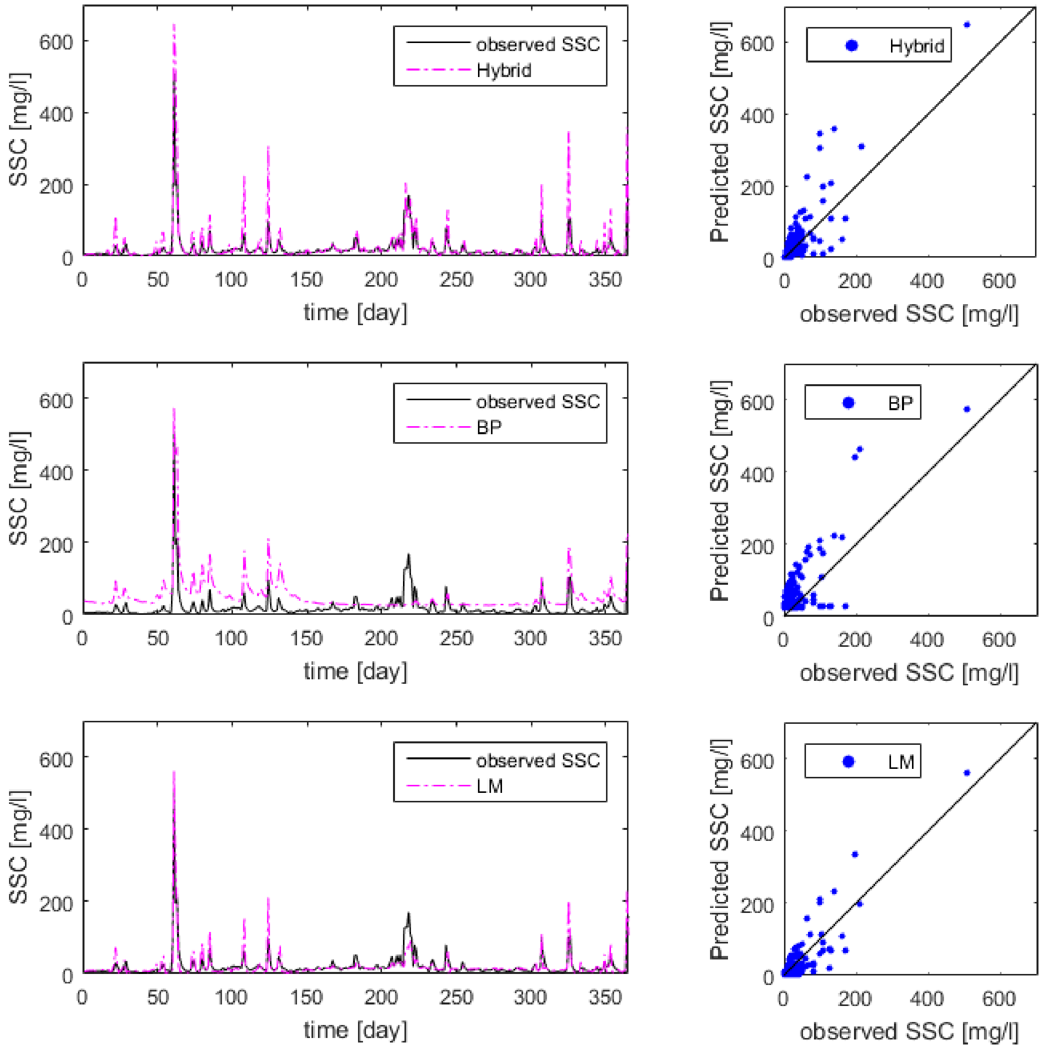


Figure 7.7: SSC predicted by ANFIS models with combination 2 for testing data set (Kaveh et al. 2017)

It can be said that ANFIS models trained with the Hybrid and LM algorithms are comparable in terms of the prediction accuracy. However, in general, the networks trained with the LM algorithm perform better than those trained by the Hybrid algorithm.

The temporal variations of the observed and predicted SSC using all three training algorithms for the combinations 2, 5, and 6 are shown in Figs.7.7, 7.8 and 7.9, respectively. Moreover, the predicted SSC are plotted against observed SSC. As can be seen, ANFIS models trained with the LM and Hybrid algorithms yield better results for SSC prediction than those trained by the BP algorithm. It is also obvious that the BP algorithm is almost always over-predicting measured values, except for a short interval. The critical result of the LM and Hybrid algorithms is that they consistently

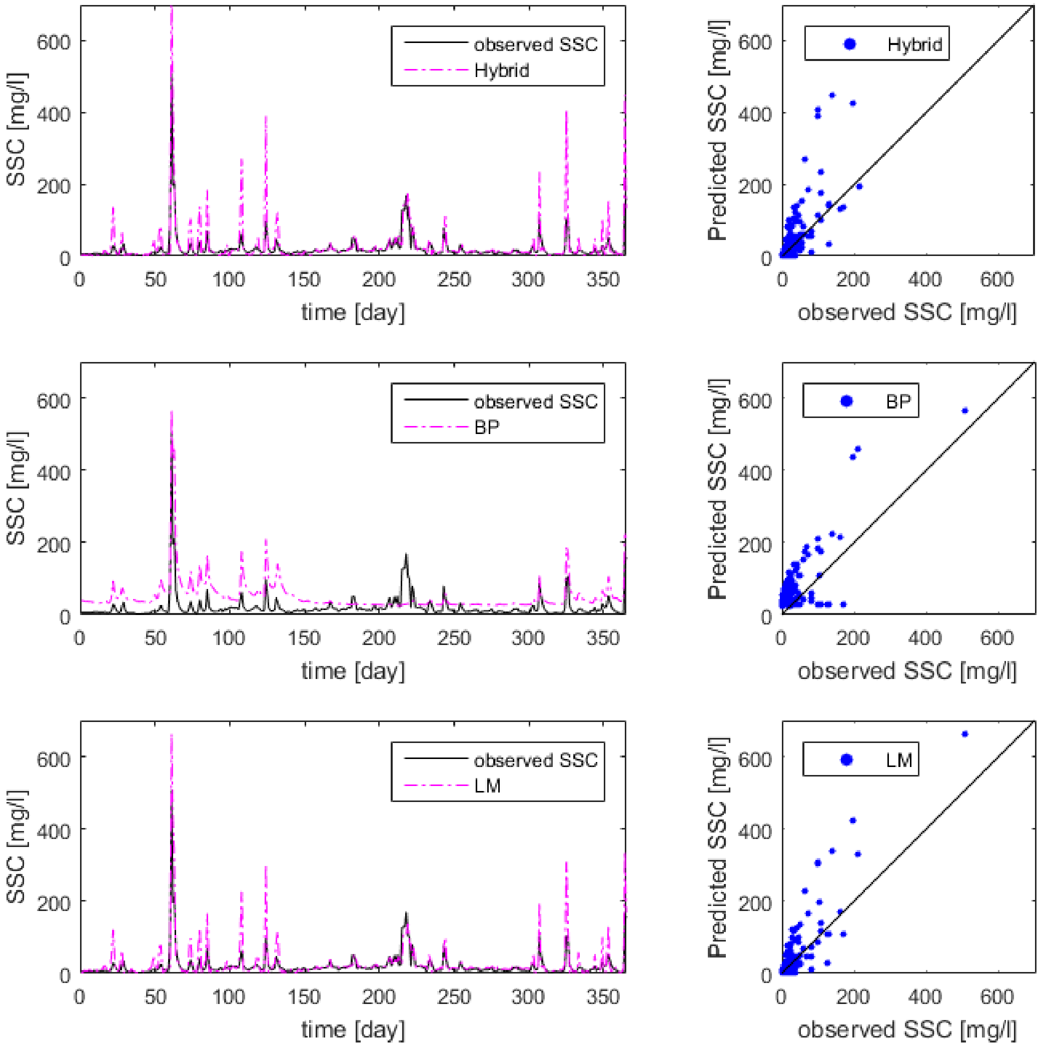


Figure 7.8: SSC predicted by ANFIS models with combination 5 for testing data set (Kaveh et al. 2017)

overestimated the SSC peaks. However, at peaks in the data, the LM algorithm shows better agreement with the observed time series than the other algorithms.
The predicted magnitudes of low, medium, and high SSC using the LM algorithm are closer to the observed values when compared to other algorithms. On the other hand, the results of ANFIS models trained by the LM algorithm are closer to 45^o straight line in the scatter plots compared with those of the other algorithms.

Fig.7.10 shows the performance indexes MAPE and generalization capability (NDEI) of the networks trained with different learning algorithms and input combinations. As can be observed, the LM algorithm performs better than the Hybrid algorithm for input combinations 1, 2, and 5 while the Hybrid algorithm provides a lower MAPE value for the other input combinations. In general, it can be said that the LM algorithm is a

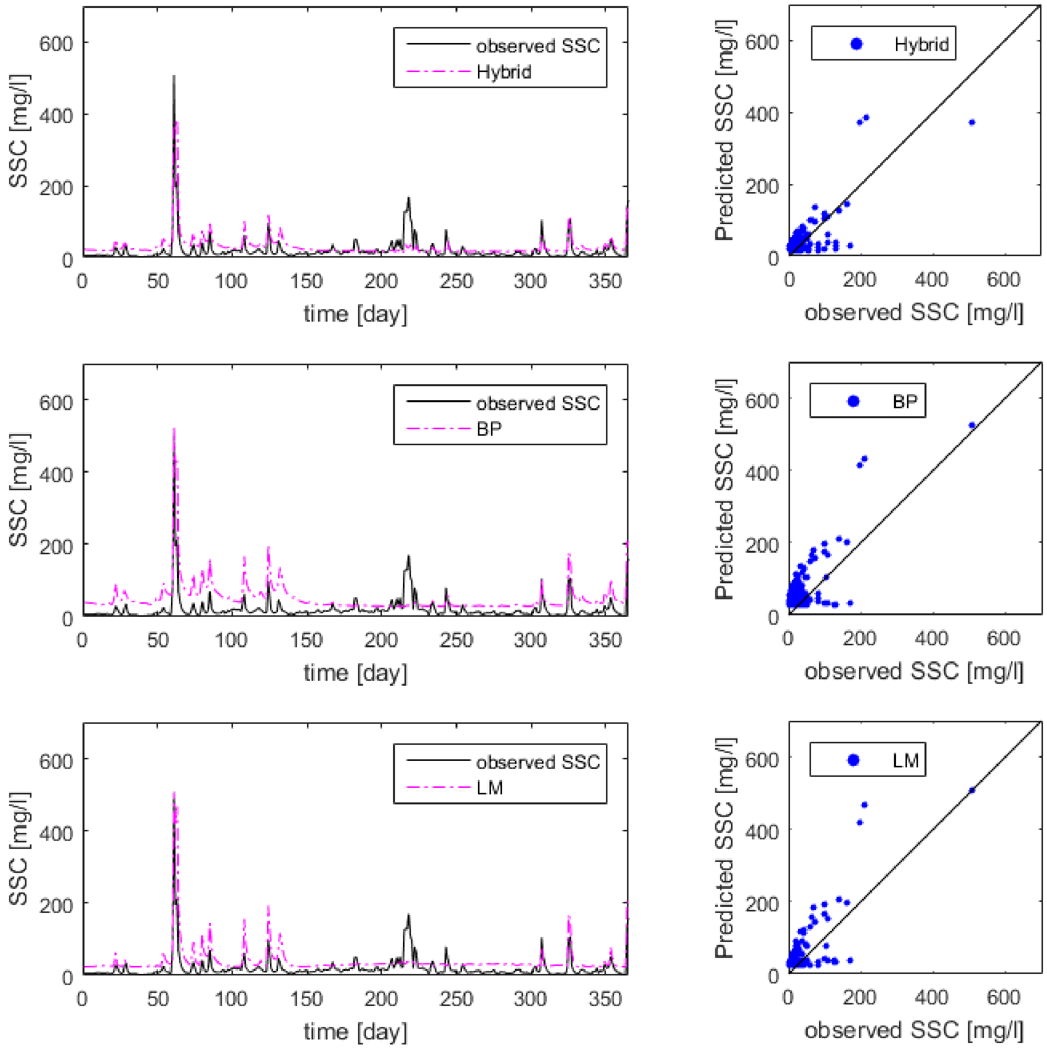


Figure 7.9: SSC predicted by ANFIS models with combination 6 for testing data set (Kaveh et al. 2017)

more stable learning rule in comparison to the Hybrid and BP algorithms. According to Fig.7.10(b), by applying the LM learning rule, the lowest NDEI value was obtained for input combination 2. Except for combination 4 where Hybrid algorithm shows a slightly better result, the LM learning rule performs better for the other combinations in term of generalization capability.

Further, the cumulative suspended sediment load (SSL) was predicted using the developed ANFIS models. The results are shown in Fig.7.11 for combinations 2, 5, and 6, as well as for the best result of each algorithm (the Hybrid algorithm with combination 5; the BP algorithm with combination 6; and the LM algorithm with combination 2). The best results for each algorithm were selected based for the comparison between the measured and predicted maximum SSL and the chosen statistical

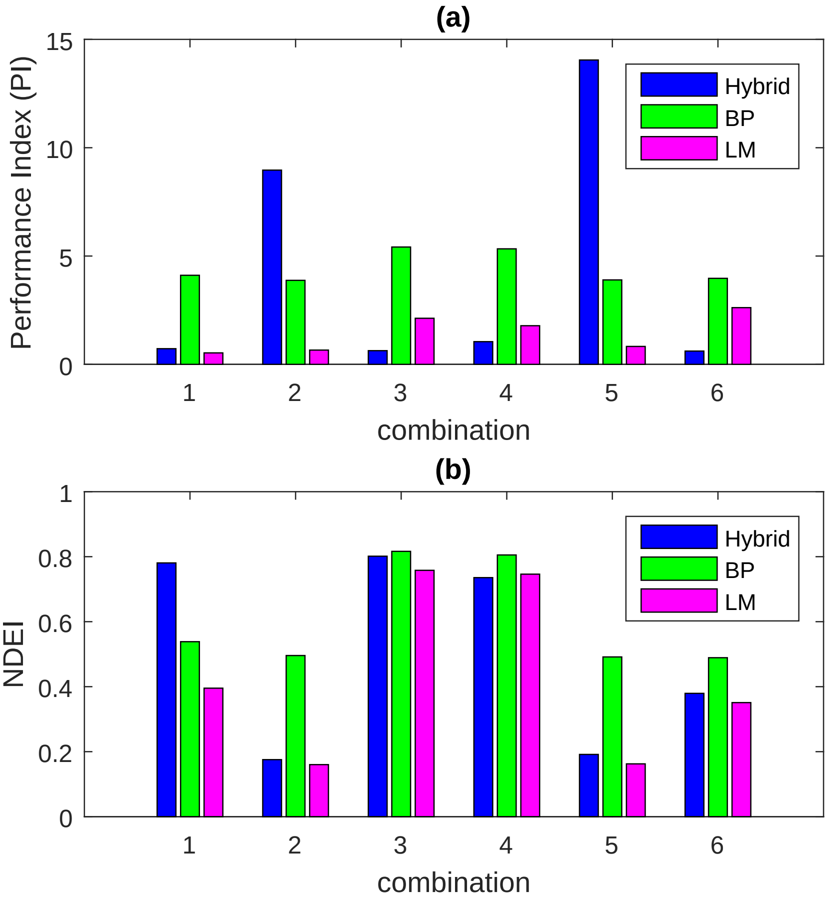


Figure 7.10: (a) Performance index and (b) generalization capability (NDEI) for different learning Algorithms and input combinations (Kaveh et al. 2017)

performance indexes.

As is obvious from Fig.7.11, models trained with the BP algorithm overestimated the cumulative SSL in all cases. As expected, the model trained by the LM algorithm has the best performance for combination 2. For combination 5, the results of the Hybrid and LM algorithms are comparable while they also overestimate the SSL values. In Fig.7.11(d), the measured cumulative SSL values are compared with the results of different ANFIS algorithms with the best combination for each model. The measured cumulative SSL in the verification period is 66,616.69 tons. The ANFIS models trained with the Hybrid, BP, and LM algorithms overestimate SSL values by 30,726.75, 82,366.88, and 7,494.97 tons, respectively. However, the LM algorithms gives better performance when compared to other algorithms. It can be concluded that, ANFIS model trained with the LM algorithm is more efficient in prediction of the cumulative SSL (Kaveh et al. 2017).

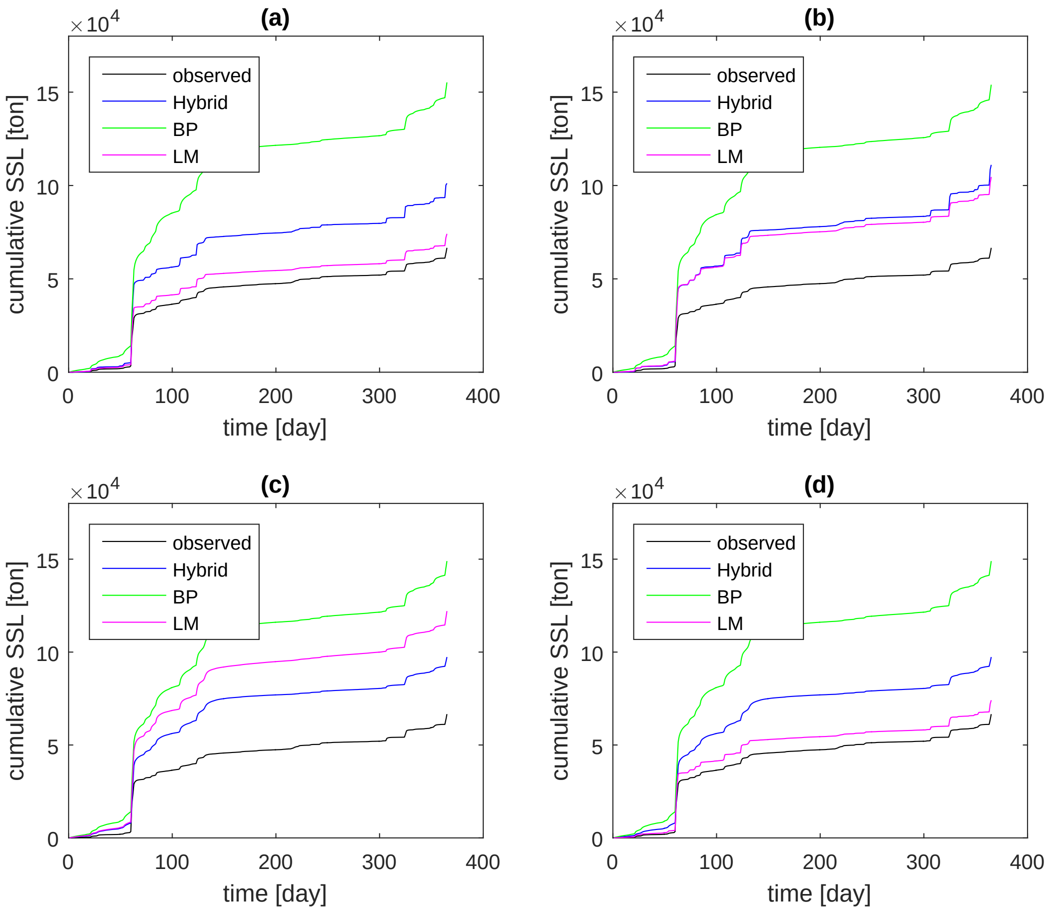


Figure 7.11: Observed and estimated cumulative SSL for: (a) combination 2, (b) combination 5, (c) combination 6, and (d) the best combination for each method (Kaveh et al. 2017)

Chapter 8

1D ANNs for modeling dynamic channel

In this chapter, the first proposed concept will be tested for 1D dynamic channel modeling. The main objective is, therefore, to develop an ANN model to predict the morphological changes in a straight alluvial channel under steady flow discharge and uniform bed material, where the bed level changes are calculated directly from the defined flow without calculation of the bed load. The prediction qualities of the designed network are studied by evaluating several statistical parameters, those of which describe the errors associated with the model in terms of the goodness-of-fit between the estimated bed change and analytical solution.

8.1 Model of a sand wave along an alluvial channel

8.1.1 Model setup

For a simple 1D model, a straight channel is considered with a length of 1000 m and a finite amplitude perturbation of the bed level near the center of the domain as illustrated in Fig.8.1. This case can represent a sand dune in a river flow. It is assumed that the bed elevation Z_b is very small in comparison to the water free surface level Z_s

This chapter was published as:
Bui, M. D.; Kaveh, K.; Rutschmann, P. (2015b): Integrating artificial neural networks into hydromorphological model for fluvial channels. In Proceedings of the 36th IAHR World Congress (pp. 8-pages).
Kaveh, K., Bui, M. D., & Rutschmann, P. (2016): A new approach for morphodynamic modeling using integrating ensembles of artificial neural networks. In “Wasserbau-mehr als Bauen im Wasser”. Beiträge zum 18. Gemeinschafts-Symposium der Wasserbau-Institute TU München, TU Graz und ETH Zürich (pp. 304-315).

and the bed form movement is only due to bed load. Assuming a steady flow discharge throughout the channel with a rigid lid $H_0 = Z_s =$ constant, one can obtain:

$$H \approx H_0 - Z_b \quad \text{and} \quad U \approx \frac{q}{H} \tag{8.1}$$

where H is the water depth, U is the mean flow velocity, and q is the constant fluid volume flux per unit width (Bui et al. 2015b).

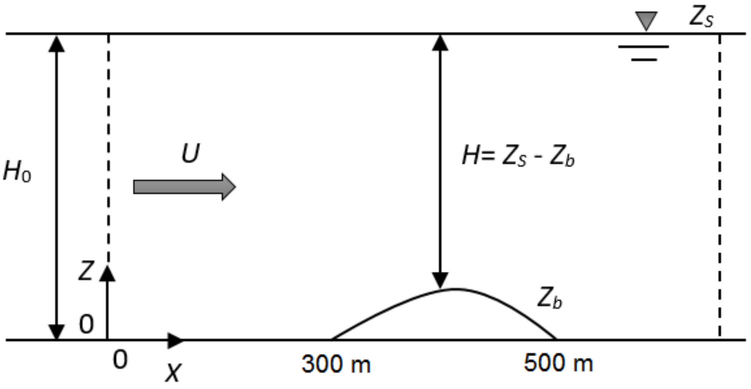


Figure 8.1: Bathymetry and coordinate system for the test case (Bui et al. 2015b)

8.1.1.1 Analytical approximation

Assuming that transport rate q_b is a power function of current speed U (Grass, 1981; van Rijn, 1993), then:

$$q_b = AU^m \tag{8.2}$$

where A is a given function and m is a given positive constant both of which are specific to the particular sediment transport formula. Note that A is typically a function of the mean flow velocity U, the total height of the water column H, and a number of constants that are based on sediment properties (e.g. sediment type and grain size) and data fitting procedures. The constant m is typically in the range of 1 $\leq m \leq 3$. The phase speed of bed form $C(Z_b)$ can be now expressed as:

$$C(Z_b) = \frac{mAq^m}{(1-p_0)(H_0 - Z_b)^{m+1}} \tag{8.3}$$

The initial condition $Z_b(x_0, 0)$ is specified as follows with $\alpha = 1(m)$:

$$Z_b(x_0,\, 0) = \begin{cases} \alpha \sin^2\left[\frac{\pi(x_0-300)}{200}\right] & \text{if}\, 300 \leq x_0 \leq 500\,(\text{m}) \\ 0 & \text{otherwise} \end{cases} \tag{8.4}$$

Further, the Exner equation may be solved approximately by the method of characteristics, which gives the result that Z_b will remain constant along characteristics given by:

$$\frac{dx}{dt} = C\left(Z_b(x_0,\, 0)\right) = \frac{mAq^m}{(1-p_0)\ \left(H_0 - Z_b\left(x_0,\, 0\right)\right)^{m+1}} \tag{8.5}$$

Further, since Z_b is constant along each characteristic, dx/dt is then constant and each characteristic is a straight line in $\{x, t\}$ with slope given by $C(Z_b(x_0, 0))$ at its intersection with the x axis. Substituting the initial bathymetry into Eq.(8.5) and integrating gives:

$$x = x_0 + \frac{mAq^m t}{(1-p_0)} \begin{cases} (H_0 - \alpha\sin^2(\frac{\pi(x_0-300)}{200}))^{-(m+1)}, \text{if } 300 \le x_0 \le 500 \text{ (m)} \\ \\ H_0^{-(m+1)}, \text{otherwise} \end{cases} \tag{8.6}$$

Unfortunately, x_0, which is a function of x and t, cannot be written in the closed form of x and t. Hence, the approximate solution of Z_b is:

$$Z_b(x,\, t) = \begin{cases} \alpha\sin^2\left[\frac{\pi(x_0-300)}{200}\right] & if\, 300 \le x_0 \le 500\, (\text{m}) \\ 0 & \text{otherwise} \end{cases} \tag{8.7}$$

where the value of x is determined by substituting values of x_0 and t into Eq.(8.6) (Bui et al. 2015b).

8.1.1.2 Data selection for ANN

The Eqs.(8.1), (8.6) and (8.7) with $0 \le t \le 60000$ sec and $250 \le x \le 550$ m are used to generate the data for ANN model. The following quantities are specified according to similar settings in Hudson et al. (2005): $m = 3$, $p_0 = 0.4$, $H_0 = 10$ m, $A = 0.001$ s^2/m, and $\alpha = 1$ m. Grid spacing is chosen to be $\Delta x = 2$ m. For all of the numerical schemes discussed in Hudson et al. (2005) and Long et al. (2008) a very small time step was chosen to satisfy stability conditions (e.g. $\Delta t = 0.1$ s), however for the ANN model, a large time step $\Delta t = 300$ s is used. To generate the data set, U must be evaluated using Eq. 8.1 at all alternate grid points i at time level n (denoted by U_i^n). Then, Z_b needs to be similarly evaluated using Eqs. 8.6 and 8.7 at the same grid points and time level (denoted by Z_{bi}^n). Once complete, the process is then repeated at time level $(n+1)$ and so on.

Finally, I have a data set of 30350 patterns, which are then divided randomly into three subsets, whereby the biggest amount of data (70%) is added randomly to the training subset. The remaining data set samples are used for validating (15%) and testing the networks (15%). The training subset is used to design the weights. The validation subset is used additionally to monitor the accuracy of training, while training is ongoing. After each epoch, the validation subset acts as a barometer for determining when the accuracy of the multilayer perceptron is at an acceptable level. After the network is considered optimally trained, the test subset is used to verify its performance (Bui et al. 2015b).

8.1.2 Design of ANN model

8.1.2.1 Model inputs

Based on different numerical schemes used in Hudson et al. (2005) and Long et al. (2008) for calculation of morphological change in alluvial channels, the following combinations including different parameters at the time step n are considered as inputs of ANN models to predict the bed level change at the grid point i and at one time step ahead :

- Combination C1: Z^n_{bi}, $Z^n_{b(i-1)}$, U^n_i, $U^n_{(i-1)}$
- Combination C2: Z^n_{bi}, $Z^n_{b(i+1)}$, $Z^n_{b(i-1)}$, U^n_i
- Combination C3: Z^n_{bi}, $U^n_{(i-1)}$, U^n_i, $U^n_{(i+1)}$
- Combination C4: Z^n_{bi}, $Z^n_{b(i+1)}$, U^n_i, $U^n_{(i+1)}$
- Combination C5: Z^n_{bi}, $Z^n_{b(i-1)}$, U^n_i, $U^n_{(i+1)}$
- Combination C6: Z^n_{bi}, $Z^n_{b(i-1)}$, $Z^n_{b(i+1)}$, U^n_i, $U^n_{(i+1)}$
- Combination C7: Z^n_{bi}, $Z^n_{b(i-1)}$, $Z^n_{b(i+1)}$, $U^n_{(i-1)}$, $U^n_{(i+1)}$
- Combination C8: Z^n_{bi}, $Z^n_{b(i-1)}$, $Z^n_{b(i+1)}$, U^n_i, $U^n_{(i-1)}$, $U^n_{(i+1)}$

8.1.2.2 Model development

There is no special rule for ANN model development. By using Kolmogorov's theorem, Marques (2001) and Hornik (1989) provided there are enough neurons in the hidden layer, therefore, only one hidden layer should be sufficient to ensure that the structure has the properties of a "universal approximator" for a number of particular problems. Further, Marques (2001), the number of neurons in the hidden layer would most likely be $(2N+1)$, where N is the number of neurons in the input layer. If there are too few hidden units there will be both a high training error and generalization error due to underfitting and high statistical bias. Correspondingly, too many hidden units result in a low training error but still in a high generalization error due to overfitting and high variance. Here a feedforward neural network is considered in which each node connects to every node in subsequent layers. During the learning phase, the network learns by adjusting the weights to be able to predict the correct class output of the input signals.

In this study, estimation of bed level change is based on an ANN model with one hidden layer. It is necessary to mention that the performance of ANN model is significantly related to the number of hidden layer nodes. Trial and error approach has been employed to choose the appropriate number of nodes in the hidden layer. For the first tests, the logistic sigmoid and linear transfer functions have been used for hidden and output layers, respectively. Further, the inputs combination C1 and the Levenberg-Marquardt learning rule have been applied. After each training process, predicted values have been compared with those calculated by analytical approximation. Statistical indicators such as, correlation coefficient (R), root mean square error (RMSE) and mean absolute error (MAE) have been used to evaluate performance of the ANN models. Based on performance indices of the ANN models given in Table.8.1 for the testing data set, it is seen that by increasing the hidden neuron numbers from 10 to 14, the performance of the ANN model becomes worse. Further, there is no significant accuracy differences between models trained with 10 and 15 hidden neurons. For simplicity and reducing the run time, this ANN configuration with ten hidden neurons is used for the next parts. Further tests have been carried out for different transfer functions used in the hidden layer and output one. As seen in Table.8.2, applying the logistic sigmoid transfer function for hidden layer and the linear function for the output one results in the best performance of the ANN model.

Table 8.1: Calculated performance indices for different hidden nodes

Number of hidden nodes	R	RMSE	MAE
1	0.998442	0.021353	0.017702
2	0.999917	0.004870	0.003703
3	0.999945	0.003976	0.003166
4	0.999993	0.001366	0.000887
5	0.999995	0.001195	0.000725
6	0.999992	0.001555	0.000862
7	0.999997	0.000935	0.000474
8	0.999998	0.000713	0.000350
9	0.999995	0.001193	0.000848
10	0.999998	0.000715	0.000302
11	0.999994	0.001330	0.000730
12	0.999993	0.001395	0.000720
13	0.999984	0.002164	0.001049
14	0.999995	0.001157	0.000527
15	0.999999	0.000628	0.000307

Table 8.2: Performance criteria for different transfer functions

Hidden layer	Output layer	R	RMSE	MAE
logsig	logsig	0.999986	0.002064	0.001557
tansig	tansig	0.999998	0.000715	0.000302
logsig	tansig	0.999994	0.001309	0.000630
tansig	logsig	0.999919	0.004907	0.004298
radbas	logsig	0.999983	0.002309	0.001906
logsig	radbas	0.999958	0.003600	0.002800
radbas	tansig	0.999987	0.001950	0.001212
tansig	radbas	0.999976	0.002696	0.002108
radbas	radbas	0.999996	0.001135	0.000877
logsig	purelin	0.999999	1.38E-05	9.13E-06
tansig	purelin	0.999999	0.000186	0.000127
radbas	purelin	0.999999	1.82E-05	1.49E-05

It is well known that the Levenberg-Marquardt method is one of the fastest learning algorithms chosen. However, its convergence speed is not necessarily linked to the optimal performance behavior. Hence, the ANN configuration mentioned above is used to test against different training methods. Table.8.3 presents the statistical performances of ten different learning rules, where the quasi-Newton Back-Propagation is denoted by BFG; Resilient Back-Propagation by RP; Gradient descent with adaptive learning rate Back-Propagation by GDA; Gradient descent with momentum Back-Propagation by GDM; Gradient descent with momentum and adaptive learning rate Back-Propagation by GDX; Gradient descent back-propagation by GD; One-step secant back-propagation by OSS; Conjugate gradient Back-Propagation with Fletcher-Reeves updates by CGF; Levenberg-Marquardt back-propagation by LM; and Conjugate gradient Back-Propagation with Powell-Beale restarts CGB. Based on the calculated performance indices, it is obvious that the LM training method is best suitable for this study case.

Table 8.3: Calculated performance indices for different learning algorithms

	BFG	RP	GDA	GDM	GDX
R	0.9999	0.9999	0.9905	0.9900	0.9964
RMSE	0.00059	0.00246	0.0513	0.05276	0.03202
MAE	0.00046	0.00181	0.04317	0.03638	0.02242
	GD	OSS	CGF	LM	CGB
R	0.9903	0.9999	0.9985	0.9999	0.9999
RMSE	0.05169	0.00359	0.01997	1.30E-05	0.00311
MAE	0.0394	0.00285	0.01459	9.10E-06	0.00238

Table 8.4: Calculated performance indices for different learning algorithms

Combination	R	RMSE	MAE
C1	0.999999999	1.38E-05	9.13E-06
C2	0.999999976	9.11E-05	6.98E-05
C3	0.999999556	0.000353	0.000282
C4	0.999999976	8.40E-05	6.22E-05
C5	0.999999710	0.000286	0.000208
C6	0.999999995	3.72E-05	2.62E-05
C7	0.999999945	0.000124	9.37E-05
C8	0.999999864	0.000195	0.000160

According to Table.8.4, the designed ANN model with the inputs combination C1 provides the best performance. Understood from Eq.(8.3), the phase speed of bed form is always positive. Hence, in this study case the morphological change at the point i depend mostly on the bed level and water velocity at this point and at the upstream neighbor point $(i-1)$ (Bui et al. 2015b).

8.1.3 Results and discussion

Applying this designed ANN model, the values of the weights and biases have been specified after a successful learning and validation process. They represent the stored knowledge of the ANN for bed level change modeling, which are separated in one input weight matrix $\mathbf{IW_{1,1}}$, one hidden-layer weight matrix $\mathbf{LW_{2,1}}$, one bias vector $\vec{\mathbf{b_1}}$ and one bias value b_2 with the following values:

$$\mathbf{IW_{1,1}} = \begin{bmatrix} 5.8720 & -6.2430 & 26.1509 & 34.6502 \\ 4.4199 & 5.1539 & -45.8799 & -46.2625 \\ -4.6946 & 6.2918 & 43.7671 & -27.4119 \\ 5.5531 & 0.0688 & 51.5353 & -54.2679 \\ 2.5215 & 5.7512 & 29.1313 & -65.7119 \\ -5.5153 & -4.7864 & 33.8444 & 42.4838 \\ -6.4566 & -2.6638 & 56.3371 & 45.2583 \\ 1.0147 & 8.6364 & -20.8286 & -35.2006 \\ 6.4408 & 4.4430 & 19.5797 & 56.1871 \\ 5.5915 & 5.4314 & -34.2711 & -48.4119 \end{bmatrix} \tag{8.8a}$$

$$\begin{aligned} \mathbf{LW_{2,1}} = \Big[\ & 0.176690 \quad -0.419430 \quad -0.068665 \quad 0.071825 \quad 0.197520 \\ & -1.445500 \quad 0.601230 \quad -0.180750 \quad -0.000003 \quad 1.134200 \ \Big] \end{aligned} \tag{8.8b}$$

$$\vec{\mathbf{b_1}} = \begin{bmatrix} -69.1472 \\ 88.7288 \\ -14.0323 \\ -2.0066 \\ 34.0812 \\ -76.1308 \\ -103.4181 \\ 56.2286 \\ -80.3684 \\ 86.0117 \end{bmatrix} \tag{8.8c}$$

$$b_2 = [-0.22516] \tag{8.8d}$$

Using the designed network, the following equation is received for the bed level change:

$$Z_{bi}^{(n+1)} = \mathbf{LW_{2,1}} \times \text{logsig}\left(\mathbf{IW_{1,1}} \times \begin{bmatrix} Z_{bi}^{n} \\ Z_{b(i-1)}^{n} \\ U_{i}^{n} \\ U_{(i-1)}^{n} \end{bmatrix} + \vec{\mathbf{b_1}}\right) + b_2 \tag{8.9}$$

Since the characteristic time scale of bed-evolution and bed load transport processes is normally much greater than that of fluid flow, it can be assumed that changes in the bed elevation during one computational time step do not significantly influence the flow field. This assumption leads to the computationally attractive possibility of coupling flow and sediment computations in an iterative manner. Hereby, the flow and sediment-transport modules communicate through a quasi-steady morphodynamic time-stepping mechanism: during the flow computation, the bed level is assumed constant and during the computation of the bed level the flow and sediment transport are assumed invariant to the bed level changes. Based on this coupling concept, the main calculation procedure implemented in this study is shown in Fig.8.2 (Bui et al. 2015b).

Now the capability of the designed ANN model for prediction of bed level changes in the new coupling model can be analyzed. First, the initial values (at time $t = 0$) are defined at every grid point i. The bed levels at one time step ahead, $t = 5$ min, are calculated using the ANN-based Eqs.(8.8) and (8.9). The water velocities at this time step are updated using Eq.(8.1). This procedure is repeated until the last time step is reached. Table.8.5 presents the statistical performances indices of the model based on a comparison between the predicted bed levels and analytical approximation at different times.

Table 8.5: Calculated performance indices for different times

Performance	500 (min)	1000 (min)	1500 (min)	2000 (min)
R	0.999974	0.999976	0.999866	0.999066
RMSE	0.003169	0.004032	0.007505	0.016973
MAE	0.002571	0.003245	0.006234	0.012875

Given the morphological change in the channel, the results indicate that the de-

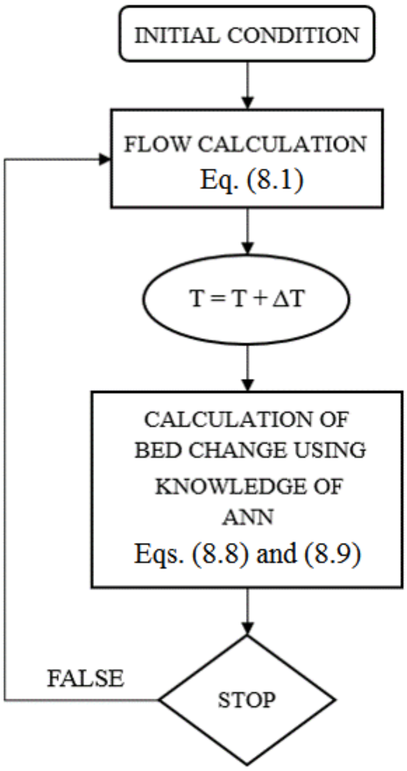


Figure 8.2: Flowchart of the coupling system (Bui et al. 2015b)

signed ANN performs well. This can be deduced from the high values of R along with small values of RMSE and MAE. Further, it should be noted that many time series problems involve the task of predicting a sequence of future values using the values observed in the past. A typical approach to solve this problem is to construct a single model from historical values of the time series and then applies the model step-by-step to predict its future states. This approach is known as multi stage prediction. As it uses the predicted values from the past, multi stage prediction could be susceptible to the error accumulation problem, i.e. errors committed in the past are propagated into future predictions. The same error problem can be found also in the conventional numerical models for morphodynamic.
Fig.8.3 is a plot of the bed level calculated by analytical approximation and prediction of the designed ANN model at after 1000 and 2000 minutes. The fit is extremely good, and it is rather difficult to discern one graph from the other at any time $t \leq$ 1500 minutes.

The predicted results show that for the simple case of 1D morphological problems, artificial neural networks can provide good performance. It is also demonstrated that the ANN could perform well for long-term time series prediction and could be integrated in the hydromorphological model systems. The following sections encompasses

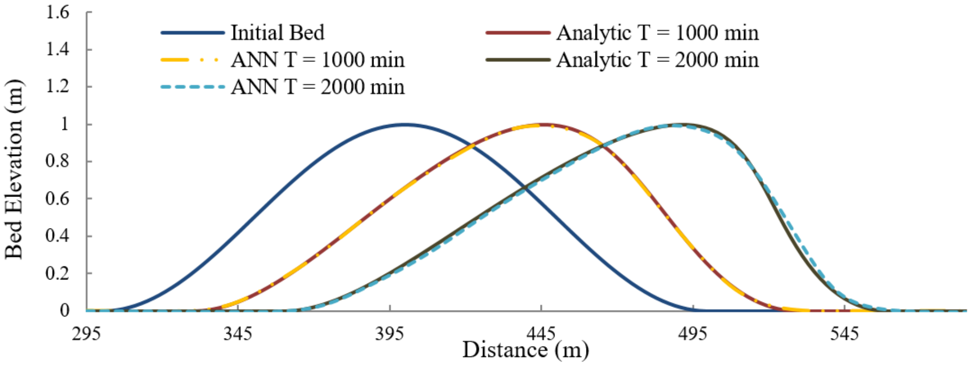


Figure 8.3: Bed elevation after 1000 and 2000 minutes

further model developments to obtain a universal ANN-based approximator for morphological change in alluvial channels (Bui et al. 2015b).

8.2 Model of dynamic bed forms

8.2.1 Model setup

To develop a more comprehensive universal ANN-based approximator for 1D morphological change in alluvial channels, a broader range of input data is required for network training. Consequently, I consider different time steps, various discharge values, and three different hump shapes including sinusoidal, Gaussian, and fractional, with a wide range of heights and widths for data generation. These different initial conditions are determined as follows with heights (α) and widths (β):

$$Z_{b1}(x_0,\, 0) = \begin{cases} \alpha \sin^2\left[\frac{\pi(x_0-300)}{\beta}\right] & if\, 300 \leq x_0 \leq 500\,(\mathrm{m}) \\ 0 & \text{otherwise} \end{cases} \tag{8.10a}$$

$$Z_{b2}(x_0,\, 0) = \alpha \times e^{-\beta(x_0-400)^2} \tag{8.10b}$$

$$Z_{b3}(x_0,\, 0) = \frac{\alpha\beta}{(x_0-400)^2+\beta} \tag{8.10c}$$

The range of changeable parameters for each hump shape, such as discharge, time step, height and width, is presented in Table.8.6.

Similar to the previous section, it is assumed that the bed elevations Z_{bi} are very small in comparison to the water free surface level Z_s, and the bed form movement is only due to bed load. It is also assumed that the flow discharges throughout the channel are steady. As a result of these assumptions the Eq.(8.1) will be valid here as well.

Table 8.6: Range of varying parameters for data generation

Hump shape	Range	Discharge (m^3/s)	Time step (min)	Height (α)	Width (β)
Sinusoidal	Lower limit	6.0	30	0.6	180
	Upper limit	12.0	50	1.2	240
Gaussian	Lower limit	6.0	30	0.6	0.0005
	Upper limit	12.0	50	1.2	0.0010
Fractional	Lower limit	6.0	30	0.3	140
	Upper limit	12.0	50	0.7	200

8.2.1.1 Analytical approximations

As the same transport rate q_b formula (Grass equation) is used here, the Eq.(8.5) which is solution of the Exner equation by the method of characteristics can be applied in this section as well. Substituting each initial bathymetry into Eq.(8.5) and integrating provides the analytical approximation of bed forms.

8.2.1.2 Data selection for ANN

The obtained equations with $0 \leq t \leq 60000$ s and $250 \leq x \leq 550$ m are used to generate the data for ANN model. The following constant parameters are specified for calculations: $m = 3$, $p_0 = 0.4$, H_0 =10 m, and $A = 0.001$ s^2/m. Then each equation is solved for different discharge, time step, height and width values. These values are quantified as: Q = 6-12 m^3/s with increment of 2 m^3/s, Δt = 30-50 min with increment of 5 min, $\alpha_1 = \alpha_2$ = 0.6-1.2 with increment of 0.2, α_3 = 0.3-0.7 with increment of 0.1, β_1 = 180-240 with increment of 20, β_2 = 0.0005-00.1 with increment of 0.00025, and β_3 = 140-240 with increment of 20. The indices 1, 2, and 3 are representing Sinusoidal, Gaussian and Fractional humps, respectively. Grid spacing is chosen to be $\Delta x = 2$ m. Similarly, U is evaluated using Eq.(8.1) at all alternate grid points i at time level n (denoted by U_i^n) for each discharge and bathymetry. Then, Z_b is evaluated using the obtained analytical approximations for each bathymetry at the same grid points and time level (denoted by Z_{bi}^n). Once complete, the process is then repeated at time level $(n+1)$ and so on.

Finally, a huge unique data set including the required information is obtained based on different discharges and bathymetries. The data set is then divided randomly into three subsets, whereby the biggest amount of data (70%) is added randomly to the training subset. The remaining data set samples are used for validating (15%) and testing the networks (15%).

8.2.2 Design of ANN model

8.2.2.1 Model inputs

The best input combination resulted from section 8.1 can be used here as well. However, it is noteworthy to mention that Δt has to be added to the input parameters, as it is not fixed for data generation. The following parameters are considered as inputs of ANN model:

- Z^n_{bi}, $Z^n_{b(i-1)}$, U^n_i, $U^n_{(i-1)}$, Δt

8.2.2.2 Model development

In this part, no further work has been conducted to determine the network best structure. The estimation of bed level change is based on the optimum ANN model found in the previous section. Therefore, the ANN is trained using Levenberg-Marquardt technique due to the fact this technique is more powerful and efficient than the conventional gradient descent technique. One hidden layer consisting of 10 neurons is used in this application. The logistic sigmoid and linear transfer functions are considered for the hidden layer and the output one, respectively.

8.2.3 Results and discussion

The values of the weights and biases, which represent the stored knowledge of the ANN, have been established after a successful training and validating process. They are separated and defined in one input weight matrix $\mathbf{IW_{1,1}}$, one hidden-layer weight matrix $\mathbf{LW_{2,1}}$, one bias vector $\vec{\mathbf{b}_1}$ and one bias value b. The following equation can be finally obtained:

$$Z_{bi}^{(n+1)} = \mathbf{LW_{2,1}} \times \text{logsig}\left(\mathbf{IW_{1,1}} \times \begin{bmatrix} Z^n_{bi} \\ Z^n_{b(i-1)} \\ U^n_i \\ U^n_{(i-1)} \\ \Delta t \end{bmatrix} + \vec{\mathbf{b}_1} \right) + b_2 \tag{8.11}$$

Now, the Eqs.(8.1), (8.11) and the coupling system explained in the previous section can be applied for bed level changes prediction. To assess the effectiveness of the ANN-based approximator, three different test cases are considered using new test parameters, which ANN is not trained with. The prediction is performed for all test cases including sinusoidal, Gaussian and fractional shape, whose results are presented in Table.8.7. The results indicate that the designed ANN performs well, given the morphological change in the channel, with high values of R and small values of RMSE and MAE, even for the new data sets provided to it. The predicted bed levels are compared with the analytical ones for testing data sets in Fig.8.4 at after approximately 1000 minutes.

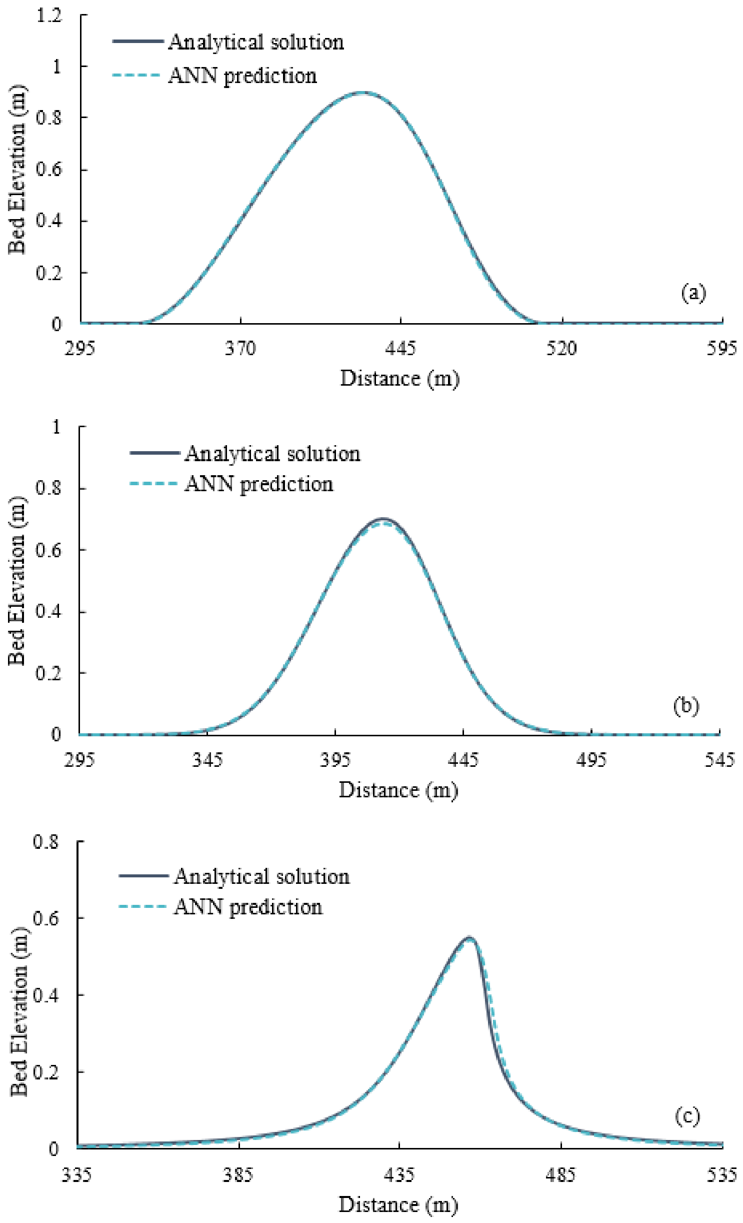


Figure 8.4: Comparison between the calculated and predicted bed elevation on the testing data; (a) sinusoidal, (b) Gaussian, and (c) fractional shape

Table 8.7: Statistical performance of ANN model for different test cases

	New test parameters				Performance index		
Shape	Q (m^3/s)	Δt (min)	α	β	R	RMSE	MAE
Sinusoidal	9.0	42.0	0.90	190	0.99993	0.00419	0.00338
Gaussian	7.0	47.0	0.70	0.0009	0.99994	0.00278	0.00177
Fractional	11.0	33.0	0.55	190	0.99788	0.00814	0.00371

Chapter 9

2D/3D ANN models for hydro-morphodynamic processes

This section presents a study on the development of ANN models for predicting bed level changes in which the water surface elevation is inconstant. The motivation behind the research is to study the efficiency and effectiveness of ANN for more complicated case studies. In contrast to the 1D model where an analytical approximation is used for data generation, the open-source finite-element TELEMAC-MASCARET system will be used for 2D/3D models to simulate the morphodynamic evolution in a channel. Obtained results are then used as input data set for ANN training process.

9.1 Flow and bed change in a straight channel

As the first case, a straight channel is considered in TELEMAC-3D modeling system with a length of 20 m and a finite amplitude perturbation of the bed level with a height of 0.3 m close to the center of the domain. The initial bathymetry is illustrated in Fig.9.1.

To build the geometry, the mesh generator from the free licensed program BLUEKENUE is applied. It is developed by the Canadian Hydraulics Centre of the National Research Council. It can be used as a Pre- and Post-processor of TELEMAC. The mesh includes nodes and elements in 2D coordinates of x and y. However the bathymetry includes additional information of the elevations in z-direction. This can be done by interpolation of the mesh, which requires a point set consisting of x-, y- and z-coordinates.

Parts of this chapter were published as:
Kaveh, K., Bui, M.D., & Rutschmann, P. (2018): Development of an Artificial-Neural-Network-based concept for hydro-morphodynamic modelling in rivers, The 5th IAHR Europe Congress, 2018,Trento, Italy.

For the hump the elevation of points in z-direction are extracted from the Gaussian function. Finally, the mesh as in Fig.9.1 shown is developed. The final geometry is then saved in a selafin file (.slf).

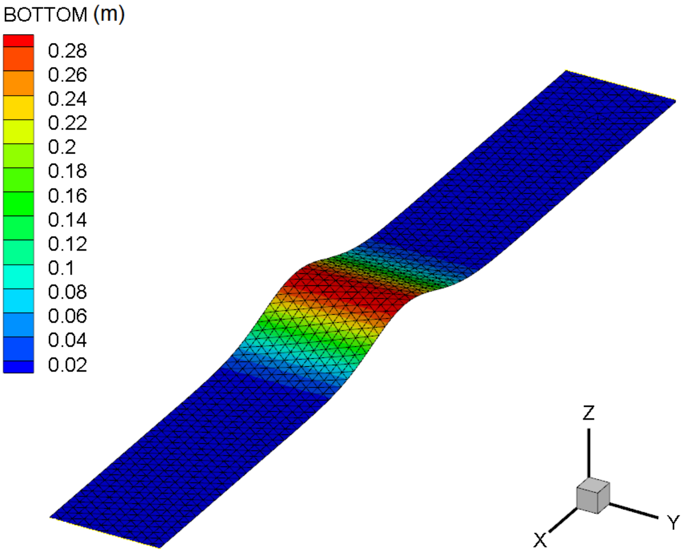


Figure 9.1: Initial bathymetry

The hydrodynamic model is based on the 3D Reynolds-Averaged Navier-Stokes (RANS) equations in unstructured mesh grids. The prescribed flow rate and elevation are considered as inlet and outlet boundary condition, respectively. To start the simulation the definition of initial conditions is also required. By default, two options of constant water depth or constant elevation over all the domain are available. Here, a constant elevation of 3 m is applied as an initial condition. The turbulence components are calculated by the accurate standard $k - \varepsilon$ model. The diffusion of velocity parameter for the 3D simulation must be given in horizontal and vertical directions. The value of 1e-6 is considered for this part.
Another important parameter to be chosen for the TELEMAC simulation is the bed roughness or bed friction. The application of friction requires the definition of the friction law and its corresponding value in the steering file. The applied law in this model is the law of Nikuradse which uses a logarithmic velocity profile in z-direction. The friction coefficient for the bottom is 0.00015, whereas a high friction coefficient is considered for lateral solid boundaries (0.045) to get a fully 3-dimensional model. As a result, during the simulation, the middle of the hump changes faster than channel sides, which makes the modelling more difficult for ANN because it must predict asymmetric movement of bed.
The sediment transport model is calculated based on semi-empirical formulae, which

includes bedload computation and bed evolution. Non-cohesive sediment with uniform grain size of 0.15 mm is used for sediment transport model. Suspended load is not considered in this model. Of the most important simulation parameters is the definition of the numerical solvers, schemes and related parameters. The most important parameter is the time step. A minor time step $\Delta t = 1$ s with repetition of 10000 times is chosen. Further, an implication parameter is defined which corresponds to the discretization scheme. To get a more stable result, a full implicit scheme is applied.

9.1.1 Data selection for ANN

After a successful simulation of the test case using TELEMAC-3D modeling system, the obtained results can be used for networks training. To do this, the program was run for 10000 sec. For this model a very small time step was chosen to satisfy stability conditions (e.g. $\Delta t = 1$ s), however for the ANN model, a large time step $\Delta t = 500$ s is used. The obtained results from TELEMAC models consist of water depth, scalar velocity, and bed elevation at all spatial nodes and time steps. It is noteworthy that all modules of the TELEMAC system are based on unstructured grids, but for ANN training I tend to use all information at structured grids. Thus, the interpolation technique was used to gain all required information at our desired structured grid points. Considering $0 \leq t \leq 10000$ s and $0 \leq x \leq 20$ m and specified values for Δt, and Δx, a data set is received, that is divided randomly into three subsets: training, validation and testing.

Finally, I have a data set for each network, which are then divided randomly into three subsets, whereby the biggest amount of data (70%) is added randomly to the training subset. The remaining data set samples are used for validating (15%) and testing the networks (15%).

9.1.2 ANN design

Based on different numerical schemes used in Hudson et al. (2005), Long et al. (2008) and Bui et al. (2015) for calculation of morphological change in alluvial channels, the following combinations, including different parameters at the time steps t and $t-1$, are considered as inputs of ANN models to predict the water depth, scalar velocity and bed level change at the grid point (i, j) and at one time step ahead $t+1$. After a trial and error procedure, the best input combinations for the models is found as follows:

- For water depth and scalar velocity integrators: bed elevation, water depth, and velocity in x-direction at grid points $(i,j), (i-1,j), (i,j-1), (i+1,j)$, and $(i,j+1)$ and at times $(t-1)$ and (t):

$$\vec{\mathbf{A}} = \big[Z^t_{b(i,j)}, Z^t_{b(i-1,j)}, Z^t_{b(i,j-1)}, U^t_{(i,j)}, U^t_{(i-1,j)}, U^t_{(i,j-1)}, H^t_{(i,j)}, H^t_{(i-1,j)}, H^t_{(i,j-1)},\\ Z^{t-1}_{b(i,j)}, Z^{t-1}_{b(i-1,j)}, Z^{t-1}_{b(i,j-1)}, U^{t-1}_{(i,j)}, U^{t-1}_{(i-1,j)}, U^{t-1}_{(i,j-1)}, H^{t-1}_{(i,j)}, H^{t-1}_{(i-1,j)}, H^{t-1}_{(i,j-1)}\Big]$$

- For bed level integrator: bed elevation at grid points $(i,j), (i-1,j), (i,j-1), (i+1,j)$, and $(i,j+1)$ and at times $(t-1)$ and (t) and scalar velocity at grid points $(i,j), (i-1,j), (i,j-1), (i+1,j)$, and $(i,j+1)$ and at times $(t+1)$ and (t):

$$\vec{\mathbf{B}} = \Big[Z^t_{b(i,j)}, Z^t_{b(i-1,j)}, Z^t_{b(i,j-1)}, U^t_{(i,j)}, U^t_{(i-1,j)}, U^t_{(i,j-1)}, Z^{t-1}_{b(i,j)}, Z^{t-1}_{b(i-1,j)}, Z^{t-1}_{b(i,j-1)},\\ U^{t+1}_{(i,j)}, U^{t+1}_{(i-1,j)}, U^{t+1}_{(i,j-1)}\Big]$$

As mentioned before, the optimal number of hidden layers and dependent nodes for the ANN models is not known and a trial and error procedure must be applied to find the best network's structure for each model. This was done in the same way as explained in chapter 8. As a result, a detailed description of model development is skipped in this section.

Similarly, the number of hidden layers and neurons were changed until the optimal architecture was determined based upon minimizing the difference among the neural network predicted values and the desired outputs. The training of the neural network models was stopped when either the acceptable level of error was achieved or the number of iterations exceeded a prescribed value. The neural network model configurations that minimized the MAE and RMSE and optimized the R were selected as the optimum networks and the whole analysis was repeated several times. After extensive trial and error processes, three optimal ANNs for velocity, water depth, and bed level change modeling were found. All models consist of only one hidden layer with ten neurons. The logistic sigmoid and linear transfer functions have been used for hidden and output layers, respectively. The Levenberg-Marquardt is applied to all three networks.

9.1.3 Results and discussion

After the training process, the weights and biases parameters of the configured networks are modified and get fixed, which represent the stored knowledge of the ANN models. This knowledge is stored in one input weight matrix $\mathbf{IW_{1,1}}$, one hidden layer weight matrix $\mathbf{LW_{2,1}}$, one bias vector $\vec{\mathbf{b}}_1$, and one bias value b_2 for each network. Finally, using the designed networks, the following equations are received for the normalized velocity, water depth, and bed level change calculations:

$$U_{(i,j)}^{t+1} = \mathbf{LW}_{2,1}^{u} \times \log sig(\mathbf{IW}_{1,1}^{u} \times \mathbf{A}^{\mathrm{T}} + \vec{b}_1^{u}) + b_2^{u} \tag{9.1a}$$

$$H_{(i,j)}^{t+1} = \mathbf{LW}_{2,1}^{h} \times \log sig(\mathbf{IW}_{1,1}^{h} \times \mathbf{A}^{\mathrm{T}} + \vec{b}_1^{h}) + b_2^{h} \tag{9.1b}$$

$$Z_{b(i,j)}^{t+1} = \mathbf{LW}_{2,1}^{z} \times \log sig(\mathbf{IW}_{1,1}^{z} \times \mathbf{B}^{\mathrm{T}} + \vec{b}_1^{z}) + b_2^{z} \tag{9.1c}$$

where superscripts u, h, and z denote the corresponding ANN model (velocity, water depth and bed level integrator, respectively).

Since in this example the water surface elevation is not fixed, the Eq.(8.1) is no longer valid for the flow calculations. As a result, two more ANN models replace Eq.(8.1) for hydrodynamic calculations (velocity and water depth integrators). The integration method for the hydro-morphodynamic modelling is shown in Fig.9.2. One of the advantages of the proposed scheme is that the calculation of sediment transport rate and bed shear stress is not required and the prediction of bed level evolution is only based on the bed elevation and hydrodynamic characteristics. This reduces the computational costs significantly.

To assess the proficiency of the designed ANN models in the new proposed coupling model, the initial values at times $t = 0$ and 500 s are determined at every grid point explained above. The velocity and water depth at one time step ahead $t = 1000$ s are updated using the ANN models (ANN_1 and ANN_2). Using calculated values and applying the next ANN-based approximator (ANN_3) the bed change is calculated. The results are sent back for flow field calculations at the next time step. The procedure is repeated until the last time step is reached. Table.9.1 presents the statistical performances of the model based on a comparison between the predicted bed levels and simulation of TELEMAC-3D modeling system at different times. Given the morphological change in the channel, the results indicate that the designed ANN performs well. This can be deduced from the high values of R along with small values of RMSE and MAE. According to this table, the model performance reduces as time increases. As explained before, the reason is an error accumulation problem, i.e. errors committed in the past are propagated into future predictions. This is more serious here as three ANNs exist instead of one. Figs.9.3 and 9.4 show the simulation of TELEMAC

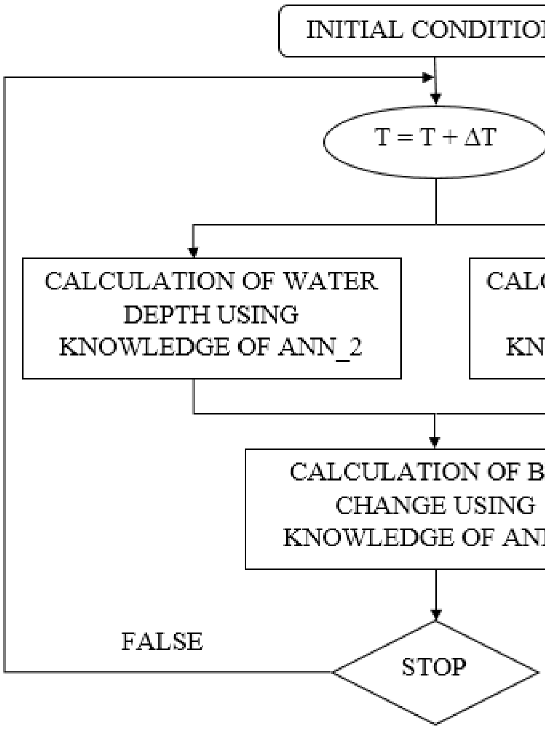

INITIAL CONDITION
T = T + ΔT
CALCULATION OF WATER DEPTH USING KNOWLEDGE OF ANN_2
CALCULATION OF SCALAR VELOCITY USING KNOWLEDGE OF ANN_1
CALCULATION OF BED CHANGE USING KNOWLEDGE OF ANN_3
FALSE
STOP

Figure 9.2: Proposed hydromorphodynamic modelling system

and proposed hydrodynamic-morphologic modeling system after 15000 and 25000 seconds, respectively. In order to make a better comparison between the models and analyze our ANN-based scheme, 2D view of Figs.9.3 and 9.4 on xy-plane are plotted in Figs.9.5 and 9.6, respectively.

Table 9.1: Statistical performance of the first proposed concept modeling the bed change in a straight channel at different times

Performance index	15000 sec	25000 sec
R	0.9876	0.9854
RMSE	0.0119	0.0129
MAE	0.0055	0.0064

As can be seen from these figures, there is a good agreement between the bed level simulated using the ANN-based model and TELEMAC model. The difference between two models are depicted in Figs.9.3(c) and 9.4(c) shows that the difference between the models are very low.

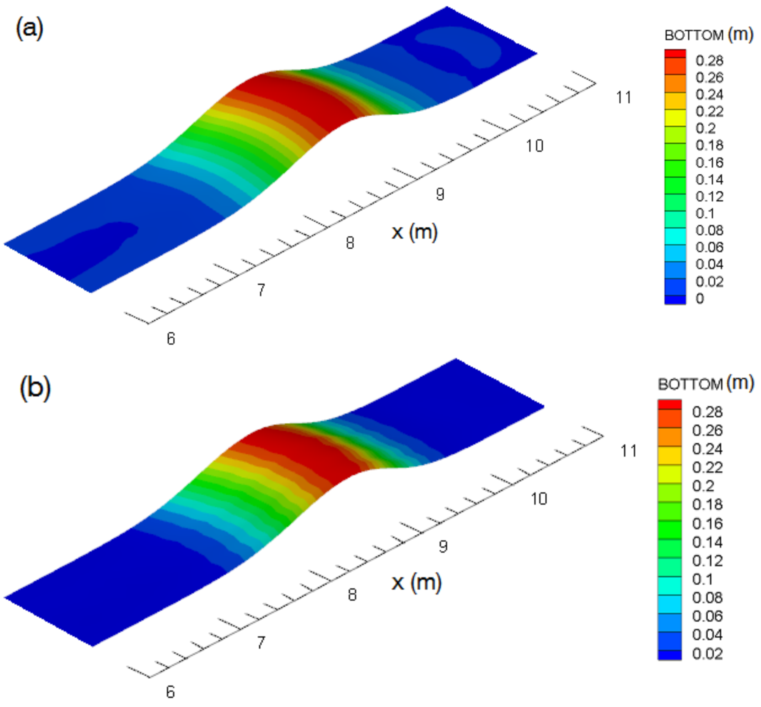


Figure 9.3: Bed level simulation after 15000 sec; (a) TELEMAC-3D, and (b) ANN-based model

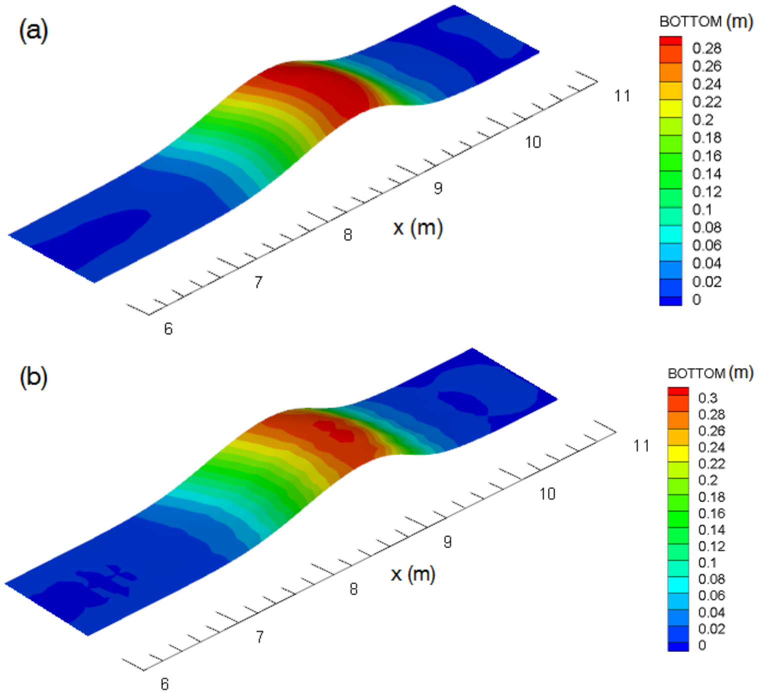


Figure 9.4: Bed level simulation after 25000 sec; (a) TELEMAC-3D, and (b) ANN-based model

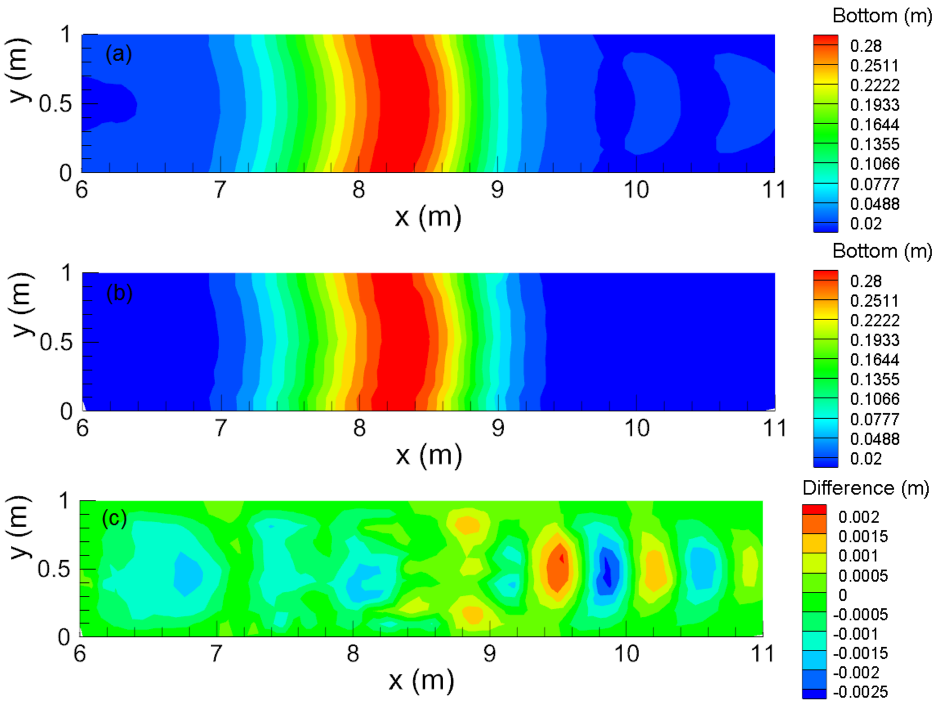


Figure 9.5: 2D comparison of bed level simulation after 15000 sec; (a) TELEMAC-3D, and (b) ANN-based model, (c) Difference between two models

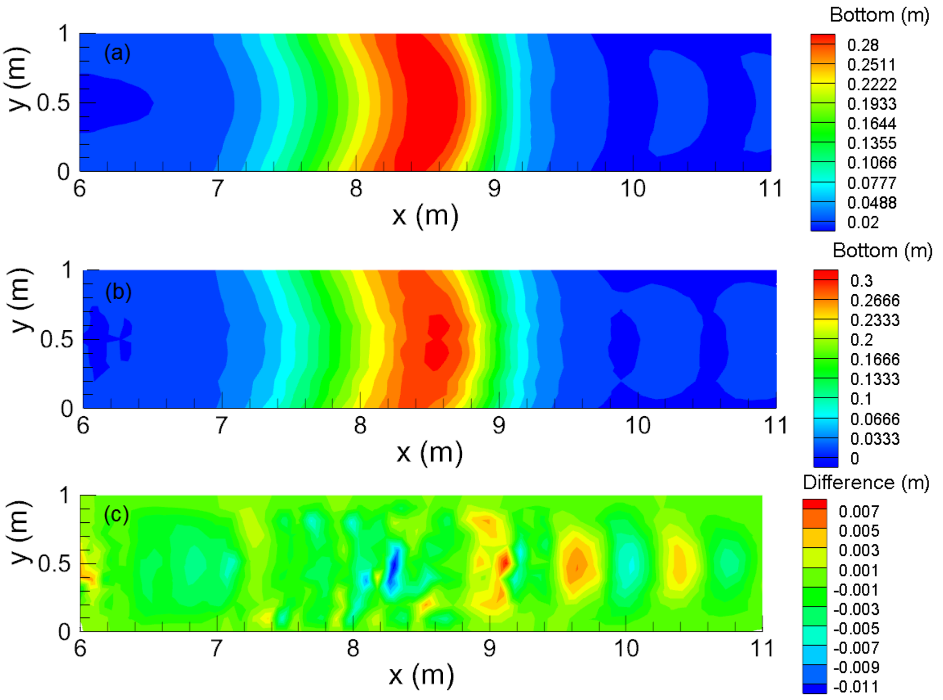


Figure 9.6: 2D comparison of bed level simulation after 25000 sec; (a) TELEMAC-3D, and (b) ANN-based model, (c) Difference between two models

9.2 Flow and bed change in a curved channel

The mechanics of sediment transport in channel bends, frequently appearing in natural rivers, is much more complex than that in straight channels. Therefore, the second test case models the evolution of the bed in a 180^o channel bend under unsteady-flow conditions with non-uniform sediment (Yen and Lee, 1995). The test case is calibrated and already available in TELEMAC program system. The unsteadiness of flow in this test case certainly has some effects on the structure of the flow field, thereby affecting the motion of sediment particles. The initial geometry of the test case is shown in Fig.9.7.

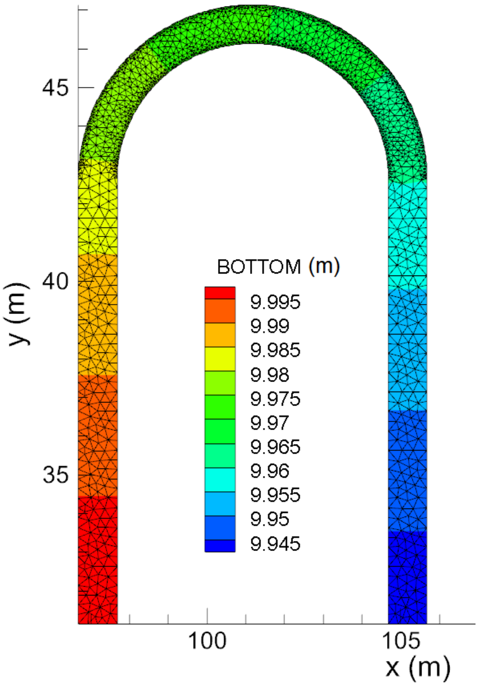


Figure 9.7: Initial geometry of the channel

The width of the channel is 1 m and the radius of curvature 4 m. At both ends of the channel, there is a straight reach of 11.5 m. Starting from a horizontal flat bed in the transverse direction with a longitudinal bed slope of 2%, a typical bank cross-section with a scour at the outer bank and deposition at the inner bank is formed without imposing any sediment discharge upstream. In the experiment, a layer of sand around 20 cm thick, with d_{50} of 1.0 mm is placed on the bed. The hydrodynamic model is based on the 2D Shallow Water Equations (SWE) in unstructured mesh grids. The base flow is set at 0.02 m^3/s, determined according to the condition of incipient motion for the sediment with d_{50} of 1.0 mm.

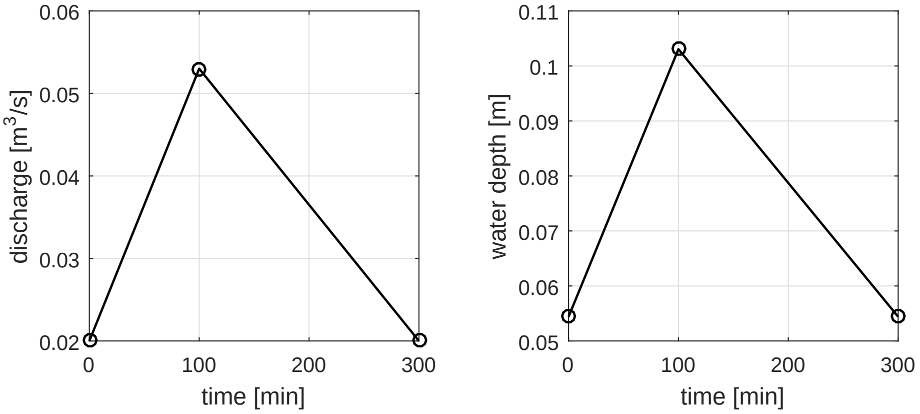


Figure 9.8: Hydrograph used in the experiment and simulation

The flow discharge first linearly increased during the 100 min up to 0.053 m^3/s and then progressively decreased back to its initial value. The associated water depth varied between 0.0544 and 0.103 m. Fig.9.8 presents the open boundary file used in TELEMAC including flow rate and water depth varying over time. As an initial condition, constant depth is applied, which initializes the water depth at the given value (0.0544 m in this example). The bedload formulae of Meyer-Peter and Müller (1948) is applied with a critical Shields parameter of 0.047, corresponding to the mean grain size. The bed roughness is taken about three times the mean diameter (k_s = 0.035 m), which corresponds to flatbed conditions.

9.2.1 Data selection for ANN

In this test case, the program was run for 6000 s. For the ANN models, a large time step of Δt = 200 s is used. Similar to the first case, the required results from TELEMAC-2D model including water depth, scalar velocity, and bed elevation are obtained at all spatial nodes and time steps. To get the required data on our desired structured points, the interpolation technique was applied. Fig.9.9 shows schematically the desired structured grids on the study domain. Lastly. the obtained data set for each network are randomly divided into three subsets: training, validation and testing.

9.2.2 ANN design

The same input parameters as section (9.1.2) for flow field and bed change calculation are used. All three ANN models are structured with only one hidden layer consisting of 10 neurons. The logistic sigmoid and linear transfer functions are considered for the hidden layer and the output one, respectively.

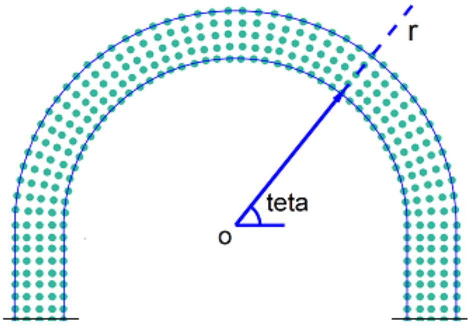


Figure 9.9: Schematic of the structured grids

9.2.3 Results and discussion

After the training process and updating the weights and biases parameters, three ANN-based equations are obtained. Since the same number of hidden layer and neurons, and the same transfer functions are applied here, these equations are comparable to Eqs.(9.1a), (9.1b), and (9.1c). The only difference is that these equations are not identical in terms of weights and biases parameters.
The initial values at time $t = 0$ and 200 s are defined at every grid point (i, j). Similarly, the velocity and water depth at one time step ahead $t = 400$ s are calculated using the ANN-based Eqs.(9.1a) and (9.1b). Using calculated values and applying the ANN-based Eq.(9.1c) the bed change is calculated. The results are sent back for flow field calculations at the next time step. The procedure is repeated until the last time step is reached. Table.9.2 presents the statistical performance indices of the model based on a comparison between the predicted bed levels and simulation of the TELEMAC-2D modeling system at different times.

Table 9.2: Statistical performance of the first proposed concept modeling the bed change in a channel curve at different times

Performance index	6000 sec	10000 sec	20000 sec
R	0.9999	0.9959	0.9865
RMSE	0.0002	0.0016	0.0035
MAE	0.0001	0.0011	0.0026

Figs.9.10(a) and (b) plot the bed level simulated by the TELEMAC-2D model and the ANN-based model after 6000 s, respectively. The difference between two models is illustrated in Fig.9.10(c) to make a comparison between the models easier. As can be seen from this figure, the difference between the bed level calculations by two models

is small which proves quality of the proposed model.

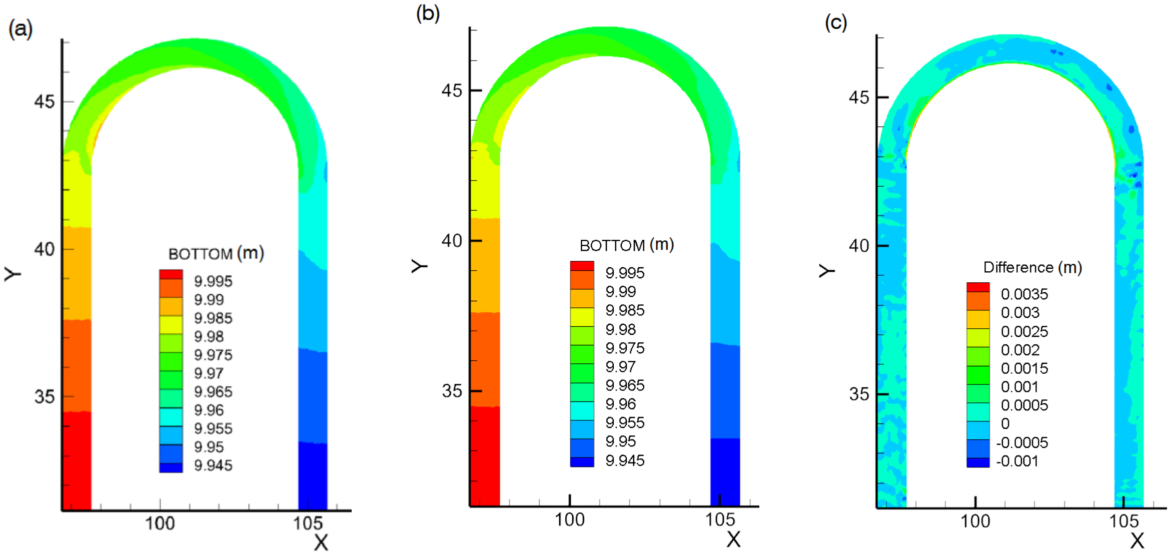


Figure 9.10: Bed level simulation after 6000 sec; (a) TELEMAC-2D, and (b) ANN-based model, (c) Difference between two models

The long-term differences between the aforementioned models after 10000 and 20000 sec simulations are illustrated in Figs.9.11(a) and (b), respectively. According to Table.9.2, the R, RMSE, and MAE are in the range of 0.9999 to 0.9865, 0.0002 to 0.0035, and 0.0001 to 0.0026, respectively. These values and figures confirm good quality of results. It is noteworthy to mention that the simulation takes only few seconds by applying this new proposed concept while it would take the TELEMAC system few minutes.

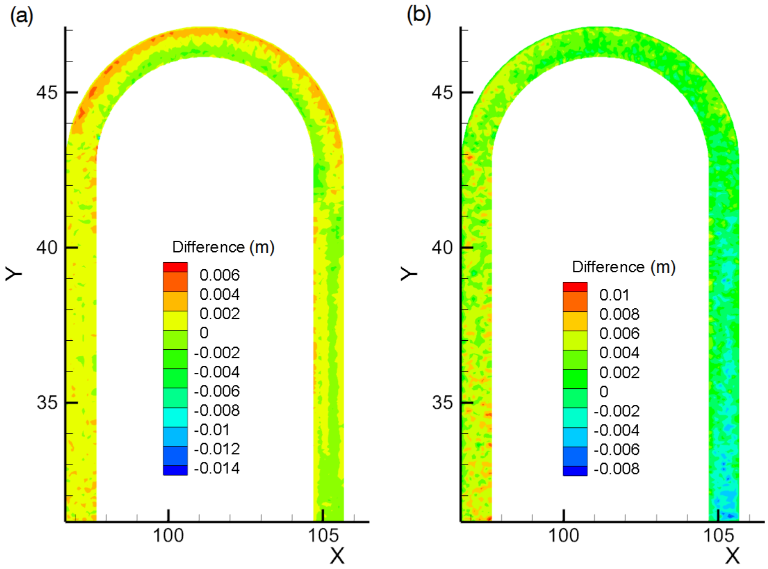


Figure 9.11: Difference between two models; (a) After 10000 sec, (b) After 20000 sec

Chapter 10

Integrating ANN model into TELEMAC system

All the simulation modules of TELEMAC open source software are written in Fortran 90, with no use of the specific language extensions in a given machine. When using a simulation module from the TELEMAC system, the user may have to program specific subroutines that are not in the code's standard release. In particular, that is made through a number of so-called user subroutines. These subroutines are written so that they can be modified, with the help of the guide for programming in the TELEMAC system. The procedure to be carried out in that case is comprised of the following steps:

1. Recovering the standard version of the user subroutines provided with the system, and copying them into a file that will be the specific Fortran file of the given case,
2. Modifying the subroutines according to the model we wish to build,
3. Linking up the set of subroutines into a single Fortran file that will be compiled during the TELEMAC start procedure.

During this programming phase, users may need to have access the various software variables. This is possible from any subroutines by using the structures of Fortran 90 gathered into a module type component. This feature of TELEMAC open source software is used in this chapter to introduce a new proposed scheme for hydro-morphodynamic modeling.

According to our second proposed concept for hydro-morphodynamic modeling, the required hydrodynamic variables (water depth (H) and scalar velocity (U)) calculated by TELEMAC-2D or -3D are sent to ANN based approximator at each coupling period Δt_{sed} (see chapter 5). This is achieved by modification of the subroutines according to the model we wish to build, and replacing our obtained explicit equation for calculation of bed evolution. One of the most important subroutines of SISYPHE, which is required to be modified in this new scheme, is the bedload_solves_fe.f. This subroutine solves the Exner equation to calculate the bed evolution. After the training process of ANN model, the weights and biases parameters of the configured network are adjusted so that an explicit equation for calculation of morphological bed level changes is obtained. This equation, which calculates the new bed level for each node at every time step, will be used to modify subroutine bedload_solves_fe.f. In this chapter, the efficiency of the model is tested by two examples used in chapter 9.

10.1 Flow and bed change in a straight channel

10.1.1 Model setup

The test case of section 9.1, is used here to show the ability of the proposed integration model to reproduce the morphodynamic modeling. The same data set that was generated in the section 9.1.1, can be used for networks training. The most important difference here is that the data on unstructured grids are needed. The reason is that all modules of the TELEMAC system are based on unstructured grids. Therefore, the new bed level calculated by ANN based approximator must be sent to TELEMAC-2D/3D for every unstructured grid points at each coupling period (Δt_{sed}).
The bed elevation and scalar velocity at each grid and its neighbor points are considered as the most important input parameters. Since the location of neighbor points around a node is irregular, another parameter must be introduced to represent the strength of each neighbor point. As a result, the area of each element around a node is considered as the other input parameter. Finally, after a trial and error procedure, the following input matrix including the mentioned parameters at the time steps t and $t-1$ are considered for ANN model to predict the bed level change at the grid point i and at one time step ahead:

$$\mathbf{B} = \left[\begin{matrix} Z_1^t & U_1^t & Z_{11}^t & U_{11}^t & \cdots & Z_{1m}^t & U_{1m}^t & Z_1^{t-1} & U_1^{t-1} & Z_{11}^{t-1} & U_{11}^{t-1} & \cdots \\ Z_2^t & U_2^t & Z_{21}^t & U_{21}^t & \cdots & Z_{2m}^t & U_{2m}^t & Z_2^{t-1} & U_2^{t-1} & Z_{21}^{t-1} & U_{21}^{t-1} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Z_p^t & U_p^t & Z_{p1}^t & U_{p1}^t & \cdots & Z_{pm}^t & U_{pm}^t & Z_p^{t-1} & U_p^{t-1} & Z_{p1}^{t-1} & U_{p1}^{t-1} & \cdots \end{matrix}\right.$$

$$\left.\begin{matrix} \cdots & Z_{1m}^{t-1} & U_{1m}^{t-1} & A_{11} & \cdots & A_{1m} \\ \cdots & Z_{2m}^{t-1} & U_{2m}^{t-1} & A_{21} & \cdots & A_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & Z_{pm}^{t-1} & U_{pm}^{t-1} & A_{p1} & \cdots & A_{pm} \end{matrix}\right]$$

where p is the maximum number of grid points, m is the maximum number of neighbor points around a node in entire domain, Z_i is bed elevation at grid point i, U_i is scalar velocity at grid point i, A_{ij} is the area of j-th element around i-th node, and Z_{ij} and U_{ij} are the bed elevation and scalar velocity of j-th neighbor node around i-th node, respectively.

10.1.2 Model development

It was found after a trial and error procedure and trying different model structures that the accuracy of the ANN model does not significantly change when varying the number of hidden layers and neurons. On the other hand, the generated explicit equation forms a model with only one hidden layer, which is much simpler than those including more layers. Therefore, I came up with the conclusion to use a model consisting of only one hidden layer with ten neurons. The logistic sigmoid and linear transfer functions have been used for hidden and output layers, respectively. The Levenberg-Marquardt was also selected as the most suitable algorithm for the model training.

10.1.3 Results and discussion

Now we analyze the capability of the proposed concept and newly modified subroutine that implements the designed ANN model for prediction of bed level changes into the TELEMAC-2D and -3D system. Applying this designed ANN model, the weights and biases values have been specified after successful learning and validating processes for each test case. They represent the stored knowledge of the ANN for bed level change modeling, which are separated in one input weight matrix $\mathbf{IW_{1,1}}$, one hidden-layer weight matrix $\mathbf{LW_{2,1}}$, one bias vector $\vec{\mathbf{b_1}}$ and one bias value b_2. Finally, the equation for calculation of the bed changes is received as follows:

$$Z_{b(i)}^{t+1} = \mathbf{LW}_{2,1} \times \log sig(\mathbf{IW}_{1,1} \times \mathbf{B}^{\mathrm{T}} + \vec{b_1}) + b_2 \tag{10.1}$$

This equation is then used in subroutine bedload_solves_fe.f to calculates the new bed level for each node at every time step.

After modification of the subroutine, we have to recompile the code. Consequently the program can be ready to be launched with the user-defined functions in the subroutine. Using the initial values, the program can simulate the bed levels at every grid point i at one time step ahead. The minor time step of $\Delta t_{hyd} = 1$ s is chosen for hydrodynamic calculations while the coupling period with the morphologic module is $\Delta t_{sed} = 500$ s. The bed level remains fixed during the hydrodynamic computation. After each 500 sec, the results are sent to bed level change module and during the computation of the bed level the flow is invariant to the bed level changes. After updating the bed changes using the knowledge of ANN model, the results are sent back to the TELEMAC-3D for flow field calculations. The procedure is repeated until the last time step is reached.

The performance of the model is analyzed using statistical indicators such as Correlation Coefficient (R), Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). The Table 10.1 presents the accuracy of the proposed hydromorphodynamic model after 15000 and 25000 sec. Figs.10.1 and 10.2 show the simulation of the original TELEMAC-3D and the proposed modeling system after 15000 and 25000 seconds, respectively. A 2D comparison between the models are illustrated in Figs.10.3 and 10.4, respectively.

Table 10.1: Statistical performance of the second proposed concept modeling the bed change in a straight channel at different times

Performance index	15000 sec	25000 sec
R	0.8938	0.9149
RMSE	0.0266	0.0602
MAE	0.0146	0.0384

According to Table.10.1, the proposed concept, which is integrating ANN model into TELEMAC system, performs poorly for the first test case. The statistical performance of the model indicates insufficient quality of results. The poor results can be also noticed from the figures. This might be due to the location of neighbor nodes as they play a crucial role in this concept.

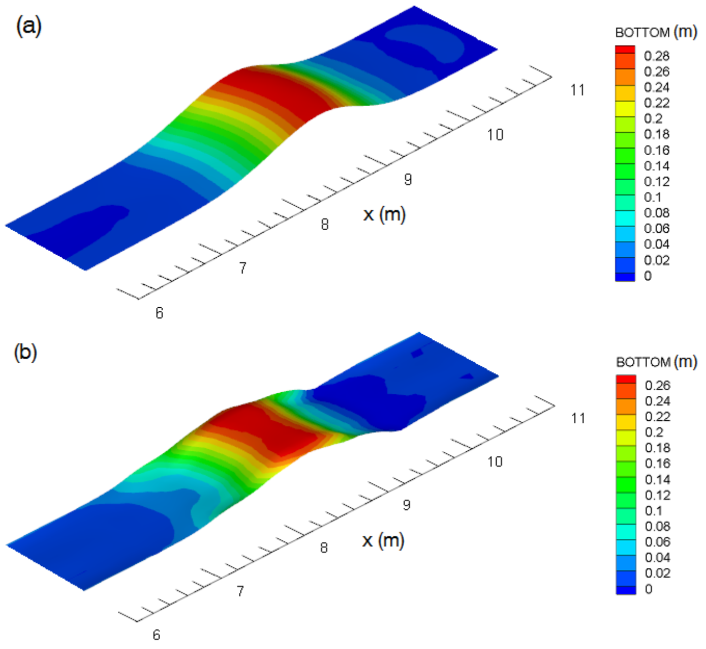


Figure 10.1: Bed level simulation after 15000 sec; (a) TELEMAC-3D, and (b) TELEMAC-3D implementing ANN-based model

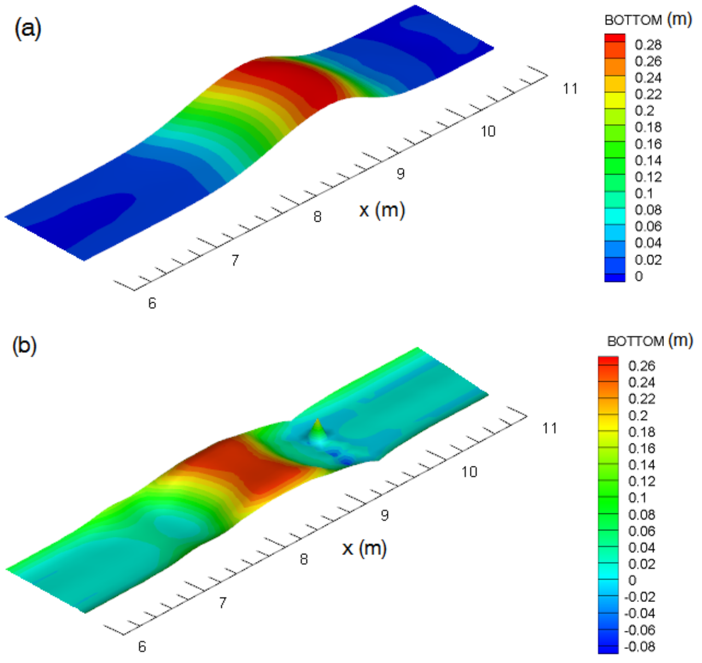


Figure 10.2: Bed level simulation after 25000 sec; (a) TELEMAC-3D, and (b) TELEMAC-3D implementing ANN-based model

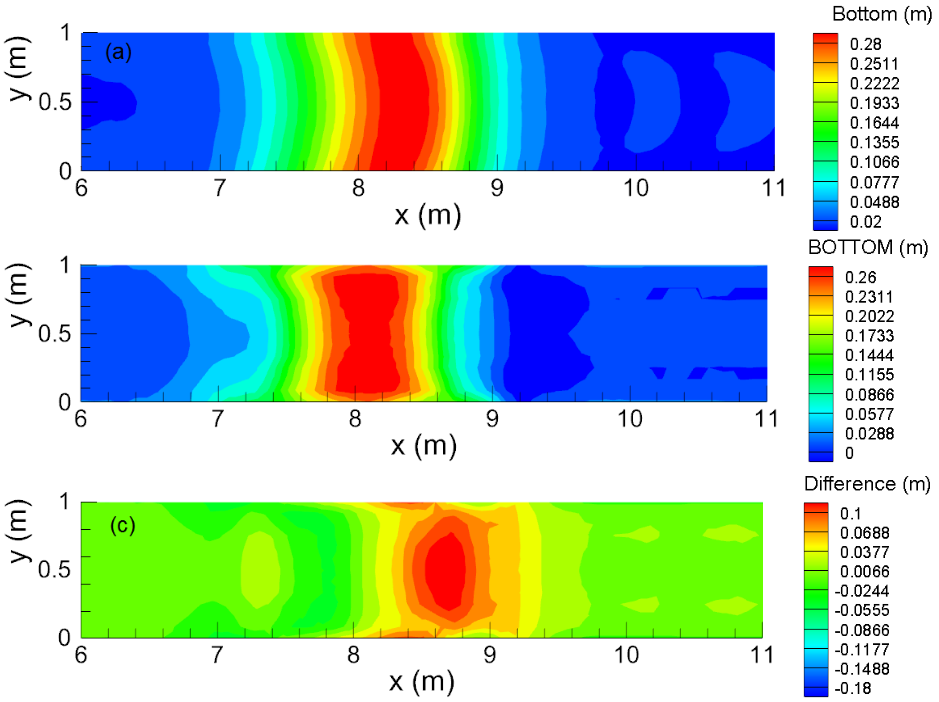


Figure 10.3: 2D comparison of bed level simulation after 15000 sec; (a) TELEMAC-3D, (b) TELEMAC-3D implementing ANN-based model, and (c) Difference of models

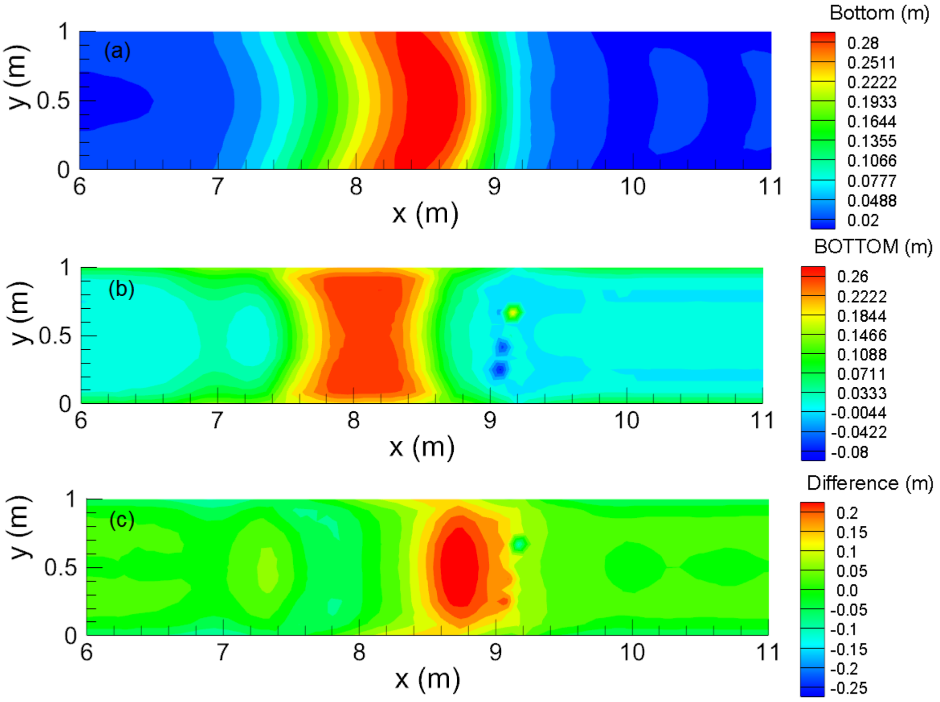


Figure 10.4: 2D comparison of bed level simulation after 25000 sec; (a) TELEMAC-3D, (b) TELEMAC-3D implementing ANN-based model, and (c) Difference of models

10.2 Flow and bed change in a curved channel

10.2.1 Model setup

In this part, we use the test case of section 9.2 and apply the same data set generated in the section 9.2.1 for networks training but according to the unstructured grids. The same input matrix (B) can be used here as well.

10.2.2 Model development

No changes are applied here to the network structure. The same model architecture as section 10.1.2 is used for this example.

10.2.3 Results and discussion

First, the initial values are defined at every grid point i. The bed levels at one time step ahead are calculated using the explicit equation implemented in the modified subroutine. We choose a minor time step for hydrodynamic calculations ($\Delta t_{hyd} = 0.1$ s) and a big time step for the bed level change simulation ($\Delta t_{sed} = 200$ s). Table 10.2 shows the tabulation of the statistical performance of the proposed hydromorphodynamic system which implements ANN into TELEMAC open source software after 6000, 10000, and 20000 s. The results confirm that there is a good agreement between the proposed scheme of hydrodynamic-morphologic system and the original TELEMAC modelling system. The differences between the original TELEMAC-2D model and the modified one after 6000, 10000 and 20000 s are illustrated in Figs.10.5(a), (b) and (c), respectively.

Table 10.2: Statistical performance of the second proposed concept modeling the bed change in a channel curve at different times

Performance index	6000 sec	10000 sec	20000 sec
R	0.9859	0.9821	0.9700
RMSE	0.0020	0.0023	0.0030
MAE	0.0011	0.0017	0.0023

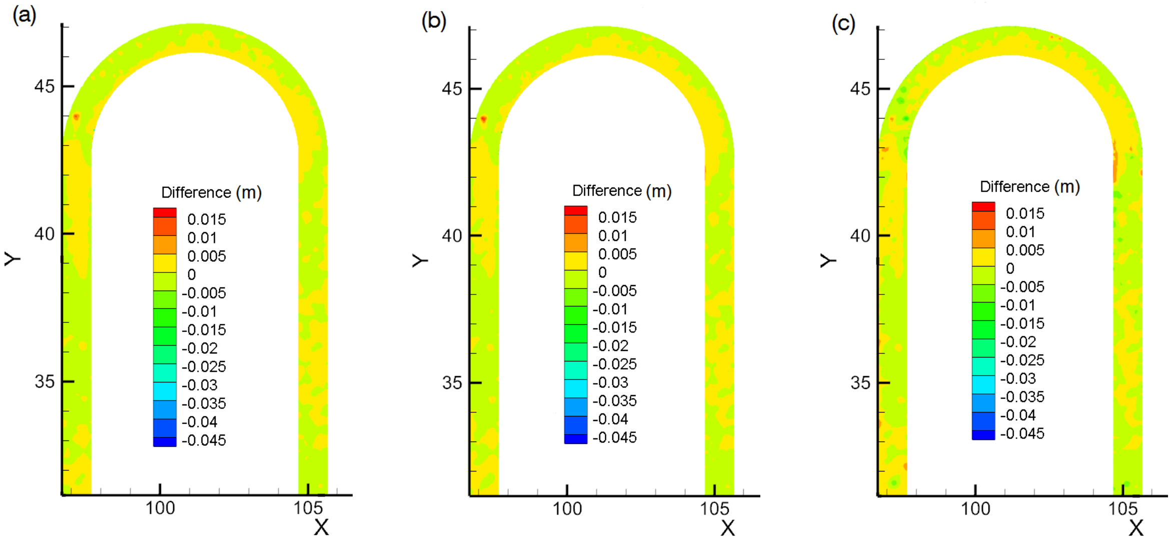


Figure 10.5: Difference between original TELEMAC-2D and TELEMAC-2D implementing ANN based-model; (a) After 6000 sec, (b) After 10000 sec, and (c) After 20000 sec

The predicted results show that the newly proposed scheme could perform well for long-term time series prediction and ANN results could be integrated in the hydro-morphological model systems.

Chapter 11

Results

11.1 Summary of the results

Current research focuses on developing new schemes of hydromorphodynamic modeling systems that integrate data-driven models, particularly Artificial Neural Networks, into hydromorphological models for fluvial channels. As the first step, the fundamentals of data-driven methods were discussed and several learning algorithms such as MATLAB existing learning rules and those which were developed in FORTRAN were applied to these models to improve their efficiency and effectiveness. Two test cases were considered and their performance in the context of prediction and function approximation was analyzed.

In case of the contraction scour depth prediction, an ANN and ANFIS were developed using the data set previously generated by various investigators for clear-water contraction scours. The designed networks applied the learning algorithm proposed by Levenberg-Marquardt in batch mode. The predicted results demonstrate the capacity of the new approaches for the estimation of contraction scour. They indicate that using suitable ANN and ANFIS networks for the scour modeling can lead to more accurate predictions than using other existing methods. The errors of the ANFIS model are less than those of the ANN model. However, it is too complicated to obtain a general ANFIS-based equation for scour depth which may be useful for engineers working in practice.

An attempt was also made to investigate the application of the LM learning algorithm in ANFIS to predict the daily suspended sediment concentration for the Schuylkill River in the United States. The FORTRAN programming language was utilized to construct ANFIS models using the LM algorithm, and the MATLAB toolbox was used to build ANFIS models based on the Hybrid and BP learning algorithms.

Comparison of the predicted results indicated that the models trained with the LM algorithm better predicted the SSC than the other algorithms. To recognize the best algorithm between the LM, Hybrid, and BP, evaluations of the performance indexes and the generalization capability of the network were presented. The results show that the LM learning rule could improve the performance and generalization of ANFIS model. It was also found that the ANFIS model, which used the LM algorithm as the learning rule, had the best accuracy when predicting the cumulative suspended sediment load. The ANFIS models trained with the LM algorithm showed an improvement in the coefficient of determination, the root mean square error, and the mean absolute error values compared to the Hybrid model.

An analytical approximation based on the equation of conservation of sediment was applied to generate data used for training and testing different ANN models. By evaluating the calculated results, the related parameters for an optimal network structure was obtained for the simulation of bed level changes in a straight alluvial channel under a steady flow discharge and simple assumption for sediment transport processes. The predicted results showed that for the simple case of 1D morphological problems considered in this study, artificial neural networks perform well. The calculated results also showed that the ANN could perform well for long term time series prediction. Moreover, the study demonstrated that a well-trained ANN model could be integrated into the hydromorphological model systems and perform like conventional numerical operators.

After a successful application of data-driven models for different hydraulic test cases, especially 1D simulation of bed level changes, two new integration methods for the hydromorphodynamic modelling were proposed. In the first scheme, which is simpler, 2 or 3 ANN models can be utilized for hydrodynamic calculations. At each time step, the hydrodynamics variables (velocity field and water depth) are transferred into the morphodynamic model (another ANN-based approximator), which then sends back the updated bed elevation to the hydrodynamic model. In the second scheme, the sediment transport model SISYPHE was replaced by an ANN model coupled with either the 2D or 3D flow models. The time-step of the morphodynamic part (ANN-based approximator) is much larger than the time step of the TELEMAC-2D/3D. During the flow computation, the bed level is assumed constant. The main advantages of the proposed schemes are that there is no need for bed shear stress and sediment transport rate calculations.

To test the efficiency and effectiveness of each proposed scheme, two different test cases were considered in this thesis. The first test case included a straight channel with a finite amplitude perturbation of the bed level in the TELEMAC-3D modeling system. The evolution of the bed in a 180°channel bend under unsteady-flow conditions with uniform sediment was also considered as the second test case. The results showed that the model predictions of the first proposed scheme for both test cases agreed closely with the TELEMAC simulation. It was concluded from the results that the proposed scheme performed well for short and long-term time series predictions and would significantly reduce the computational time. The comparison of the prediction accuracies of the second proposed scheme and TELEMAC simulation indicated that the proposed model could sufficiently simulate and predict the morphological bed level changes for the second test case. However the second scheme did not perform as well for the first test case as compared to the first scheme. In general, it can be concluded that applying the schemes developed in this research would reduce the computational costs and simplify the computational procedure, as they did not require the shear stress and sediment transport rate calculations. It is also proved that these schemes could perform well for long-term time series prediction, even though the ANN models were not trained for that length of time.

11.2 Recommendation and further work

The current research applies a very simple structure of Artificial Neural Network with only one hidden layer and ten neurons in its hidden layer. A useful direction would be to use deep learning which is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain i.e. Artificial Neural Networks. Deep learning methods aim at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features. Automatically learning features at multiple levels of abstraction allow a system to learn complex functions by mapping the input to the output directly from data, without depending completely on human-crafted features. The hierarchy of concepts allows the computer to learn complicated concepts by building them out of simpler ones. It is also recommended to test and develop the new proposed concepts for real rivers.

References

Alam, M. (1998): Application of MacCormack scheme to the study of aggradation-degradation in alluvial channels. *M. Sc. Engineering thesis work, Department of Water Resources Engineering, BUET, Dhaka.*

Altun, H.; Bilgil, A.; Fidan, B. C. (2007): Treatment of multi-dimensional data to enhance neural network estimators in regression problems. *Expert Systems with Applications*, *32*(2), 599-605.

Amoudry, L. (2008): A review on coastal sediment transport modelling.

Araghinejad, S. (2013): *Data-driven modeling: using MATLAB®in water resources and environmental engineering* (Vol. 67). Springer Science & Business Media.

Beale, M.H.; Hagan M.T.; Demuth, H.B. (2013): neural network toolbox–user's guide R2013b, Natick, Massachusetts, the Mathworks, Inc.

Black, N. T.; Ertel, W. (2011): Introduction to artificial intelligence.

Blumer, A.; Ehrenfeucht, A.; Haussler, D.; Warmuth, M. K. (1989): Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM (JACM)*, *36*(4), 929-965.

Bogárdi, J.; Bogárdi, J. (1974): Sediment transport in alluvial streams (No. 04; TC175. 2, B6.). Budapest: Akademiai Kiado.

Bose, N.K.; Liang, P. (1996): Neural network fundamentals with graphs, algorithms, and application. *McGraw-Hill*, New York.

Brown, C. B. (1950): Sediment transportation. Engineering hydraulics, 12.

Bui, M. D.; Kaveh, K., Penz, P.; Rutschmann, P. (2015a): Contraction scour estimation using data-driven methods. Journal of Applied Water Engineering and Research, 3(2), 143-156.

Bui, M. D.; Kaveh, K.; Rutschmann, P. (2015b): Integrating artificial neural networks into hydromorphological model for fluvial channels. In Proceedings of the 36th IAHR World Congress (pp. 8-pages).

Bui, M. D.; Kaveh, K.; Rutschmann, P. (2017): Performance Analysis Of Different Model Architectures Utilized In An Adaptive Neuro Fuzzy Inference System For Contraction Scour Prediction. IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE).

Chanson, H. (2004): The hydraulics of open channel flow: An introduction. Elsevier Butterworth Heinemann, Amsterdam, 2nd edn.

Chien, N. (1956): The present status of research on sediment transport. Trans. ASCE, 121; 833-68.

Cunge, J.A.; Holly, F.M.; Verway, A. (1980): Practical aspects of computational river hydraulics. Pitman.

Dey, S.; Raikar, R. V. (2005): Scour in long contractions. *Journal of hydraulic engineering*, *131*(12), 1036-1049.

Duboys, M. P. E. (1879): Etudes du régime du Rhone et de l'action exercé par les eaux sur un lit á fond de graviers indéfiniment affouillable. Ann Ponts et Chaussées, Ser, 5(18), 141-95.

Einstein, H. A. (1941): Formulas for the transportation of bed load. Transactions of ASCE, 561-597.

Elman, J. L. (1990): Finding structure in time. Cognitive science, 14(2), 179-211.

Exner, F.M. (1925): Uber die wechselwirkung zwischen wasser und geschiebe in flussen. *Akad. der Wiss in Wien, Math-Naturwissenschafliche Klasse, Sitzungsberichte, Abt IIa*, *134*, 165-203.

Fischler, M. A.; Firschein, O. (1987): Intelligence: the eye, the brain, and the computer.

Gessler, J. (1971): Critical shear stress for sediment mixtures. Proceedings of the 14th Congress of International Association for Hydraulic Research, Vol. 3, C1_1-C1_8.

Gilbert, G. K.; Murphy, E. C. (1914): The transportation of debris by running water (No. 86). US Government Printing Office.

Gill, M.A. (1981): Bed erosion in rectangular long contraction. *Journal of the Hydraulics Division*, *107*(3), 273-284.

Godunov, S. K. (1959). A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. Matematicheskii Sbornik, 89(3), 271-306.

Graf, W. H. (1971): *Hydraulics of sediment transport*, McGraw-Hill, New York, 287–291.

Grass, A.J. (1981): Sediment Transport by Waves and Currents, SERC London Cent. Mar. Technol., Report No: FL29.

Haykin, S. (2005): Neural networks, a comprehensive foundation, Second Edition, Pearson Education (Singapore) Pte. Ltd., Indian Branch.

Hervouet, J. M. (2007): Hydrodynamics of free surface flows: modelling with the finite element method. John Wiley & Sons.

Hornik K. (1989): Multilayer Feed-forward Networks are Universal Approximator. Neural Networks, 2, 359-66.

Huber, D. (2014): Data driven approaches to predicting the morphology of alluvial rivers, Master's thesis at the Technical University Munich (Department of Hydraulic and Water Resources Engineering).

Hudson, J. (2001): Numerical techniques for morphodynamic modelling (Doctoral dissertation, University of Reading).

Hudson, J.; Damgaard, J.; Dodd, N.; Chesher, T.; Cooper, A. (2005): Numerical approaches for 1D morphodynamic modelling. *Coastal Engineering*, *52*(8), 691-707.

HydroQual Inc., (2002): User's Manual, Version 1.3. HydroQual, Inc., Mahwah, NJ, USA 188 pp.

Jain, A. K.; Mao, J.; Mohiuddin, K. M. (1996): Artificial neural networks: A tutorial. Computer, 29(3), 31-44.

Jang, J.S. (1993): ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, *23*(3), 665-685.

Jang, J.S.R.; Mizutani, E. (1996): Levenberg-Marquardt method for ANFIS learning. In *Fuzzy Information Processing Society. NAFIPS., 1996 Biennial Conference of the North American* (pp. 87-91). IEEE.

Jang, J.S.; Sun, C.T. (1995): Neuro-fuzzy modeling and control. *Proceedings of the IEEE*, *83*(3), 378-406.

Jang, J.S.; Sun, C.T.; Mizutani, E. (1997): Neuro-fuzzy and soft computing, a computational approach to learning and machine intelligence.

Johnson, H.K.; Zyserman, J.A. (2002): Controlling spatial oscillations in bed level update schemes. *Coastal Engineering*, *46*(2), 109-126.

Julien, P. Y. (2010): Erosion and sedimentation. Cambridge University Press.

Kandel, A. (1991): *Fuzzy expert systems*. CRC press.

Kassem, A.A.; Chaudhry, M.H. (2002): Numerical modeling of bed evolution in channel bends. *Journal of Hydraulic Engineering*, *128*(5), 507-514.

Kaveh, K.; Bui, M. D.; Rutschmann, P. (2015a): Improvement of ANFIS model by developing of novel hybrid learning algorithms for contraction scour modeling. Mathematics in Engineering, Science & Aerospace (MESA), 6(4).

Kaveh, K.; Bui, M. D.; Rutschmann, P. (2015b): New hybrid learning algorithms in adaptive neuro fuzzy inference systems for contraction scour modeling. In Proc. of the 14th International Conference on Environmental Science and Technology Rhodes, Greece.

Kaveh, K., Bui, M. D., & Rutschmann, P. (2016): A new approach for morphodynamic modeling using integrating ensembles of artificial neural networks. In “Wasserbau-mehr als Bauen im Wasser". Beiträge zum 18. Gemeinschafts-Symposium der Wasserbau-Institute TU München, TU Graz und ETH Zürich (pp. 304-315).

Kaveh, K.; Bui, M. D.; Rutschmann, P. (2017): A comparative study of three different learning algorithms applied to ANFIS for predicting daily suspended sediment concentration. International Journal of Sediment Research, 32(3), 340-350.

Kaveh, K., Bui, M.D., & Rutschmann, P. (2018): Development of an Artificial-Neural-Network-based concept for hydro-morphodynamic modelling in rivers, The 5th IAHR Europe Congress, 2018,Trento, Italy.

Kisi, Ö. (2004): Multi-layer perceptrons with Levenberg-Marquardt training algorithm for suspended sediment concentration prediction and estimation. *Hydrological Sciences Journal*, *49*(6).

Kisi, Ö.; Haktanir, T.; Ardiclioglu, M.; Ozturk, O.; Yalcin, E.; Uludag, S. (2009): Adaptive neuro-fuzzy computing technique for suspended sediment estimation. *Advances in Engineering Software*, *40*(6), 438-444.

Komura, S. (1966): Equilibrium depth of scour in long constrictions. *Journal of the Hydraulics Division*, *92*(5), 17-37.

Kubatko, E.J.; Westerink, J.J.; Dawson, C. (2006): An unstructured grid morphodynamic model with a discontinuous Galerkin method for bed evolution. *Ocean modelling*, *15*(1), 71-89.

Latteux, B. (1995): Techniques for long-term morphological simulation under tidal action. *Marine Geology*, *126*(1-4), 129-141.

Laurien, E.; Oertel Jr., H. (2013): Numerische Strömungsmechanik: Grundgleichungen und Modelle-Lösungsmethoden-Qualität und Genauigkeit. Springer-Verlag.

Laursen, E.M. (1963): An analysis of relief bridge scour. *Journal of the Hydraulics Division*, *89*(3), 93-118.

Lawrence, S.; Giles, C.L.; Tsoi, A.C. (1998): *What size neural network gives optimal generalization? Convergence properties of backpropagation.*

Lawrence, S.; Giles, C.L. (2000): Overfitting and neural networks: conjugate gradient and backpropagation. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on* (Vol. 1, pp. 114-119). IEEE.

Legates, D.R.; McCabe, G.J. (1999): Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation. *Water resources research*, *35*(1), 233-241.

Leliavsky, S. (1955): *An introduction to fluvial hydraulics*, Constable, London, 24–33.

Lesser, G.R.; Roelvink, J.A.; Van Kester, J.A.T.M.; Stelling, G.S. (2004): Development and validation of a three-dimensional morphological model. *Coastal engineering*, *51*(8), 883-915.

Levenberg, K. (1944): A method for the solution of certain non-linear problems in least squares. Quarterly of applied mathematics, 2(2), 164-168.

Lim, S.Y. (1993): Clear water scour in long contractions. *Proceedings of the Institution of Civil Engineers-Water Maritime and Energy*, *101*(2), 93-98.

Lim, S.Y.; Cheng, N.S. (1998): Scouring in long contractions. *Journal of irrigation and drainage engineering*, *124*(5), 258-261.

Long, W.; Kirby, J.T.; Shao, Z. (2008): A numerical scheme for morphological bed level calculations. *Coastal Engineering*, *55*(2), 167-180.

Marquardt, D. W. (1963): An algorithm for least-squares estimation of nonlinear parameters. Journal of the society for Industrial and Applied Mathematics, 11(2), 431-441.

Marques de Sá J.P. (2001): Pattern Recognition. Concepts, Methods and Applications. Berlin, Heidelberg, New York: Springer-Verlag.

Meyer-Peter, E.; Müller, R. (1948): Formulas for bed-load transport. In IAHSR 2nd meeting, Stockholm, appendix 2. IAHR.

Mohammad, J.R.; Al-Saffar, M.S.A. (2010): Estimation of scour in long contractions using artificial neural networks. *J Duhok Univ*, *13*(1), 102-108.

Nagy, H.M.; Watanabe, K.A.N.D.; Hirano, M. (2002): Prediction of sediment load concentration in rivers using artificial neural network model. *Journal of Hydraulic Engineering*, *128*(6), 588-595.

Neural Network ToolboxTM, Matlab®R2013b, Mathworks, (2013).

Nicolle, A.; Karpytchev, M.; Benoit, M. (2009): Amplification of the storm surges in shallow waters of the Pertuis Charentais (Bay of Biscay, France). Ocean Dynamics, 59(6), 921.

Nicholson, J.; Broker, I.; Roelvink, J.A.; Price, D.; Tanguy, J.M.; Moreno, L. (1997): Intercomparison of coastal area morphodynamic models. *Coastal Engineering*, *31*(1-4), 97-123.

Ocampo-Duque, W.; Schuhmacher, M.; Domingo, J.L. (2007): A neural-fuzzy approach to classify the ecological status in surface waters. *Environmental Pollutio8n*, *148*(2), 634-641.

Papanicolaou, A.T.N.; Elhakeem, M.; Krallis, G.; Prakash, S.; Edinger, J. (2008): Sediment transport modeling review-current and future developments. *Journal of Hydraulic Engineering*, *134*(1), 1-14.

Pedrycz, W. (1993): *Fuzzy control and fuzzy systems (2nd.* Research Studies Press Ltd.

Penz, P. (2013): Contraction scour modelling using artificial neural networks [MSc. thesis]. Munich: Technische Universität München.

Pletcher, R.H.; Tannehill, J.C.; Anderson, D. (2012): *Computational fluid mechanics and heat transfer*. CRC Press.

Raghuwanshi, N.S.; Singh, R.; Reddy, L.S. (2006): Runoff and sediment yield modeling using artificial neural networks: Upper Siwane River, India. *Journal of Hydrologic Engineering*, *11*(1), 71-79.

Rajaee, T.; Mirbagheri, S.A.; Zounemat-Kermani, M.; Nourani, V. (2009): Daily suspended sediment concentration simulation using ANN and neuro-fuzzy models. *Science of the Total Environment*, *407*(17), 4916-4927.

Raudkivi, A.J. (1967): *Loose boundary hydraulics*, Pergamon, Oxford, U.K., 176–177.

Raudkivi, A.J.; Ettema, R. (1982): Stability of armor layers in rivers. *Journal of the Hydraulics Division*, *108*(9), 1047-1057.

Rezzolla, L. (2011): Numerical methods for the solution of partial differential equations. *Lecture Notes for the COMPSTAR School on Computational Astrophysics*, 8-13.

Richardson E.V.; Davis S.R. (2001): Evaluating scour at bridges. Hydraulic Engineering Circular No. 18. Washington, DC: Federal Highway Administration.

Roe, P. L. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. Journal of computational physics, 43(2), 357-372.

Ross, T.J. (2009): *Fuzzy logic with engineering applications.* John Wiley & Sons.

Salas, J.D. (1980): *Applied modeling of hydrologic time series.* Water Resources Publication.

Sivanandam, S.N.; Sumathi, S.; Deepa, S.N. (2007): *Introduction to fuzzy logic using MATLAB* (Vol. 1). Berlin: Springer.

Sleath, J.F.A. (1984): *Seabed mechanics*, Wiley, New York.

Shields, A. (1936): Anwendung der Aehnlichkeitsmechanik und der Turbulenzforschung auf die Geschiebebewegung. PhD Thesis Technical University Berlin.

Straub, L.G. (1934): Effect of channel-contraction works upon regimen of movable bed-streams. *Eos, Transactions American Geophysical Union*, *15*(2), 454-463.

Takagi, T.; Sugeno, M. (1985): Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems, man, and cybernetics*, (1), 116-132.

Tu, J. V. (1996): Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. Journal of clinical epidemiology, 49(11), 1225-1231.

Van der Wegen, M.; Roelvink, J.A. (2008): Long-term morphodynamic evolution of a tidal embayment using a two-dimensional, process-based model. *Journal of Geophysical Research: Oceans*, *113*(C3).

Van Rijn, L.C. (1993): Principles of Sediment Transport in Rivers, Estuaries and Coastal Seas, Aqua Publications.

Warren, I.R.; Bach, H. (1992): MIKE 21: a modelling system for estuaries, coastal waters and seas. *Environmental Software*, *7*(4), 229-240.

Warner, J.C.; Sherwood, C.R.; Signell, R.P.; Harris, C.K.; Arango, H.G. (2008): Development of a three-dimensional, regional, coupled wave, current, and sediment-transport model. *Computers & Geosciences*, *34*(10), 1284-1306.

Webby, M.G. (1984): General scour at contraction. *RRU bulletin*, *73*, 109-118.

Wilson, K. C. (1966): Bed-load transport at high shear stress. Journal of the hydraulics division, 92(6), 49-59.

Yang, C.T. (1996): *Sediment transport: Theory and practice*, McGraw Hill, New York.

Yen, C. L.; Lee, K. T. (1995): Bed topography and sediment sorting in channel bend with unsteady flow. Journal of Hydraulic Engineering, 121(8), 591-599.

Yu, H.; Wilamowski, B.M. (2011): Levenberg–Marquardt training. *Industrial Electronics Handbook*, *5*(12), 1.

Zhu, Y.M.; Lu, X.X.; Zhou, Y. (2007): Suspended sediment flux modeling with artificial neural network: An example of the Longchuanjiang River in the Upper Yangtze Catchment, China. *Geomorphology*, *84*(1), 111-125.

Zounemat-Kermani, M.; Teshnehlab, M. (2008): Using adaptive neuro-fuzzy inference system for hydrological time series prediction. *Applied Soft Computing*, *8*(2), 928-936.

List of Figures

List of Tables

Nomenclature

α	Coefficient adjusting height of humps
β	Coefficient adjusting width of humps
η	Learning constant
Γ_m	Higher order terms of Taylor series expansions
λ	Constant coefficient for ANFIS
λ_k	Eigen values of the Jacobian matrix
μ	Combination coefficient for ANN
μ	Dynamic viscosity
ω_0	Fall velocity
$\overline{Fr}$	Densimetric Froude number
ρ	Density of water
ρ_m	Fluid density
ρ_s	Density of sediment
ρ_s	Sediment density
σ_g	Geometric standard deviation
τ^*	Shields parameter
τ^{Re}	Reynolds stress tensor
τ_0	Bed shear stress
τ_c^*	Critical shields parameter

τ_{ij}	Three dimensional shear stress
θ	Threshold
υ	Kinematic viscosity of water
$\vec{\mathbf{R}}$	inhomogeneous term
A	Function of the mean flow velocity
a	Bed layer thickness
a	Premise parameter
b_1	Approaching channel width
b_2	Bias value
b_2	Contracted channel width
C	Bed celerity
C	Volumetric sediment concentration
c	Premise parameter
c	Solid concentration
$C(Z_b)$	The phase speed of bed form
C_a	Reference sediment concentration at a reference elevation a
CFL	Courant-Friedrichs-Levy number
d_m	Median sediment particle size
d_s	Equilibrium scour depth
d_s	Grain diameter
d_{50}	Mean grain size
E	Error function
e	Error vector
f	Output of ANFIS model

$F(w)$	Numerical flux
$f(x)$	Continuous function
f_i	Forces in i-direction
Fr^*	Grain Froude number
g	Gravity
H	Water depth
h	Water depth
h_1	Approaching flow depth
h_2	Flow depth in contracted depth
I	Sum of the weighted inputs
k	Iteration
k	Von Karman constant
L	Length of contraction
m	Constant parameter
N	Premise parameter
n	Number of pairs of data
O_1^5	The output of the ith node in the fifth layer of ANFIS
O_i^1	The output of the ith node in the first layer of ANFIS
O_i^2	The output of the ith node in the second layer of ANFIS
O_i^3	The output of the ith node in the third layer of ANFIS
O_i^4	The output of the ith node in the fourth layer of ANFIS
o_{pn}	Correct output values for pattern p
P	Pressure
p_0	Porosity

p_1, p_2	Constant parameters of first-order Sugeno function
q	Constant fluid volume flux per unit width
q_1, q_2	Constant parameters of first-order Sugeno function
q_b	Unit sediment discharge (bedload)
q_s	Unit sediment discharge (suspended load)
q_t	Unit sediment discharge (total load)
q_{b_*}	Dimensionless sediment discharge
r_1, r_2	Constant parameters of first-order Sugeno function
ra_{max}	Upper range
ra_{min}	Lower range
$Re*$	Grain Reynolds number
s	Relative density
t_{pn}	Target values for pattern p
U	Mean flow velocity
u^*	Friction velocity
U_x	Depth-averaged quantities of local velocity in x-direction
U_y	Depth-averaged quantities of local velocity in y-direction
V	Total volume
v_1	Approaching flow velocity
V_s	Solid volume
v_x	Velocity in x-direction
v_y	Velocity in y-direction
v_z	Velocity in z-direction
x	Input of ANFIS model

x	Original variable
X_A	Membership functionMembership function
x_i	ith predicted value
x_{max}	Maximum of variable x
x_{min}	Minimum of variable x
xpo	Post-processed variable
xpr	Preprocessed variable
y	Input of ANFIS model
y_i	ith measured value
z	Bed level above a fixed datum
z_0	Distance from the flat boundary at which the logarithmic velocity hypothetically equals zero
Z_b	Bed elevation
θ	Parameter vector of ANFIS
S	Set of total parameters (premise and consequent)
S_1	Set of premise parameters
S_2	Set of consequent parameters
$\mathbf{b}_1$	Bias vector
$\mathbf{g}$	Gradient (the first order derivative of total error function)
$\mathbf{H}$	Hessian matrix
$\mathbf{IW_{1,1}}$	Input weight matrix
$\mathbf{I}$	Identity matrix
$\mathbf{J}$	Jacobian matrix
$\mathbf{LW_{2,1}}$	Hidden layer weight matrix

$\mathbf{w}$	Weight matrix
$\mathbf{x}$	Input vector
$\mathbf{z}$	Input vector for ANFIS
b	Bias
Cs	Skewness coefficient
MAE	Mean absolute error
MAPE	Mean absolute percentage error
NDEI	Non-dimensional error index
Q_t	River discharge at time t
R	Correlation coefficient
R^2	Coefficient of determination
R1, R2, R3	Autocorrelations from 1 day lag to 3 day lag
RMSE	Root mean square error
Sd	Standard deviation
SSCt	Suspended sediment concentration at time t
SSL	Suspended sediment load