

## A Hermite Neural Network Incorporating Artificial Bee Colony Optimization to Model Shoreline Realignment at a Reef-Fronted Beach

George E. Tsekouras<sup>1</sup>, Vasilis Trygonis<sup>2</sup>, Andreas Maniatopoulos<sup>3</sup>, Anastasios Rigos<sup>1</sup>, Antonios Chatzipavlis<sup>2</sup>, John Tsimikas<sup>4</sup>, Nikolaos Mitianoudis<sup>3</sup>, Adonis F. Velegrakis<sup>2</sup>

<sup>1</sup>Department of Cultural Technology and Communication, University of the Aegean, Greece

<sup>2</sup>Department of Marine Sciences, University of the Aegean, Greece

<sup>3</sup>Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece

<sup>4</sup>Department of Statistics and Actuarial-Financial Mathematics, University of the Aegean, Greece

[gtsek@ct.aegean.gr](mailto:gtsek@ct.aegean.gr), [vtrygonis@marine.aegean.gr](mailto:vtrygonis@marine.aegean.gr), [andrmani@ee.duth.gr](mailto:andrmani@ee.duth.gr), [a.rigos@aegean.gr](mailto:a.rigos@aegean.gr),  
[a.chatzipavlis@marine.aegean.gr](mailto:a.chatzipavlis@marine.aegean.gr), [tsimikas@aegean.gr](mailto:tsimikas@aegean.gr), [nmitiano@ee.duth.gr](mailto:nmitiano@ee.duth.gr), [beachtour@aegean.gr](mailto:beachtour@aegean.gr)

### Abstract

This paper investigates the potential of using a novel Hermite polynomial neural network to model shoreline realignment along an urban beach fronted by a highly irregular beachrock reef. Modeling takes place on the basis of a number of input variables related to reef morphology and wave forcing, whereas the output variable is time series of shoreline position that have been recorded in high spatio-temporal resolution using a coastal video monitoring system. The main network functionality is the generation of Hermite truncated polynomial series of linear combinations of the input variables, and output is calculated as the weighted sum of these truncated series. It is shown that the proposed network can approximate any continuous function defined on a compact set of the multidimensional Euclidean space to arbitrary accuracy. The network is optimized in terms of a modified artificial bee colony method. For comparative reasons, three more related neural networks have been tested that have been optimized by employing different swarm intelligence-based algorithms. Comparison between the four networks has been carried out by standard performance criteria and detailed parametric statistical analysis. Main results of the study are: (a) polynomial orders 3 and 4 are able to effectively handle reasonably well the high nonlinear effects imposed by the presence of the reef; (b) the statistical analysis indicates that the proposed network outperforms the other networks tested; and (c) model efficiency improves noticeably when beach sections behind

reef inlets and/or particular wide sections of the reef that introduce high shoreline variability are not considered.

**Keywords:** Hermite polynomials, polynomial neural network, artificial bee colony method, beach shoreline realignment, perched beach, coastal video monitoring.

## **1. Introduction**

In embayed ('pocket') beaches, periodic beach alongshore sediment movement towards opposing directions can occur in response to shifts in the direction of incident waves; this can result in realignment of the shoreline [56, 57] that could manifest as an apparent rotation of the beach planform [28]. In recent decades, this process has been documented in many beaches (e.g. [52, 53, 57]) and suggested to be an important self-regulatory beach mechanism [28] under changing wave regimes [10, 46, 55]. Beach shoreline realignment is a non-linear process, dependent not only on the longshore sediment fluxes, but also on the longshore variability in the cross-shore sediment fluxes [18, 57]. It is also controlled by the nearshore bed morphology, with particularly complex patterns occurring at perched beaches (i.e. beaches underlain and/or fronted seaward by shallow buried and/or outcropping natural reefs or engineered hard structures [13, 16]), a beach type that proliferates along the global coastline. At such beaches, in addition to the changing wave forcing, shoreline realignment is also dependent on the nearshore bed morphology (e.g. the distance and dimensions of the offshore natural reefs or engineered structures) the interaction of which with the incoming waves can produce complex, non-linear nearshore flows [14, 50, 59]. It appears that modeling of the shoreline realignment in perched beaches requires approaches capable of capturing the high non-linearities of the system and which can model beach morphological response on the basis, if possible, of a minimum number of environmental parameters. In recent years, a promising approach has been to employ neural networks [2, 19, 23, 27, 43].

Polynomial neural networks (PNNs) constitute a class of neural networks that use high-order functional representations to establish highly nonlinear input-output relationships. They are formed by embedding polynomial functions into the network's topology. PNNs have been shown to form effective frameworks in a wide range of applications, such as stock market predictions [31], nonlinear channel equalization [42], classification problems [36], solving differential equations [34], system identification [41, 45], and shoreline extraction from video images [43, 48, 49, 60].

Various types of PNNs have been developed so far. Pao [39] introduced the functional link neural network (FLNN), a single-layer network without hidden layers. FLNNs are formed by tensor-based series expansions of the input variables, whereas the output is estimated as the aggregate of the weighted sum of the above expansions. Later studies incorporated Chebyshev [29, 41, 45] and Legendre polynomials [31, 34, 42] into the structure of FLNNs. The use of ridge polynomials in constructing feedforward networks was discussed in [51]. Oh et al [38] employed Kolmogorov-Gabor polynomials to design radial basis function (RBF) networks trained by sophisticated evolutionary computation procedures, whereas Rigos et al [48] defined the connection weights in the output layer of an RBF network through a specialized Chebyshev polynomial structure. The implementation of the group method of data handling [24, 35] in the designing fuzzy PNNs was studied in [22, 40]; the network was obtained through a dynamic process and included several layers, each of which consisted of a set of multi-input single-output nodes with polynomial activation functions. A major difficulty associated with the implementation of these networks has been that the number of parameters grows fast with the number of input variables.

To address the above difficulty and reduce the number of polynomial parameters, Ma and Khorasani [33] used Hermite polynomials in a more straightforward manner, i.e. as node activation functions, with the inputs to the polynomials being linear combinations of the input variables. A similar approach was investigated by Rigos et al [47], who embedded non-linear constraint optimization to

appropriately normalize the input-side weight parameters of Legendre polynomial activation functions. A problem associated with both the above approaches is that the inputs to different polynomial activation functions are different to one another, as each linear combination of the input variables enters a single polynomial activation function, meaning that the resulting series expansion is not exact [33]. Hence, the polynomial representations might be compromised, influencing negatively the approximation capabilities of the network.

In the present contribution, Hermite polynomials are employed to define the activation functions, and the above problem is addressed as follows. Linear combinations of the input variables are created, each of which corresponds to an inner product operator. Then, contrary to the approaches developed in [33, 47], we do not insert each linear combination into a single polynomial activation function, but expand it into a Hermite polynomial truncated series and the output is defined as the weighted sum of these series. The proposed network is used to model the shoreline realignment/rotation of an urban island beach fronted by a natural reef (Ammoudara, Crete, Greece, see Fig.1) using a number of environmental variables. The input variables of the proposed network are basic morphological characteristics of the fronting reef and the offshore wave forcing, whereas output is the shoreline position time series, which were recorded in high spatio-temporal resolution using a coastal video monitoring system.

Main contributions of the study are related to the: development of a novel Hermite polynomial NN to model the shoreline realignment at a reef-fronted beach that exhibits complex nearshore processes; and the acquisition of accurate nearshore bathymetric information and shoreline position and offshore wave data time-series which are then systematically analysed to generate input-output data sets that can meaningfully describe beach shoreline realignment/rotation.

The paper is structured as follows. The experimental setup and network's input/output variables are described in Section 2, whereas Section 3 describes the structure of proposed neural network. The study results and discussion are reported in Section 4. Finally, the paper concludes in Section 5.

## **2. Study Area and Data**

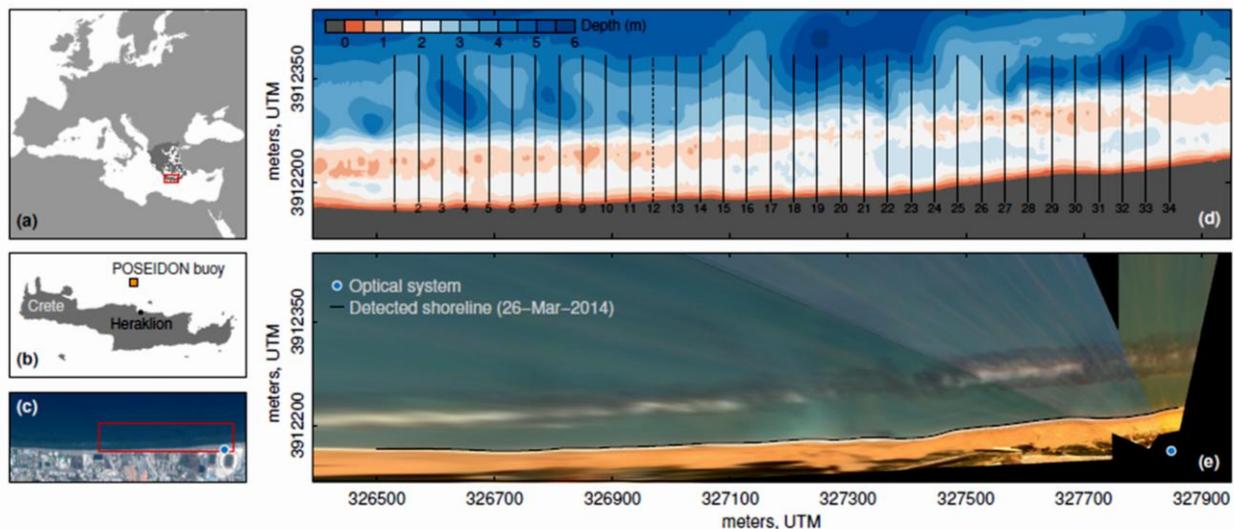
### ***2.1 Study area***

Ammoudara is a 6.1 km long, urban beach, located to the west of the port of Heraklion, Crete, Greece (see Fig. 1). The perched beach can be divided into two main sectors. At the western sector (about 1.5 km), beachrock (i.e. lithified beach sediments [61]) is found at or close to the beach face, whereas at the eastern sector (about 4.6 km long) beachrock forms a submerged reef oriented almost parallel to the coast with a varying width and distance from the shoreline (Fig. 1). Dry beach widths range between 22 and 75 m along the beach, with the inner beach associated with low sand dunes, as well as extensive human development. Beach face gradients vary, with the steeper gradients ( $5-8^\circ$ ) found at the east, where the dry beach forms on sands and gravely sands. Ammoudara beach is exposed to winds and waves from the northern sector and analysis of the available historical information has shown that it has experienced significant shoreline retreat and sediment loss since 1960. Shoreline retreat of 10-60 m has been recorded with longer retreats found at the eastern section, suggesting that, at least, part of the now submerged reef was originally at the beach face and left behind due to beach erosion [3, 4]. Recent research has shown that the shoreline position at the eastern section of the beach, is characterized by high spatio-temporal variability [59].

### ***2.2 Experimental Setup and Data Acquisition***

The nearshore bed morphology of the eastern Ammoudara beach was resolved by a highly detailed bathymetric survey in late October 2014. The nearshore bed (down to water depths of about 6 m)

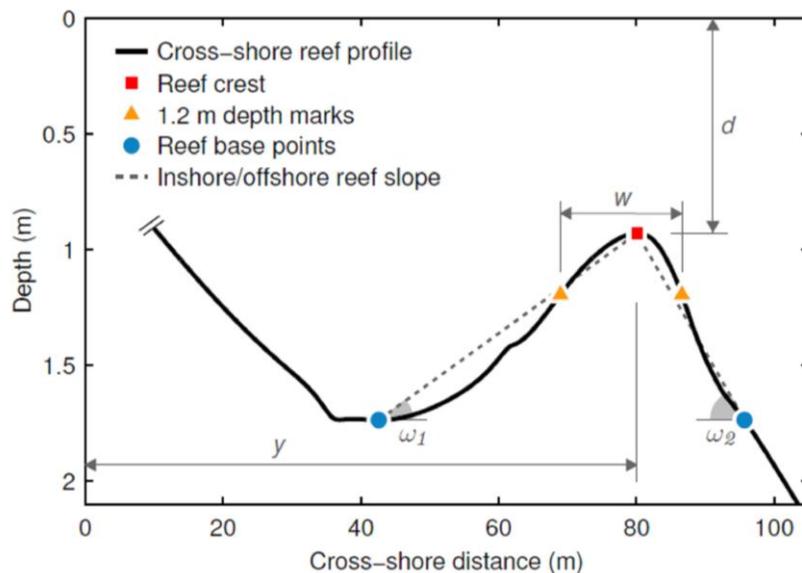
was surveyed using a dense grid (up to 20 m spacing) of cross- and long-shore bathymetric transects; further offshore (to a depth of about 12 m), a coarser grid was used.



**Figure 1.** (a), (b) and (c) Ammoudara beach, Crete and locations of the offshore POSEIDON E1-M3A wave buoy and of the video monitoring (optical) system at the eastern margin of the beach; the monitored beach area is shown by a red square. (d) Detailed bathymetry of the monitored area showing the complex reef architecture; vertical lines correspond to the 34 cross-shore sections analysed. (e) TIMEX image, showing also the location of video monitoring system, and the extent of the analysed shoreline (noted by a black curve superimposed on the shoreline).

Data were obtained through a single-beam digital Hi-Target HD 370 echosounder and a Differential GPS (Topcon Hipper RTK-DGPS) deployed from a shallow draft inflatable boat; additional information on seabed elevation was collected by divers over areas of very shallow/outcropping reef crests. The dry beach morphology (down to water depths of about 0.8 m to overlay/‘tie’ the topographic and bathymetric transects) was recorded using a dense grid of RTK-DGPS cross- and long-shore topographic transects. Detailed shallow side scan sonar and bed sediment sampling surveys were also carried out, which have shown that the bed sediments inshore of the reef consist of patches of poor/moderately sorted, gravely sands and sandy gravels (median size diameters ( $d_{50}$ s) between 1.8 and 2.7 mm) and offshore of the reef mostly of sands. Seabed sediment thickness varies, with extensive areas of the seabed forming on outcropping beachrock, being devoid of surficial sediments; generally the beach sediment wedges seaward towards the outcropping beachrock reef.

The morphological information, following standard corrections, was used to generate an accurate beach digital elevation model (DEM)/map. Morphological characteristics of 34 cross-shore profiles along or very close to actual topographic/bathymetric transects were recorded to provide input parameters for the proposed neural network (Fig. 1(d)).



**Figure 2.** Selected cross-shore section (dashed line in Fig. 1(d)), showing the variables used to describe the reef architecture and the morphological inputs to the proposed network. The network’s output  $y$ , i.e. the crest reef-shoreline distance, is also shown. Note: the reef profile has been smoothed through interpolation.

The morphological characteristics selected as input variables are (Fig.2): the water depth  $d$  of the crest of the submerged reef, the reef onshore and offshore slopes ( $\omega_1$  and  $\omega_2$ , respectively) and the reef width  $w$  at 1.2 m water depth. The aforementioned morphological features are considered as important controls on wave breaking and transmission over both natural and engineered submerged reefs (e.g. [8, 14, 32]) and, thus, on the wave energy distribution along the shorelines they front. Note that no input variable related to the texture of beach sediments has been selected, although sediment texture also forms a significant control of beach morphodynamics. This is due to the very patchy distribution (and, in many areas, absence) of the seabed sediments that does not allow

selection of a sediment size that could reasonably represent the bed sediments along the cross-shore sections.

With regard to wave forcing inputs, these were abstracted from the information collected by an offshore wave buoy (POSEIDON E1-M3A buoy), located about 35 km to the north of the study area (35.66° N and 24.99° E) at 1440 m water depth (Fig. 1(b)) and installed/operated by the Greek National Centre for Marine Research (NCMR). Three wave parameters were considered as network input variables, namely the significant wave height ( $H_s$ , in m), the peak wave period ( $T_p$ , in seconds), and the wave steepness ( $So$ ) that quantifies the wave height/wave length ratio. Their daily values were used, for periods with waves from the northern sector (i.e. waves affecting the beach) and for which there was available concurrent information on beach shoreline positions from the coastal video monitoring system.

Time series of shoreline positions at the 34 selected cross-sections were provided by a coastal video monitoring system deployed at the eastern margin of Ammoudara beach (Fig. 1). The system consisted of three PointGrey FLEA-2 cameras, installed at 26 m above the mean sea level, and a field station PC. Hourly 10-min bursts were obtained during daylight with an image acquisition rate of 5 Hz. From this imagery, time-averaged (TIMEX) images, i.e. time averages of the bursts' 3000 snapshots defined on the red–green–blue (RGB) colour model [1, 21], were produced. As accuracy decreases with the distance from the camera due to the increasing pixel footprint, images from the proximal beach stretch (1400 m long) have been considered (Fig. 1(e)); in this area, pixel footprint (and accuracy of shoreline detection) was always  $< 0.5$  m.

Images were geo-rectified through standard photogrammetric methods, including the calibration of the cameras' intrinsic parameters (distortion) and estimation of extrinsic parameters on the basis of an extensive set of ground control points (GCPs), collected during a dedicated RTK-DGPS topographic survey. All images were projected to real-world coordinates, and as multiple cameras

were used, geo-referenced mosaics were generated with 0.5 m resolution. Since image intensity at the shoreline is associated with the wave foam on the swash zone, the weighted mean positions of the innermost zones of high intensity (as manifested in the TIMEX imagery) were used as records of the shoreline position at the 34 selected cross-shore sections (Fig. 1) during the hourly 10-min bursts (for further details of the image processing methodology see [59]). In all, 1430 TIMEX mosaics in 122 days were obtained in the period between the 1<sup>st</sup> January and the 5<sup>th</sup> November 2014; periods of system downtime and/or periods with no recorded waves from the northern sector were ignored. Using these mosaics, daily average shorelines were estimated, and shoreline positions at the 34 cross-sections recorded. From this information, the cross-shore distances  $y$  between the shoreline and the fronting (fixed) reef crest (see Fig. 2) were estimated; these distances and their variability in time and space define shoreline realignment/rotation and form the output to the proposed network.

In summary, the input variables selected for the proposed neural network are  $x_1 = d$ ,  $x_2 = \tan \omega_1$ ,  $x_3 = \tan \omega_2$ ,  $x_4 = w$ ,  $x_5 = H_S$ ,  $x_6 = T_p$ , and  $x_7 = S_o$ , and the output variable  $y$ , (the distance (in m) between the reef crest and the shoreline). In total,  $N = 4148$  input-output data were generated, having the form:  $S = \{(\mathbf{x}_k, y_k) | \mathbf{x}_k = [x_{k1}, x_{k2}, x_{k3}, x_{k4}, x_{k5}, x_{k6}, x_{k7}]^T, y_k \in \mathfrak{R}, k = 1, 2, \dots, N\}$ . The input-output data are available in <http://erabeach.aegean.gr/datasets/>.

### 3. The proposed Neural Network

The proposed neural network comprises four layers (Fig. 3). In a nutshell, Layer 1 creates linear combinations of the input variables. In Layer 2, each linear combination enters a set of Hermite polynomials up to the  $n$ -th order and in Layer 3 the corresponding truncated series are generated. Finally, Layer 4 sums up the weighted truncated series and estimates the network's output.

A modified artificial bee colony (ABC) algorithm is used to minimize the network's square error, by optimizing all weight parameters. There are several reasons for using the ABC method [17, 25, 63]. First, ABC is a stochastic optimization paradigm, part of the assembly of swarm intelligence-based algorithms. In addition, it is capable to effectively optimize non-differential and non-continuous functions with continuous and/or discrete variables. Secondly, the strategy used to exploit possible solutions can avoid premature convergence. Thirdly, it can be used as a global optimizer which in effect can efficiently cope with highly complex and non-linear problems. Finally, it appears to be insensitive to initialization, regardless the location/distribution of the initial values.

### 3.1 Artificial Bee Colony (ABC) Method

The ABC method has been gaining increasing popularity during the last decade [17, 25, 30]. It involves  $SN$  food sources, the positions of which are represented by  $q$ -dimensional vectors:  $fs_i \in \mathfrak{R}^q$ . Different types of bees, are considered, i.e. employed and onlooker bees, as well as scout bees. Each employed bee is associated with a specific food source (i.e. there exist  $SN$  employed bees) and, typically, the number of onlooker bees is set equal to the number of employed bees; thus, the size of the colony is  $CS = 2SN$ . Food sources are randomly initialized and the employed bees produce candidate solutions according to the rule:

$$v_{ij} = fs_{ij} + \psi_{ij} (fs_{ij} - fs_{kj}) \quad (1 \leq i \leq SN) \quad (1)$$

where  $v_i = [v_{i1} \ v_{i2} \ \dots \ v_{iq}]^T$  is the position of the  $i$ th food source recommended by the corresponding employed bee,  $\psi_{ij}$  is randomly generated by a uniform distribution within the interval  $[-1, 1]$ ,  $j$  is a random integer in  $[1, q]$ , and  $k$  is a randomly generated integer in  $[1, SN]$ , which

must be different than  $i$ . Based on the evaluated fitness functions, a greedy selection between  $v_i$  and  $fs_i$  takes place. Hence, for each food source a probability is calculated:

$$Pr_i = \frac{Fitness_i}{\sum_{j=1}^{SN} Fitness_j} \quad (1 \leq i \leq SN) \quad (2)$$

where  $Fitness_i$  is the value of the fitness function for the  $i$ th food source. Based on these probabilities, each onlooker bee visits a food source according to a roulette wheel selection mechanism, and generates a candidate solution according to eq. (1). Once again, a greedy selection takes place. If the number of times a food source fails to be updated is greater than a predefined integer number, symbolized as  $Limit$ , the corresponding employed bee becomes a scout bee and commences searching for a new food source from scratch, by reinitializing its position.

In general, two properties decide the effectiveness of a swarm intelligence algorithm, namely exploitation and exploration. The former refers to the effective incorporation of old good solutions in the search for improved solutions. The latter concerns the ability of exploring different regions in the feature space. Implementation of eq. (1) along with the onlooker and scout bees' phases increases the exploration performance of the method [30]. On the other hand, exploitation capabilities may be compromised, as, due to the rule described by eq. (1), an old solution may move towards a randomly selected food source without any guarantee for its quality. To improve the ABC exploitation capability, Zhu and Kwong [63] modified eq. (1) as:

$$v_{ij} = fs_{ij} + \psi_{ij} (fs_{ij} - fs_{kj}) + \theta_{ij} (fs_{best,j} - fs_{ij}) \quad (3)$$

where  $fs_{best} = [fs_{best,1}, fs_{best,2}, \dots, fs_{best,q}]^T$  is the food source position with the best solution so far, and

$\theta_{ij}$  is a random number within the interval  $[0, \lambda]$ , with  $\lambda > 0$  chosen by trial and error.

However, implementation of eq. (3) is carried out only in terms of a randomly chosen dimension (i.e. the  $j$ th dimension). Therefore, while eq. (3) is expected to improve the exploitation performance, its effect may be weak. In order to address this difficulty and inspired by particle swarm optimization (PSO) (see Appendix B), we propose a modification in the learning condition (eq. (3)) by considering the following vector-based approach. The point-wise product between two vectors

$\mathbf{x} = [x_1, x_2, \dots, x_q]^T$  and  $\mathbf{y} = [y_1, y_2, \dots, y_q]^T$  is defined as:

$$\mathbf{x} \square \mathbf{y} = [x_1 y_1, x_2 y_2, \dots, x_q y_q]^T \quad (4)$$

Then, the vector-based modification of (3) is given by the subsequent rule:

$$\mathbf{v}_i = \mathbf{f}\mathbf{s}_i + \boldsymbol{\Psi}_i \square (\mathbf{f}\mathbf{s}_i - \mathbf{f}\mathbf{s}_k) + \boldsymbol{\Theta}_i \square (\mathbf{f}\mathbf{s}_{best} - \mathbf{f}\mathbf{s}_i) \quad (5)$$

with  $\boldsymbol{\Psi}_i = [\psi_{i1}, \psi_{i2}, \dots, \psi_{iq}]^T$ , where  $\psi_{ij}$  is a random number in  $[-1, 1]$ , and  $\boldsymbol{\Theta}_i = [\theta_{i1}, \theta_{i2}, \dots, \theta_{iq}]^T$

where  $\theta_{ij}$  is a random number in  $[\lambda, \lambda + 2]$  and  $\lambda \in [-1, 0]$  with a typical value  $\lambda = 0$ . Note that, according to eq. (5), both employed and onlooker bees attempt to update the position of a food source with respect to all dimensions of the vector. This, in turn, will effect movement of the food source  $\mathbf{f}\mathbf{s}_i$  towards the vectors  $\mathbf{f}\mathbf{s}_{best}$  and  $\mathbf{f}\mathbf{s}_k$ , ensuring an efficient exploitation of the feature space.

### 3.2 Structure of the Proposed Network

Before presenting the proposed network, a brief description of the Hermite polynomials is provided.

The Hermite polynomials emerge from the differential operation [6]:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} (e^{-x^2}) \quad (6)$$

with  $n$  being the polynomial order. They are orthogonal in  $(-\infty, \infty)$  with respect to the exponential

function  $e^{-x^2}$  [6]:

$$\langle H_n(x), H_r(x) \rangle = \int_{-\infty}^{+\infty} e^{-x^2} H_n(x) H_r(x) dx = \begin{cases} 0, & \text{if } n \neq r \\ 2^n n! \sqrt{\pi}, & \text{if } n = r \end{cases} \quad (7)$$

where  $\langle \square, \square \rangle$  stands for the inner product. The individual polynomials can be derived by the following recurrence relations:

$$H_0(x) = 1 \quad (8)$$

$$H_1(x) = 2x \quad (9)$$

$$H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x) \quad (10)$$

It can be easily shown that the  $n$ -th order polynomial can be expanded as truncated series,

$$H_n(x) = \sum_{k=0}^n \rho_{nk} x^k \quad (11)$$

where the coefficients are recurrently estimated as:

$$\rho_{nk} = \begin{cases} 2\rho_{n-1,k-1} - (k+1)\rho_{n-1,k+1}, & \text{if } k > 0 \\ -\rho_{n-1,k+1}, & \text{if } k = 0 \end{cases} \quad (12)$$

with  $\rho_{00} = 1, \rho_{10} = 0$ , and  $\rho_{11} = 2$ .

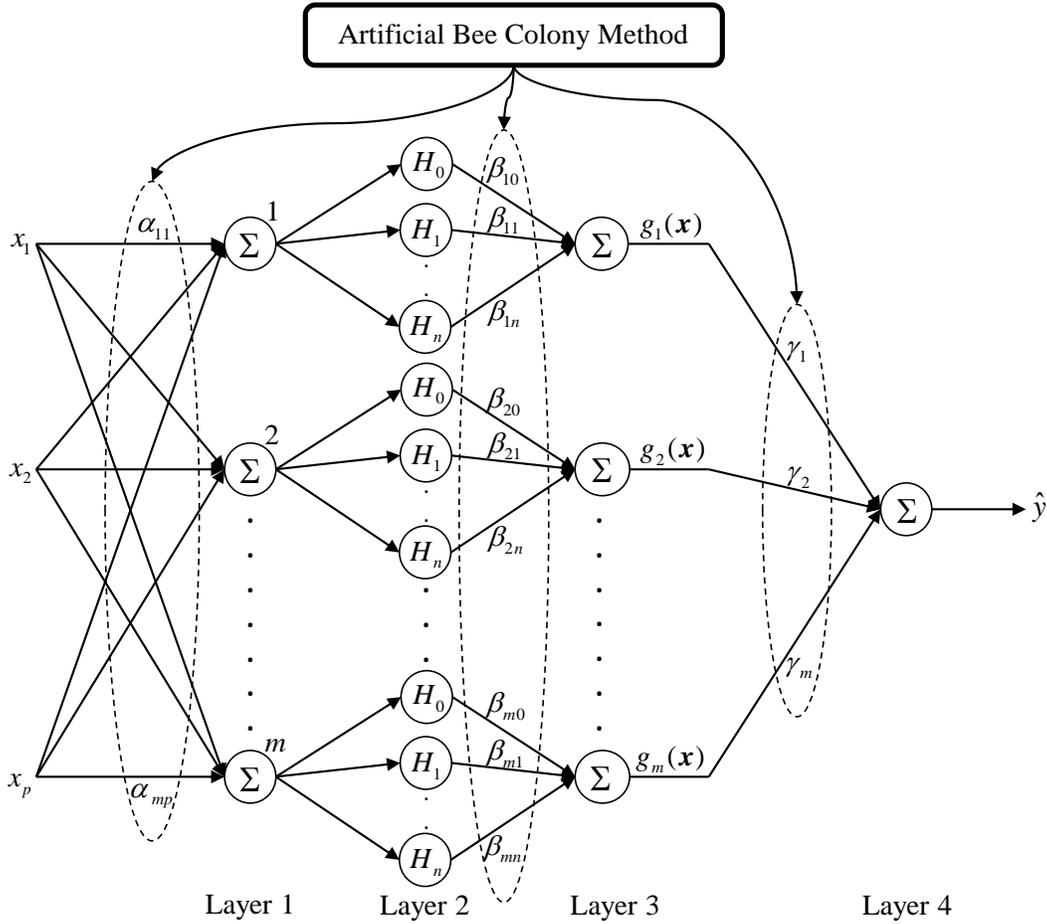
Utilization of Hermite polynomials in the design of neural networks appears to have certain advantages compared to other orthogonal polynomials like e.g. the Chebyshev, Legendre, and Laguerre polynomials which are orthogonal within bounded intervals. Using polynomial activation functions with bounded input ranges requires normalization of the corresponding input-side weight vectors which may negatively affect the efficiency of the PNN, particularly when the procedure involves regression and classification tasks [33]. In contrast, Hermite polynomials are orthogonal in the whole range of real numbers (i.e. in  $(-\infty, +\infty)$ ), a property that makes the learning process unconstrained and easier to handle.

The proposed network structure comprises four layers (Fig. 3). Layer 1 includes a set of  $m$  nodes, each of which sums up the weighted input variables. By defining the matrix of the input-side weight parameters as:

$$\mathbf{A} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1p} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2p} \\ \dots & \dots & \dots & \dots \\ \alpha_{m1} & \alpha_{m2} & \dots & \alpha_{mp} \end{bmatrix} = \begin{bmatrix} \mathbf{\alpha}_1^T \\ \mathbf{\alpha}_2^T \\ \dots \\ \mathbf{\alpha}_m^T \end{bmatrix} \quad (13)$$

the  $i$ -th node creates the inner product:

$$\langle \mathbf{\alpha}_i, \mathbf{x} \rangle = \mathbf{\alpha}_i^T \mathbf{x} = \sum_{j=1}^p \alpha_{ij} x_j \quad (1 \leq i \leq m; 1 \leq j \leq p) \quad (14)$$



**Figure 3.** Topology of the proposed Hermite polynomial neural network.

In Layer 2, each inner product enters a set of  $n$  neurons with activation functions the Hermite polynomials up to the  $n$ -th degree:  $H_\ell(\langle \mathbf{a}_i, \mathbf{x} \rangle)$  for  $\ell = 0, 1, \dots, n$ . The above polynomials are, then, weighted by the parameters  $\beta_{i\ell}$  ( $1 \leq i \leq m, 1 \leq \ell \leq n$ ) and processed in Layer 3 that consists of  $m$  summation nodes. Each of these nodes generates a truncated Hermite polynomial series of the corresponding inner product:

$$g_i(\mathbf{x}) = \sum_{\ell=0}^n \beta_{i\ell} H_\ell(\langle \mathbf{a}_i, \mathbf{x} \rangle) \quad (15)$$

The above functions are once more weighted using the parameters  $\gamma_i$  ( $1 \leq i \leq m$ ) and elaborated further in Layer 4 that finally provides the network's output:

$$\tilde{y} = \sum_{i=1}^m \gamma_i g_i(\mathbf{x}) = \sum_{i=1}^m \sum_{\ell=0}^n \gamma_i \beta_{i\ell} H_\ell(\langle \mathbf{a}_i, \mathbf{x} \rangle) \quad (16)$$

Next it is demonstrated that the proposed network can uniformly approximate any continuous function defined on a compact subset of the Euclidean space  $\mathfrak{R}^p$ .

**Definition 1** (Shin and Ghosh [51]). Let  $C(f)$  be the vector space of all continuous real valued functions defined on a compact subset  $\Theta$  of the Euclidean space  $\mathfrak{R}^p$ , and  $\mathbf{a} \in \mathfrak{R}^p$  be a parameter vector. Then, all functions of the form  $f(\langle \mathbf{x}, \mathbf{a} \rangle): \Theta \rightarrow \mathfrak{R}$ , with  $\mathbf{x} = [x_1, x_2, \dots, x_p]^T \in \Theta$ , are called ridge functions. In addition, a ridge polynomial is a ridge function, which is expanded as

$$P(\mathbf{x}) = \sum_{k=0}^n \sum_{i=1}^m \phi_{ik} \langle \mathbf{x}, \mathbf{a}_{ik} \rangle^k \quad (17)$$

with  $\mathbf{a}_{ik} \in \mathfrak{R}^p$ , and  $\phi_{ik} \in \mathfrak{R}$ .

Shin and Gosh [51] have shown that ridge polynomials are able to uniformly approximate any continuous function defined on a compact subset of the Euclidean space  $\mathfrak{R}^p$  to arbitrary accuracy.

It can be shown (Appendix A) that the network's output (i.e. eq. (16)) can be equivalently written in the form

$$\tilde{y} = \sum_{k=0}^n \sum_{i=1}^m \phi_{ik} \langle \mathbf{a}_i, \mathbf{x} \rangle^k \quad (18)$$

with

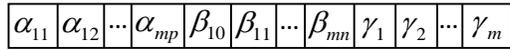
$$\phi_{ik} = \gamma_i \sum_{\ell=k}^n \beta_{i\ell} \rho_{\ell k} \quad (19)$$

It thus appears that the output  $\tilde{y}$  is a ridge polynomial and, therefore, the proposed network is able to approximate any continuous function defined on a compact subset of  $\mathfrak{R}^p$  to arbitrary accuracy.

Given a set of  $N$  input-output data pairs  $S = \{(\mathbf{x}_k, y_k) | \mathbf{x}_k \in \mathfrak{R}^p, y_k \in \mathfrak{R}, k = 1, 2, \dots, N\}$ , the performance of the network can be evaluated by the corresponding square error:

$$J_{SE} = \sum_{k=1}^N (y_k - \tilde{y}_k)^2 \quad (20)$$

where  $\tilde{y}_k$  is calculated in eq. (16) or eq. (18).



**Figure 4.** The structure of the food source.

Optimization of the network's parameters is achieved by employing the ABC method using the updating rule (eq. (5)) to minimize the above error. In each food source, the whole set of the weight parameters is encoded, yielding the string structure depicted in Fig. 4, where the dimension of the resulting search space is:

$$q = m(p + n + 2) \quad (21)$$

### 3.3 Computational Complexity

The computational complexity due to the addition and multiplication operations performed by the proposed network is now discussed. The starting point is eq. (14), for which the number of multiplications and additions is  $2p-1$ . Thus, the total number of operations involved in the calculation of all of the quantities  $\langle \alpha_i, \mathbf{x} \rangle$  for  $i=1,2,\dots,m$  is

$$T_1 = m(2p-1) \quad (22)$$

In addition, the multiplication and addition operations involved in the estimation of the sum

$\sum_{l=k}^n \beta_{il} \rho_{lk}$  (eq. (19)) are equal to  $n-(k-1)$  and  $n-k$ , respectively, and if the multiplication of the

parameter  $\gamma_i$ , is considered, then  $2(n-k+1)$  operations are required for determining the parameter

$\phi_{ik}$ . Thus, to calculate all parameters  $\phi_{ik}$  for  $k=0,1,\dots,n$  and  $i=1,2,\dots,m$ , the total number of

operations is

$$T_2 = m \sum_{k=0}^n (2(n-k+1)) = m(n+1)(n+2) \quad (23)$$

Having calculated the quantities  $\phi_{ik}$  ( $1 \leq i \leq m, 1 \leq k \leq n$ ) and  $\langle \alpha_i, \mathbf{x} \rangle$  ( $1 \leq i \leq m$ ), the numbers of

additions and multiplications for estimating the sum  $\sum_{k=0}^n \phi_{ik} \langle \alpha_i, \mathbf{x} \rangle^k$  are equal to  $n$  and  $\sum_{k=1}^n k$ ,

respectively. Thus, the number of operations involved in eq. (18) is

$$T_3 = m \left( n + \sum_{k=1}^n k \right) = m \frac{2n + n(n+1)}{2} \quad (24)$$

In view of (22)-(24), the total number of operations performed by the proposed network is,

$$T = T_1 + T_2 + T_3 = \frac{1}{2} \left( (4mp + 2m) + (3mn^2 + 9mn) \right) \quad (25)$$

Since,  $(4mp + 2m) \square mp$  and  $(3mn^2 + 9mn) \square mn^2$ , the computational complexity of the network is,

$$O(mp + mn^2) = O(m(n^2 + p)) \quad (26)$$

Generally, in order to effectively capture the nonlinearities of a given data set,  $n$  is not required to take large values, particularly as overfitting problems may also emerge.

With regard to the implementation of the ABC method, a most dominant effect is associated with the network's evaluation process in eq. (25)). Since, in each iteration of the ABC, the network is evaluated as many times as the number of employed plus the onlooker bees (scout bees form a subset of the employed bees' set), the overall computational complexity per iteration of the proposed methodology becomes  $O(CS m (n^2 + p))$ , where  $CS$  the colony size in the ABC algorithm.

## 4. Results and Discussion

### 4.1 Simulations

Based on the analysis presented in Section 2, the available data set includes  $N = 4148$  input-output data pairs of the form  $\{\mathbf{x}_k; y_k\}_{k=1}^N$  with  $\mathbf{x}_k = [x_{k1} \ x_{k2} \ \dots \ x_{k7}]^T$  and  $y_k \in \mathfrak{R}$ . Thus, the dimensionality of the feature space is  $p = 7$ . The ABC parameters were selected as  $CS = 20$ ,  $\lambda = 0$ , and  $Limit = 0.5 * SN * q$ , where  $q$  is given in (21).

To assess the effectiveness of the method, three more networks were considered for comparative purposes. First, the Hermite feedforward PNN developed by Ma and Khorasari in [33] which encompasses one hidden layer of  $m$  nodes (to avoid confusion, this Hermite network will be referred to hereafter as 'Hermite PNN' and the one developed in the present study as 'proposed network'). In the Hermite PNN, the input layer creates  $m$  linear combinations of the input variables using a set of weights. The  $i$ -th combination enters only the node with activation function the Hermite polynomial  $H_i(x)$  ( $1 \leq i \leq m$ ) and the network's output is calculated as the weighted sum of the above

polynomials (for a detailed analysis see [33]). Herein, all weight parameters of the Hermite PNN are optimized by minimizing the square error using the differential evolution (DE) algorithm [44] and the parameters of the DE have been chosen as:  $Np = 20$ ,  $Fr = 0.6$ , and  $Cr = 0.9$  (for a brief description of DE and the nomenclature used refer to Appendix B).

The second network tested is a Legendre PNN introduced by Rigos et al in [47]. As in the case of the Hermite PNN, it includes a hidden layer that comprises  $m$  nodes with activation functions Legendre polynomials up to the  $m$ -th polynomial order. The input layer produces linear combinations of the input variables, each of which enters only one polynomial activation function. Since Legendre polynomials are orthogonal in the interval  $[-1, 1]$ , their inputs are appropriately normalized by confining the input-side weight parameters within a predefined domain of values. All synaptic weights are optimized by a standard ABC algorithm that uses the learning rule of eq. (3). The parameter values for the ABC are selected as  $CS = 20$ , and  $Limit = 0.5 * SN * q$ .

The third network is a Radial Basis Function neural network (RBFNN), the parameters of which are estimated by employing particle swarm optimization (PSO) [9] (see also Appendix B). Using the notation of Appendix B, the parameters for the PSO were selected as:  $Np = 20$ ,  $\omega$  was randomly selected in  $[0.5, 1]$ ,  $\varphi_1 = \varphi_2 = 2$ , and  $L = 2$ .

The performance of the tested networks was evaluated by the root mean square error:

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N |y_k - \tilde{y}_k|^2} \quad (27)$$

In the experiments, data were randomly divided into a training set consisting of the 60%, and a testing set consisting of the remainder 40% of the original data set. The four neural networks were compared for numbers of hidden nodes varying between 2 and 10, i.e.  $m = 2, 3, \dots, 10$ . In addition, for the proposed network, the polynomial order was varying between 2 and 5, i.e.  $n = 2, 3, 4, \text{ and } 5$ .

Table 1. RMSEs and corresponding standard deviations obtained by the proposed network for various numbers of nodes ( $m$ ) and polynomial orders ( $n$ )

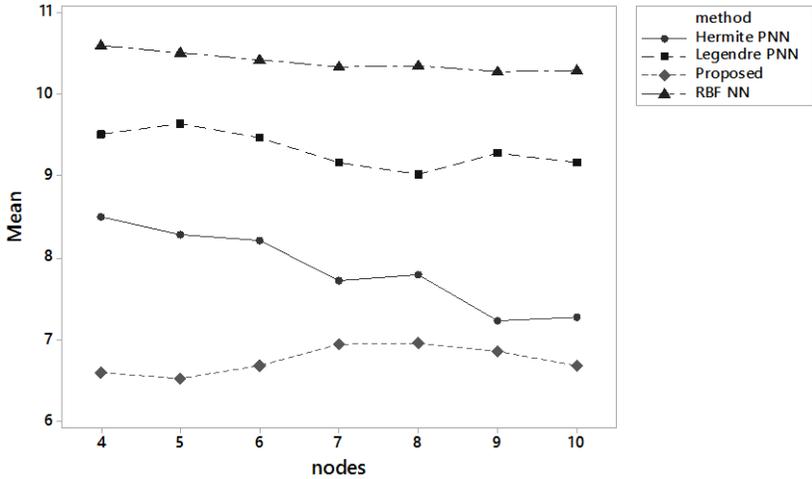
# Nodes ( $m$ )	$n = 2$	$n = 3$	$n = 4$	$n = 5$
<b>Training Data</b>				
2	6.5500±0.3404	6.0714±0.2319	6.1233±0.2614	6.1859±0.2813
3	6.1277±0.2702	5.6156±0.3072	5.7621±0.4877	5.8440±0.5054
4	6.0395±0.2316	5.5343±0.4975	5.5139±0.5943	5.6155±0.5650
5	6.1283±0.2354	5.5114±0.3671	5.4801±0.3927	5.7297±0.3942
6	6.0513±0.1949	5.6428±0.4227	5.6273±0.4044	5.8285±0.4809
7	6.1968±0.1956	5.8202±0.4059	5.8167±0.4375	5.7872±0.4472
8	6.1363±0.3143	5.8520±0.4272	5.7423±0.3681	5.9915±0.3088
9	6.3233±0.2751	5.8260±0.3679	5.7284±0.4465	5.9468±0.4257
10	6.3548±0.2406	5.9114±0.3471	5.6539±0.3900	5.9404±0.3598
<b>Testing Data</b>				
2	7.5409±0.2717	7.3954±0.3003	7.3077±0.4142	7.4799±0.2917
3	7.1779±0.4349	6.6997±0.4182	6.8686±0.5798	6.9410±0.6657
4	7.2024±0.3278	6.5705±0.6185	6.6016±0.7260	6.8036±0.4992
5	7.2014±0.3083	6.5704±0.4906	6.5290±0.3972	7.0524±0.3001
6	7.2011±0.2014	6.6770±0.5412	6.6779±0.4642	7.0057±0.5233
7	7.3183±0.2676	6.9225±0.4660	6.9391±0.4031	7.0949±0.3668
8	7.2598±0.3451	6.9553±0.4789	6.9552±0.3671	7.2300±0.3421
9	7.4904±0.2885	7.0209±0.4031	6.8551±0.5299	7.0951±0.2701
10	7.4395±0.3869	6.8815±0.3577	6.6856±0.5446	6.9202±0.3256

Table 2. RMSEs and the corresponding standard deviations obtained by the proposed network for various numbers of nodes ( $m$ )

# Nodes ( $m$ )	Hermite PNN	Legendre PNN	RBF NN
<b>Training Data</b>			
2	13.4106±2.8782	10.3176±0.8567	10.9265±0.4690
3	9.4066±1.8315	10.0382±1.0957	10.6796±0.0879
4	7.7167±1.9167	9.2737±0.8174	10.6412±0.0890
5	7.6683±1.3192	9.3614±1.3644	10.5746±0.0976
6	7.5587±1.3157	9.2451±1.5286	10.5318±0.0766
7	6.8030±0.7428	9.0974±1.2664	10.4765±0.1111
8	6.9952±0.8232	8.9176±1.5656	10.4345±0.1059
9	6.3589±0.4126	9.2072±1.1807	10.4458±0.1002
10	6.5115±0.5813	8.8851±1.3194	10.4378±0.0755
<b>Testing Data</b>			
2	14.1771±2.5615	10.4748±0.8061	10.9447±0.5937
3	9.8489±1.7116	10.1846±0.9512	10.6387±0.1498
4	8.4968±1.5980	9.5094±0.7539	10.5882±0.1592
5	8.2854±0.9858	9.6321±1.9384	10.5088±0.1275
6	8.2095±0.9066	9.4664±1.3467	10.4139±0.1557
7	7.7270±0.4408	9.1570±1.2250	10.3272±0.1559
8	7.7946±0.5844	9.0250±1.4221	10.3509±0.1410
9	7.2342±0.4188	9.2797±1.2280	10.2689±0.1693
10	7.2725±0.5427	9.1666±1.1610	10.2813±0.0807

For each network, each number of nodes and each polynomial order we evaluated all the networks by using 20 distinct runs with random initializations. Tables 1 and 2 report the resulting RMSEs and corresponding standard deviations.

The best results for all nodes  $m$  were obtained by the proposed network for polynomial orders  $n=3,4,$  and  $5$ . On the other hand, the Hermite PNN is competitive to the proposed network for  $n=2$ , whereas it clearly outperforms the Legendre PNN and the RBF NN which shows the worst performance in all cases. It appears that the most effective polynomial orders of the proposed network (showing the best performances in both the training and testing data) are orders  $n=3$  and  $n=4$ .



**Figure 5.** Mean values of the RMSEs (see Tables 1 and 2) for the testing data versus the number of nodes for the four networks.

A more rigorous statistical analysis of the performance of the four networks was conducted with respect to the testing data, only. The case  $n=4$  is considered for the proposed network, and values for the number of nodes as:  $m=4,5,\dots,10$ . Fig. 5 shows the average RMSEs for each method by node. The proposed network does not show a decreasing trend in the average RMSE, whereas the other three networks do (particularly the Hermite PNN).

The distribution of RMSE is summarized in the boxplots shown in Fig. 6. It is evident that the RMSE values exhibit different variability for the four networks studied with the RBFNN showing the lowest variability. The pooled standard deviations in RMSE were equal to 0.5146, for the proposed network, 0.8938 for the Hermite PNN, 1.2265 for the Legendre PNN and 0.1472 for the RBF NN.

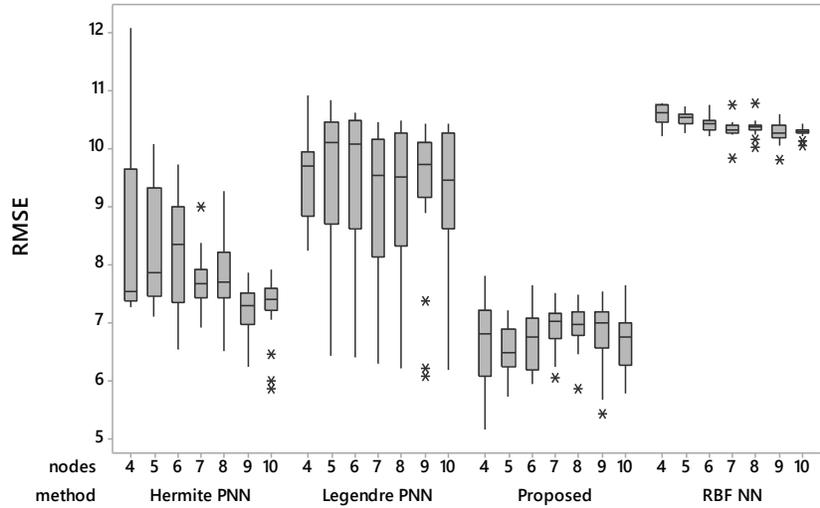


Figure 6. Boxplots of the RMSEs for the four methods.

A weighted two-way ANOVA model was fit to the data:

$$RMSE_{ij} = \mu + \nu_i + \tau_j + (\nu\tau)_{ij} + \varepsilon_{ij} \quad (i = 1, 2, 3, 4; j = 4, 5, \dots, 10),$$

where  $i$  denotes the method and  $j$  the number of nodes,  $\nu_i$  and  $\tau_j$  are the main effects for method and number of nodes, respectively,

and  $(\nu\tau)_{ij}$  is the method by node interaction effect. Heteroscedasticity due to differences in variance

between methods was modelled assuming that  $\varepsilon_{ij}$  were independent error terms that followed a

Gaussian distribution with mean 0 and variance  $\sigma_i^2$ . The weights used in the two-way ANOVA

were estimates of the inverse variance for each individual RMSE. Method-specific variances were

estimated by pooling over the seven node levels for each method and equaled to the square of the

pooled standard deviations given in the previous paragraph (each estimate is associated with 133 degrees of freedom). The weighted two-way ANOVA results are provided in Table 3.

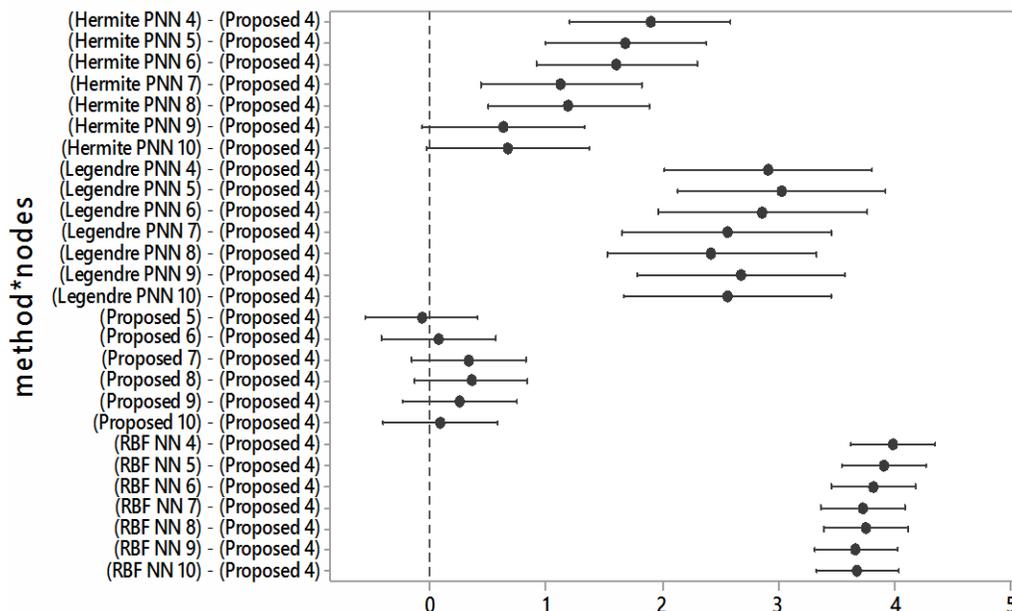
Table 3. Analysis of variance (ANOVA)

Source	Degrees of Freedom (DF)	Adj SS	Adj MS	F-Value	P-Value
Method	3	7400.60	2466.87	2448.46	0.000
Nodes	6	21.15	3.52	3.50	0.002
Method*Nodes	18	54.66	3.04	3.01	0.000
Error	532	536.00	1.01		
Total	559	8069.25			

The highly statistically significant interaction effects ( $p\text{-value} < 0.001$ ) confirm that the effect of increasing the number of nodes is dependent on the method used (see also Fig. 5). It also shows that differences between methods depend on the number of nodes used and, thus, we cannot give a single estimate of the mean RMSE differences between two methods after controlling for the number of nodes. Instead, a multiple comparison procedure was carried out as a follow-up where the mean RMSE of the proposed network with  $m=4$  nodes (the simplest computationally) was compared to all others. A Dunnett multiple comparison procedure yielded simultaneous 95% confidence intervals for the mean difference in RMSE between the proposed method with  $m=4$  nodes and all other methods by nodes combinations (Fig. 7).

On the basis of these results, it is suggested that there is: (1) no statistically significant effect for the proposed NN when using more than  $m=4$  nodes, i.e. there is no significant gain in increasing the number of nodes above 4; (2) clear superiority of the proposed method when using  $m=4$  nodes over the Legendre PNN and RBF NN methods irrespective of their number of nodes (all adjusted  $p$ -values associated with the comparisons were  $< 0.001$ ); (3) superiority of the proposed network (4 nodes) over the Hermite PNN with  $< 9$  nodes (all adjusted  $p$ -values  $< 0.001$ ); (4) a marginally significant difference in average RMSE in favor of the Proposed method (4 nodes) over the Hermite

PNN with 10 nodes (estimated difference = 0.671, adjusted 95% confidence interval (-0.023, 1.365), adjusted p-value = 0.067); and (5) a non-significant difference in average RMSE in favor of our Proposed method (4 nodes) over the Hermite PNN with 9 nodes (estimated difference = 0.633, adjusted 95% confidence interval (-0.062, 1.327), adjusted p-value = 0.107).



**Figure 7.** Dunnett simultaneous 95% CIs for mean difference in RMSE between all method by node combinations (illustrated as method followed by number of nodes) and the proposed method with 4 nodes (control mean). If an interval does not contain zero, the corresponding mean is significantly different from the control mean.

In summary, the proposed network performs efficiently for a small number of nodes (i.e.  $m = 4, 5$ ), while it appears to reasonably model the nonlinearities of the problem utilizing a small polynomial orders (i.e.  $n = 3, 4$ ). Finally, in most of the simulation cases, it significantly outperforms the other networks.

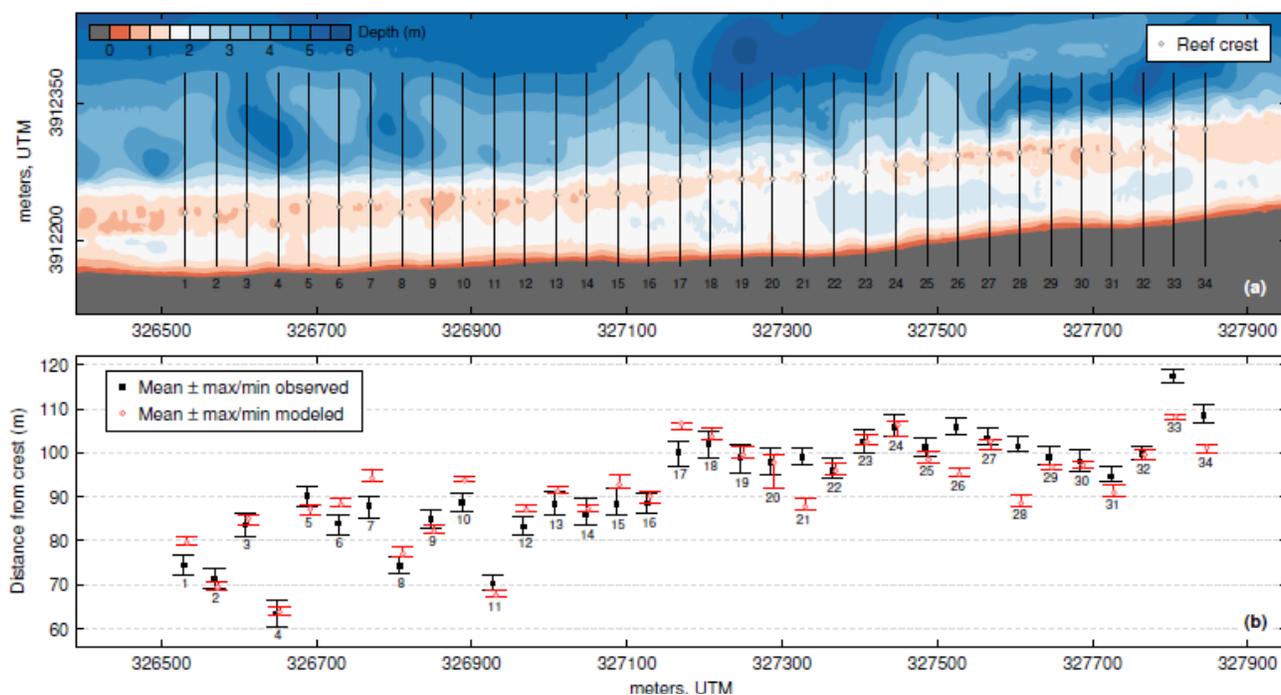
#### ***4.2 Shoreline Realignment: Observations and Model Results***

In the previous section it was shown that the proposed network, and particularly its 4<sup>th</sup> polynomial order, can model more efficiently shoreline realignment at the reef-fronted Ammoudara beach than all other neural networks tested. In this section, the ability of the proposed methodology to deal with the non-linearities and predict actual shoreline realignment on the basis of a limited number of environmental variables in this complex nearshore system is discussed.

Concerning the spatial characteristics and their variability of the modeled beach the following can be observed. The fronting reef shows a complex architecture and large spatial variability. The distance  $y$  of the reef crest from the shoreline and its depth  $d$  generally increase towards the east (Fig. 8). The reef width also varies, being greater towards the western and eastern margins of the monitored area, whereas there are also areas where the continuity of the reef is broken, forming reef inlets (Figs. 1 and 8) with the main inlet being approximately 130 m wide (at  $x$  about 327300-327450, cross-sections 20-24). The cross-shore shoreline position also showed significant variability in relation to the fronting reef along the 34 selected cross-sections during the 10-month monitoring period (Fig. 7); changes in  $y$  (i.e. differences between the minimum and maximum  $y$  at a cross-section) were found to range between 3.1 and 6.7 m. Main areas of increased  $y$  variability are associated with the reef inlets, with the variability being greater at the shoreline behind and to the west of the main reef inlet (at profiles 14-20, Fig. 8). In these areas, previous research has shown that complex, non-linear wave induced flows dominate [59].

RMSEs between the observed and the modeled distances ( $y$ ) between the reef crest and the shoreline were estimated to be less than 6 and 7 m for the training and testing data sets, respectively (Table 1), which are considered reasonable if the complexity of the domain and the small number of input variables utilized are taken into account. Model efficiency improves when cross-sections associated with inlets and/or particular wide sections of the reef are not considered. For example, in

the case shown in Fig. 8, RMSEs between modeled and observed shoreline position are 4.67 and 5.48 m for the training and testing experiments, respectively. When cross-sections that are associated with reef inlets or the particularly wide section of the reef at the eastern margin of the beach (profiles 20, 21, 22, 26, 33, 34) where field observations and modeling have shown particular complex across-reef flows are not considered, RMSE decreases to 4 m which is close to the lower end of the minimum cross-shore shoreline change (3.15 m) detected in Ammoudara during the monitoring period; this is a reasonable prediction considering the complexity of the domain. It is also interesting to note that the relative trend of the modeled and observed shoreline positions changes to the east of the main inlet (to the east of profiles 24: here, the model appears to underestimate shoreline retreat, i.e. its range of predictions for  $y$  are generally lower than those observed).



**Figure 8.** Example run showing the comparison between observed and modeled ranges between the minimum and maximum reef crest – shoreline distances during the monitoring period

Part of the difference found between the observed and modeled shorelines may be related to the nature/characteristics of the input and output variables used in the proposed network. Regarding the inputs, the morphological variables used (i.e. the  $d$ ,  $w$ ,  $\omega_1$ , and  $\omega_2$  see Fig. 2) form only geometric approximations of the highly uneven cross-sections of the natural reef. Secondly, these variables have been selected due to their significant control on wave breaking and energy transmission over the reef and, thus, the nearshore hydrodynamics [12]. However, as submerged ‘hard’ reefs have fixed crest levels, crest submergence ( $d$ ) variability due to the astronomical/baric tides, influences these processes (e.g. [21]) and, ultimately, the patterns of shoreline change behind the reef. In the present study, crest submergence ( $d$ ) has been constant, representing conditions at the mean sea level; therefore, although Ammoudara is a microtidal beach, with an astronomical tidal range of up to 0.10 - 0.15 m tidal oscillations over the very shallow water depths over the crest reef (Fig. 8) may introduce noticeable effects on the nearshore wave processes. Thirdly, the wave forcing used did not contain detailed directional information being based on offshore, deepwater records. Although previous research has shown good correlation between the buoy’s wave parameter trends and those recorded at the inshore waters (at 10 m water depths) in terms of wave heights and periods [59], it is likely to have been wave directional changes that could have affected back-reef hydrodynamics and, ultimately, beach shoreline realignment/rotation. Finally, data on sediment texture which is a significant control on beach morphodynamics have not been available (and for the wave direction rarely are) to be introduced as an input variable to the neural network.

On the other hand, the data forming the output of the network ( $y$ ) are not only very accurate (the shoreline position, and, thus, the cross-shore distance to the fixed reef crest ( $w$ ) recorded to an accuracy of  $< 0.5$  m, see Section 2.2), but also represent the integrated beach response to the nearshore hydrodynamics and sediment dynamics and their interaction with the actual, complex reef architecture. Therefore, differences between the observed and modeled shorelines at both the

training and testing exercises are expected, particularly for a nearshore environment of such complexity. Accuracy of the neural network predictions is expected to increase substantially in beaches protected by engineered reefs as, in these cases, the offshore structures have simpler, designed cross-sections (e.g. [2,23]).

Under a changing climate, large changes are projected for the global shoreline with most of the changes associated with beaches, the vast majority of which is expected to be under severe erosion [5, 7, 15, 20, 62]. Different options should be considered to protect the beaches, including offshore submerged structures (breakwaters), the functional design of which is still developing [2]. Neural networks that can model shoreline evolution on the basis of a relatively few and easily obtained environmental variables could be efficient tools to test designs of technical responses to beach realignment/erosion under changing climatic conditions. In this context, the proposed neural network, which has been able to reasonably predict shoreline realignment at a very challenging beach that is fronted by a highly irregular reef, may present a promising way forward.

## **5. Summary and Conclusions**

The study forms an investigation on the potential of a novel Hermite polynomial neural network to model shoreline realignment along an urban beach fronted by a highly irregular beachrock reef which introduces significant non-linearity in the nearshore hydrodynamics and morphodynamics, on the basis of a small number of input variables related to reef morphology and the wave forcing. The main functionality of the proposed network is to generate Hermite truncated polynomial series of linear combinations of the input variables and the network's output is calculated as the weighted sum of the above truncated series. It is analytically shown that the proposed network can approximate any continuous function defined on a compact set of the multidimensional Euclidean space to arbitrary accuracy and the network is optimized in terms of a modified artificial bee colony method. It is

shown that the proposed network can model shoreline realignment reasonably well and generally much better than the 3 other neural networks tested, i.e. a Hermite PNN, a Legendre feedforward PNN and a Radial Basis Function NN. It is also shown that there is no significant gain in increasing the number of nodes above 4; at this setting, the proposed network achieves its best performance and is clearly superior to all other networks tested.

Concerning the model efficiency, RMSEs between the observed and modeled shorelines (i.e. their proxies  $y$  - the distance between the reef crest and the shoreline) are estimated to be less than 6 and 7 m for the training and testing data sets, respectively, in all simulations. Model efficiency improves when cross-sections associated with the reef inlets and/or particular wide sections of the reef are not considered; in this case, RMSE decreases to 4 m, a reasonable prediction considering the complexity of the domain.

Part of the difference found between the observed and modeled by the proposed method shorelines may be related to the input and output variables used. Regarding the inputs, the morphological variables related to the reef form only geometric approximations of the highly uneven cross-sections of the natural reef, and there are no input variables related to significant morphodynamic controls such as tidal oscillation, wave direction and beach sediment texture. On the other hand, the data forming the output of the network ( $y$ ) are not only very accurate (accuracy of  $< 0.5$  m), but also represent the integrated beach response to the nearshore hydrodynamics and sediment dynamics and their interaction with the actual, complex reef architecture. Therefore, differences between the observed and modeled shorelines are expected with the accuracy of the method predictions expected to increase substantially in beaches protected by engineered reefs with simpler, designed cross-sections. In this context, the proposed neural network, which has been able to reasonably model shoreline realignment at a very challenging coastal environment may present a promising way forward.

## Acknowledgements

This research has been co- financed in 85% by the EEA GRANTS, 2009-2014, and 15% by the Public Investments Programme (PIP) of the Hellenic Republic. Project title ERABEACH: “Recording of and Technical Responses to Coastal Erosion of Touristic Aegean island beaches”.

## Appendix A. Proof of eq. (18)

In view of eq. (11), the quantity  $H_\ell(\langle \mathbf{a}_i, \mathbf{x} \rangle)$  can be precisely evaluated by the following relation,

$$H_\ell(\langle \mathbf{a}_i, \mathbf{x} \rangle) = \sum_{k=0}^{\ell} \rho_{\ell k} \langle \mathbf{a}_i, \mathbf{x} \rangle^k \quad (\text{A.1})$$

Therefore, the eq. (15) is modified as

$$g_i(\mathbf{x}) = \sum_{\ell=0}^n \beta_{i\ell} \sum_{k=0}^{\ell} \rho_{\ell k} \langle \mathbf{a}_i, \mathbf{x} \rangle^k = \sum_{\ell=0}^n \sum_{k=0}^{\ell} \beta_{i\ell} \rho_{\ell k} \langle \mathbf{a}_i, \mathbf{x} \rangle^k \quad (\text{A.2})$$

By expanding  $g_i(\mathbf{x})$  using the individual terms for an increasing order of the index  $\ell$ , we get

$$\begin{aligned} g_i(\mathbf{x}) = & \beta_{i0} \rho_{00} \\ & + \beta_{i1} \rho_{10} + \beta_{i1} \rho_{11} \langle \mathbf{a}_i, \mathbf{x} \rangle \\ & + \beta_{i2} \rho_{20} + \beta_{i2} \rho_{21} \langle \mathbf{a}_i, \mathbf{x} \rangle + \beta_{i2} \rho_{22} \langle \mathbf{a}_i, \mathbf{x} \rangle^2 \\ & \dots \\ & + \beta_{in} \rho_{n0} + \beta_{in} \rho_{n1} \langle \mathbf{a}_i, \mathbf{x} \rangle + \beta_{in} \rho_{n2} \langle \mathbf{a}_i, \mathbf{x} \rangle^2 + \dots + \beta_{in} \rho_{nn} \langle \mathbf{a}_i, \mathbf{x} \rangle^n \end{aligned}$$

which implies that,

$$g_i(\mathbf{x}) = \sum_{\ell=0}^n \beta_{i\ell} \rho_{\ell 0} + \sum_{\ell=1}^n \beta_{i\ell} \rho_{\ell 1} \langle \mathbf{a}_i, \mathbf{x} \rangle + \sum_{\ell=2}^n \beta_{i\ell} \rho_{\ell 2} \langle \mathbf{a}_i, \mathbf{x} \rangle^2 + \dots + \beta_{in} \rho_{nn} \langle \mathbf{a}_i, \mathbf{x} \rangle^n \quad (\text{A.3})$$

and thus,

$$g_i(\mathbf{x}) = \sum_{k=0}^n \sum_{\ell=k}^n \beta_{i\ell} \rho_{\ell k} \langle \mathbf{a}_i, \mathbf{x} \rangle^k \quad (\text{A.4})$$

In the above equation the term  $\langle \mathbf{a}_i, \mathbf{x} \rangle^k$  does not depend on index  $\ell$ . Therefore, by setting

$$b_{ik} = \sum_{\ell=k}^n \beta_{i\ell} \rho_{\ell k} \quad (\text{A.5})$$

we arrive at

$$g_i(\mathbf{x}) = \sum_{k=0}^n b_{ik} \langle \mathbf{a}_i, \mathbf{x} \rangle^k \quad (\text{A.6})$$

In view of eq. (16), the network's output comes in the form,

$$\tilde{y} = \sum_{i=1}^m \gamma_i g_i(\mathbf{x}) = \sum_{i=1}^m \sum_{k=0}^n \gamma_i b_{ik} \langle \mathbf{a}_i, \mathbf{x} \rangle^k \quad (\text{A.7})$$

Taking into account eq. (A.5) and choosing,

$$\phi_{ik} = \gamma_i b_{ik} = \gamma_i \sum_{\ell=k}^n \beta_{i\ell} \rho_{\ell k} \quad (\text{A.8})$$

Finally, by substituting (A.8) and rearranging the sums, the eq. (A.7) is identical to eq. (18).

## Appendix B

### B.1 Differential Evolution

The differential evolution (DE) involves a population of  $Np$  individuals that search the space for an optimal solution [44, 54]. Each individual is a  $q$ -dimensional vector  $\mathbf{d}_i = [d_{i1}, d_{i2}, \dots, d_{iq}]^T$   $i=1, 2, \dots, Np$ , which is randomly initialized. It employs three basic learning phases namely, mutation, crossover, and selection. In this paper, to perform the mutation phase, we randomly select two distinct individuals  $\mathbf{d}_a, \mathbf{d}_b$  different from  $\mathbf{d}_i$ ; the individual  $\mathbf{d}_{best}$  that corresponds to the best solution found so far; and a random number  $Fr \in (0, 1)$ . Then, the mutation vector

$\boldsymbol{\delta}_i = [\delta_{i1}, \delta_{i2}, \dots, \delta_{iq}]^T$  is estimated according to the next rule [38],

$$\delta_i = d_{best} + Fr(d_a - d_b) \quad (\text{B.1})$$

The crossover phase is typically used to increase the diversity of the population by generating a trial

vector  $\mathbf{v}_i = [v_{i1}, v_{i2}, \dots, v_{iq}]^T$  as follows,

$$v_{ij} = \begin{cases} \delta_{ij}, & \text{if } (rand \leq Cr) \vee j = j_0 \\ d_{ij}, & \text{otherwise} \end{cases} \quad (\text{B.2})$$

where  $rand$  is a random number in  $(0, 1)$ ,  $Cr \in (0, 1)$  is the crossover factor, and  $j_0$  is a random integer in the interval  $[1, q]$ .

Finally, in the selection phase, if  $\mathbf{v}_i$  comes with a better solution than the  $\mathbf{d}_i$  then it is transferred to the next iteration.

## ***B.2 Particle Swarm Optimization***

The particle swarm optimization (PSO) involves a swarm of  $Np$  vectors  $\mathbf{p}_i \in \mathfrak{R}^q$  ( $1 \leq i \leq Np$ ), called particles [9, 11, 26]. Each particle is assigned a velocity  $\mathbf{h}_i \in \mathfrak{R}^q$ . The positions with the best solution obtained so far by the particle  $\mathbf{p}_i$  and by all particles are respectively denoted as  $\mathbf{p}_i^{best}(t)$  and  $\mathbf{p}_{best}(t)$ . Then, the velocity is calculated as [26],

$$\mathbf{h}_i(t+1) = \omega \mathbf{h}_i(t) + \varphi_1 U(0,1) \square (\mathbf{p}_i^{best}(t) - \mathbf{p}_i(t)) + \varphi_2 U(0,1) \square (\mathbf{p}_{best}(t) - \mathbf{p}_i(t)) \quad (\text{B.3})$$

where  $\square$  is the vector point-wise product defined in eq. (4),  $U(0,1)$  is a vector with elements randomly generated in  $[0, 1]$ ;  $\omega$ ,  $\varphi_1$ , and  $\varphi_2$  are positive constant numbers called the inertia, cognitive and social parameter, respectively. Finally, the position of each particle is updated as,

$$\mathbf{p}_i(t+1) = \mathbf{p}_i(t) + \mathbf{h}_i(t+1) \quad (\text{B.4})$$

The elements of the particle are confined in the range [38],

$$p_j^{\min} \leq p_{ij} \leq p_j^{\max} \quad (\text{B.5})$$

where  $p_j^{\min}$  and  $p_j^{\max}$  are the boundaries of the domain of values in the  $j$ th dimension of the particles' search space. As stated in [37, 58], the above version of the PSO lacks a mechanism responsible to control the magnitude of the velocities, which fosters the danger of swarm explosion and divergence. Therefore, the velocities are restricted in the intervals  $-H_j^{\max} \leq h_{ij} \leq H_j^{\max}$  with

$$H_j^{\max} = (p_j^{\max} - p_j^{\min})/L \quad (\text{B.6})$$

with  $L$  being a positive integer [37, 58].

## References

- [1] S. G. J Aarninkhof, I. L. Turner, T. D. T. Dronkers, M. Caljouw, L. Nipius, A video-based technique for mapping intertidal beach bathymetry, *Coastal Engineering* 49(4) (2003) 275–289.
- [2] A. S. Ahmadian, R.R. Simons, Estimation of nearshore wave transmission for submerged breakwaters using a data-driven predictive model, *Neural Computing and Applications* (2016) 1–15, DOI: 10.1007/s00521-016-2587-y.
- [3] G. Alexandrakis, S. E. Poulos, G. Ghionis, G. Leivaditis, A morphological study of a reef with beachrock characteristics, in association with the recent evolution of the Ammoudara beach zone (Heraklion, Crete), *Bulletin of the Geological Society of Greece* 39(3) (2006) 146-155.
- [4] G. Alexandrakis, G. Ghionis, S. E. Poulos, The effect of beach rock formation on the morphological evolution of a beach. The case study of an Eastern Mediterranean beach: Ammoudara, Greece, *Journal of Coastal Research*, SI 69 (2013) 47-59.
- [5] K. Allenbach, I. Garonna, C. Herold, I. Monioudi, G. Giuliani, A. Lehmann, A.F. Velegrakis, Black Sea beach vulnerability to sea level rise, *Environmental Science & Policy* 46 (2015) 95–109.
- [6] G. E. Andrews, R. Askey, R. Roy, *Special Functions*, Cambridge University Press, UK, 2000.
- [7] C. Armaroli C, E. Grotoli, M. D. Harley, P. Ciavola, Beach morphodynamics and types of foredune erosion generated by storms along the Emilia-Romagna coastline, Italy, *Geomorphology* 199 (2013) 22–35.
- [8] C. E. Blenkinsopp, J. R. Chaplin, The effect of relative crest submergence on wave breaking over submerged slopes, *Coastal Engineering* 55 (2008) 967 – 974.
- [9] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation* 6 (1) (2002) 58-73.
- [10] F. Dolique, E. J. Anthony, Seasonal-term profile changes of sandy pocket beaches affected by Amazon-Derived Mud, Cayenne, French Guiana, *Journal of Coastal Research* 21 (6) (2005) 1195-1202.
- [11] R. C. Eberhart, Y. Shi, Tracking and optimizing dynamic systems with particle swarms, in: *Proceedings of the*

IEEE Congress on Evolutionary Computation (2001) 94-100, Seoul, Korea.

- [12] H. C. Friebel, L.E. Harris, A new wave transmission coefficient model for submerged breakwaters, in: Proceedings of the 29th International Conference on Coastal Engineering (2004) 19-24, Lisbon, Portugal.
- [13] S. L. Gallop, C. Bosserelle, I. Eliot, C. B. Pattiaratchi, The influence of limestone reefs on storm erosion and recovery of a perched beach, *Continental Shelf Research* 47 (2012) 16-27.
- [14] S. L. Gallop, C. Bosserelle, I. Eliot, C. B. Pattiaratchi, The influence of coastal reefs on spatial variability in seasonal sand fluxes, *Marine Geology* 344 (2013) 132-143.
- [15] S. L. Gallop, M. Collins, C. Pattiaratchi, M. Eliot, C. Bosserelle, M. Ghisalberti, L. B. Collins, I. Eliot, P. L. A. Erfemeijer, P. Larcombe, I. Marigomez, T. Stul, D. White, Challenges in transferring knowledge between scales in coastal sediment dynamics, *Frontiers in Marine Science - Coastal Ocean Processes* 2 (2015), Article 82.
- [16] M. R. Gourlay, Wave transformation on a coral reef, *Coastal Engineering* 23 (1-2) (1994) 17-42.
- [17] H. Habbi, Y. Boudouaoui, D. Karaboga, C. Ozturk, Self-generated fuzzy systems design using artificial bee colony optimization, *Information Sciences* 295 (2015) 145-159.
- [18] M. D. Harley, I. L. Turner, A. D. Short, New insights into embayed beach rotation: The importance of wave exposure and cross-shore processes, *Journal of Geophysical Research* 120 (8) (2015) 16.
- [19] M.R. Hashemi, Z. Ghadampour, S.P. Neill, Using an artificial neural network to model seasonal changes in beach profiles, *Ocean Engineering* 37 (2010) 1345-1356.
- [20] J. Hinkel, R. J. Nicholls, R. S. J. Tol, Z. B. Wang, J. M. Hamilton, G. Boot, A. T. Vafeidis, L. McFadden, A. Ganopolski, R. J. T. Klein, A global analysis of erosion of sandy beaches 1207 and sea-level rise: An application of DIVA, *Global and Planetary Change* 111 (2013) 150-158.
- [21] R. A. Holman, J. Stanley, The history and technical capabilities of Argus, *Coastal Engineering* 54(6-7) (2007) 477-491.
- [22] W. Huang, S.-K. Oh, W. Pedrycz, Design of hybrid radial basis function neural networks (HRBFNNs) realized with the aid of hybridization of fuzzy clustering method (FCM) and polynomial neural networks (PNNs), *Neural Networks* 60 (2014), 166-181.
- [23] G. Iglesias, I. López, A. Castro, R. Carballo (2009). Neural network modelling of planform geometry of headland-bay beaches, *Geomorphology*, 103, 577-587,
- [24] A. G. Ivakhnenko, Polynomial theory of complex systems, *IEEE Transactions on Systems, Man and Cybernetics* 1 (4) (1971) 364-378.
- [25] D. Karaboga, B. Bastrurk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization* 39 (2007) 459-471.
- [26] J. Kennedy, R. C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, Berlin, 2001.
- [27] T. Kerh, H. Lu, R. Saunders, Shoreline change estimation from survey image coordinates and neural network approximation, *International Journal of Civil, Environmental, Structural, Construction and Architectural Engineering* 8(4) (2014 ) 381-386.
- [28] A. H. F Klein., F. L. Benedet, D. H. Schumacher, Seasonal-term beach rotation processes in distinct headland bay systems, *Journal of Coastal Research* 18 (3) (2002) 442-458.
- [29] T.-T. Lee and J.-T. Jeng, The Chebyshev-polynomials-based unified model neural networks for function

- approximation, *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics* 28 (6) (1998) 925-935.
- [30] Z. Li, W. Wang, Y. Yan, Z. Li, PS-ABC: A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems, *Expert Systems with Applications* 42 (2015) 8881–8895.
- [31] F. Liu, J. Wang, Fluctuation prediction of stock market index by Legendre neural network with random time strength function, *Neurocomputing* 83 (2012) 12–21.
- [32] R. J. Lowe, C. Hart, C. B. Pattiaratchi, Morphological constraints to wave-driven circulation in coastal reef-lagoon systems: A numerical study, *Journal of Geophysical Research* 115 (2010) C09021.
- [33] L. Ma, K. Khorasani, Constructive feedforward neural networks using Hermite polynomial activation functions, *IEEE Transactions on Neural Networks* 16 (4) (2005) 821–833.
- [34] S. Mall, S. Chakraverty, Application of Legendre Neural Network for solving ordinary differential equations, *Applied Soft Computing* 43 (2016) 347–356.
- [35] I. Maric, Optimization of self-organizing polynomial neural networks, *Expert Systems with Applications* 40 (2013) 4528–4538.
- [36] B. Naik, J. Nayak, H. S. Behera, A. Abraham, A self adaptive harmony search based functional link higher order ANN for non-linear data classification, *Neurocomputing* 179 (2016) 69–87.
- [37] S.-K. Oh, H.-J. Jang, W. Pedrycz, A comparative experimental study of type-1/type-2 fuzzy cascade controller based on genetic algorithms and particle swarm optimization, *Expert Systems with Applications* 38 (2011) 11217–11229.
- [38] S.-K. Oh, W.-D. Kim, W. Pedrycz and S.-C. Joo, Design of K-means clustering-based polynomial radial basis function neural networks (pRBF NNs) realized with the aid of particle swarm optimization and differential evolution, *Neurocomputing* 78 (1) (2012), 121-132.
- [39] Y.-H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Reading MA: Addison-Wesley, 1989.
- [40] B.-J. Park, W. Pedrycz, S.-K. Oh, Fuzzy polynomial neural networks: hybrid architectures of fuzzy modeling, *IEEE Transactions on Fuzzy Systems* 10 (5) (2002) 607-621.
- [41] J. C. Patra, A. C. Kot, Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* 32 (4) (2002) 505-511.
- [42] J. C. Patra, P. K. Meher, G. Chakraborty, Nonlinear channel equalization for wireless communication systems using Legendre neural networks, *Signal Processing* 89 (2009) 2251–2262.
- [43] N. G. Plant, S.G.J. Aarninkhof, I.L. Turner, K.S. Kingston, The performance of shoreline detection models applied to video imagery, *Journal of Coastal Research* 23(3) (2007) 658–670.
- [44] K. V. Price, R. M. Storn, J. A. Lampinen, *Differential evolution: a practical approach to global optimization*, Springer-Verlag, 2005.
- [45] S. Purwar, I. N. Kar, A.N. Jha, On-line system identification of complex systems using Chebyshev neural networks, *Applied Soft Computing* 7 (2007) 364–372.
- [46] R. Ranasinghe, R. McLoughlan, A. Seasonal, G. Symonds, The southern oscillation index, *Wave Climate and Beach Rotation, Marine Geology*, 204 (3-4) (2004) 273-287.
- [47] A. Rigos, G. E. Tsekouras, A. Chatzipavlis, A. F. Velegrakis, Modeling beach rotation using a novel Legendre polynomial feedforward neural network trained by nonlinear constrained optimization, in: *Proceedings of the*

- 12th IFIP International Conference on Artificial Intelligence Applications and Innovations (2016) 167-179, Thessaloniki, Greece.
- [48] A. Rigos, G. E. Tsekouras, M. I. Vousdoukas, A. Chatzipavlis, A. F. Velegrakis, A Chebyshev polynomial radial basis function neural network for automated shoreline extraction from coastal imagery, *Integrated Computer-Aided Engineering* 23 (2016) 141–160.
- [49] A. Rigos, M. I. Vousdoukas, G. E. Tsekouras, O. P. Andreadis, A. F. Velegrakis, On the systematic implementation of artificial neural networks in the classification of variance images and shoreline extraction. *Fresenius Environmental Bulletin* 23(2014) 2677-2686.
- [50] A. Sancho-García, J. Guillén, E. Ojeda, Storm-induced readjustment of an embayed beach after modification by protection works, *Geo-Marine Letters* 33 (2013) 159-172.
- [51] Y. Shin, J. Gosh, Ridge polynomial networks, *IEEE Transactions on Neural Networks* 6 (3) (1995) 610-622.
- [52] A. D. Short, G. Masselink, Embayed and structurally controlled beaches, in: A. D. Short (Ed.), *Handbook of Beach and Shoreface Morphodynamics*, John Wiley and Sons Ltd., Chichester, 1999, pp. 230-250.
- [53] A. D. Short, A. C. Trembanis, Decadal Scale Patterns of Beach Oscillation and Rotation: Narabeen Beach, Australia, *Time Series PCA and Wavelet Analysis, Journal of Coastal Research* 20 (2) (2004) 523-532.
- [54] R. Storn, K. Price, Differential evolution- a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* 11 (4) (1997) 341-359.
- [55] T. Thomas, M. R. Phillips, A. T. Williams, R. E. Jenkins, Decadal timescale beach rotation: Gale climate and offshore island influences, *Geomorphology* 135 (2011) 97-107.
- [56] T. Thomas, M. R. Phillips, A. T. Williams, A centurial record of beach rotation, *Journal of Coastal Research* 65 (2013) 594-599.
- [57] T. Thomas, N. Rangel-Buitrago, M. R. Phillips, G. Anfuso, A. T. Williams, Mesoscale morphological change, beach rotation and storm climate influences along a macrotidal embayed beach, *Journal of Marine Sciences and Engineering* 3 (2015) 1006-1026.
- [58] G. E. Tsekouras, A simple and effective algorithm for implementing particle swarm optimization in RBF network's design using input-output fuzzy clustering, *Neurocomputing* 108 (2013) 36-44.
- [59] A. F. Velegrakis V. Trygonis, A. E. Chatzipavlis, T. Karambas, M. I. Vousdoukas, G. Ghionis, I. N. Monioudi, T. Hasiotis, O. Andreadis, F. Psarros, Shoreline variability of an urban beach fronted by a beachrock reef from video imagery. *Natural Hazards* 83 (1) (2016) 201–222.
- [60] M. I. Vousdoukas, P.M. Ferreira, L.P. Almeida, G. Dodet, F. Psaros, U. Andriolo, R. Taborda, A.N. Silva, A. Ruano, O.M. Ferreira, Performance of intertidal topography video monitoring of a meso-tidal reflective beach in South Portugal, *Ocean Dynamics* 61(10) (2011) 1521–1540.
- [61] M. I. Vousdoukas, A. F. Velegrakis, C. Plomaritis, Beachrock occurrence, characteristics, formation mechanisms and impacts, *Earth Science Reviews* 85 (2007) 23-46.
- [62] P. P. Wong, I.J. Losada, J.P. Gattuso, J. Hinkel, A. Khattabi, K.L.McInnes, Y. Saito, A. Sallenger, Coastal systems and low-lying areas, in: In: C.B.Field, V.R. Barros, D.J. Dokken, K.J. Mach, M.D. Mastrandrea, T.E. Bilir, M. Chatterjee, K.L. Ebi, Y.O. Estrada, R.C.Genova, B. Girma, E.S. Kissel, A.N. Levy, S. MacCracken, P.R. Mastrandrea, L.L. White (Eds.), *Climate Change 2014: Impacts, Adaptation, and Vulnerability. Part A: Global and Sectoral Aspects, Contribution of Working Group II to the Fifth Assessment Report of the Intergovernmental Panel of Climate Change*, Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2011, pp. 361–409.

- [63] G. Zhu, S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, *Applied Mathematics and Computation* 217 (2010) 3166-3173.