

2007 Special Issue

Recurrent neural network modeling of nearshore sandbar behavior

Leo Pape^{a,*}, B.G. Ruessink^a, Marco A. Wiering^b, Ian L. Turner^c

^a Department of Physical Geography, Faculty of Geosciences, IMAU, Utrecht University, P.O. Box 80.115, 3508 TC Utrecht, Netherlands

^b Intelligent Systems Group, Department of Information and Computing Sciences, Utrecht University, Utrecht, Netherlands

^c School of Civil and Environmental Engineering, Faculty of Engineering, University of New South Wales, Sydney, Australia

Abstract

The temporal evolution of nearshore sandbars (alongshore ridges of sand fringing coasts in water depths less than 10 m and of paramount importance for coastal safety) is commonly predicted using process-based models. These models are autoregressive and require offshore wave characteristics as input, properties that find their neural network equivalent in the NARX (Nonlinear AutoRegressive model with eXogenous input) architecture. Earlier literature results suggest that the evolution of sandbars depends nonlinearly on the wave forcing and that the sandbar position at a specific moment contains ‘memory’, that is, time-series of sandbar positions show dependencies spanning several days. Using observations of an outer sandbar collected daily for over seven years at the double-barred Surfers Paradise, Gold Coast, Australia several data-driven models are compared. Nonlinear and linear models as well as recurrent and nonrecurrent parameter estimation methods are applied to investigate the claims about nonlinear and long-term dependencies. We find a small performance increase for long-term predictions (>40 days) with nonlinear models, indicating that nonlinear effects expose themselves for larger prediction horizons, and no significant difference between nonrecurrent and recurrent methods meaning that the effects of dependencies spanning several days are of no importance.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Sandbar; Recurrent neural networks; Time-series modeling; ARX; NARX; Nonlinear

1. Introduction

The hydrodynamic and sediment transport processes that operate in the nearshore, defined as the innermost region of the coastal ocean with water depths less than 10 m, are highly nonlinear, causing a nontrivial response of the sediments on the sea bed. The most obvious example thereof is the organization of sandy bed material in spatially quasi-periodic patterns known as alongshore sandbars, 0.5–3 m high submarine ridges of sand located approximately parallel to shore and occurring singularly or in multiples of up to four or five bars. The cross-shore oriented dynamics of sandbars constitutes an important part of the morphological dynamics of the entire nearshore zone and comprises variability on timescales of a few days to years (e.g., Gallagher, Elgar, and Guza (1998), Plant, Holman, Freilich, and Birkemeier (1999), Ruessink, Wijnberg, Holman, Kuriyama, and Van Enckevort (2003)).

The desired ability to predict temporal variability in sandbar characteristics on all relevant timescales drives much of the present-day fundamental research in coastal processes and is also an important issue in coastal management. Sandbars are the last natural line of defence against the attack of storm waves on the coasts and, accordingly, human measures to combat coastal erosion often involve changes in sandbar height or position by means of sand nourishments (Hamm et al., 2002). The prediction of the evolution of sandbars is required to plan and evaluate the effectiveness of nourishments in mitigating coastal erosion.

Most existing nearshore bathymetric evolution models are based on knowledge of nearshore water motion (waves and currents) and sediment transport (Roelvink & Brøker, 1993). Existing knowledge is incomplete and, accordingly, process models struggle to reproduce natural bar behavior on timescales of a few days to weeks (Plant, Holland, & Puleo, 2004; Van Rijn et al., 2003) and have uncertain skill on longer scales (Roelvink, Meijer, Houwman, Bakker, & Spanhoff, 1995; Van Rijn et al., 2003). While most models are, to some extent, capable of predicting the rather rapid offshore

* Corresponding author. Tel.: +31 30 2532982; fax: +31 30 2531145.

E-mail addresses: l.pape@geo.uu.nl (L. Pape), g.ruessink@geo.uu.nl (B.G. Ruessink), marco@cs.uu.nl (M.A. Wiering), ian.turner@unsw.edu.au (I.L. Turner).

Notation

c_g	wave group velocity
ζ	breaker parameter
Θ	parameter vector of a process-based model
$\hat{\theta}$	parameter estimate
G	parameter update function
g	gravitational acceleration
H_{rms}	root-mean-squared wave height
H_b	breaker height
I	model input
M	autoregressive model order
\mathcal{M}	process-based model
N	exogenous model order
\mathcal{N}	neural network
p	statistical significance level
T_{peak}	peak wave period
t	time
ϕ	wave direction relative to the shore normal
U	model state, model output
\mathbf{W}	weight matrix of a neural network
\bar{X}	alongshore average observed sandbar position
z_{tide}	astronomical tide

sandbar migration under breaking waves (1–20 m day⁻¹), they fail to reproduce the subsequent recovery of a beach profile (e.g., onshore bar migration, <1–5 m day⁻¹) under relatively quiescent wave conditions, possibly because of missing relevant processes. Therefore, it appears appropriate to investigate the suitability of other model types, such as data-driven models, to predict sandbar behavior. In contrast to process models, data-driven models need no process knowledge to operate, but are constructed by establishing statistical relations in field data. Until the mid- to late 1990s such data were rare, but they are now becoming available through the high-resolution (hourly), long-term (years) monitoring of the nearshore zone with automated video systems (Aarninkhof & Holman, 1999).

Existing analyses of time-series of cross-shore sandbar position have suggested a nonlinear dependence of sandbar migration on various environmental factors, such as the height of the incident waves at breaking, see, for instance, Plant et al. (1999). Furthermore, ‘memory’ is sometimes suggested to be present in time-series of sandbar position, meaning dependencies in sandbar position spanning longer time periods which might even exceed the duration of individual storms. An example of modeling such temporal dependencies can be found in Wright, May, Short, and Green (1985), who use hydrodynamic forcings weighted over several days to provide for a measure of relaxation time in predicting the evolution of sandbar variability. Based on cross-shore sediment transport modeling, Plant, Ruessink, and Wijnberg (2001) also suggested that the response time of beach profiles is exceedingly long compared to the timescale of the variability of the offshore wave forcing. O’Hare and Huntley (2006) later on debated this suggestion, indicating that, while it may be true for

quiescent wave conditions, the morphological timescale under storm conditions might be of the same order as the nearshore hydrodynamics. This would imply that the storm response of a sandbar would not depend on previous wave forcings and sandbar characteristics but is determined by the storm characteristics solely.

The autoregressive nature of process models as well as their forcing with offshore wave information finds its equivalent in the NARX (Nonlinear AutoRegressive model with exogenous inputs) neural network architecture. In addition, this architecture is known to be capable of representing nonlinear dynamics as well as long-term dependencies in time-series data (Lin, Horne, Tñio, & Giles, 1996). The aim of the present paper is to identify to what extent these claims of nonlinear dynamics and long-term dependency apply to time-series of sandbar position by constructing a NARX neural network model and by comparing its performance to other less sophisticated data-driven models. The applied data are observations of an outer sandbar collected daily for over seven years at Surfers Paradise, the Gold Coast, Australia, and concurrent measurements of offshore wave parameters (height, period, direction) and water level.

2. Observations

The sandbar data set used in the present work is acquired with an Argus coastal imaging station located at the double-barred Surfers Paradise, northern Gold Coast, Queensland, Australia (Turner, Aarninkhof, Dronkers, & McGrath, 2004). The station consists of four cameras pointed obliquely along the beach, providing 180° uninterrupted coverage of the beach and the nearshore zone. Each daylight hour, the cameras acquire a time-exposure image (Fig. 1(a)), created by averaging over 1200 consecutive images collected at 2 Hz. This smooths out moving objects such as ships, vehicles and people, and averages the individual breaking waves to reveal one or more smooth white bands of breaking waves. These bands serve as a reasonable estimate for the submerged sandbars (Lippmann & Holman, 1989; Van Enckevort & Ruessink, 2001). The four oblique images are rectified (Holland, Holman, Lippmann, Stanley, & Plant, 1997) and merged to yield a single plan-view image (e.g., Fig. 1(b)). As detailed in Van Enckevort and Ruessink (2001), the crest lines of the inner and outer bar are extracted from these plan-view images by the automated alongshore tracking of the intensity maxima across each bar (Fig. 1(c)). The alongshore average of a crest line, in the present work based on plan-view images with an alongshore extent of 3000 m, is referred to in the following as the barcrest position \bar{X} . The actual (in situ) bar position, however, is known to deviate from the position of the breaking waves by a factor of O(10) m, and varies in time and alongshore distance with the wave height, the water level, and the bathymetry (Van Enckevort & Ruessink, 2001).

To eliminate the time-varying nature of the difference between the measured and the actual bar position caused by several factors, the data set is reduced to a single observation each day, at the lowest tide of that day (when the breaking patterns are most pronounced in the images). Occasionally, the

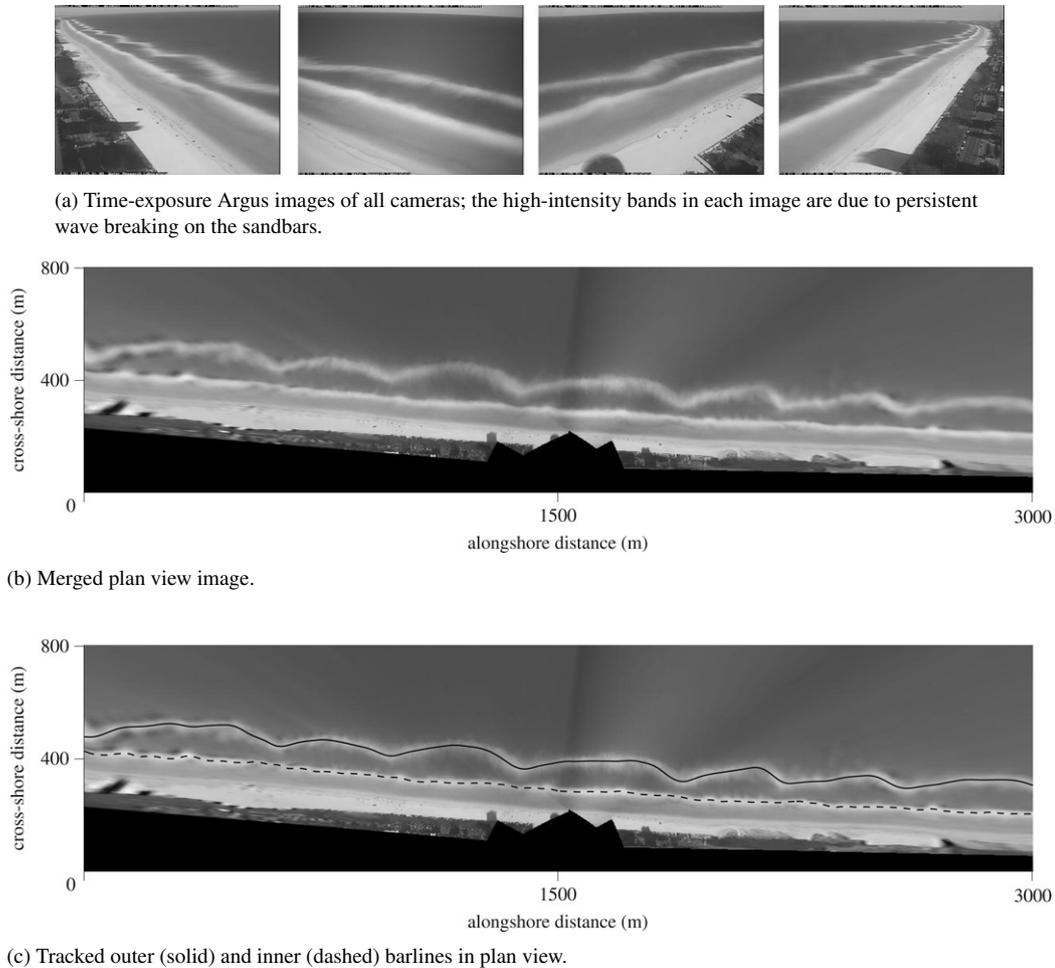


Fig. 1. Argus camera images, merged plan view and tracked barlines.

bar position cannot be computed due to poor image quality (fog or rain droplets on one of the camera lenses), conditions when waves were too low to break, or the malfunctioning of the video acquisition system. To create a continuous time-series data set, the gaps are filled with observations from the last breaking-based observation, assuming that bar migration is insignificant under low-energy conditions.

Image data at the Gold Coast station is available for over seven years starting at July 15, 1999. During this period, five different sandbars can be distinguished (Fig. 2(a)). The life cycle of a bar at the Gold Coast begins close to the shore as an inner sandbar. When the existing outer bar disappears, the inner bar moves seaward and becomes the new outer bar. After one or more major storms (high wave energy conditions in Fig. 2(b)) the outer bar moves further offshore than usual (e.g., early 2006 in Fig. 2; see also *Castelle, Turner, Ruessink, and Tomlinson (2007)*), after which it disappears, and a new cycle begins. As the outer bar decays, the beach is particularly prone to erosion (*Castelle et al., 2007*), highlighting the relevance of sandbars as a natural defence mechanism.

The predominant exogenous inputs for driving sandbar variability in process-based models are the offshore waves, which can be represented by their root-mean-squared wave height H_{rms} , peak wave period T_{peak} and wave direction relative

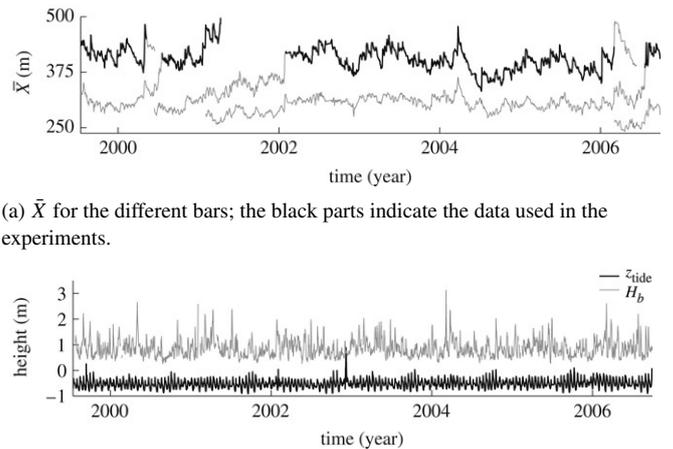


Fig. 2. Overview of the variables in the data set.

to the shore normal ϕ . To reduce the number of inputs for the models, these variables are combined into the wave height at breaking (*Plant et al., 1999*):

$$H_b = \left(\frac{\gamma}{g} \right)^{\frac{1}{5}} \left(H_{\text{rms}}^2 \cdot c_g \cdot \cos(\phi) \right)^{\frac{2}{5}}. \quad (1)$$

The offshore group velocity c_g is computed using linear wave theory involving T_{peak} and the water depth at the location of the offshore wave measurement. The gravitational acceleration g is 9.81 m s^{-2} and the breaker parameter γ is set to 0.4, a typical field value (Thornton & Guza, 1983). The variables H_{rms} and T_{peak} are obtained every 30 minutes from the Gold Coast waverider buoy, located approximately 2 km offshore of the study area, in 16 m water depth. Directional information ϕ , is collected hourly by the Brisbane waverider buoy located some 10 km offshore in 70 m water depth, about 100 km north of the study area. The H_b values are computed from the observed values of H_{rms} , T_{peak} and ϕ , and are then averaged to yield a single value between two subsequent sandbar observations. The time-averaged H_b value is given the same time index as the second observation. Missing H_b values are filled by linear interpolation between adjacent points in time of the 7-year data set. The time-series of the hydrodynamic data are depicted in Fig. 2(b).

As shown by Van Enckevort and Ruessink (2001), the magnitude of the difference between the video-observed and the actual sandbar position depends on the tidal water level z_{tide} . Because our observed sandbar positions are collected at different tidal levels (Fig. 2(b)), this creates artificial sandbar migration, that is, migration due to varying tidal levels rather than due to actual sandbar migration. The artificial day-to-day migration rate depends linearly on the water-level difference between the two images (e.g., Van Enckevort and Ruessink (2001)), and can therefore be removed easily from the data by projecting each observed sandbar position on a fixed water-level. Here, the average low-tide value (approximately 0.5 m below mean sea level) is used. This approach is similar to that described in Alexander and Holman (2004).

3. Models

3.1. Autoregressive models

The presently most common approach to the modeling of nearshore bathymetry is by means of process-based models (Roelvink & Brøker, 1993). A formal description of these models can be given in terms of inputs, outputs, model states, functions and parameters. In discrete time, the internal state of the model at a certain time t , called the system state $\hat{U}(t)$ depends on the states of the model in the past and the external forcing $I(t)$ to the model at time t :

$$\hat{U}(t) = \mathcal{M}(\hat{U}(t-1), \dots, \hat{U}(t-M); I(t), \dots, I(t-N)|\vec{\Theta}), \quad (2)$$

where \mathcal{M} represents the model's process knowledge, $\vec{\Theta}$ the adjustable model parameters, M the autoregressive order of the model state, and N the order of the external forcings. Because process-based models are based on first-order time differential equations, the autoregressive and exogenous terms always span one time step ($M = 1$ and $N = 0$, given the definition in Eq. (2)). The parameters $\vec{\Theta}$ must be determined by calibration from observed model behavior or the literature.

The general name for linear data-driven models in which previous model states are fed back into the model together with

additional inputs, is AutoRegressive models with eXogenous inputs (ARX). The parameters of the ARX model can be determined by an algorithm that finds the values for which the sum of the squared differences between all samples in the data and the model output is minimized. It must be noted that other methods exist in which several assumptions about the nature of the data are relaxed (Draper & Smith, 1998), which will not be dealt with here. Another way in which the parameters can be determined is to apply a recursive algorithm in which parameter updates are computed iteratively for each sample. In the general case the parameter update rule can be written as:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + G(t)(U(t) - \hat{U}(t)), \quad (3)$$

where $\hat{\theta}(t)$ is the parameter estimate at time t , $U(t)$ is the observed value at t and $\hat{U}(t)$ is the model output at time t , which might have been influenced by different model parameters at earlier time steps. $G(t)$ is the factor that determines how the difference between the model output and the observed data affects the parameter update, and is often based on the gradient of $\hat{U}(t)$ with respect to $\hat{\theta}(t)$. The model is adjusted each time step and can adapt to variations in the properties of the observed data over time, making it suitable for online model estimation. Furthermore, the recursive parameter estimation method for ARX models is very similar to other algorithms that iteratively compute parameter updates (such as neural network training algorithms), and can often be written in terms of such models.

The ARX model is based on establishing linear relations between the different variables. In case the transfer functions of the model are nonlinear such a model is called a Nonlinear AutoRegressive model with eXogenous input (NARX). Although a linear ARX model might be a good starting point to study several features of observed data, it is suspected that because of the nonlinear nature of the nearshore dynamics only nonlinear models will be able to represent the characteristics of such a system to their full extent.

While ARX models are fairly easy to create, the construction of nonlinear models is less straightforward. As described by Narendra and Parthasarathy (1990) various methods exist to build nonlinear data-driven models. One method that has become very popular in recent years is the use of neural networks. Whereas nonrecurrent neural networks can be used to model nonlinear relations, recurrent neural networks are also able to model temporal dependencies in the data (Rumelhart, Hinton, & Williams, 1986). Because of the supposed nonlinearity and temporal dependencies in the sandbar system, a recurrent neural network known as the NARX neural network will be used here. The NARX neural network is equivalent to autoregressive process-based models by replacing the process knowledge \mathcal{M} in Eq. (2) by a neural network \mathcal{N} and the adjustable model parameters $\vec{\Theta}$ by the weight matrix \mathbf{W} of the neural network:

$$\hat{U}(t) = \mathcal{N}(\hat{U}(t-1), \dots, \hat{U}(t-M); I(t), \dots, I(t-N)|\mathbf{W}). \quad (4)$$

The transfer function \mathcal{N} is now a neural network and the weight matrix \mathbf{W} is to be determined by learning from the data, as is

the case for data-driven models. The autoregressive terms in Eq. (2) are represented by recurrent connections (connections going back from the output to the input layer), which can now be employed by a recurrent training algorithm to learn temporal dependencies in the data.

3.2. Parameter estimation for ARX and NARX models

While backpropagation (Rumelhart et al., 1986) is a commonly used technique for training feedforward neural networks such as the MultiLayer Perceptron (MLP), several interpretations exist for the application of this method in recurrent neural networks. RealTime Recurrent Learning (RTRL) and Back-Propagation Through Time (BPTT) (Rumelhart et al., 1986; Williams & Peng, 1990) are well-known training algorithms. Whereas RTRL is of more theoretical interest, BPTT is the algorithm that is used to train the NARX neural network architecture (Lin et al., 1996). When BPTT is applied for training a NARX neural network, the recurrent connections are unfolded in time and the resulting network is treated as an MLP with injected errors. Instead of unfolding the temporal operation of a network into an MLP that grows by one layer each step, as is common practice in BPTT (Rumelhart et al., 1986), in the NARX architecture the entire network is unfolded at the recurrent connections. In the unfolded network, the recurrent connections appear as jump-ahead connections, providing for a shorter path for backpropagating the error through the network, thus diminishing the problem of the vanishing gradient (Lin et al., 1996). After presenting the data to the network, the error is backpropagated through the unfolded network. In the output units of the recurrent network, the local error is computed and added to the backpropagated value from the subsequent input unit. As the error in the present time step is reduced while taking into account the errors made in the future, the NARX neural network is able to learn long-term dependencies in the data.

Because in its basic form the BPTT algorithm is very computationally intensive, the number of time steps h over which the recurrent connections are unfolded (or the number of network states that is saved) can be reduced, which is called truncated BPTT. Furthermore, the algorithm can compute weight updates for each step, but the updates can also be delayed an additional number of steps h' . The application of both changes to the BPTT algorithm will reduce computation time, while still following the true gradient closely (Williams & Peng, 1990). This efficient algorithm called BPTT ($h; h'$) will be used in the present work and is exactly the same as batch-wise BPTT if h is chosen to equal h' . Since the time-series data of alongshore average positions of the sandbars are not naturally segmented into independent batches, the batch size h (which is now equal to h') must be carefully chosen. If the batch size is too small, any temporal dependencies that span a longer period than the selected batch size cannot be learned, and if it is too large the algorithm converges slowly.

In a NARX neural network it is possible to disregard the long-term dependencies by not backpropagating the error over the recurrent connections. In such a network the errors made in the future are not considered in the computation of

the current weight updates, meaning the network will not be able to learn any long-term dependencies. This nonrecurrent training algorithm for a recurrent neural network can be seen as an autoregressive MLP that still uses the previously predicted outputs as input, but does not backpropagate any error information over the autoregressive terms. The importance of long-term dependencies in the data can be studied empirically by comparing the results of an autoregressive MLP that is trained with a nonrecurrent method and a NARX neural network that is trained using the BPTT algorithm.

As discussed in Section 3.1, the parameters of an ARX model can be established by least-squares linear regression, but also with recursive methods. The recursive method has much in common with the backpropagation training method for neural networks, but can also be implemented as a recurrent training algorithm that is equivalent to BPTT. When the NARX neural network employs only one layer with units that have linear transfer functions instead of one or more layers of units with nonlinear functions, the network is reduced to a linear ARX model, while the BPTT algorithm can still be used to compute iterative weight updates. The equation for this ARX model becomes:

$$\hat{U}(t) = \theta_0 + \sum_{i=1}^M \theta_i \hat{U}(t-i) + \sum_{j=0}^N \theta_{(1+j+M)} I(t-j), \quad (5)$$

where $\theta_1 \dots \theta_{(M+N+1)}$ are the model parameters which correspond to the connections between the input and output layer, and are determined by the recurrent learning algorithm that is now based on the gradients of the parameters with respect to model outputs over multiple time steps. Note that in this definition the ARX network also includes a bias term θ_0 that is not present in the original ARX formulation. Using BPTT for estimating ARX models enables such a model to learn temporal dependencies, and allows for a fair comparison between other linear and nonlinear ARX models.

In summary, four different data-driven models will be used to test the claim of nonlinear dynamics and long-term dependencies in time-series of sandbar positions from an empirical point of view: an ARX model solved with nonrecurrent parameter estimation to test for linearity in absence of temporal dependencies, an ARX model solved with BPTT representing a linear method that can learn temporal dependencies, an autoregressive MLP that can model nonlinearities, but no temporal dependencies and a NARX neural network trained with BPTT that can model both nonlinear and long-term dependencies. Because of its computational efficiency, the ARX model solved with least squares is used to determine optimum values for the autoregressive model and exogenous input order (see Section 4.3 below), assuming that these optimum values also apply for the other model types.

4. Experiments

4.1. Setup

For the purpose of the present work, we focus on the cross-shore behavior of the outer bar. As described in Section 2,

the outer bar occasionally decays following an extreme storm event, after which the inner bar migrates offshore to become the new outer bar. In a time-exposure image a decaying bar shows up as a series of discontinuous, vague white blobs from which an accurate alongshore average position cannot be estimated. Furthermore, [Castelle et al. \(2007\)](#) noted that the behavior of a decaying bar (e.g., its response to subsequent minor storms) is different from a well-developed bar. Therefore, outer bar data starting at the major storm after which the bar started to decay are discarded. The offshore migrating inner bar is considered to be a new outer bar, when it migrates beyond 350 m from a local baseline (the approximately most shoreward position of a well-developed outer bar, see [Fig. 2\(a\)](#)). Given these considerations and the availability and quality of the image data set, a total of 2131 days of alongshore average outer bar positions is selected, as indicated by the black lines in [Fig. 2\(a\)](#). This data set, comprising the tide-corrected positions and associated wave breaker heights with a resolution of one day, is scaled to zero mean and unit variance for the experiments with the data-driven models.

All four models described in the previous section use the same input and output data structure. The alongshore average bar position \bar{X} is to be predicted at time t :

$$U(t) = \bar{X}(t). \quad (6)$$

The input to a model consists of the model's output during the past M time steps, together with the forcing factors at present and the previous $N - 1$ time steps, which are the wave forcings represented by H_b :

$$I(t), \dots, I(t - N) = H_b(t), \dots, H_b(t - N). \quad (7)$$

Note that $H_b(t)$ is defined here as the mean wave forcing corresponding to the time between observations of $\bar{X}(t - 1)$ and $\bar{X}(t)$.

During the experiments the data set is divided into 20 sets of approximately 100 days each. The performance of the four models is tested on each of the 20 sets, while one randomly permuted set is used as validation set, and the remaining 18 sets are used as training data. The performance of a model is tested by running the model starting with observed conditions at the first time step and model outputs during the next 10 time steps. This process is repeated each 10 samples where the observed values preceding those samples are used as initial values. The root-mean-squared difference between 10-step-ahead model predictions and observed values, averaged over the 20 sets is taken as the performance measure in all experiments, unless noted otherwise. The choice for 10-step-ahead predictions is motivated by the observation in the data that on this time span notable changes in bar position take place during both high (storm) and relatively low H_b values. The sensitivity of our results to this particular choice is discussed in [Section 4.5](#).

Both the recurrent and nonrecurrent training algorithms update the parameter estimates with small steps each iteration, and the data need to be presented to the network several times (epochs) before the algorithm converges. Whereas establishing a model with least-squares linear regression is straightforward,

a stopping criterion has to be established for the neural network training algorithms. The basic idea of the stopping criterion used here is that training is halted when the performance on the validation set stops to increase. However, when training recurrent neural networks the performance on the validation set (and sometimes even on the training set) does not decrease steadily, but might increase and decrease again several times during training. Therefore the training process is continued for a large number of epochs, during which the performance on test and validation sets is constantly monitored. The performance on the test set that corresponds to the best performance on the validation set is taken as the outcome of an experiment.

The nonrecurrent backpropagation and BPTT algorithms as well as the neural networks have a number of parameters that need to be adjusted to the task at hand. Since training recurrent neural networks takes a considerable amount of time, several experiments were performed to find reasonable settings for the network architecture and training algorithm. We found that a network with one hidden layer containing four sigmoid units yields a reasonable balance between learning speed and performance. Furthermore, we used a learning rate of 10^{-6} , a momentum factor of 0.2 and performed the training process over 10^6 epochs. Optimum values for the batch size and for the autoregressive model and exogenous input order are determined in the next two subsections. While nonrecurrent neural networks might not need to be trained for such a large number of epochs, the fact that the best performance during the entire training process is used, ensures that comparable results are achieved.

4.2. Batch size

The batch size in the BPTT($h; h'$) algorithm needs to be carefully chosen because it determines the length of the temporal dependencies that can be learned. The computation time for the BPTT algorithm decreases as the batch size increases, but this comes at the expense of a reduction in the learning speed; the optimum batch size has, therefore, to be determined empirically. A number of experiments was performed to investigate the effect of batch size on the performance and learning speed of the NARX neural network. Training and testing were performed as described in [Section 4.1](#), with $M = 1$ and $N = 0$. [Fig. 3](#) shows the course of the error for several batch sizes during the first 0.5×10^6 epochs. Because for batch sizes larger than approximately 20 the size does not affect the performance or learning speed anymore, a batch size of 25 is used in all further experiments.

As becomes clear from [Fig. 3](#), training a network takes a large number of epochs before it converges. Because [Fig. 3](#) shows only average performances over 20 networks while individual networks might need even more epochs for convergence, all networks in further experiments are trained for 10^6 epochs. While other learning methods using second or higher order derivatives or approximates, such as quasi-Newton methods, can improve learning speed, they also decrease the learning capacity of recurrent architectures because of fast-vanishing gradient information over recurrent connections ([Lin](#)

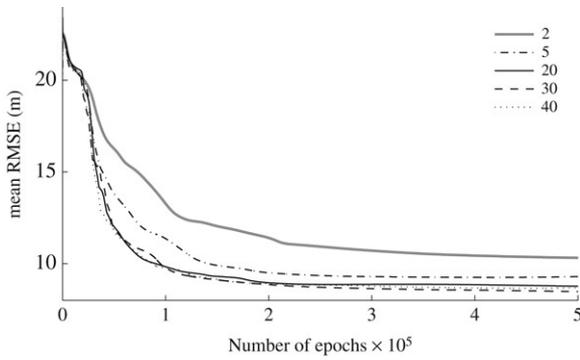


Fig. 3. Error during training on all test sets as a function of batch size.

et al., 1996). Because the ability of the NARX neural network to learn long-term dependencies is studied, we only use gradient descent based on the first-order derivative.

4.3. Model orders

Whereas process-based models always use one autoregressive term ($M = 1$) and one exogenous term ($N = 0$), data-driven models might benefit from additional information on past observations and model outputs. For most data-driven models performance will improve with increasing model orders, but this comes at the cost of additional parameters and learning time. Several information criteria exist (e.g., AIC and MDL) to determine optimal model orders, however to allow for a fair comparison between the four different models, the orders of all models are set to the same values.

Because of its computational efficiency the least-squares solved ARX model can be used easily to test a large number of different model orders. First the effect of changing the autoregressive order is tested with $M = 1 \dots 10$. Each of the 20 sets is used once as test set, while a model is fitted to the remaining 18 training sets in the same fashion as is done in the neural network experiments (no validation set is used). We found that the root-mean-squared error (RMSE) averaged over all sets for 10-step-ahead predictions yields no significant performance difference for each additional autoregressive order. Therefore an autoregressive order of unity is used in all experiments.

To find optimal values for the exogenous model order N , values from zero up to four were tested using a least-squares linear regression ARX model in the same fashion. As shown in Fig. 4, the largest performance increase for ARX models takes place with an increase from $N = 0$ to $N = 1$. To investigate whether this also holds for the neural network models, the experiments were repeated with a NARX neural network that is trained with BPTT. As becomes clear from Fig. 4, in the BPTT NARX model the error also shows the largest decrease when the exogenous order is increased from $N = 0$ to $N = 1$. Moreover, in this model the error increases for exogenous orders larger than 1. Because the best performance on the validation set in the NARX model is always achieved within 10^6 epochs, a possible reason for decreasing performance for $N > 1$ might be the relatively small number (four) of hidden units. However, in an additional experiment with 10 hidden

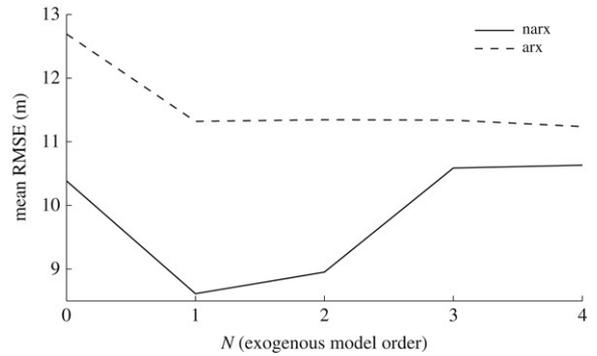


Fig. 4. Error on all test sets for different models as a function of the exogenous model order.

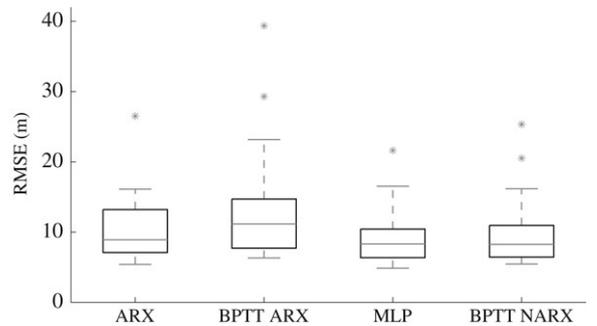


Fig. 5. Boxplots of the error distribution for 10-step-ahead predictions for different models, the asterisks represent outliers.

units, the performance also reduced for $N > 1$. It might be that NARX neural networks with even larger numbers of hidden units benefit from exogenous model orders larger than 1, but training such networks would require considerable extra computational effort. Because both models tested here benefit most from increasing the exogenous order from zero to one, the latter value is used in all further experiments.

4.4. Recurrent versus nonrecurrent and linear versus nonlinear models

As discussed in Section 3.2 the parameters of linear and nonlinear methods can be estimated using BPTT, allowing the model to learn temporal dependencies, or by nonrecurrent training methods that do not employ any information on temporal dependencies. Now that all parameter settings for the neural networks are established, the performance of the four different models can be tested on the sandbar data set. The models are trained as described in Section 4.1, and the performance is tested using the RMSE on 10-step-ahead predictions. As can be inferred from Fig. 5 there is a large variability between the results of the models on different parts of the data. Further investigation reveals that the outliers, with values that can be three times as large as their corresponding mean, are found in the same parts of the data in all experiments. These anomalies are caused by different statistical properties of the involved parts resulting in bad generalization capacities for the neural networks. A possible way to circumvent the problem of abnormalities in the data, might be to leave out the parts that deviate most from the rest of the data. Although there is

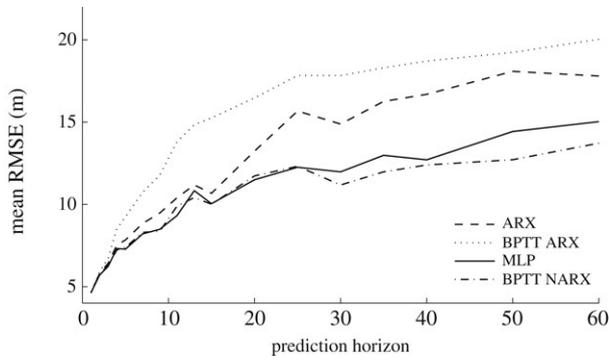


Fig. 6. Error on all test sets for different models as a function of the prediction horizon.

no significant ($p = 0.05$) difference in performance between any of the models, removing the largest outlier reveals a significant better performance of nonlinear over linear models, but still no significant difference is found between BPTT and the nonrecurrent parameter update rule. Apparently, long-term dependencies are not relevant to the 10-step-ahead predictions of the outer sandbar position.

4.5. Prediction horizon

Whereas the performance of the four models in the previous experiments was tested on 10-step-ahead predictions, it is suspected that the differences between the models become more pronounced for larger prediction horizons. Therefore the performance of the models is tested on several step-ahead predictions from 1 up to 60 (which is the size of the smallest continuous part of the data). During the training process the performance on the test and validation sets is monitored for each prediction horizon, and the performance on the test set that corresponds to the minimum error on the validation set is used for each prediction horizon separately. Fig. 6 shows the average performance over all sets (including the outliers) for the four models as a function of the prediction horizons. As suspected, the performance decreases with increasing prediction horizon and the differences between the four models grow likewise.

Whereas the differences between the autoregressive MLP and the BPTT NARX remain small for prediction horizons up to 30, larger values show that the performance of the NARX model is consistently better than that of the MLP. Due to the large variability in the results, this difference remains insignificant. The performance differences between other models are also very small for short prediction horizons, but for prediction horizons larger than 40 steps the difference between nonlinear and linear methods becomes significant ($p = 0.05$). This indicates that modeling the cross-shore outer bar migration benefits from the use of nonlinear methods, at least for long-term predictions.

Surprisingly, the BPTT ARX model performs worse than its nonrecurrent counterpart. The BPTT ARX model is trained to minimize the errors over 25 steps, but even on the 25-step-ahead prediction its performance is worse than the nonrecurrent ARX model. In both models the networks are trained for 10^6 epochs. While the BPTT ARX model slowly converges during its

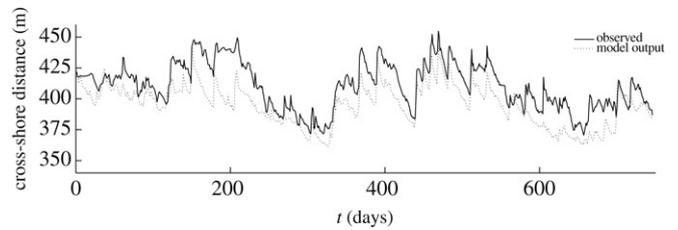


Fig. 7. Observed (solid) bar position versus model output (dotted) over a long time period (>2 years).

training for 10^6 epochs to a (local) optimum, the nonrecurrent ARX model converges much faster ($<10^5$ epochs), and then starts to oscillate (due to the momentum factor) around an optimum or between several optima. The performance on the test set that corresponds to the best performance on the validation set during the entire training period is taken as the outcome of an experiment. Therefore the chance that the BPTT training algorithm finds a better (local) optimum is much lower than in the nonrecurrent ARX, which might cause the latter to perform better.

Finally, we investigated the performance of a BPTT NARX model for even longer prediction horizons up to approximately 2 years. The result of a model that was tested on 744 days with only the first bar position provided to the model (744-step-ahead prediction), while the rest of the data was used as a training set is given in Fig. 7. As demonstrated in this figure, the model output reasonably follows the observed rapid offshore migration (during storms) and the subsequent slower onshore migration as well as the more seasonal variability in sandbar position, although, admittedly, details are occasionally predicted poorly (e.g., $t = 150$ – 200 days). It can be inferred from this experiment that the NARX model is capable of learning the general dynamics of the dependence of the alongshore average bar position on the hydrodynamic forcing. Also, the predicted positions of the models never show any values beyond the outer bar zone (between 350 and 500 m) while this was not imposed by the linear nature of the output units.

5. Discussion and conclusions

In the previous section nonlinear and linear models as well as recurrent and nonrecurrent parameter estimation methods were applied to investigate the supposed nonlinear and long-term dependencies in the data set of observed outer bar positions and hydrodynamic data. We found a small but statistically significant performance increase for long-term predictions with nonlinear models for prediction horizons above 40 days, indicating that nonlinear effects expose themselves for larger prediction horizons, and no significant difference between nonrecurrent and recurrent methods meaning that the effects of dependencies spanning more than 2 days are of no importance.

Fig. 5 shows that the behavior of the sandbars in terms of linear and nonlinear dependence on the hydrodynamic forcings exhibits large differences between separate parts of the data set. For the linear case this result was also found by Pape, Ruessink, Wiering, and Turner (2006) who attributed it to the rather small

spatial (1.8 km) and temporal (3.5 years) scales of observed sandbar positions in their Gold Coast data set. Fig. 5, based on larger scales (3 km and 6 years, respectively), implies that the suggestion of Pape et al. (2006) is incorrect. The reason for the different performances of the same model type on the various data sets is not known. It may indicate that the characterization of sandbar morphology by position alone is insufficient to predict the evolution of future positions. Potentially, sandbar volume and water depth above the bar crest, both of which cannot be determined from time-exposure video images in a straightforward manner, play a role as well. Van Enckevort et al. (2004), for instance, suggested that the response time of the alongshore crescentic patterns in sandbars (e.g., see Fig. 2) to similar storms at four sites depended on bar volume, with larger bars behaving more slowly. The water depth above the bar crest determines the relative importance of onshore versus offshore sediment transport. Thus, a bar located at the same cross-shore position and subjected to the same wave conditions may behave differently based on the water depth above the bar. If the bar volume and water depth above the bar indeed varied substantially during and between the data subsets, then this might have invoked the different performances of the same model on the data subsets.

Although the hydrodynamical and sediment transport processes in the near-shore are strongly nonlinear, the difference in performance of linear and nonlinear models is significant only for prediction horizons above 40 days, but even then RMSE differences between the nonlinear and linear methods remain relatively small. Intriguingly, other nearshore studies have also indicated a relatively small improvement in performance of nonlinear models over linear models (e.g., Ruessink, van Enckevort, and Kuriyama (2004)) or found reasonably accurate data-driven predictions of sandbar behavior during and following a single storm using linear models (Plant, Holland, & Holman, 2006). We are currently investigating how nonlinearities in the processes that govern sandbar behavior express themselves in time series of sandbar positions. Work of Laio, Porporato, Ridolfi, and Tamea (2004) on rainfall-runoff time series, for instance, demonstrates that the presence of nonlinear processes in a system does not necessarily imply that time series from specific variables in this system are nonlinear themselves.

As was to be expected from the use of first-order gradient descent methods, recurrent neural networks require a considerable amount of computational effort (10^6 epochs) for convergence. Other methods using second or higher order derivatives or approximates, can dramatically improve the learning speed, but also decrease the learning capacity of recurrent architectures. Feedforward networks suffer less from the problem of vanishing gradients because they do not employ recurrent connections. Our conclusion from Fig. 6 that the effect of memory on the cross-shore migration of the outer Gold Coast bar is negligible may advocate the use of MLPs for our future work on sandbar behavior. However, whether this conclusion also holds for the cross-shore migration of the inner bar (which is sheltered from the offshore waves by wave breaking on the outer bar and thus may respond differently to

offshore wave forcing than the outer bar (Plant et al., 2001; Southgate & Möller, 2000)) and for the evolution of the above-mentioned crescentic patterns remains to be investigated.

Acknowledgments

This work was supported by the Netherlands Organization for Scientific Research (NWO) under contract 864.04.007.

All wave and water level data were kindly made available by Doug Anderson (University of New South Wales) and the Environmental Protection Agency in Queensland. The Gold Coast City Council is acknowledged for ongoing funding support to Ian Turner that enables the continued operation of the Argus video station at the northern Gold Coast.

References

- Aarninkhof, S. G. J., & Holman, R. A. (1999). Monitoring the nearshore with video. *Backscatter*, 10, 8–11.
- Alexander, P. S., & Holman, R. A. (2004). Quantification of nearshore morphology based on video imaging. *Marine Geology*, 208, 101–111.
- Castelle, B., Turner, I. L., Ruessink, B. G., & Tomlinson, R. B. (2007). Impact of storms on beach erosion: Broadbeach Gold Coast, Australia. In *Paper accepted for presentation at the 2007 international coastal symposium*.
- Draper, N. R., & Smith, H. (1998). *Series in probability and statistics, Applied regression analysis* (third ed.). New York: Wiley.
- Gallagher, E. L., Elgar, S., & Guza, R. T. (1998). Observations of sand bar evolution on a natural beach. *Journal of Geophysical Research*, 103, 3203–3215.
- Hamm, L., Capobianco, M., Dette, H. H., Lechuga, A., Spanhoff, R., & Stive, M. J. F. (2002). A summary of European experience with shore nourishment. *Coastal Engineering*, 47, 237–264.
- Holland, K. T., Holman, R. A., Lippmann, T. C., Stanley, J., & Plant, N. G. (1997). Practical use of video imagery in nearshore oceanographic field studies. *Journal of Oceanic Engineering*, 22, 81–92.
- Laio, F., Porporato, A., Ridolfi, L., & Tamea, S. (2004). Detecting nonlinearity in time series driven by non-Gaussian noise: The case of river flows. *Nonlinear Processes in Geophysics*, 11, 463–470.
- Lin, T., Horne, B. G., Tiño, P., & Giles, C. L. (1996). Learning long-term dependencies in NARX recurrent neural networks. *IEEE Transactions on Neural Networks*, 7(6), 1329–1338.
- Lippmann, T. C., & Holman, R. A. (1989). Quantification of sand bar morphology: A video technique based on wave dissipation. *Journal of Geophysical Research*, 94(C1), 995–1011.
- Narendra, K. S., & Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1, 4–27.
- O'Hare, T. J., & Huntley, D. A. (2006). Comment on “morphologic properties derived from a simple cross-shore sediment transport model” N.G. Plant, B.G. Ruessink, K.M. Wijnberg. *Journal of Geophysical Research*, 111, C07003. doi:10.1029/2004JC002808.
- Pape, L., Ruessink, B. G., Wiering, M. A., & Turner, I. L. (2006). Neural network modeling of nearshore sandbar behavior. In *Proceedings of the 2006 international joint conference on neural networks* (pp. 8735–8742).
- Plant, N. G., Holland, K. T., & Holman, R. A. (2006). A dynamical attractor governs beach response to storms. *Geophysical Research Letters*, 33, L17607. doi:10.1029/2006GL027105.
- Plant, N. G., Holland, K. T., & Puleo, J. A. (2004). Prediction skill of nearshore profile evolution models. *Journal of Geophysical Research*, 109. doi:10.1029/2003JC001995.
- Plant, N. G., Holman, R. A., Freilich, M. H., & Birkemeier, W. A. (1999). A simple model for interannual sandbar behavior. *Journal of Geophysical Research*, 104(15), 755–776.

- Plant, N. G., Ruessink, B. G., & Wijnberg, K. M. (2001). Morphologic properties derived from a simple cross-shore sediment transport model. *Journal of Geophysical Research*, *106*, 945–958.
- Roelvink, J. A., & Brøker, I. (1993). Cross-shore profile models. *Coastal Engineering*, *21*, 163–191.
- Roelvink, J. A., Meijer, T. J. G. P., Houwman, K., Bakker, R., & Spanhoff, R. (1995). Field validation and application of a coastal profile model. In *Proceedings of coastal dynamics* (pp. 818–828). New York: ASCE.
- Ruessink, B. G., van Enkevort, I. M. J., & Kuriyama, Y. (2004). Nonlinear principal component analysis of nearshore bathymetry. *Marine Geology*, *203*(1), 185–197.
- Ruessink, B. G., Wijnberg, K. M., Holman, R. A., Kuriyama, Y., & Van Enkevort, I. M. J. (2003). Intersite comparison of interannual nearshore bar behavior. *Journal of Geophysical Research*, *108*, 3249. doi:10.1029/2002JC001505.
- Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning internal representations by error propagation. *Parallel Distributed Processing*, *1*, 318–362.
- Southgate, H. N., & Möller, I. (2000). Fractal properties of beach profile evolution at Duck, North Carolina. *Journal of Geophysical Research*, *105*(C5), 11489–11507.
- Thornton, E. B., & Guza, R. T. (1983). Transformation of wave height distribution. *Journal of Geophysical Research*, *88*(C10), 5925–5938.
- Turner, I. L., Aarninkhof, S. G. J., Dronkers, T. D. T., & McGrath, J. (2004). CZM applications of Argus coastal imaging at the Gold Coast, Australia. *Journal of Coastal Research*, *20*, 739–752.
- Van Enkevort, I. M. J., & Ruessink, B. G. (2001). Effects of hydrodynamics and bathymetry on video estimates of nearshore sandbar position. *Journal of Geophysical Research*, *106*, 16969–16979.
- Van Enkevort, I. M. J., Ruessink, B. G., Coco, G., Suzuki, K., Turner, I. L., Plant, N. G., et al. (2004). Observations of nearshore crescentic sandbars. *Journal of Geophysical Research*, *109*, C06028.
- Van Rijn, L. C., Walstra, D. J. R., Grasmeyer, B., Sutherland, J., Pan, S., & Sierra, J. P. (2003). The predictability of cross-shore bed evolution of sandy beaches at the time scale of storms and seasons using process-based profile models. *Coastal Engineering*, *47*, 295–327.
- Williams, R. J., & Peng, J. (1990). An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural Computation*, *2*(4), 490–501.
- Wright, L. D., May, S. K., Short, A. D., & Green, M. O. (1985). Beach and surf zone equilibria and response times. In *Proceedings of the 19th international conference on coastal engineering* (pp. 2150–2164).