Neuro-Fuzzy Network for Modeling the Shoreline Realignment of the Kamari Beach, Santorini, Greece

George E. Tsekouras[1(✉)], Vasilis Trygonis[2], Anastasios Rigos[1],
Antonios Chatzipavlis[2], Dimitrios Tsolakis[1],
and Adonis F. Velegrakis[2]

[1] Department of Cultural Technology and Communication,
University of the Aegean, 81100 Mytilene, Lesbos Island, Greece
gtsek@ct.aegean.gr, a.rigos@aegean.gr
[2] Department of Marine Sciences, University of the Aegean, 81100 Mytilene,
Lesbos Island, Greece
{vtrygonis,a.chatzipavlis}@marine.aegean.gr,
beachtour@aegean.gr

Abstract. In this paper, a novel multiple-layer neuro-fuzzy network is proposed to model/predict shoreline realignment at a highly touristic island beach (Kamari beach, Santorini, Greece). A specialized experimental setup was deployed to generate a set of input-output data that comprise parameters describing the beach morphology and wave conditions and the cross-shore shoreline position at 30 cross-sections of the beach extracted from coastal video imagery, respectively. The proposed network consists of three distinct modules. The first module concerns the network representation of a fuzzy model equipped with a typical inference mechanism. The second module implements a novel competitive learning network to generate initial values for the rule base antecedent parameters. These parameters are, then, used to facilitate the third module that employs particle swarm optimization to perform a stochastic search for optimal parameter estimation. The network is compared favorably to two other neural networks: a radial basis function neural network and a feedforward neural network. Regarding the effectiveness of the proposed network to model shoreline re-alignment, the RMSE found (7.2–7.7 m, depending on the number of rules/nodes), reflects the high variability of the shoreline position of the Kamari beach during the period of observations: the RMSE is of a similar order to the standard deviation (up to 8 m) of the cross-shore shoreline position. The results are encouraging and the effectiveness of the proposed network could be further improved by changes (fine-tuning) of the input variables.

Keywords: Shoreline realignment · Beach morphodynamics · Neuro-fuzzy network · Competitive learning · Particle swarm optimization

1   Introduction

Beaches, i.e. the low-lying coasts formed on unconsolidated sediments, are generally under erosion [1–3]. These significant on their own right ecosystems are critical components of the coastal zone, as they provide very important ecosystem services, such as flood protection to the valuable coastal assets/infrastructure they front [4] as well as substantial socio-economic benefits as they form the pillar of the ever-increasing coastal tourism [5]. Therefore, the long- and short-term beach morphological evolution (morphodynamics) and/or erosion and its controls have important implications for the sustainable development of the coastal zone.

Beach morphodynamics is controlled by complex forcing-response processes that operate at various spatio-temporal scales [6]. An important determinant of the current and future beach morphological evolution is the shoreline position. This is generally characterized by high spatio-temporal variability, mainly controlled by complex interactions between the beach morphology and sediments and the incident waves. Such interactions can lead to large localized shoreline position changesas well as an overall short- and long-term shoreline retreat (beach erosion) particularly under increasing sea levels and changing storminess [7, 8]. However, the study of these changes through traditional morphodynamic modeling (e.g. [7, 9, 10]) is subject to limitations arising from the: (a) high non-linearities involved in the shoreline re-alignment phenomenon due to complex nearshore hydrodynamical, sedimentological and morphological processes, and (b) high computational costs involved.

To address the above problem, we propose to use a novel neuro-fuzzy network (NFN), designed to model shoreline realignment on the basis of a small number of environmental variables that are based on high frequency observations of the shoreline position and the nearshore hydrodynamics. NFNs attempt to offset the approximate reasoning of fuzzy systems through the learning mechanisms and connectionist structures of neural networks [11–14]. The proposed network comprises three modules. The first module is a network representation of a standard fuzzy model. The second module is a competitive learning network structure, which generates a set of initial values for the fuzzy rule base antecedent parts. These values are then used to assist to a particle swarm optimization-based learning procedure that finally carries out the network's optimal parameter estimation. The model is trained and tested at Kamari beach (Santorini), one of the most touristic beaches of the Aegean Archipelago.

The material is organized as follows. Section 2 describes the experimental setup and the data acquisition process. Section 3 presents the analytical structure of the proposed neuro-fuzzy network. The simulation study is given in Sect. 4. Finally, the paper concludes in Sect. 5.

2   Experimental Setup and Raw Data Extraction

Kamari beach is a microtidal beach, located at the island of Santorini (Fig. 1). The southern section of beach (length of about 600 m) forms on coarse sediments, which along the shoreline have mean grain sizes of 2.2–9.5 mm and in the nearshore seabed mean sizes of 0.17–1.39 mm. The beach has a SE orientation and could be exposed to energetic wave conditions.
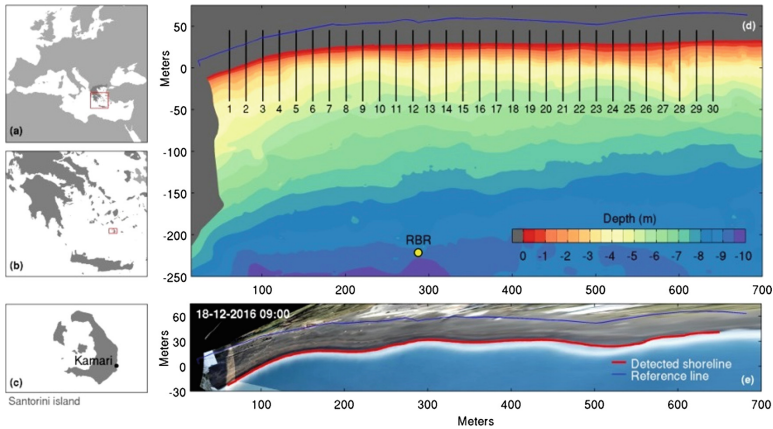
Fig. 1. (a)–(c) Location of Kamari beach in Santorini, Greece. (d) Bathymetric map of the studied beach part showing also the 30 cross-shore sections and the location of the deployed RBR pressure sensor; and (e) example of a detected shoreline plotted on the corresponding TIMEX image showing also the reference line (the blue line located above the shoreline) used to define the output parameter in the experiments. (Color figure online)

The experimental methodology consists of the collection/analysis of (a) beach topographic data and (b) time series of shoreline position and contemporaneous nearshore wave activity during a two-week period at the end of December 2016 (16-31/12/2016).

More specifically, nearshore bathymetric data were obtained through a single beam digital echo-sounder (Hi-Target HD 370) and a Differential GPS (Topcon Hipper RTK-DGPS) deployed from an inflatable boat, whereasland topographic data were obtained through a RTK-DGPS survey at the beginning of the examined period (16/12/2016). From these data, 30 cross-shore (perpendicular to the shoreline) profiles were extracted with a spacing of 20 m (Fig. 1(d)). For each of these cross-shore sections, the wet beach slope (W_S), a critical morphological variable of beach morphodynamics [7, 9], has been estimated forming the first input variable of the proposed network. A pressure sensor (RBR*virtuoso*) deployed offshore at about 9 m water depth of the beach (Fig. 1(d)) provided high frequency (4 Hz) nearshore wave data, during 10 min bursts each hour, from which hourly values of significant wave height (H_S), peak wave period (T_P), wave energy (E) and tidal elevations (T_S) were estimated. In addition, mean wave directions during these bursts were estimated from video records obtained from the coastal video system deployed in the area to monitor in high frequency the shoreline position (see below). Wave directions were expressed with regard to the North-South (denoted as W_{NS}) and East-West (denoted as W_{EW}) vector components. All the above hydrodynamic characteristics control shoreline position as they impose direct effects on beach morphodynamics [15], and form the remainder of the network's input variables. Based on the above nomenclature, the input variables taken into account are described as: $x_1 = W_S$, $x_2 = H_s$, $x_3 = T_p$, $x_4 = E$, $x_5 = T_s$, $x_6 = W_{NS}$, and $x_7 = W_{EW}$.

An autonomous coastal video monitoring system was installed at the beach (at the (0,0) point shown in Fig. 1(e)) to monitor shoreline position. The system, which allows automatic and low-cost monitoring of key beach features, comprised a station PC and two fixed Vivotek IP8362 video cameras (center of view elevation of about 22 m above mean sea level), operating at a sampling rate of 5 frames per secondand in hourly 10-minute bursts. Images were corrected for lens distortion and georectified and pro-jected on real-world coordinates using standard photogrammetric methods and ground control points (GCPs) collected during the dedicated RTK-DGPS survey. Shoreline detection was performed during post-processing, using an automated coastal feature detector that records the mean shoreline position over each 10-minute burst on the basis of the obtained TIMEX images (Fig. 1(e)); TIMEX images are the time-averages of the 3000 snapshots collected in each burst defined on the red–green–blue (RGB) colour model. A detailed description of the system and the automated procedure developed to extract the shoreline from the TIMEX imagery can be found in [10]. On the basis of this information, the network's output variable (y) (for each of the 30 cross-shore profiles and each burst) was defined as the distance between the video extracted shoreline position and a reference line (see Fig. 1(e)); this variable expresses the shoreline position change (re-alignment).

To summarize, the experimental setup generated $n = 3480$ input-output data symbolized as $\boldsymbol{x}_k = \begin{bmatrix} x_{k1} & x_{k2} & x_{k3} & x_{k4} & x_{k5} & x_{k6} & x_{k7} \end{bmatrix}^T$, and $y_k \in \mathrm{R}$. These data are elaborated by the neuro-fuzzy network in order to model the shoreline realignment during the period of the experiment.

3   The Proposed Neuro-Fuzzy Network

The proposed network comprises three modules (see Fig. 2) namely: (a) the Takagi-Sugeno-Kang (TSK) fuzzy network, (b) the competitive learning network, and (c) the particle swarm optimization (PSO). In what follows, it is assumed that there are n input-output data pairs $\{\boldsymbol{x}_k; y_k\}|_{k=1}^{n}$ with $\boldsymbol{x}_k \in \mathrm{R}^p$ and $y_k \in \mathrm{R}$.
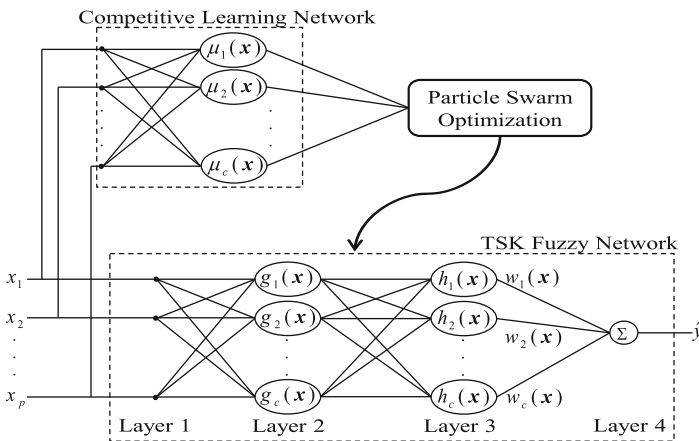

Fig. 2. The structure of the TSK neuro-fuzzy network.

3.1 TSK Fuzzy Network

The typical TSK fuzzy model establishes the input-output relations in terms of c fuzzy rules written as:

$$R_\ell : \ \text{If } x_1 \text{ is } A_{\ell 1} \text{ and } \ldots. \text{ and } x_p \text{ is } A_{\ell p} \text{ then } y \text{ is } w_\ell(\boldsymbol{x}), \quad (1 \le \ell \le c) \qquad (1)$$

where, $A_{\ell j}$ are Gaussian fuzzy sets:

$$A_{\ell j}(x_j) = \exp\left(-\left((x_j - v_{\ell j})/\sigma_{\ell j}\right)^2\right) \qquad (2)$$

and

$$w_\ell(\boldsymbol{x}) = a_{\ell 0} + a_{\ell 1}x_1 + \ldots + a_{\ell p}x_p \qquad (3)$$

with $v_{\ell j}$ and $\sigma_{\ell j}$ being the fuzzy set centers and widths, and $a_{\ell j} \in \mathrm{R}$. The proposed network representation of a TSK fuzzy model is illustrated in Fig. 2.

There are four layers involved in the inference process. Layer 1 is the input layer. Layer 2 calculates the rule firing degrees as indicated next,

$$g_\ell(\boldsymbol{x}) = \prod_{j=1}^{p} A_{\ell j}(x_j) = \exp\left(-\sum_{j=1}^{p}\left((x_j - v_{\ell j})/(\sigma_{\ell j})\right)^2\right) \qquad (4)$$

In Layer 3, the fuzzy basis functions are calculated as follows:

$$h_\ell(\boldsymbol{x}) = g_\ell(\boldsymbol{x}) \bigg/ \sum_{i=1}^{c} g_i(\boldsymbol{x}) \qquad (5)$$

Finally, Layer 4 estimates the output of the network:

$$\hat{y} = \sum_{\ell=1}^{c} h_\ell(\boldsymbol{x})\, w_\ell(\boldsymbol{x}) \qquad (6)$$

3.2 Competitive Learning Network

The motivation of using the competitive learning network (CLN) is to facilitate the evolutionary process of the PSO algorithm. To do so, the CLN elaborates the input-output data and generates a set of parameter values for the rule antecedent parts. To provide an accurate initialization of the learning mechanism, the above values are then codified in the best position of the PSO algorithm (see Sect. 3.3). The structure of CLN (see Fig. 2) includes a hidden layer that comprises c nodes, the activation functions of which are given as,

$$\mu_\ell(\boldsymbol{x}_k) = \left[\sum_{i=1}^{c} \left(\frac{\|\boldsymbol{x}_k - \boldsymbol{v}_\ell\|}{\|\boldsymbol{x}_k - \boldsymbol{v}_i\|}\right)^{\frac{2}{m-1}}\right]^{-1}, \qquad (1 \le \ell \le c;\ 1 \le k \le n) \qquad (7)$$

where $m > 1$, and $\boldsymbol{v}_\ell = \begin{bmatrix} v_{\ell 1}, & v_{\ell 2}, & \dots & v_{\ell p} \end{bmatrix}^T \in R^p$ are the codewords obtained by the competitive learning process. In this sense the set $U = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_c\}$ is realized as the set of the center elements of the partition of the data set $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$ into c clusters. The CLN falls in the realm of batch learning vector quantization (BLVQ) introduced by Kohonen in [16]. The objective of the CLN is to provide an estimation of the $\boldsymbol{v}_\ell\ (1 \le \ell \le c)$ by minimizing the following distortion function [16–18],

$$D = \frac{1}{n} \sum_{k=1}^{n} \min_{1 \le \ell \le c} \left\{ \|\boldsymbol{x}_k - \boldsymbol{v}_\ell\|^2 \right\} \qquad (8)$$

Assume that we are given a partition of the data set X into c clusters with centers (i.e. codewords) $U = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_c\}$. Then, the mean of the set U is evaluated as,

$$\tilde{v} = \frac{1}{c} \sum_{\ell=1}^{c} \boldsymbol{v}_\ell \qquad (9)$$

In addition, we define the $p \times (c + 1)$ matrix

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 & \dots & \boldsymbol{b}_c & \boldsymbol{b}_{c+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_c & \tilde{v} \end{bmatrix} \qquad (10)$$

The codeword positions are directly affected by the distribution of the clusters across the feature space [17, 18]. An optimal cluster distribution should possess well separated and compact clusters. These two properties depend on the relative positions of the codewords [18, 19]. To quantify the effect the codeword relative positions impose on the quality of the partition, we view each codeword as the center of a multidimensional fuzzy set, the elements of which are the rest of the codewords. In this direction, the membership degree of the codeword $\boldsymbol{b}_\ell = \boldsymbol{v}_\ell$ in the cluster with center the codeword $\boldsymbol{b}_i = \boldsymbol{v}_i$ is defined as [18]

$$u_{i\ell} = \left[\sum_{\substack{l=1 \\ l \ne \ell}}^{c+1} \left(\frac{\|\boldsymbol{b}_\ell - \boldsymbol{b}_i\|}{\|\boldsymbol{b}_\ell - \boldsymbol{b}_l\|}\right)^{\frac{2}{m-1}}\right]^{-1}, \qquad (1 \le \ell \le c;\ 1 \le i \le c) \qquad (11)$$

where $m \in (1, \infty)$ is the fuzziness parameter. Note that the indices ℓ and i are not assigned the value $c+1$, which corresponds to the position $\boldsymbol{b}_{c+1} = \tilde{v}$. If the quantity $\tilde{v}$ was not taken into account, then in case there were only two codewords (i.e. $c = 2$) the function in $u_{i\ell}$ would not work properly because it would give one in all cases. Therefore the presence of $\boldsymbol{b}_{c+1} = \tilde{v}$ is important. Next, for each $\boldsymbol{x}_k$ the closest codeword $\boldsymbol{v}_{i_k} = \boldsymbol{b}_{i_k}$ is detected

$$\|\boldsymbol{x}_k - \boldsymbol{v}_{i_k}\| = \min_{1 \le \ell \le c} \{\|\boldsymbol{x}_k - \boldsymbol{v}_\ell\|\} \tag{12}$$

To this end, the codewords are updated by the subsequent learning rule,

$$\boldsymbol{v}_\ell^{t+1} = \boldsymbol{v}_\ell^t + \eta f_{i_k,\ell}^t \left(\boldsymbol{x}_k - \boldsymbol{v}_\ell^t\right) \tag{13}$$

where t is the iteration number, $\eta > 0$ is the learning rate, and $f_{i_k,\ell}$ is the neighborhood function:

$$f_{i_k,\ell} = \begin{cases} 1 & , \; if \; \ell = i_k \\ u_{i_k,\ell}, & otherwise \end{cases} \tag{14}$$

with $i_k \in \{1, 2, \ldots, c\}$, $1 \le \ell \le c$, and $u_{i_k,\ell}$ is calculated in (11). The function $f_{i_k,\ell}$ gives the relative excitation degree of each codeword having as reference point the winning codeword and the relative positions of the rest of the codewords. For a more detailed analysis of the properties of the function $f_{i_k,\ell}$ the interested reader is referred to [18]. As easily seen, the learning rule in Eq. (13) constitutes an on-line process. In a similar way to the BLVQ [16], we can produce a batch mechanism by employing the subsequent expectation measure:

$$E\left[f_{i_k,\ell}^t \left(\boldsymbol{x}_k - \boldsymbol{v}_\ell^t\right)\right] = 0 \;\; as \; t \to \infty \tag{15}$$

The condition in (15) enables us to modify the codeword updating rule as,

$$\boldsymbol{v}_\ell^{t+1} = \sum_{k=1}^{n} f_{i_k,\ell}^t \boldsymbol{x}_k \left/ \sum_{k=1}^{n} f_{i_k,\ell}^t \right. \tag{16}$$

The above CLN appears to be less sensitive to initialization when compare to the BLVQ. This remark is justified by the fact that, based on the functions in (11) and (14), a specific codeword moves towards its new position considering the relative positions of the rest of codewords. Thus, before obtaining the new partition, the algorithm takes into account the overall current partition and forces all codewords to be more competitive. Therefore, in a single iteration all of the codewords are moving in an attempt to win as much as training vectors they can. This behavior enables the whole updating process to avoid undesired local minima.

Based on the nomenclature given in Eqs. (1)–(6), to extract initial values for the rule antecedents, the fuzzy set centers are determined by projecting the codewords $\boldsymbol{v}_\ell$ $(1 \le \ell \le c)$ on each dimension:

$$v_{\ell j} = v_{\ell j}, \quad (1 \le \ell \le c; \; 1 \le j \le p) \tag{17}$$

while the corresponding fuzzy set widths by the next relation

$$\sigma_{\ell j} = (Diag(\boldsymbol{FC}_\ell))^{1/2} \tag{18}$$

where $\boldsymbol{FC}_\ell$ is the fuzzy covariance matrix [20], which based on Eq. (7) is

$$\boldsymbol{FC}_\ell = \sum_{k=1}^{n} (\mu_\ell(\boldsymbol{x}_k))^m (\boldsymbol{x}_k - \boldsymbol{v}_\ell)(\boldsymbol{x}_k - \boldsymbol{v}_\ell)^T \Big/ \sum_{k=1}^{n} (\mu_\ell(\boldsymbol{x}_k))^m \tag{19}$$

3.3 Particle Swarm Optimization

The particle swarm optimization (PSO) elaborates on a population of N particles $\boldsymbol{p}_i \in \mathrm{R}^q$ [21–23]. Each particle is assigned a velocity $\boldsymbol{h}_i \in \mathrm{R}^q (1 \leq i \leq N)$. The positions with the best values of the objective function obtained so far by the ith particle and by all particles are respectively denoted as $\boldsymbol{p}_i^{best}$ and $\boldsymbol{p}_{best}$. The velocity is updated as

$$\begin{aligned} \boldsymbol{h}_i(t+1) = {}& \omega \boldsymbol{h}_i(t) + \varphi_1 \, \boldsymbol{U}(0,\,1) \otimes \left(\boldsymbol{p}_i^{best}(t) - \boldsymbol{p}_i(t)\right) + \varphi_2 \, \boldsymbol{U}(0,\,1) \\ & \otimes \left(\boldsymbol{p}_{best}(t) - \boldsymbol{p}_i(t)\right) \end{aligned} \tag{20}$$

where $\otimes$ is the point-wise vector multiplication; $\boldsymbol{U}(0,\,1)$ is a vector with elements randomly generated in [0, 1]; ω, φ_1, and φ_2 are positive constant numbers called the inertia, cognitive and social parameter, respectively. The position of each particle is:

$$\boldsymbol{p}_i(t+1) = \boldsymbol{p}_i(t) + \boldsymbol{h}_i(t+1) \tag{21}$$

In our case, to implement the PSO, each particle codifies the antecedent parameters, i.e. the fuzzy set centers and widths. Therefore, the dimension of the particles' search space is equal to $q = 2\,c\,p$ (i.e. $\boldsymbol{p}_i \in \mathrm{R}^q$, $i = 1, 2, \ldots, N$). All particles are randomly initialized. In addition, the values of the antecedent parameters that were calculated in Eqs. (17) and (18) are codified in the best overall position $\boldsymbol{p}_{best}$, which is expected to guide more efficiently the particles in their search for optimal parameter estimation. On the other hand, there are $c(p+1)$ consequent parameters, described as

$$\boldsymbol{a} = \left[a_{10}, a_{11}, \ldots, a_{1p}, a_{20}, a_{21}, \ldots, a_{2p}, \ldots, a_{c0}, a_{c1}, \ldots, a_{cp}\right]^T \tag{22}$$

The estimation of these parameters is carried out in terms of ridge regression [24]. To do so, we employ the following matrices:

$$\Lambda = \left[\Lambda_1 \quad \Lambda_2 \quad \ldots \quad \Lambda_n\right]^T \tag{23}$$

$$\Lambda_k = \left[\boldsymbol{\lambda}_{k1} \quad \boldsymbol{\lambda}_{k2} \quad \ldots \quad \boldsymbol{\lambda}_{kc}\right] \quad (1 \leq k \leq n) \tag{24}$$

$$\boldsymbol{\lambda}_{k\ell} = \left[h_\ell(\boldsymbol{x}_k) \quad h_\ell(\boldsymbol{x}_k)\, x_{k1} \quad \ldots \quad h_\ell(\boldsymbol{x}_k)\, x_{kp}\right] \quad (1 \leq \ell \leq c) \tag{25}$$

Note that the dimensionality of matrix Λ is $n \times c(p+1)$. The objective of ridge regression is to minimize the following error function [24]:

$$E = \|\Lambda \boldsymbol{a} - Y\|^2 + \|\Gamma a\|^2 \tag{26}$$

where $Y = [y_1, y_2, \ldots, y_n]^T$, $\Gamma = \beta I$ with $\beta > 0$, and I the $c(p+1) \times c(p+1)$ identity matrix. The parameter β is called regularization parameter and, in this paper, its value is adjusted manually. The solution to the above problem reads as [24],

$$\tilde{\boldsymbol{a}} = \left(\Lambda^T \Lambda + \Gamma^T \Gamma\right)^{-1} \Lambda^T Y \tag{27}$$

4 Simulation Study and Discussion

Based on the analysis described in Sect. 2, the data set includes $n = 3480$ input-output data pairs (corresponding to 30 beach cross-sections) of the form $\{\boldsymbol{x}_k; y_k\}|_{k=1}^n$ with $\boldsymbol{x}_k = \begin{bmatrix} x_{k1} & x_{k2} & \ldots & x_{k7} \end{bmatrix}^T$ and $y_k \in \Re$. The data set was divided into a training set consisting of the 60% of the original data, and a testing set consisting of the remainder 40%. Parameter setting for the proposed methodology is as follows: $m = 2$, and $\beta = 20$. For the PSO we set $\varphi_1 = \varphi_2 = 2$, the parameter ω was randomly selected in [0.5, 1], and the population size was $N = 20$.

For comparison, two more neural networks were designed and implemented. The first is a radial basis function neural network (RBFNN). The parameters of the basis functions were estimated in terms of the input-output fuzzy clustering algorithm developed in [20], while the connection weights were calculated by the orthogonal least squares method. The second is a feedforward neural network (FFNN), the activation functions of which read as:

$$f(x) = \tanh \frac{x}{2} \tag{28}$$

To train the FFNN we used the PSO algorithm (see previous Section) with the above parameter setting. All networks were implemented using the Matlab software.

The performance index used in the simulations was the root mean square error:

$$RMSE = \sqrt{\frac{1}{n} \sum_{k=1}^n |y_k - \hat{y}_k|^2} \tag{29}$$

For the three networks, we considered various numbers of rules/nodes, whereas for each number of rules/nodes we run 10 different initializations. The results are shown in Table 1. The proposed network appears to have the best performance compared with the other two networks. The best result for both the training and testing data sets is obtained by the proposed network for a number of rules equal to $c = 10$.

The number of rules plays an important role for the validity of the method since as it increases the RMSE decreases. The results reported in Table 1 are visualized also in Fig. 3. There are some interesting observations related to this figure. First, the superiority of the proposed method is clear, in comparison with the other two tested networks. Second, the behavior of the FFNN and the RBFNN appears to be similar, but with the latter achieving better performance in both the training and testing data. Third, in all networks, the general tendency is a decrease (as expected) of the RMSEs as the number of rules/nodes increases.

Table 1. RMSEs and the corresponding standard deviations for the three networks with respect to various numbers of rule/nodes

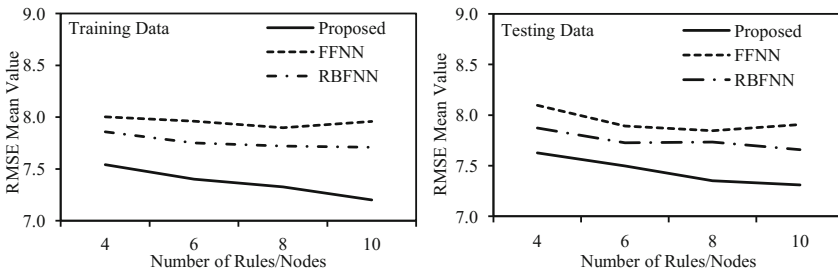| Method		$c = 4$	$c = 6$	$c = 8$	$c = 10$
Proposed	Train	7.540 ± 0.059	7.402 ± 0.044	7.327 ± 0.109	7.201 ± 0.099
	Test	7.626 ± 0.095	7.497 ± 0.079	7.351 ± 0.114	7.308 ± 0.115
FFNN	Train	8.003 ± 0.076	7.960 ± 0.055	7.899 ± 0.060	7.958 ± 0.108
	Test	8.097 ± 0.083	7.893 ± 0.116	7.846 ± 0.077	7.907 ± 0.102
RBFNN	Train	7.858 ± 0.004	7.751 ± 0.003	7.720 ± 0.005	7.710 ± 0.002
	Test	7.871 ± 0.005	7.727 ± 0.006	7.658 ± 0.004	7.734 ± 0.005

Fig. 3. Mean values of the RMSE as a function of the number of rules/nodes for the training and testing data.

With regard to the effectiveness of the proposed network to model shoreline re-alignment, the following should be noted. The RMSEs found, although smaller than those of the other tested networks is still considered high, being between 7.2–7.7 m, depending on the number of rules/nodes. However, it should be noted that the variability of the shoreline position of the Kamari beach was indeed high during the 15-day period of observations, showing, in some cross-sections, differences between the minimum and maximum cross-shore shoreline position in excess of 15 m and a standard deviation of up to 8 m. Therefore, it seems that the obtained high RMSEs reflect the high spatio-temporal variability of the Kamari shoreline position observed during the period of observations; thus, the proposed network could model/predict the shoreline re-alignment to accuracy similar to that of the actual variability of the beach.

The results suggest that in order to improve the proposed model effectiveness in the case of beaches the shorelines of which are characterised by high spatio-temporal variability, there should be some changes in the input variables. First, another input variable might be introduced that can define the 'antecedent' cross-shore position of the shoreline; a good guess should be the average shoreline position of the previous 24 h for each tested cross-section that could be easily defined from the video imagery. It is thought that the inclusion of such a parameter will improve the behavior of the proposed model. Secondly, as the tidal range at the Kamari beach was found to be rather small (<0.08 m), it may be that nearshore total water level during energetic events is controlled mostly by the storm-induced level increases. In this case, it makes sense for an input variable reflecting this process (i.e. the total nearshore water level) to be used instead of the tidal elevation (T_S). Generally, the encouraging results of the modelling exercise suggest that the proposed neural network, after a fine-tuning of the input variables, can be used to model the shoreline re-alignment of beaches characterised by highly variable morphodynamics, on the basis of a small number of environmental variables.

5   Summary and Conclusions

In the present contribution we present a systematic methodology that uses a sophisticated experimental setup and a novel neuro-fuzzy network to model and predict the phenomenon of shoreline realignment at a highly touristic beach (Kamari, Santorini). A set of morphological and wave variables were identified that can affect shoreline realignment, which together with records of hourly shoreline positions obtained using a coastal video imagery system, were used to generate the network's input-output training data. The proposed network consists of three modules. The main task of the first module is to provide the inference mechanism of a typical fuzzy system represented as network structure. The second module constitutes a competitive learning network able to provide an initial partition of the input feature space. This partition is used to extract a set of values for the fuzzy rule antecedent parameters that are codified in the overall best position of a PSO approach. Then, the PSO runs properly and optimizes the network's parameters. In each step of the iterative implementation of the PSO algorithm, the fuzzy rule consequent parameters are evaluated using ridge regression. Simulation experiments were carried and the results were compared with those by two other neural networks: a radial basis function neural network and a feedforward neural network. The comparison showed that the proposed network performs better in all cases. Regarding the effectiveness of the proposed network to model shoreline re-alignment, the RMSE found (7.2–7.7 m, depending on the number of rules/modes), reflects the high variability of the shoreline position of the Kamari beach during the period of observations: the RMSE is of a similar order to the standard deviation (up to 8 m) of the cross-shore shoreline position. The results are encouraging and the effectiveness of the proposed network could be further improved by changes (fine-tuning) of the input variables.

References

1. Eurosion: Living with coastal erosion in Europe: Sediment and Space for Sustainability. Part II. DG Environment, EC (2004). http://www.eurosion.org/reports-online/part2.pdf
2. Seneviratne, S.I., Nicholls, N., Easterling, D., et al.: Changes in climate extremes and their impacts on the natural physical environment. In: Field, C.B. et al. (eds.) Managing the Risks of Extreme Events and Disasters to Advance Climate Change Adaptation (Chap. 3). A Special Report of Working Groups I and II of the Intergovernmental Panel on Climate Change (IPCC), pp. 109–230. Cambridge University Press, Cambridge, UK, and New York, NY, USA (2012)
3. IPCC Climate Change: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. In: Stocker, T. F., Qin, D., Plattner, G.-K., Tignor, M., Allen, S.K., Boschung, J., Nauels, A., Xia, Y., Bex, V., Midgley, P.M. (eds.) Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA (2013)
4. Neumann, B., Vafeidis, A.T., Zimmerman, J., Nicholls, R.J.: Future coastal population growth and exposure to sea-level rise and coastal flooding - a global assessment. PLoS One (2015). doi:10.1371/journal.pone.0118571
5. Handmer, J., Honda, Y., Kundzewicz, Z.W., et al.: Changes in impacts of climate extremes: human systems and ecosystems. In: Field, C.B. et al. (eds.) Managing the Risks of Extreme Events and Disasters to Advance Climate Change Adaptation (Chap. 4). A Special Report of Working Groups I and II of the Intergovernmental Panel on Climate Change (IPCC), pp. 231–290. Cambridge University Press, Cambridge, UK, and New York, NY, USA (2012)
6. Short, A.D., Jackson, D.W.T.: Beach morphodynamics. Treatise Geomorphol. **10**, 106–129 (2013)
7. Monioudi, I.N., Velegrakis, A.F., Chatzipavlis, A.E., Rigos, A., Karambas, T., Vousdoukas, M.I., Hasiotis, T., Koukourouvli, N., Peduzzi, P., Manoutsoglou, E., Poulos, S.E., Collins, M.B.: Assessment of island beach erosion due to sea level rise: the case of the Aegean archipelago (Eastern Mediterranean). Nat. Hazards Earth Syst. Sci. **17**, 449–466 (2017)
8. Mentaschi, L., Vousdoukas, M.I., Voukouvalas, E., Dosio, A., Feyen, L.: Global changes of extreme coastal wave energy fluxes triggered by intensified teleconnection patterns. Geophys. Res. Lett. (2017). doi:10.1002/2016GL072488
9. Roelvink, D., Reniers, A., van Dongeren, A., van Thiel de Vries, J., Lescinsky, J., McCall, R.: Modeling storm impacts on beaches, dunes and barrier islands. Coastal Eng. **56**, 1133–1152 (2009)
10. Velegrakis, A.F., Trygonis, V., Chatzipavlis, A.E., Karambas, T., Vousdoukas, M.I., Ghionis, G., Monioudi, I.N., Hasiotis, T., Andreadis, O., Psarros, F.: Shoreline variability of an urban beach fronted by a beachrock reef from video imagery. Nat. Hazards **83**(1), 201–222 (2016)
11. Banakar, A., FazleAzeemb, M.: Parameter identification of TSK neuro-fuzzy models. Fuzzy Sets Syst. **179**, 62–82 (2011)

12. Chen, C., Wang, F.W.: A self-organizing neuro-fuzzy network based on first order effects sensitivity analysis. Neurocomputing **118**, 21–32 (2013)
13. Rigos, A., Tsekouras, G.E., Vousdoukas, M.I., Chatzipavlis, A., Velegrakis, A.F.: A Chebyshev polynomial radial basis function neural network for automated shoreline extraction from coastal imagery. Integr. Comput.-Aided Eng. **23**, 141–160 (2016)
14. Tsekouras, G.E., Rigos, A., Chatzipavlis, A., Velegrakis, A.: A neural-fuzzy network based on hermite polynomials to predict the coastal erosion. Commun. Comput. Inf. Sci. **517**, 195–205 (2015)
15. Thomas, T., Phillips, M.R., Williams, A.T.: A centurial record of beach rotation. J. Coastal Res. **65**, 594–599 (2013)
16. Kohonen, T.: The self-organizing map. Neurocomputing **21**, 1–6 (2003)
17. Tsolakis, D., Tsekouras, G.E., Niros, A.D., Rigos, A.: On the systematic development of fast fuzzy vector quantization for grayscale image compression. Neural Netw. **36**, 83–96 (2012)
18. Tsolakis, D.M., Tsekouras, G.E.: A fuzzy-soft competitive learning approach for grayscale image compression. In: Celebi, M., Aydin, K. (eds.) Unsupervised Learning Algorithms, pp. 385–404. Springer, Heidelberg (2016). doi:10.1007/978-3-319-24211-8_14
19. Tsolakis, D., Tsekouras, G.E., Tsimikas, J.: Fuzzy vector quantization for image compression based on competitive agglomeration and a novel codeword migration strategy. Eng. Appl. Artif. Intell. **25**, 1212–1225 (2012)
20. Tsekouras, G.E.: Fuzzy rule base simplification using multidimensional scaling and constrained optimization. Fuzzy Sets Syst. **297**, 46–72 (2016)
21. Clerc, M., Kennedy, J.: The particle swarm-explosion: stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2002)
22. Eberhart, R.C., Shi, Y.: Tracking and optimizing dynamic systems with particle swarms. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 94–100, Seoul, Korea (2001)
23. Tsekouras, G.E.: A simple and effective algorithm for implementing particle swarm optimization in RBF network's design using input-output fuzzy clustering. Neurocomputing **108**, 36–44 (2013)
24. Tikhonov, A.N., Goncharsky A.V., Stepanov V.V., Yagola A.G.: Numerical methods for the solution of ill-posed problems. Kluwer Academic Publishers, Berlin (1995)