Simulation of Significant Wave Height by Neural Networks and Its Application to Extreme Wave Analysis

A. AMINZADEH-GOHARI
EECS Department, University of California, Berkeley, California

H. BAHAI
School of Design and Engineering, Brunel University, Uxbridge, Middlesex, United Kingdom

H. BAZARGAN
College of Engineering, Shahid-Bahonar University of Kerman, Kerman, Iran

(Manuscript received 19 July 2007, in final form 29 August 2008)

ABSTRACT

The derivation of the long-term statistical distribution of significant wave heights \( (H_s) \) is discussed in this paper. The distribution parameters are estimated using artificial neural networks (ANNs) trained with the help of a simulated annealing algorithm and operated in an autoregressive mode. The ANNs were utilized in estimating the parameters of a conditional probability distribution related to a desired \( H_s \), given its preceding \( H_s \)s, approximated by a proposed distribution called the hepta-parameter spline. The performance function during training was based on the likelihood function of the statistical method of maximum likelihood estimation (MLE). Given the observed dataset, the most probable weights and biases of the neural networks were determined in such a way that the performance function was optimized. The distribution could be used in the simulation and forecasting of \( H_s \). This paper also presents an extreme wave analysis using the simulated \( H_s \). The extreme analysis conducted in this study using the maxima method offers an alternative approach, avoiding the unrealistic hypothesis that annual \( H_s \)s are identically distributed, as is conventionally assumed when using the Fisher–Tippet theorem.

1. Introduction

The knowledge of wind wave characteristics is essential in many ocean engineering activities. In the past a number of models have been developed to simulate and forecast these characteristics. These simulations are predominantly based on approximated analytical models. An alternative approach is based on developing artificial neural networks (ANNs). ANNs have effectively been utilized in forecasting natural phenomena that are commonly characterized by uncertain interrelationships among their physical parameters. Jain and Deo (2006) took stock of the research studies reported so far in this area. They reported that in general neural networks provide a better alternative, either substitutive or complementary, compared to traditional computational schemes of regression, time series analysis, pattern matching, and numerical methods. Some recent chronological applications of the ANNs to forecasting wave characteristics are reviewed here.

Deo et al. (2001) used a three-layered feed-forward ANN to obtain significant wave height and mean zero-up-crossing wave period (output) from wind speeds (input). They concluded that an appropriately trained network could provide satisfactory results for certain types of predictions and that, unlike the deterministic models, wind fetch and duration do not seem to be the required inputs to be given to ANNs. Agrawal and Deo (2002) dealt with online forecasting of wave heights by ANNs as well as by first-order autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) models. They reported that the ANNs resulted in a more accurate prediction of wave...
heights than the two time series models for short prediction intervals. For the long-term prediction, however, both approaches had similar performance. Altunkaynak and Ozger (2004) forecast significant wave height ($H_s$) from wind speed using the Kalman filter technique. The parameters of the Kalman filter were estimated by neural networks. Makarynskyy et al. (2005) developed and tested an ANN-based methodology to predict $H_s$, the mean zero-up-crossing wave period, and the peak wave period of sea states. They dealt with forecasting subsequent 3-hourly intervals by training a set of ANNs with one hidden layer given the 24-h $H_s$ history as input. They concluded that the closer the $H_s$s are to the time being, the better their forecasts are. They employed two different ANN strategies to forecast the sea-state characteristics for 3, 6, 12, and 24 h in advance. In the first approach, eight separate ANNs were implemented to simulate wave parameters for each of the above four prediction intervals. In the second approach only two networks provided simultaneous forecasts of the wave characteristics for the four prediction intervals. The suitability of ANNs has been demonstrated through verifying the short-term forecasts against the observed data. The results of the simultaneous forecasts exhibited less accuracy than those obtained separately.

The focus of the present study is not on the short-term or long-term forecasts of $H_s$ and, furthermore, the desired optimization function is not taken as the mean-squared error between the observed and simulated $H_s$s. The main objective of this work is to develop an ANN model for the statistical behavior of the stochastic process of the $H_s$s.

The use of ANNs also differs from that of the previous works. ANNs have essentially been used previously to predict a future $H_s$ using its past observations. Consider an algorithm that predicts $H_s$ using some trained ANNs and gives the predicted $H_s$ as a feedback to the ANNs in order to predict further $H_s$s. In other words, preceding forecasts are recursively used to predict future $H_s$s. While one could repeat this procedure to predict the $H_s$s for future years, the algorithm will become useless for this purpose due to the “propagation of errors” effect. The predictions become less and less accurate as we move forward in time. This difficulty of losing the precision could be overcome by using the following proposed approach. Assuming that one has access to the conditional probability distribution of a future $H_s$ given the values of its past $H_s$s and starting with some initial observations, it is possible to find the conditional probability distribution of next $H_s$ given its preceding $H_s$s. A random variate is generated from this distribution. This random variate is then used in the next iteration to find the conditional distribution of the next $H_s$. This process could be continued in an autoregressive manner resulting in a random sequence of $H_s$s.

It could be proved that the sequence will have the same statistical characteristics as the observed $H_s$s. In this work ANNs were used to estimate the parameters of the “conditional probability distribution” of a future $H_s$ given its past $H_s$s. The ANNs were trained using an extensive set of observed data. The distribution could be used for simulating the $H_s$s by generating variates. The simulated variates could, in turn, be utilized for other purposes such as forecasting and extreme value analysis.

Since the present work deals with the long-term distribution of $H_s$, some related previous works are briefly discussed here. The existing methods commonly assume that the $H_s$s in a given month or day of the year follow a particular parametric distribution such as lognormal distribution or the Weibull distribution fitted to the relevant $H_s$ data. Since there is no theoretical basis for favoring one parametric distribution over another, the goodness of the selected distribution depends on how the data fit the distribution. Some distributions adopted for long-term $H_s$ behavior are the lognormal distribution (Jaspers 1956), the Weibull distribution (Battjes 1971; Goda et al. 1993; van Vladder et al. 1993; Guedes-Soares and Henriques 1996), and a mixture of lognormal and Weibull (Harver 1985). The beta and gamma distribution have also been suggested by Ferreira and Guedes Soares (1999, 2000). Jaspers (1956) successfully fitted the lognormal distribution while Battjes (1971) suggested the Weibull distribution for the data upper range (Ferreira and Guedes-Soares 1998). However, the adequacy of these two distributions for extreme wave analysis has been questioned by Guedes-Soares and Henriques (1996). Ochi (1992) utilized several datasets to show that the generalized gamma distribution provides a better fit than the two distributions.

This research assumes that $H_s$s belong to a nonstationary Markov model and that the characteristics of the Markov model do not change from year to year. In the proposed algorithm an assumption has been made on the order of the Markov model, following Makarynskyy et al.’s (2005) assumption that a 24-h history of $H_s$ has sufficient information for predicting future $H_s$s. The study, therefore, assumes that the 24-h $H_s$ history would be a fairly good choice for the order. A 24-h $H_s$ history comprises a sequence of 8 successive 3-hourly $H_s$s.

Full characterization of the Markov model requires the distribution of a desired future $H_s$ given its immediate eight preceding 3-hourly $H_s$s. To meet this requirement, a class of seven-parameter distributions named the hepta-parameter spline has been introduced in appendix B. These distributions, designed to be very flexible, approximate the conditional distribution of $H_s$. 

Unauthenticated | Downloaded 04/15/24 12:49 AM UTC
ANN modeling was utilized to estimate the parameters of the hepta-parameter spline distributions. Seven four-layer feed-forward networks were developed to estimate the seven parameters of the proposed distribution for the $H_s$ of a desired time given its 24-h history of the preceding 3-hourly $H_s$. The performance function for network training was a logarithmic function, based on the maximum likelihood estimation (MLE) method.

This study also conducts an extreme wave height analysis. The method used for this purpose employs the empirical distribution of the simulated maxima, avoiding the traditional need for unrealistic assumption that the maxima are drawn from identically distributed $H_s$.

2. **Neural networks**

Biological neural networks comprise units called neurons with inputs and outputs called dendrites and axons, respectively. Artificial neurons are extremely simple abstractions of the complex structure of biological neurons. Figure 1 shows a simplified mathematical model of a neuron.

Equation (1) holds for the neuron in Fig. 1:

$$o = f_{tr}(v) = f_{tr}(w \times p + s),$$  \hspace{1cm} (1)

where $p$ is the input to the neuron, $w$ is the synaptic weight, $s$ is bias, $v$ is the output of the summing junction, $f_{tr}$ is the transfer function, and $o$ is the neuron output.

ANNs can roughly be divided into three categories: feed-forward networks, feedback networks, and self-organizing networks (Kohonen models). Haykin (1998) and Fausett (1994), among many others, present an overview of ANNs. In this work one of the most common neural network modeling techniques (i.e., feed-forward multilayer networks) have been used. The weights and biases of ANNs are determined by a training process. During this process the weights and biases are continually modified using an algorithm to optimize a desired function called performance (cost) function. The networks utilized in this research were trained with two algorithms [i.e., the deterministic algorithm of the gradient descent and simulated annealing (SA) algorithm, which is a stochastic algorithm]. The latter proved to be much faster than the former in optimizing the performance function. Therefore, SA was chosen as the preferred algorithm.

3. **Simulated annealing**

The SA algorithm is a numerical global optimization technique, inspired by the principles of thermodynamics. The SA is motivated by an analogy to annealing in solids. The idea stems from the work by Metropolis et al. (1953). Kirkpatrick et al. (1983) suggested that the simulation idea of Metropolis could be used to search for the feasible solution space of an optimization problem, with the objective of converging to an optimal solution. Table 1 shows the flowchart of the standard SA algorithm used for training in this research.

The cooling schedule of a simulated annealing algorithm consists of some components including a parameter called temperature. There are different methods to reduce the temperature iteratively. The method used in this work, first suggested by Lundy and Mees (1986), only carries out one iteration at each temperature and decreases the temperature very slowly using Eq. (2):

$$T_{k+1} = \frac{T_k}{1 + \beta \times T_k},$$  \hspace{1cm} (2)

where $T_k$ is the current temperature, $T_{k+1}$ is the temperature of the next iteration, and $\beta$ is a suitably small value (e.g., $10^{-6}$, $10^{-7}$).

For an overview of SA there are many references including Dowsland (1995) and Salamon et al. (2002).

4. **Data**

A set of nearly 21-yr 3-hourly significant wave heights was used for training the networks. The data (available online at http://www.nodc.noaa.gov/BUOY/46005.html) have been measured and compiled by U.S. National Oceanographic Data Center (NODC) Buoy 46005 in the northeast Pacific from 1 January 1978 for 21 yr and 84 days (i.e., 7754.375 days). This set of data was expected to consist of $7754.375 \times 8 = 62035$ 3-hourly $H_s$. However, there are some gaps in the data record. For example, the data for the following periods were missing: February–April 1985, November–December 1985, October–November 1986, December 1990, and April–June 1993 (Anderson et al. 2001). From this data those values of $H_s$ were extracted whose immediate eight preceding consecutive 3-hourly values of $H_s$ were available; this led to 49736 $H_s$. The extracted $H_s$, a vector of length 49,736, was used as the target vector for training the networks. An additional dataset of size $1 \times 8771$ observed in the 4-yr period 1999–2002 was also prepared for comparison purposes (test data). This 4-yr
Table 1. The standard simulated annealing algorithm (after Moins 2002).

<table>
<thead>
<tr>
<th>Calculation of the Current_Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialization</strong> (Current_solution, temperature)</td>
</tr>
<tr>
<td>Initialization (Current_solution, temperature)</td>
</tr>
<tr>
<td>Calculation of the New_Cost</td>
</tr>
<tr>
<td>New_State</td>
</tr>
<tr>
<td>Calculation of the New_Cost</td>
</tr>
<tr>
<td>IF (Current_Cost–New_Cost) ≤ 0 THEN</td>
</tr>
<tr>
<td>Current_State = New_State</td>
</tr>
<tr>
<td>ELSE</td>
</tr>
<tr>
<td>IF exp ( \left( \frac{\text{Current_Cost} – \text{New_Cost}}{\text{temperature}} \right) ) &gt; Random (0, 1) THEN</td>
</tr>
<tr>
<td>– Accept</td>
</tr>
<tr>
<td>Current_solution = New_solution</td>
</tr>
<tr>
<td>ELSE</td>
</tr>
<tr>
<td>– Reject</td>
</tr>
<tr>
<td>Decrease the temperature</td>
</tr>
<tr>
<td>EXIT when STOP_CRITERION</td>
</tr>
<tr>
<td>END LOOP</td>
</tr>
</tbody>
</table>

5. Network specifications

Seven feed-forward networks were created for estimating the seven parameters of the proposed distribution. The architecture of the networks should be sufficiently complex and involve an adequate number of neurons to accurately approximate the behavior of the functions they are dealing with. While an insufficient number of neurons will not allow the network to track closely the changes of the function it is estimating, an excessive number of neurons, compared to the size of the input data, will cause an undesirable effect known as “overfitting.” More than 15 different network architectures were tested and eventually the size \( 13 \times 24 \times 13 \times 1 \) was chosen. The transfer functions of the two hidden layers and the output layer were of “logsig,” “logsig,” and “purelin” type, respectively. For estimating the seven parameters, networks with two hidden layers proved to be more efficient than networks with one hidden layer having the same number of weights and biases.

6. Input matrix, target vector, output and training algorithm, and performance function

a. Input matrix

The input for training the networks is a matrix of size \( 13 \times 49 \, 736 \) described as follows: each column of the input matrix for training the networks contains 13 pieces of information related to the corresponding \( H_s \) in the target vector [i.e., its 8 successive 3-hourly \( H_s \) measurements, the time, and the four membership values (defined later) for the following 3-hourly \( H_s \)].

In the present research it is assumed that 8 preceding successive 3-hourly \( H_s \) measurements \( (h_{t-1}, h_{t-2}, \ldots, h_{t-8}) \) and also the day of the occurrence of an \( H_s \), denoted by \( t_0 \), affect the conditional distribution of the \( H_s \). The reason for considering only the 8 preceding successive 3-hourly \( H_s \) measurements was given in the introduction. The day of the occurrence of an \( H_s \) (but not the year) was included in the input set. This is based on the assumption that the \( H_s \) Markov model does not change from year to year. Therefore, it was concluded that the year in which the desired \( H_s \) would occur contains no useful information and hence the year value was ignored.

The day values (lying in the range 0–365.25 with 0 = 1 January and step 0.125) were given to the networks at the first trials of training. The results of simulations, however, showed that using the “time” (ranging between 0 and 24) and the 4-value season (defined later) instead of day as input, even though redundant, increases the convergence of the neural networks and decrease their size. Intuitively, the \( H_s \)'s vary with the changes in day through the changes in time (ranging between 0 and 24) and the season of the year; therefore, providing the values of time and season as inputs could improve network training. These five values were added to the set of inputs. Hence, since for every \( H_s \) in the target vector there were 13 values available as input, the input for training the networks is a matrix of \( 13 \times 49 \, 736 \). Although the time values lie in the range 0–24, they were divided by 24 to increase the efficiency of training.

The behavior of \( H_s \) changes from one season to another. To distinguish the degree of associativity of each day to a given season, a degree of membership to each season was assigned to a given day. In other words, the four seasons were conceived as fuzzy sets. Hence, a particular day of the year was arranged to belong to all the four seasons with four membership values each lying in the interval 0–1. For example the vector \((1, 0, 0, 0)\) is assigned to midspring which is at 0730 local time (LT) on 127th day of Gregorian calendar or 5 April and \((0.0246, 0, 0, 0.9754)\) is assigned to 0600 LT on 6 Feb or 0600 LT of the 38th day of the calendar. Figure 2 shows the fuzzy diagram of the membership values. As depicted, the spring value is zero on 4 February. It increases linearly until it reaches its maximum (which is one) on 5 April and then linearly decreases until it becomes zero on 6 August and so on.

b. Target vector

The target vector consists of those \( H_s \)'s whose 24-h history (i.e., 8 immediate preceding successive 3-hourly
Since there were only 49,736 $H_s$s whose 8 preceding $H_s$s were available, the following product was formed:

$$P_1 = \prod_{i \in A} f(H_s(t_i)|H_s(t_{i-1}) - h_{i-1}, \ldots, H_s(t_{i-8}) - h_{i-8}(h_i)). \quad (4)$$

where $A$ is the set of indices of the available $H_s$s whose immediate 8 preceding consecutive 3-hourly values were available in the dataset. Equivalently

$$P_2 = \sum_{i \in A} \log f(H_s(t_i)|H_s(t_{i-1}) - h_{i-1}, \ldots, H_s(t_{i-8}) - h_{i-8}(h_i)), \quad (5)$$

was used. The relation between maximizing $P$, $P_1$, $P_2$, and the MLE function is discussed in appendix A.

The initial value of $P_2$, at the beginning of the training process, was less than $-100,000$. As time went on, $P_2$ gradually increased. The increase rate, however, became smaller over time. In optimization problems such as this, a decision needs to be made regarding the stopping time of the algorithm. On the one hand, it is important to ensure that the algorithm has reached a desirable performance while, on the other hand, one has to be careful not to run into the so-called overfitting problem. Overfitting occurs when the number of free parameters (here the networks weights and biases) is relatively high compared to the size of the dataset. To avoid this problem, different approaches have been proposed in the literature. These include stopping the training process once overfitting starts or, alternatively, restarting the training process with a simpler network architecture (e.g., see Jain and Kacprzyk 2002, p. 321). The first approach was adopted in this study. During the training procedure the optimization algorithm was stopped to check the networks against overfitting. A long-term simulation of an overfitted network produces $H_s$ values that are, on the whole, either very high or very low as compared to the observed $H_s$s. A long-term simulation of an underfitted network, on the other hand, will create irregularly scattered points that would not closely follow the annual cycled structure of $H_s$s.

To ensure a desirable performance for the algorithm and to prevent overfitting, the networks were frequently stopped and the observed $H_s$s of 1978–2002 were compared with a 25-yr set of $H_s$s simulated by the model as described in section 7. The training was discontinued when the performance reached a steady point at $P_2 = -9375$. However, the networks with $P_2 = -22174$ was selected in this work. With this performance, the simulated $H_s$s showed a better fit to the corresponding observed values (see Figs. 3 and B1). It should be pointed out that networks of size $13 \times 24 \times 13 \times 1$ are not necessarily the best network architecture that could be trained for this purpose. More experiments with other...
network structures, other inputs, other measures of terminating the training process, and other distributions, along with performing sensitivity analysis, may result in better-trained networks. Also, other algorithms such as the genetic algorithms may be used instead.

7. Simulation of $H_s$

Once the networks are trained, we could simulate the Markov model representing the statistical behavior of $H_s$ over time. To simulate the significant wave heights for a time period using the trained networks, the starting day of the period and the 8 initial 3-hourly $H_s$'s are given to a Matlab program. To simulate the first unknown 3-hourly $H_s$, the program computes Input-Vect, a vector of size 13, consisting of the 8 initial $H_s$'s, the time value, and the 4-season membership values corresponding to the $H_s$ being predicted. The program then estimates the parameters of the conditional distribution of the desired $H_s$ given its 8 preceding successive $H_s$'s from the seven outputs of the seven trained networks using the formulas given in appendix C. The conditional distribution is then completely known.

The simulation proceeds by replacing Input-Vect($i$) with Input-Vect($i+1$), for $i = 1, 2, \ldots, 7$ and Input-Vect (8) with the $H_s$ simulated in the previous stage. Other elements of Input-Vect (i.e., the time and four fuzzy season values) are updated appropriately. The new input vector Input-Vect is used in the next iteration. This process continues until the simulation of the $H_s$'s for the desired period is completed. Figure 3 shows a typical simulation together with the corresponding observed dataset.

It could be shown that the simulated and the observed $H_s$'s are both random samples from the same joint distribution, if the networks are accurately trained. That is why one could use the simulated significant wave heights to deal with any statistical problem concerning the $H_s$ such as forecasting or calculating return values.

8. Extreme wave analysis

Since the statistical analysis of extreme waves is an important tool in determining the design wave for offshore structures, the following section is devoted to this subject. More specifically, the model proposed here was used in calculating the 100-yr $H_s$ return value.

It is worth noting that this research has in fact developed a statistical model of $H_s$ based on the whole 3-hourly dataset. Since the networks are trained using all the datasets, in a sense the whole past 3-hourly observed data have been used to calculate such characteristics as the return value. In some conventional extreme wave analysis methods, the main body of data is filtered at the beginning through (e.g., selecting the values greater than a high threshold). Such methods do not use a huge
chunk of information relevant to the internal structure of the stochastic process of \( H_s \).

Some methods of extreme wave analysis are illustrated here.

\section*{a. Total sample method (initial distribution method)}

1) \textbf{USING THE EMPIRICAL CDF OF THE SIMULATED DATA}

The \( N \)-year return value, \( H_{sNyr} \), is calculated by Eq. (6), following the work of Barltrop (1998):

\[
Pr(H_s > H_{sNyr}) = \frac{m}{365.25 \times 24 \times N},
\]

where \( m \) is the sampling interval and \( N \) is the return period.

Equation (6) gave \( H_{s100yr} = 14.77 \) m when applied on the empirical cumulative distribution function (CDF) of all of the 1000-time-simulated data for 1 yr or 1000-yr simulated data beginning 1 January.

2) \textbf{CONVENTIONAL TOTAL SAMPLE METHOD}

To apply the conventional total sample method, a distribution such as the Weibull distribution has to be fitted to the entire simulated \( H_s \). A 3-parameter Weibull distribution, having the CDF of the following formula:

\[
F(x) = 1 - e^{-\left(\frac{x - A}{B}\right)^C},
\]

with location parameter \( A = 0.14 \), scale parameter \( B = 4.04 \), and shape parameter \( C = 1.92 \) was fitted to all the 1000-time-simulated data. A Rayleigh distribution with parameter 4.203 was also fitted. Figure 4 shows the empirical PDF of the simulated data together with the Weibull and the Rayleigh distributions fit. Figure 5 shows the quantile–quantile (Q–Q) plot for the Weibull distribution. Applying Eq. (6) on the Weibull distribution yields \( H_{s100yr} = 15.29 \) m.

3) \textbf{TOTAL SAMPLE METHOD FOR 1996–99 SIMULATED DATA}

In total, 3000 runs of simulation were conducted for the period 1996–99. The reason for choosing this period is that its data had not been used for training. A 3-parameter Weibull distribution fitted very well to each of the 3000 simulated dataset. Figure 6 shows a Q–Q plot for a typical simulation. In total, 3000 return values were obtained for the 3000 runs using Eq. (6) with maximum and minimum equal to 17.35 and 14.73, respectively, and a mean of 16.12 m. A return value computed by Anderson et al. (2001) for this period using the observed value is 15.74 m.

\section*{b. Maxima method}

To perform the extreme wave analysis using the maxima method, 39 996 runs of simulation for a year beginning 1 January were performed. Each simulation of \( H_s \) would be a random sample drawn from the joint distribution of the real \( H_s \). The law of large number ensures that the CDF of the simulated \( H_s \) approaches the CDF of the statistical model of the observed \( H_s \), as the number of simulations becomes large. Having carried out 39 996 runs of simulation, the empirical CDFs of the simulated \( H_s \) were taken as the representative of the actual CDFs of the observed \( H_s \).

1) \textbf{ANNUAL MAXIMA METHOD}

In this section the conventional maxima method was implemented as well as a proposed method that used the empirical distribution of the maxima of the 39 996 time-simulated \( H_s \).

\( i \) \textbf{Using the annual maxima empirical CDF}

The following formula was used for the calculation of \( H_{sNyr} \) (Goda 2000):

\[
H_{sNyr} = F^{-1}\left(1 - \frac{1}{N}\right),
\]

where \( F^{-1} \) is the inverse function of the CDF of the annual maxima and \( N \) is the return period.

Figure 7 depicts the empirical PDF of the 39 996 annual maxima and its closest approximation by a generalized extreme value (GEV) distribution. The parameters of the GEV distribution were found to be location \( A = 11.9880 \), scale \( B = 0.7187 \), and shape \( C = 0.0904 \). Applying Eq. (8) to the empirical CDF of 39 996 annual maxima would yield a 100-yr \( H_s \) return value equal to...
14.86 m. Here it was not necessary to assume that the annual $H_s$ time series, from which the maxima are drawn, are independent and identically distributed (i.i.d.). This assumption, made in the conventional maxima method, is not realistic because the distribution of the significant wave height is time dependent (e.g., the $H_s$ during the winter are known to be generally higher). This fact has been mentioned and discussed in many works including Anderson et al. (2001, p. 44), Goda (2000, p. 378), Ferreira and Guedes-Soares (1998, p. 165), and Carter and Challenor (1981, p. 259).

(ii) Conventional maxima method

The Fisher–Tippet theorem was used to fit the 39,996 annual maxima, extracted from the simulated data, to the above GEV assuming that the 39,996 time-simulated data are i.i.d. As calculated from Eq. (8), $H_{s100yr} = 14.67$ m. Also a Weibull distribution with parameters $A = 9.7253$, $B = 2.9225$, and $C = 3.2692$ was fitted to the 39,996 simulated annual maxima (see Fig. 8) resulting in $H_{s100yr} = 14.39$ m using Eq. (8).

2) COMPounding MONTHLY MAXIMA METHOD

Carter and Challenor (1981) derived the cumulative distribution of annual maxima as the product of 12 monthly cumulative distributions (as referenced by Goda 2000):

$$F_{\text{annual maxima}}(x) = \prod_{i=1}^{12} F_{\text{the maxima of month }} (x).$$

(9)

For simplicity a year was divided into 12 equal parts and 12 empirical CDFs were computed from the 12 sets of data, each consisting 39,996 monthly maxima. The empirical PDFs for the first and ninth month are shown in Figs. 9 and 10 as typical PDFs. The CDF of the annual maxima was calculated using Eq. (9) and the 100-yr return value was computed by applying Eq. (8) on the CDF resulting in $H_{s100yr} = 14.85$ m.

c. Peaks-over-threshold method

The peaks-over-threshold (POT) analysis was performed using the simulated data and the following two methods: a method in which the empirical CDF of the
peaks was used and the conventional POT method. The 39 996 time-simulated $H_s$ data for a year beginning from 1 January were also used in this method.

1) CONVENTIONAL POT METHOD

The conventional POT method requires finding a threshold for which the simulated data above it fit very well to a generalized Pareto distribution (GPD). The tool used for recognizing the well fit was the Q–Q plot. Several thresholds, selected between 8 m to the maximum of the data, were tentatively applied on the data-set. Finally 14.3 m was selected as the suitable threshold yielding the best GPD fit; that is, the GPD with the parameters, $A = 14.3$, $B = 0.65185$, and $C = 0.1684$ fitted very well to the 989 peaks of the 39 996-time-simulated $H_s$ data over threshold $= 14.3$ m. A Q–Q plot for the GPD fit is shown in Fig. 11. The following formula was used for the calculation of $N$-year return value in the POT method (Goda 2000):

$$H_{s_{N\text{yr}}} = F^{-1} \left( 1 - \frac{1}{N\lambda} \right),$$ \hspace{1cm} (10)

where $F^{-1}$ is the inverse function of the CDF of the peaks, $\lambda$ is the mean number of peaks per year, and $N$ is the return period.

Equation (10) gives $H_{s_{100\text{yr}}} = 14.84$ m when applied on the above GPD.

2) USING THE EMPIRICAL CDF OF THE PEAKS

Among the 39 996 runs of $H_s$ simulation, 989 $H_s$s exceeded the threshold 14.3 m. Finding the empirical CDF (ECDF) of these peaks yields a 100-yr return value of $H_{s_{100\text{yr}}}$ equal to 14.85 m using Eq. (10).

Table 2 compares the calculated return values with those reported by Anderson et al. (2001). They selected several samples from the full 1978–99 dataset in the northeast Pacific for estimating the 100-yr return value. Depending on the method and dataset used, the return value is reported to vary between 13.34 and 17.3 m. For example, they applied the total sample method by fitting all the hourly observed $H_s$s during 1996–99 to a Fisher–Tippett type-I distribution. The $H_{s_{100\text{yr}}}$ obtained by this approach equals 15.74 m. Applying the compounding monthly maxima method, however, yielded $H_{s_{100\text{yr}}} = 17.3$ m.

9. Conclusions

1) Multilayer feed-forward ANNs proved to be an efficient modeling technique for estimating the parameters of a proposed distribution related to long-term simulation of significant wave heights.
2) Simulated annealing proved to be an effective algorithm for training the networks used in the works of this kind.

3) ANNs may be employed to estimate the parameters of a conditional probability distribution related to a random variable, instead of predicting its values directly. To that end, during the training stage, one needs to maximize an appropriately chosen performance function. The appropriate performance function defined in this work, which is based on the MLE method, proved to be very appropriate.

4) Three points are worth noting in the extreme value analysis part of this work:

- First, in order to find the internal behavior of significant wave heights (high or low, in winter or in summer) the simulated data used in the extreme value analysis has been obtained using the output of the ANNs trained by a whole set of observed data. Therefore, it could be concluded, in a sense, that the POT and maxima methods in the present study have used all the observed data. This is an advantage over the conventional POT method and maxima methods where only the high $H_s$ are used and most of the observed data are filtered at a very early stage. These approaches ignore the internal behavioral structure of $H_s$.

- Second, in this study there was no need to use the Fisher–Tippet theorem for the maxima method to calculate the $H_s$ return value. Application of the theorem used in extreme value analysis requires some assumptions, which are not realistic in the cases such as wave characteristics.

- The procedure introduced in this article for dealing with the long-term statistical distribution of the $H_s$ process using ANNs is applicable to other non-stationary processes having periodic variations in a variety of applications and analyses. These include simulation and forecasting of the likelihood of events, determination of mean values of various processes, and performing extreme value analysis. A typical algorithm for conducting such studies is given below.

Assume that there exists a duration $T$, which is a natural number, such that for any $q \geq 0$, the non-stationary sequence $X_n$ satisfies

\[
\Pr(X_n = x_n, X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots, X_{n-q} = x_{n-q}) = \Pr(X_{n+T} = x_n, X_{n+T-1} = x_{n-1}, X_{n+T-2} = x_{n-2}, \ldots, X_{n+T-q} = x_{n-q}).
\]

The procedure could be then described as follows.

1) Find a natural number $m$ in such a way that

\[
\Pr(X_n = x_n|X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots, X_1 = x_1) \cong \Pr(X_n = x_n|X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots, X_{n-m} = x_{n-m}).
\]

2) Choose a flexible parametric distribution for the random variable

$X_n|X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots, X_{n-m} = x_{n-m}$.

3) Each parameter of the distribution when fitted to

$(X_n|X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots, X_{n-m} = x_{n-m})$ is

<table>
<thead>
<tr>
<th>Method</th>
<th>Return value (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present study—simulated data</td>
<td>14.77</td>
</tr>
<tr>
<td>Anderson et al. (2001)—observed data</td>
<td>15.29</td>
</tr>
<tr>
<td>Tot sample method using the empirical CDF</td>
<td>14.73–17.35 (mean 16.12)</td>
</tr>
<tr>
<td>Conventional tot sample using all simulated data</td>
<td>15.74 (p. 43)</td>
</tr>
<tr>
<td>Compounding monthly max</td>
<td>14.85</td>
</tr>
<tr>
<td>Conventional POT</td>
<td>14.84</td>
</tr>
<tr>
<td>Empirical CDF of the peaks</td>
<td>14.85</td>
</tr>
<tr>
<td>Two 100-yr return value for 2 months</td>
<td>14.92 and 13.34 (p. 44)</td>
</tr>
</tbody>
</table>

Fig. 11. A GPD fit to 989 peaks over 14.3 of the 39 996 runs of simulating $H_s$ time series (i.e., Q–Q plot).
a deterministic function of \(x_{n-1}, x_{n-2}, \ldots, x_{n-m}\), and \((n \text{ modulo duration } T)\). To determine the function of each parameter, create a neural network whose inputs are \(x_{n-1}, x_{n-2}, \ldots, x_{n-m}\) and \((n \text{ modulo duration } T)\). The output is the corresponding parameter.

4) Given the observed dataset, the most probable weights and biases of the networks could be found through maximizing the likelihood function of the MLE method. The long-term distribution of the nonstationary sequence \(X_n\) is then derived using the trained networks when operated in an autoregressive mode. The distribution could be used in the simulation and forecasting of future \(X_n\)s.

**Acknowledgments.** The authors would like to express their gratitude to Mr. Carter, the joint author of Anderson et al. (2001), for being kind enough to obtain the permission for the report from Shell International and for answering all of our questions. Indeed, the present study stems from the report Mr. Carter sent us.

**APPENDIX A**

**The Relationship between Maximizing Eqs. (3)–(5) and the MLE Function**

To estimate the seven parameters of the proposed density function by the MLE method, the joint density function of the 3-hourly \(H_s\)s in the period 1978–99 [i.e., the joint PDF of \(H_s(t_0), H_s(t_1), H_s(t_2), H_s(t_3), \ldots, H_s(t_{62035})\)] must be maximized (i.e., to maximize the likelihood function \(L\)):

\[
L = f_{H_s(t_0), H_s(t_1), \ldots, H_s(t_{62035})}(h_0, h_1, \ldots, h_l, \ldots, h_{62035}),
\]

(A1)

where \(H_s(t_i)\) is the random variable associated with 3-hourly \(H_s\) measured at \(t_i, i = 0, 1, \ldots, 62035\), and \(h_i\) is a particular value for \(H_s(t_i)\).

Applying the chain rule for factorization to (A1) results in

\[
L = \prod_{i=62035}^{62035} f_{H_s(t_i) | H_s(t_{i-1}), H_s(t_{i-2}), \ldots, H_s(t_0)}(h_0, h_1, \ldots, h_l, \ldots, h_{62035}),
\]

(A2)

where

\[
f_{H_s(t_i) | H_s(t_{i-1}), H_s(t_{i-2}), \ldots, H_s(t_0)} = h_0(h_i)
\]
denotes the conditional PDF of \(H_s(t_i)\) given \(H_s(t_{i-1}), H_s(t_{i-2}), \ldots, H_s(t_0)\) = \(h_0(h_i)\) and

\[
f_{H_s(t_0), H_s(t_1), \ldots, H_s(t_{62035})}(h_0, h_1, \ldots, h_l)
\]
is the joint PDF of the first initial 8 \(H_s\)s.

It is assumed here that there exists a value for \(m\) such that all necessary information concerning the history of \(H_s(t_i)\), the observed \(H_s\) of \(t_i\), is conveyed by the sequence \(H_s(t_{i-1}), H_s(t_{i-2}), \ldots, H_s(t_{i-m})\).

Therefore, for all \(i = 62035\) to \(m\), we could write

\[
f_{H_s(t_i) | H_s(t_{i-1}), H_s(t_{i-2}), \ldots, H_s(t_{i-m})} = h_0(h_i)
\]

\[
\cong f_{H_s(t_i)} = h_{i-1}(H_s(t_{i-1}), h_{i-2}(H_s(t_{i-2}), \ldots, h_{i-m}(H_s(t_{i-m})))
\]

(A3)

Finding the order \(m\) remains a problem to be tackled. However, it was assumed, in the introduction, that \(8\) immediate preceding measurements contain sufficient information to simulate \(H_s(t_i)\); in other words, the order was determined to be \(8\).

Therefore, from Eq. (A2) and (A3) it could be concluded that

\[
L \cong \prod_{i=62035}^{62035} f_{H_s(t_i)|H_s(t_{i-1}), \ldots, H_s(t_{i-8})}(h_0, h_1, \ldots, h_8),
\]

(A4)

Let

\[
J = \prod_{i=62035}^{62035} f_{H_s(t_i)|H_s(t_{i-1}), \ldots, H_s(t_{i-8})}(h_0, h_1, \ldots, h_8)
\]

then

\[
L \cong J \times \prod_{i=62035}^{62035} f_{H_s(t_i)|H_s(t_{i-1}), \ldots, H_s(t_{i-8})}(h_0, h_1, \ldots, h_8),
\]

(A5)

where

\[
f_{H_s(t_i)|H_s(t_{i-1}), h_{i-2}(H_s(t_{i-2}), \ldots, h_{i-m}(H_s(t_{i-m})))} = f_{h_{i-1}}(h_{i-1}(h_0, h_1, \ldots, h_8)),
\]

(A6)

The joint PDF \(J\), which could be factorized by chain rule into the product of 7 conditional PDFs, is negligible in comparison to the remaining 62028 terms of Eq. (A5). Hence, we are able to conclude that when maximizing

\[
P = \prod_{i=62035}^{62035} f_{h_0, h_1, \ldots, h_8} = \prod_{i=62035}^{62035} f_{h_0, h_1, \ldots, h_8}(h_0(h_i))
\]

the likelihood function \(L\) in Eq. (11) is being maximized. Out of the 62028 potential \(H_s\)s, there were, however, only 49736 ones whose immediate preceding 8 successive 3-hourly values were available in the dataset. Let \(A\) denote the set of indices of the available \(H_s\)s. The set \(A\) therefore satisfies \(A \subseteq \{8, 9, 10, \ldots, 62035\}\) and \(|A| = 49736\). The product

\[
P_1 = \prod_{i \in A} f_{h_0, h_1, \ldots, h_8}(h_0(h_i))
\]

was therefore formed instead of \(P\). Finally, the logarithm of \(P_1\), that is, the sum.
\[ P_2 = \sum_{i \in A} \log f(h_i) \quad \text{given} \quad H_s(t_i) = h_i \]

was chosen as the cost function to be maximized during the training.

**APPENDIX B**

**Hepta-Parameter Spline Distribution**

In this appendix, a parametric distribution named “hepta-parameter spline” is proposed. This very flexible distribution is used to approximate the conditional distribution of \[ H_s(t_n) \mid H_s(t_{n-1}) = h_n, \ldots, H_s(t_{n-k}) = h_{n-k} \], whose PDF is denoted by \( f(x) \). The density function \( f(x) \) should satisfy the following two properties:

1) \( \lim_{x \to -\infty} f(x) = 0 \), and 2) smoothness. These two properties imply that there are two points, say, \( a \) and \( e \) where \( f(x) \) approaches zero for \( x \leq a \) and \( x \geq e \), and also there is a point \( c \) in the interval \( <a-e> \) for which \( f'(c) = 0 \) (e.g., the maximum of the function). Since there was no a priori information available on the shape of the distribution of \( H_n \), the proposed density function had to be designed in such a way that it could closely approximate a number of possible distributions. The cubic interpolations on the four subintervals of Fig. B1 were assumed to be a good approximation for \( f(x) \). Note that \( a, b, c, d, e, m_1, \) and \( m_2 \) were chosen such that

\[ b = \frac{a + c}{2}, \quad d = \frac{c + e}{2}, \quad m_1 = f'(b), \quad \text{and} \quad m_2 = f'(d). \]

Figure B2 also shows \( a_1, a_2, a_3, a_4, \lambda, \Delta, m_1, \) and \( m_2 \) the parameters of the function used to approximate of the conditional distribution.

The density function \( g(x) \), the proposed PDF for approximating \( f(x) \), is defined as follows:

\[
g(x) = \begin{cases} 0 & x \leq a \\ \text{Cubic-Int.} \left[ \begin{array}{c} a \\ 0 \end{array} \right] \quad \text{with slope} = 0 \quad \text{and} \quad \left[ \begin{array}{c} b \\ a_2 \end{array} \right] \quad \text{with slope} m_1 > 0 \quad a < x \leq b \\ \text{Cubic-Int.} \left[ \begin{array}{c} b \\ a_2 \end{array} \right] \quad \text{with slope} m_1 > 0 \quad \text{and} \quad \left[ \begin{array}{c} c \\ a_3 \end{array} \right] \quad \text{with slope} = 0 \quad b < x \leq c \\ \text{Cubic-Int.} \left[ \begin{array}{c} c \\ a_3 \end{array} \right] \quad \text{with slope} = 0 \quad \text{and} \quad \left[ \begin{array}{c} d \\ a_4 \end{array} \right] \quad \text{with slope} m_2 < 0 \quad c < x \leq d \\ \text{Cubic-Int.} \left[ \begin{array}{c} d \\ a_4 \end{array} \right] \quad \text{with slope} m_2 < 0 \quad \text{and} \quad \left[ \begin{array}{c} e \\ 0 \end{array} \right] \quad \text{with slope} = 0 \quad d < x \leq e \\ 0 & x \geq e \end{cases}
\]

where \( a, b, c, d, e, a_2, a_3, a_4, m_1, \) and \( m_2 \) are illustrated in Fig. B2, and “Cubic-Int.” stands for the cubic interpolation of each pair of the points mentioned above according to spline method.

Having two points \( x_1 \) and \( x_2 \) and the values of a function \( f(x) \) and its derivative at \( x_1 \) and \( x_2 \), the formula for approximating \( f(x) \) according to the spline method is given by

![A typical Hs Simulation for 1999-2002.](image)
APPENDIX C
Calculation of the Distribution Parameters from the Network Outputs

Let \( x_j \) denotes the output of the \( j \)th network for \( j = 1, 2, \ldots, 7 \). Since the range of the networks outputs is \((-\infty \text{ to } \infty)\) due to the transfer function of \textit{purelin} in the output layers, we cannot directly use the outputs of any of the seven networks for calculating positive \( a_1 \). Hence, a transformation was applied on \( x_1 \) to arrive at a positive number: \( a_1 = x_1^2 \). The relationships used to calculate the parameters from the outputs of the networks are given in Eqs. (C1)–(C9).

\[
\begin{align*}
\text{TABLE C1. Constraints and relationships.} \\
\text{Constraint} & \quad \text{Relationship} \\
\hline
a_1 > 0 & \quad a_1 = x_1^2 \\
0 < a_2 < a_3 & \quad a_2 = a_3 \logsig x_2 \\
\Delta > 0 & \quad \Delta = x_3 \\
0 < a_4 < a_5 & \quad a_4 = a_5 \logsig x_4 \\
0 \leq m_1 \leq \frac{6a_2}{\Delta a} & \quad m_1 = \frac{6a_2}{\Delta a} \logsig x_5 \\
\frac{-6a_4}{(1-\lambda)\Delta} \leq m_2 \leq 0 & \quad m_2 = \frac{-6a_4}{(1-\lambda)\Delta} \logsig x_6 \\
0 \leq \lambda < 1 & \quad \lambda = \logsig x_7
\end{align*}
\]

From Eqs. (B2), (C2), (C3), (C4), and (C7) we have

\[
a_3 = \frac{2x_3^2}{(\logsig x_1)(\logsig x_2) + 0.5 + (1-\logsig x_1)(\logsig x_2)},
\]

where \( \logsig x_j = \frac{1}{1 + e^{-x_j}} \), \( j = 1, \ldots, 7 \).
In practice, in the first step, \( a_3 \) is calculated by Eq. (C8), then \( a_2 \) and \( a_4 \) are calculated from Eqs. (C2) and (C4). Other parameters are calculated simply using the appropriate relationship.

**APPENDIX D**

**Nomenclature**

- \( A \): The set of indices of the \( H_s \) data whose immediate 8 preceding consecutive 3-hourly values were available in the dataset.
- \( \text{ANNs} \): Artificial neural networks.
- \( B \): Scale parameter of Weibul, GEV, and GPD distributions.
- \( C \): Shape parameter of Weibul, GEV, and GPD distributions.
- \( CDF \): Cumulative distribution function.
- \( \text{ECDF} \): Empirical CDF.
- \( f(h_i) = f_{H_s(t_i) \text{given } H_s(t_i-1) = h_i-1, \ldots H_s(t_i-8) = h_i-8}(h_i) \): The conditional PDF of \( H_s(t_i) \) given its 8 immediate preceding 3-hourly successive \( H_s \).
- \( F^{-1} \): CDF inverse function.
- \( GPD \): Generalized Pareto distribution.
- \( \text{GEV} \): Generalized extreme value distribution.
- \( H_s \): Significant wave height.
- \( H_s(t_i) \): A random variable denoting the 3-hourly \( H_s \) measured at \( t_i \).
- \( H_NyR \): \( H_s \) N-year return value.
- \( H_s(t_i)|H_s(t_i-1) = h_{i-1}, \ldots H_s(t_i-8) = h_{i-8} \): A random variable representing \( H_s(t_i) \) given its 8 immediate consecutive previous 3-hourly \( H_s \).
- \( h_i \): A particular observed value for \( H_s(t_i) \).
- \( \text{i.i.d.} \): Independent and identically distributed.
- \( \text{Input-Vector} \): A general name for the columns of the input matrix of the ANNs corresponding to a desired \( H_s \) in the target vector.
- \( \text{MLE} \): Maximum likelihood estimation method.
- \( N \): The return period.
- \( n \): Number of data available in an \( H_s \) time series.
- \( m, m' \): 1) meter, 2) sampling interval (time interval between two successive observed \( H_s \)), and 3) the Markov chain order.
- \( o \): Output of artificial neuron.
- \( p \): Input of artificial neuron.
- \( \text{PDF} \): Probability density function.
- \( \text{Q-Q} \): Quantile-quantile.

**REFERENCES**


