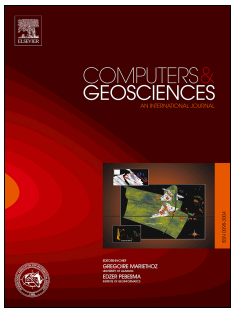
Accepted Manuscript



Sample selection via angular distance in the space of the arguments of an artificial neural network

J.M. Fernández Jaramillo, R. Mayerle

PII: S0098-3004(17)30454-5

DOI: 10.1016/j.cageo.2018.02.003

Reference: CAGEO 4096

To appear in: *Computers and Geosciences*

Received Date: 26 April 2017

Revised Date: 27 December 2017

Accepted Date: 15 February 2018

Please cite this article as: Jaramillo, J.M.Ferná., Mayerle, R., Sample selection via angular distance in the space of the arguments of an artificial neural network, *Computers and Geosciences* (2018), doi: 10.1016/j.cageo.2018.02.003.

Sample selection via angular distance in the space of the arguments of an artificial neural network.

J.M. Fernández Jaramillo[a,*], R. Mayerle[a]

[a] *Technology Center West coast, University of Kiel (CAU), Hafentörn d-25761, Buesum, Germany*

Abstract

In the construction of an artificial neural network (ANN) a proper data splitting of the available samples plays a mayor role in the training process. This selection of subsets for training, testing and validation affects the generalization ability of the neural network. Also the number of samples has an impact in the time required for the design of the ANN and the training. This paper introduces an efficient and simple method for reducing the set of samples used for training a neural network. The method reduces the required time to calculate the network coefficients, while keeping the diversity and avoiding overtraining the ANN due the presence of similar samples. The proposed method is based on the calculation of the angle between two vectors, each one representing one input of the neural network. When the angle formed among samples is smaller than a defined threshold only one input is accepted for the training. The accepted inputs are scattered throughout the sample space. Tidal records are used to demonstrate the proposed method. The results of a cross-validation show that with few inputs the quality of the outputs is not accurate and depends on the selection of the first sample, but as the number of inputs increases the accuracy is improved and differences among the scenarios with a different starting sample have and important reduction. A comparison with the K-means clustering algorithm shows that for this application the proposed method with a smaller number of samples are producing a more accurate network.

Keywords:
Sample selection, Artificial neural network, Data partitioning, Cross-validation, Early stopping, K-means.

*Corresponding author
Email addresses:
jmfernandez@corelab.uni-kiel.de (J.M. Fernández Jaramillo),
rmayerle@corelab.uni-kiel.de (R. Mayerle)

1. Introduction

Artificial neural networks are used for the prediction and reconstruction of time series of several types, specially in geosciences several applications are available, including the short or long term forecast of water levels in rivers (Bazartseren et al. (2003) and Fernández et al. (2010)), discharges in rivers (Tayyab et al. (2016), Krishna et al. (2011)), water temperature (Piotrowski et al. (2015)), in coastal areas (Bowles et al. (2012) and Fernández Jaramillo (2014)), wave forecast (Gopinath and Dwarakish (2015)), and for reconstruction of tidal records (Tirozzi et al., 2006). The effectiveness and relevance of the ANN compared with other linear methods like ARIMA (Box and Jenkins (1976)) is evident when the relation between the inputs and outputs is not linear (Wang et al. (2015)). In the case of the water level in natural areas, the influence of environmental conditions like wind, waves or local bathymetry are main causes of such non linearities (Longuet-Higgins (1970) and Komen et al. (1996)).

The process of training an ANN requires to present a set of samples to adjust the weights of the different neurons and their connection. The amount and quality of those samples will determine the computational effort required for the training and the capability to make a good generalization of the prediction (Macas et al. (2016)). If several samples are representing similar scenarios, not only it will require a longer time to determine the parameters of the artificial neural network but it could also lead to overtraining. In the case of water level predictions in coastal areas, low speed wind conditions are more frequent and prone to be measured than conditions with strong wind arriving from different directions. Samples representing heavy storms are even less frequent. Under low wind conditions the water level behaves more like astronomical tides. We proposed a simple method based in the angle formed by two vectors to filter the samples in order to reduce the size of the training set, while keeping the diversity and quality of the prediction.

2. Methodology

Feed forward ANN are commonly used for time series, (see Fig. 1), The ANN consists of an input layer, at least one hidden layer and an output layer. In the input layer each node x_i is assigned to each argument provided for the ANN. This can be represented as a n-dimensional vector x. The output layer could have one or multiple elements representing the prediction of the network. They are represented by the vector y. The number of hidden layers and the nodes contained in each element varies according to the difficulty of the problem. As the number of elements increases, it permits to handle more complex interactions, but they are harder to train and prone to be overfitted.

Overfitting occurs when the degrees of freedom of the model are increased. This happens when the number of parameters is high compared to the number of observations. Strategies to reduce the impact of overfitting in the reduction of the dimension of parameter space or the effective size of each dimension are summarized by Prechelt (1998). To

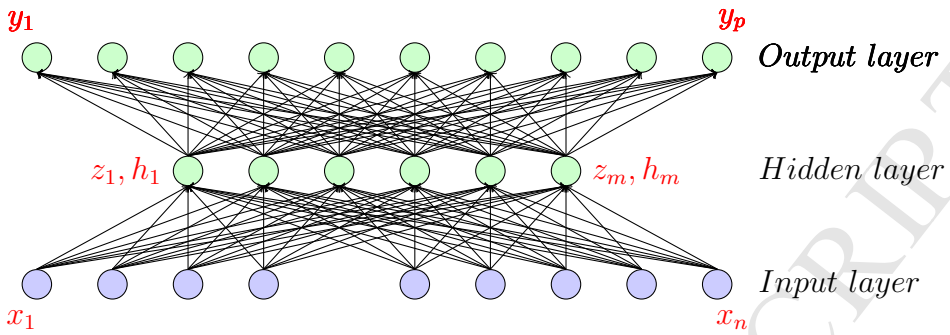


Figure 1: General Feedforward Neural Network

reduce the number of parameters of an ANN the inclusion of greedy constructive learning, pruning or weight sharing can be implemented. Moreover to reduce "the size of each parameter dimension", Prechelt suggests regularization or early stopping.

Early stopping splits the data in 3 sets, the training, the validation and the test. The training set is used to tune the parameters of the ANN. The resulting error is then compared in the validation dataset to decide when to stop. The estimation of the error in the training set is calculated at every iteration in order to figure out whether a change is accepted or not. For the validation set the error is calculated at a lower frequency to reduce the computational effort. The earlier stop will limit overfitting and the required time for training. This is because the error is not allowed to reduce in the training set when this improvement is not also reflected in a better capacity of the model to represent the untrained values. Finally the predictive capacity of the ANN is assessed against data from the test set.

Attention should be payed to the amount and quality of the samples used to adjust the model. It is a common practice to take about 50 to 60% of the available samples for training and the remaining for validation of the ANN (Prechelt, 1998). The samples are selected depending on different criteria. Training samples can be created from random subsets for cross-validation (Prechelt, 1998). For time series a continuous period, where in some cases the selected length considers the properties of the modeled process.- As an example, Tirozzi et al. (2006) selected 1000 hourly samples along 42 days, which cover 3 semi-monthly tidal periods. During an experimental design, the selection of optimal measurements can be taken in to account, for example to evaluate the performance of an ANN trained by random, equal-spaced and latin hypercube samples (Lunani et al., 1995), while Tong and Liu (2005) used samples from a Hammersley-net obtaining better results in generalization performance. Some works have been done to group the samples by the use of clustering algorithms (Sahoo et al. (2012), May et al. (2010), Faraoun and Boukelif (2006) and Ding et al. (2011))

The training of an ANN is usually done by the gradient descent method or by the resilient back-propagation (RPROP). RPROP is a modification of the gradient descent method that does not take the gradients to define the required correction (Riedmiller, 1994). This method considers the change in the sign of the error. If two consecutive errors have the same sign, the correction term will increase. If there is an inversion of the sign, the amount of correction is reduced.

3. Proposed method for the reduction of the sample pool

To reduce redundancy in the information provided by the samples during the training process, a filter has been implemented. This filtering avoids the superfluous information and it also saves computational efforts in the iterative training. For the proposed method the angle (α) between two vectors u_i and u_j (see Eq. 1) of n dimensions (with n being the number of elements of the input layer) is considered. To determine the angle formed by the elements of the input set, the origin of the vector space is considered at the mean value of all the elements as follows:

$$\cos\left(u_i, u_j\right) = \frac{\left(u_i - \bar{u}\right)\left(u_j - \bar{u}\right)}{\left\|u_i - \bar{u}\right\| \left\|u_j - \bar{u}\right\|} \qquad (1)$$

In which: u_i is one sample already selected for the training, u_j is one sample that is tested in order to be included in the training set and $\bar{u}$ is the mean of all the samples.

Whenever the angle α formed by two samples is lower than some selected reference value, the sample is rejected from the pool employed for training (see Fig. 2). The reduced set of samples provides an efficient way to speed up the procurement of an ANN with no loss of significance in the results. It also helps to keep the generalization of the obtained model.

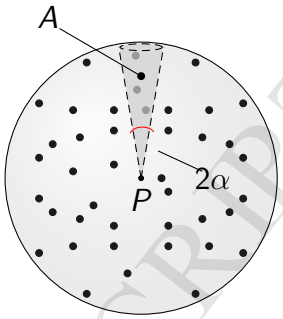


Figure 2: Sample Selection

The algorithm (see Fig. 3) to implement the method is as follows:

1. Split the list of available samples $[S]$ in two lists, in the first one store the samples candidates $[C]$ to be used during the training and validation, the other will be used for testing [W] the quality of the ANN.
2. Calculate the mean vector of all available candidates $\bar{C}$.
3. For each sample, subtract the mean value.
4. Sort the samples obtained in the previous step according to the norm in descending order.
5. Create an empty list for the training elements $\left[T'\right]$ and one for the validation $\left[V'\right]$ elements.

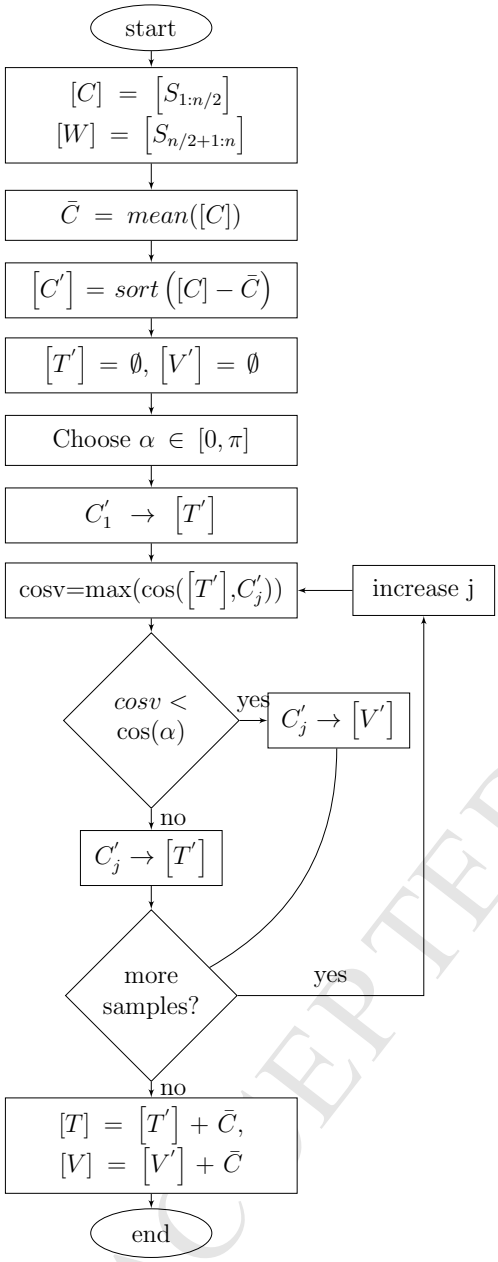


Figure 3: Adopted data assimilation scheme

6. Choose a value for $\alpha \in [0, \pi]$. Smaller values allows more samples to be included in $\left[T'\right]$.
7. Take the first available sample from the candidates list C_1 (The one with the biggest norm) and add to the training set $\left[T'\right]$.
8. Take the next available sample C_j, and for each member of the training $\left[T'\right]$ calculate the cosine of the angle between the two vectors.
9. If the cosine is smaller than the $\cos(\alpha)$ selected in step 6 for any element of $\left[T'\right]$, include the sample in the validation list $\left[V'\right]$ and stop comparing for the candidate C_j, otherwise add the sample to the training list $\left[T'\right]$.
10. Repeat step 8 until no more samples are available.
11. Add the mean value to each member of the training and validation sets and return $[T]$ and $[V]$.

The ANN is trained with the obtained set of samples. After applying the filter, early stopping is implemented by comparing the RMSE between the outputs obtained with the training samples $[T]$ and the validation set $[V]$. An upper limit is defined for the tolerated discrepancy in the RMSE between both groups. If the difference between the validation error and the one from the training is bigger than the limit, the training process is stopped. If the accuracy of the results is not satisfactory, a new filter with a less restrictive angle α is applied and the training is repeated. To save time, the RMSE for validation samples is calculated only every 1000 iterations.

The upper limit for the number of comparisons of the angle of vectors is given by $m(n-1)/2$, with "m" being the number of samples finally accepted for training and "n" the number of candidates to be used in the training. The comparison of the angle between one candidate and the elements already conforming the training set is stopped as soon as the angle is smaller than the required. Then in the worst case this comparison has a complexity of $O(n^2)$, this occurs when every single sample is accepted and the total number of comparisons is $n(n-1)/2$. The best case has a complexity of $O(n)$, when only the first sample is accepted and all the remaining are rejected.

4. Results

4.1. Study case

In order to test the proposed method, it is used to filter the inputs of ANN created for the estimation of water levels in a coastal area. Two networks are used, one for hindcast and one for a short term forecast. The inputs are water levels and wind data from a tidal gauge located at Buesum. The output is given by the predicted water levels at a tidal gauge located at Rochelsteert. Both stations are located in the German North Sea.

For the hindcast is adopted an ANN where only water level measured every 20 minutes over a period of 11 hours and 20 minutes previous to the time when the hindcast is calculated. This has a total of 35 elements in the input layer, 24 neurons in the first hidden layer and 3 in the second hidden layer. The

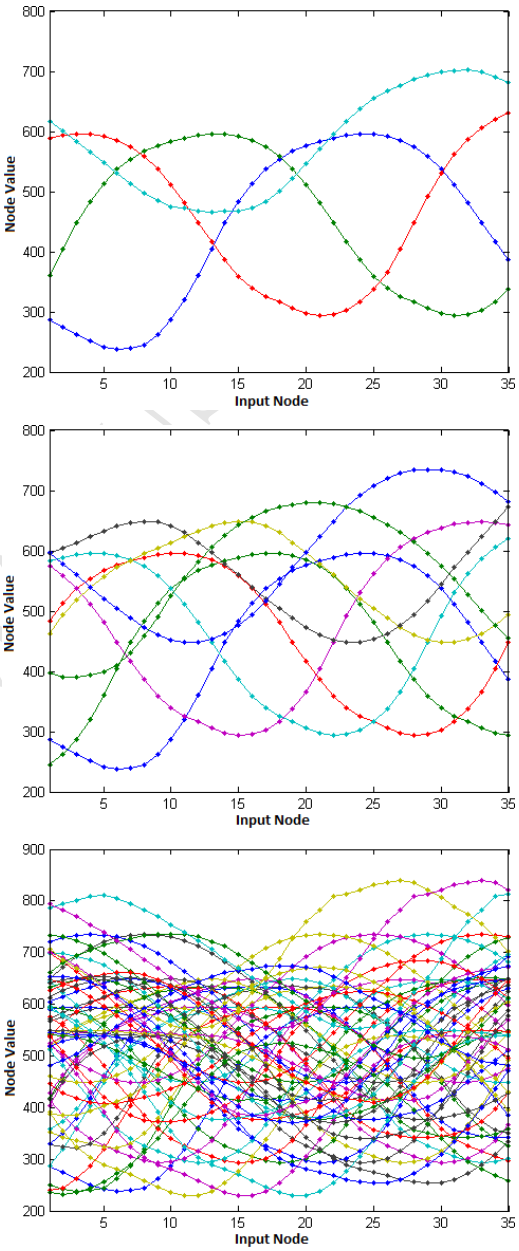


Figure 4: Filtered samples using different values of α for the hindcast. From top to bottom a) $\alpha > \pi/2$, b) $\alpha > \pi/3$ and c) $\alpha > \pi/6$

output is defined as the water level in Rochelsteert. For the forecast also wind measurements at the same times as the hindcast are used as inputs, included in the ANN as the components in North-South and West-East directions. For the forecast then the total number of elements of the input layer is 105. The output is the water level in the Rochelsteert station 12 hours in advance. More details about the design of the ANN are given in (Fernández Jaramillo, 2014).

There are more than $430,000$ measurements recorded each minute along 313 days of information. By taken samples every 20 minutes, the total number of sample elements is reduced to $21,500$. After cleaning samples containing noise or missing information, the first $8,864$ valid elements are selected as possible candidates for the training and validation set, while the remaining are kept for the test samples.

Fig. 4 illustrates three cases of training sets obtained from different values of α. From top to bottom results with the angles equal to $\pi/2$,$\pi/3$ and $\pi/6$ are presented. A bigger angle is more restrictive reducing the number of accepted samples. With $\alpha > \pi/2$ (see Fig. 4.a) only four samples are included in the training set. Each sample represents a different scenario. The most evident difference is the time shift, corresponding to different phases in the tidal cycle. Three scenarios have a similar tidal range, while the fourth covers a situation when the low tide stays in a condition of high water level due to a wind-driven surge condition. As long as more samples are allowed by relaxing the angle α, other scenarios of water level are presented to the ANN. As a result a wider range of tidal amplitudes and tidal phases is included in the process. With $\pi/6$ (see Fig. 4.c) the available phases and water levels for the training represent weather conditions including calm periods and surges.

Next the ANN is trained with the outputs from the proposed filter. For this, different values of α are selected, starting with a large angle and reducing it in order to include more samples. For the early stop, a difference of 10% is allowed between the RMSE of the training and validation samples. Also the maximum number of iterations is defined as $1,000,000$.

Fig. 5 presents the results from the proposed algorithm for a selected number of angles α for the hindcast and the forecast. The x-axis represents the number of samples that are actually used for the training, and the y-axis the RMSE in cm for the training, validation and test sets. The labels in this figure present the $cos(\alpha)$, the number of samples obtained from the application of the algorithm for this angle, the required time in hours and the number of iterations to perform the training of the network with these samples. It is evident that in the training the computational time grows by increasing the number of samples. For the RMSE it can be seen in Fig. 5.a that using less than 42 samples ($cos(\alpha) = 0.720$) the errors are relatively large, but after that point the error is decreasing. The drawback is that the time consumed to find a better solution is much larger. For example, to obtain RMSE values of 17.4, 15.0 and 13.9 cm, computational times of 0.9, 7.3 and 39.5 hours are required

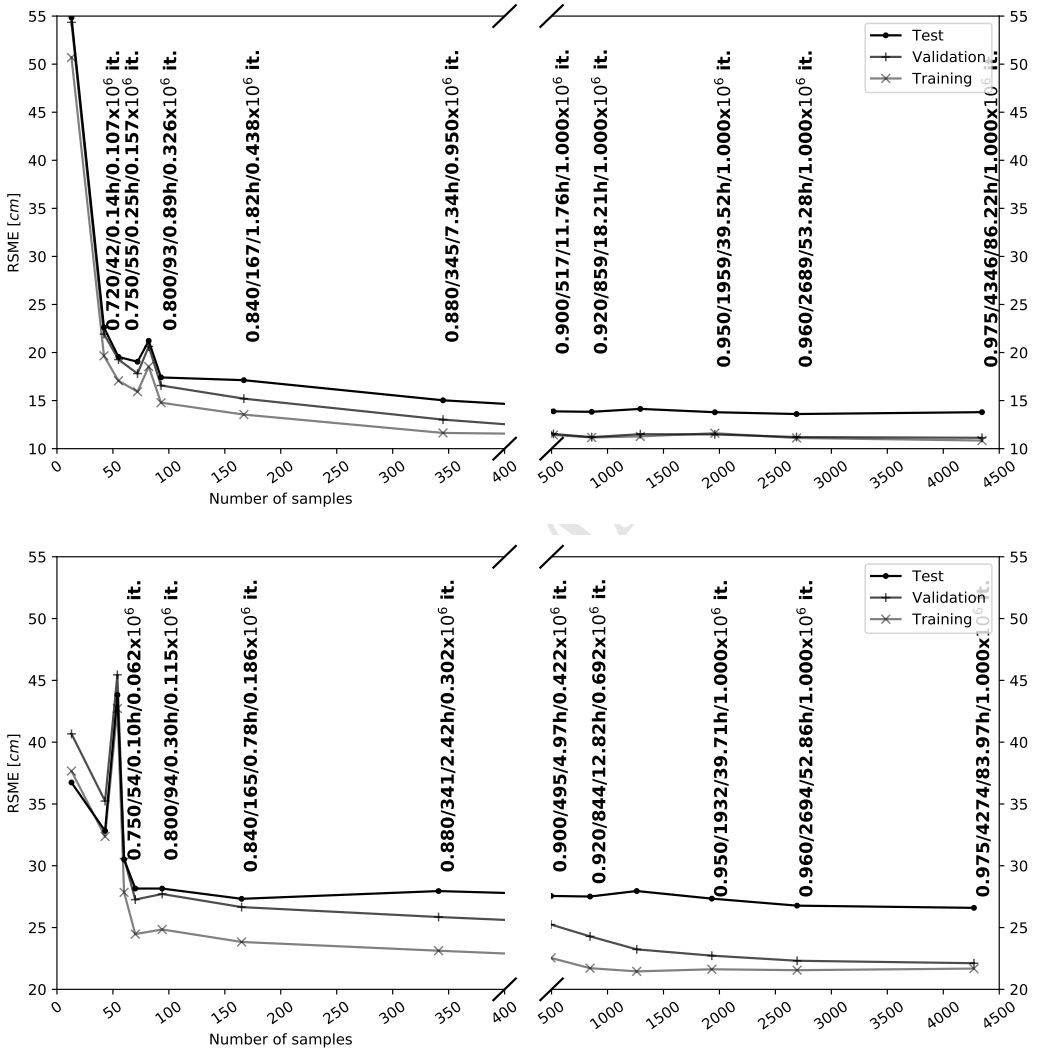


Figure 5: RMSE for different number of elements in the training set. a) Hindcast b) Forecast. Labels present the $cos(\alpha)$, the number of samples obtained, computational time and number of iterations required for the training

respectively. Reducing the RMSE only 1.1 cm takes more than 5 times longer than for a reduction of $2,4$ cm. The RMSE of 13.9 cm is obtained with 517 samples for the training, this represents only 6.3% of the available candidates.

More uncertainties could be observed during the forecast, and that is reflected in a higher RMSE than the one obtained for the hindcast (see Fig. 5.b). As during the hindcast, there is a minimum number of samples required to obtain a fast reduction of the error. In this case after 70 samples ($cos = 0.775$) there is an important improvement in the forecast error. The RMSE reduces until a local minimum of 27.3 cm when 165 samples ($cos = 0.840$) are used. After a small increment of the RMSE, it decreases again in all the remaining attempts. The next improvement is obtained with 1932 samples ($cos = 0.950$), obtaining an RMSE of 27.2 cm after $39,7$ hours of computational time. The minimum is obtained with 4274 samples ($cos = 0.975$) (the largest set of samples used and around half of the available candidates) with an RMSE of 26.6 cm but requiring 84.0 hours. It is important to note that for the same station using a physical driven model the RMSE is 33.7 cm (Fernández Jaramillo, 2014). This means that the ANN has a performance that compares and even exceeds the physical based model in this location.

4.2. Cross-validation

To verify if the selection of the samples according to the magnitude of the norm provides the best results, a cross-validation is performed. This cross-validation helps to understand the influence of the selection order of the samples for training. For one angle, 45 training sets are generated starting each one with a different sample (Starting samples are picked every 200 elements in the candidate list). This is repeated for different angles. Then the ANNs are trained and tested to obtain the performance.

Fig. 6 shows the box plot for the different experiments. The median is represented by an horizontal line inside the box. The mean is plotted as a square. Both the median and mean of each box are connected with the respective values of the next box. For few samples, the dispersion of the results is large. As more samples are considered this dispersion is reduced as can be observed from the interquartile range ($IQR = Q_3 - Q_1$).

In the case of hindcast, where the uncertainties are smaller, just 55 samples ($cos = 0.750$) produce a small IQR, after that some outliers are present. With 517 samples ($cos = 0.900$), the mean and the median converge and no more outliers are present. After this point, increasing the number of samples is not able to improve the capacity of prediction of the ANN.

For the forecast more uncertainties are present, therefore the reduction of the IQR requires the inclusion of more samples. At least 495 samples ($cos = 0.900$) are required to grant the convergence of the estimators and avoid the presence of outliers. After this point there are still small improvements in the RMSE by including extra samples. The minimum is obtained with 2694 samples ($cos = 0.960$). This shows that there is an in-

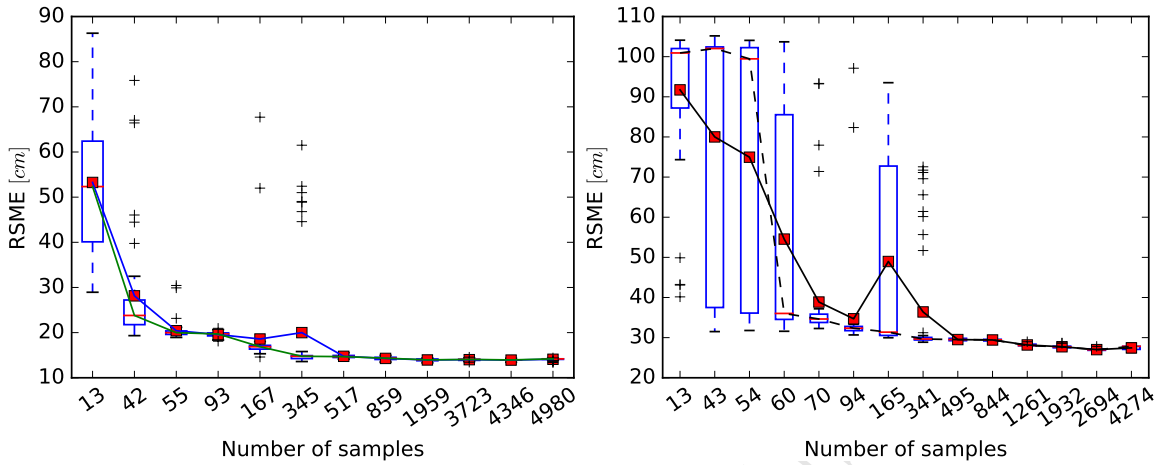


Figure 6: Box plot of the RMSE for the cross-validation for different starting samples in the training set. a) Hindcast b) Forecast

fluence in the selection of the initial sample, but when the samples are selected according to the descending order by magnitude, the results have RMSE values in the lower limit from the cross-validation, and also its effect is negligible as more samples are included. Reducing the angle α looking for the convergence of the RMSE (as shown in Fig. 5) is enough to trust the results without requiring a cross-validation every time.

4.3. Comparison with K-means clustering

The K-means clustering is an algorithm that creates a predefined number of clusters where the centroid of each is representative for several samples that share similar characteristics. Faraoun and Boukelif (2006) and Ding et al. (2011) used K-means as an unsupervised method to reduce the number of samples that are presented for doing the supervised training of the ANN. One difference between the proposed method and the K-means is that in the proposed method, the user decides an angle to filter the samples, then the total number of samples is not under his control, while in the K-mean the number of clusters to be generated are decided from the beginning. Another important difference is that with the angular distance the actual samples are used. With K-means, the centroid tends to smooth the extreme values when few clusters are selected. Then the angular distance is more likely to capture extreme events.

The following statistical relations are used to compare the results from the proposed method with the K-means. The error defined as the difference between the observed water level and the estimation from the neural network ($e = y - \hat{y}$), the mean error ($me = \bar{e}$), the mean absolute error ($mae = \overline{|e|}$), the standard deviation of the error ($\sigma_e = \sqrt{\sum (e - me)^2/n}$), the asymmetry coefficient ($ac = \sum e^3/n\sigma_e^3$) and the root square mean

error ($rmse = \sqrt{\sum e^2/n}$).

Comparisons between the two methods for selecting samples have been carried out considering the hindcast and the forecast periods measuring the performance taking into account all the measurements and outputs individually, but also the extreme values that in the case of tides are given by the low and the high tides.

Figures 7, 8 and 9 show RMSE, ME and AC for each of the hindcast and forecast cases, regarding to all water levels measurements, and only comparing low and high tide values. In order to improve legibility of the details, the vertical axes is split in two ranges with different resolution, one coarser to show the magnitude of the worst obtained values, and one at a finer scale to appreciate the differences as the two methods are converging.

Comparing the results of using the angle among vectors and the K-means, it is evident that the drop in the RMSE occurs with less elements than with the K-means. For the hindcast (see Fig. 5.a and 7.a) it occurs with 42 samples for the proposed method and 859 clusters with K-means. For the forecasting (see Fig. 5.b and 7.b) the drop appears with 70 samples and 844 clusters. In both cases the proposed method allows to train a network with a smaller RMSE independent of the number of elements used. Except for the low tide, with a small number of samples and clusters, the proposed method it is better, but after the K-means error drops, they present similar RMSE values.

Using both algorithms produce ANNs that converge faster in terms of the mean error (see Fig. 8). Still, the angular distance method requires less samples to reach a mean error close to the mean error obtained if all the samples were used to train the ANN. The long term tendency for the mean error is that the ANNs are under estimating the water level in the range of 4 and 6 cm for the hindcast and between 5 to 12 cm for the forecast.

	Hindcast			Forecast		
Par.	Ang.	Ang.	K-m.	Ang.	Ang.	K-m.
cos angle	0.800	0.920	-	0.775	0.920	-
N samples	93	859	859	84	844	844
RMSE	17.4	13.8	19.0	28.2	27.5	29.9
MAE	13.6	10.7	14.9	22.1	21.1	24.0
ME	0.3	3.9	5.6	5.3	6.9	9.6
AC	-3.4	-5.4	-2.1	-0.6	-0.3	-0.2

Table 1: Error comparison using all observations

The skewness measured by the asymmetry coefficient AC also shows (see Fig. 9) that the angular distance method is able to reach a reasonable value with a small number of samples. With a big number of samples the two methods converge to similar values. The AC is positive when only high and low tide are considered. This means that the median of the errors is smaller than the mean error and, as the mean error is positive, the median error it is closer to zero. When all the samples are considered the skewness of the hindcast is negative. This implies that most of the errors are greater than the mean error and are underestimating the water level more than the reported mean error. Finally, for the forecast the asymmetry is closer to 0, meaning that the errors are symmetrically distributed around the reported mean error.

Table 1 presents the results for the forecast and hindcast using all the observed values.

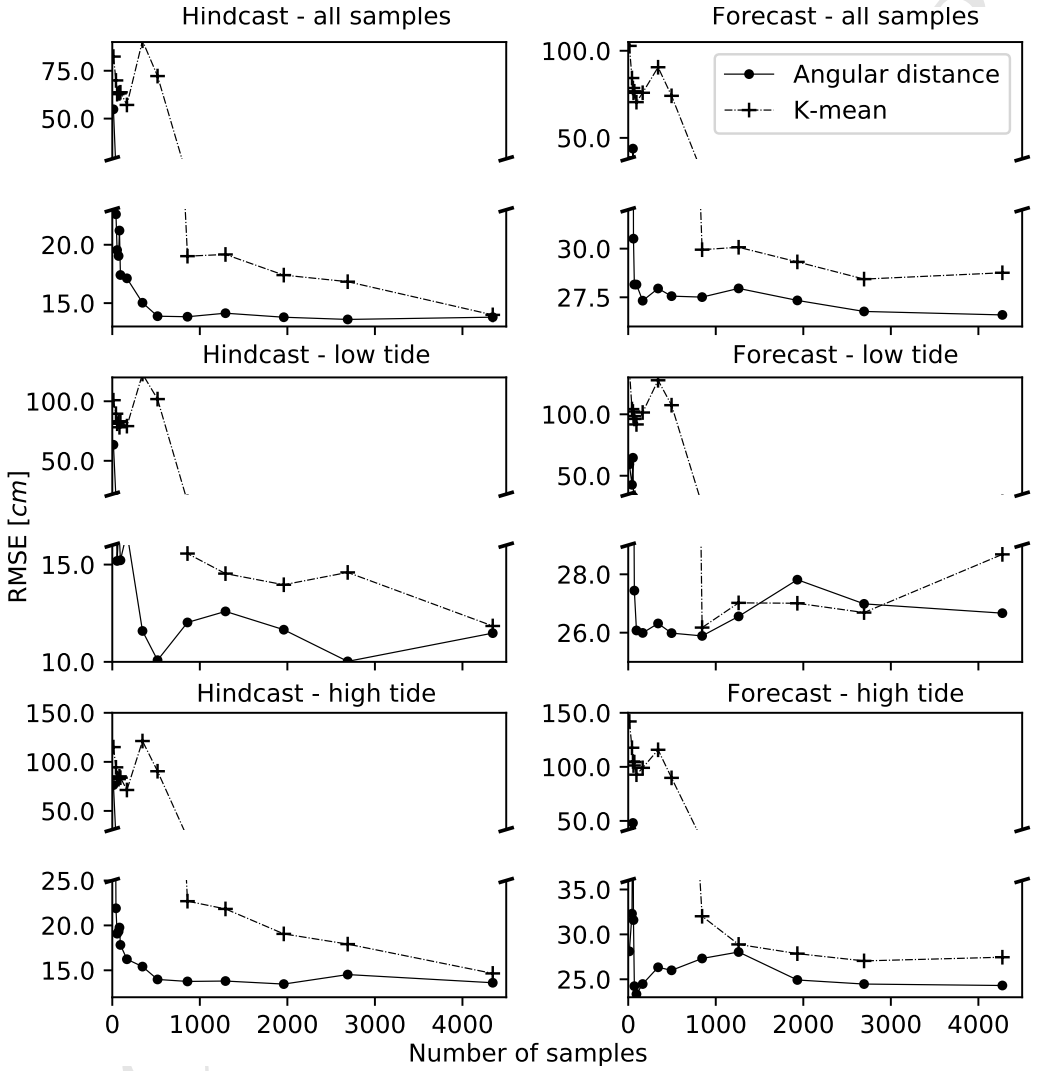


Figure 7: RMSE for different number of elements in the training set using the K-means and the proposed method. a) Hindcast b) Forecast

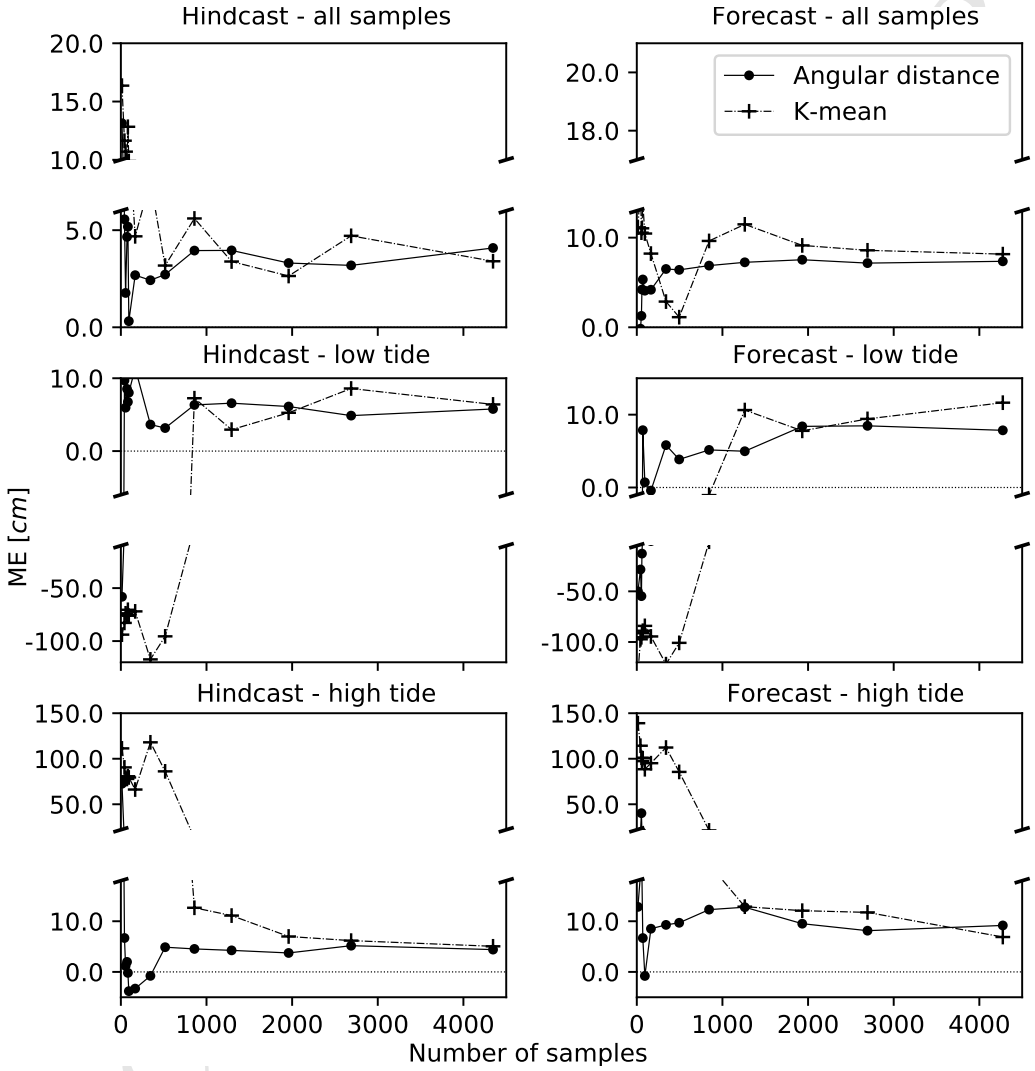


Figure 8: ME for different number of elements in the training set using the K-means and the proposed method. a) Hindcast b) Forecast

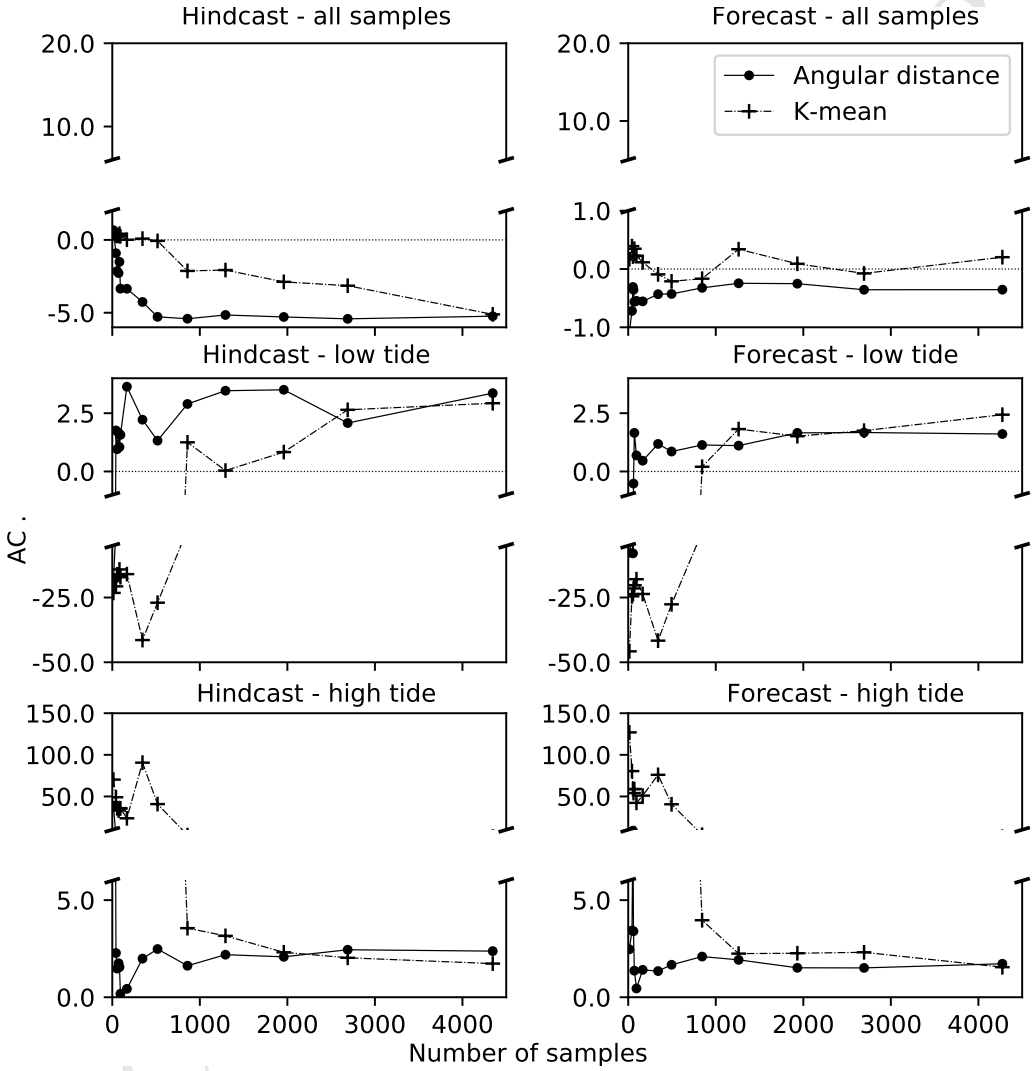


Figure 9: Asymmetry coefficient for different number of elements in the training set using the K-means and the proposed method. a) Hindcast b) Forecast

	Hindcast		Forecast	
Par.	Ang.	K-m.	Ang.	K-m.
cos angle	0.920	-	0.920	-
N samples	859	859	844	844
RMSE	13.8	22.7	27.3	32.0
MAE	10.4	16.8	20.5	25.4
ME	4.5	12.7	12.3	21.3
AC	1.6	3.6	2.1	4.0

Table 2: Error comparison for high tide

	Hindcast		Forecast	
Par.	Ang.	K-m.	Ang.	K-m.
cos angle	0.920	-	0.920	-
N samples	859	859	844	844
RMSE	12.0	15.6	25.9	26.1
MAE	9.0	12.6	19.3	19.8
ME	6.3	7.3	5.2	-1.0
AC	2.9	1.3	1.1	0.2

Table 3: Error comparison for low tide

The presented values correspond to the first K-means results with a reasonable RMSE (859 clusters for hindcast and 844 clusters for forecast) and two values for the angular distance method. The ANN trained with the smaller number of samples from the angular distance that has a lower RMSE than the K-means (99 samples that corresponds to $cos = 800$ for the hindcast and 84 samples with $cos = 0.775$ for the forecast) and also the model that trained with the same number of elements from the angular distance and K-means. In both cases the results from the proposed method are better than the K-means for the RMSE, MAE and ME, while the AC is better for the K-means. It is important to note that the angular distance method only requires 11% of the samples used by the K-means to obtain the same quality in trained ANNs (0.93% of the total number of validation samples). Comparing the computational time, it is approximately 18 times faster. On the other hand, if the number of elements are kept equal, a similar computational time is expected with lower errors provided by the angular distance method.

Table 2 presents only the values for the high tide for the angular distance and the K-means using the same number of elements. It is clear that at this point the angular distance performs much better than the K-means both for hindcast and forecast. For most of the error criteria there are no improvements for the angular distance if more samples are included beyound this point. In the case of K-means, improvements can be achieved by using more clusters, but at a higher computational cost.

Similarly Table 4.3 presents only the values for the low tide were both methods use the same number of elements. For the selected values, the angular distance method is better than the K-means even though the difference is not as big as for the high tide. For the forecast, the proposed method is slightly better than the K-means with respect to the RMSE and MAE but not for the ME and AC. In the case of ME and AC values, it is a coincidence that 844 clusters is a transition case were the values of both of them in the low tide are passing from big negative values to positive numbers (see Fig. 8 and Fig. 9). Therefore these values are not conclusive about the performance of the method in this case.

5. Conclusions

The angle between vectors is an easy method to select samples that show spreading in the space of the inputs. The proposed method enables to reduce the redundancy of information, avoiding the use of two or more samples that are similar to each other. As the selected samples are in general those that differ more from others, the resulting training dataset contains a higher diversity and is able to produce an ANN with a similar performance than networks trained with more inputs. The reduced number of samples has a clear advantage to speed-up the training process. This is relevant in the design of the architecture of a new ANN where the number of input nodes and hidden layers are still to be determined and several scenarios need to be tested to find an optimal configuration. After the architecture is defined, in some cases it is possible to increase the number of samples for the final model if still smaller errors can be obtained in a reasonable time. The proposed method is more efficient capturing extreme values than the K-means. When comparing with the clustering algorithm K-means, the proposed method is clearly providing not only an smaller dataset (with its great benefit in reducing computational time) but also an smaller RMSE.

6. Acknowledgments

The authors want to thanks Bibiana López Rodríguez and Marco Antonio Chavarria Fabila for their help and ideas to improve the proposed method.

This research did not receive any specific grant from funding agencies in the public, commercial or non-profit sectors.

7. References

B. Bazartseren, G. Hildebrandt, and K.-P. Holz. Short-term water level prediction using neural networks and neuro-fuzzy approach. *Neurocomputing*, 55(34):439 – 450, 2003. ISSN 0925-2312. doi: 10.1016/S0925-2312(03)00388-6. Evolving Solution with Neural Networks.

Z. Bowles, P. E. Tissot Tissot, P. Michaud, and A. Sadovski. Artificial neural network predictions of water levels in a gulf of mexico shallow embayment. *Revista de Matemática: Teoría y Aplicaciones*, 12(1-2):139–150, 2012. ISSN 2215-3373. doi: 10.15517/rmta.v12i1-2.258.

G. E. P. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and Control (Revised Edition)*. Holden-Day, 1976. ISBN 0816211043.

S. Ding, Q. Wu, and Y. Yang. Research on the application of rbf neural network based on k-means clustering in system identification. In *The Fourth International Workshop on Advanced Computational Intelligence*, pages 110–112, Oct 2011. doi: 10.1109/IWACI.2011.6159984.

K. Faraoun and A. Boukelif. Neural networks learning improvement using the k-means clustering algorithm to detect network intrusions. *INFOCOMP*

Journal of Computer Science, 5(3): 28–36, 2006. ISSN 1982-3363. URL `http://www.dcc.ufla.br/infocomp/index.php/INFOCOMP/article/view/140`.

N. Fernández, W. Jaimes, and E. Altamiranda. Neuro-fuzzy modeling for level prediction for the navigation sector on the magdalena river (colombia). *Journal of Hydroinformatics*, 12(1):36–50, 2010. ISSN 1464-7141. doi: 10.2166/hydro.2010.059.

J. M. Fernández Jaramillo. *Development of an operational system for a coastal area on the German North Sea using artificial intelligence*. PhD thesis, University of Kiel, 2014.

Deepthi.I. Gopinath and G.S. Dwarakish. Wave prediction using neural networks at new mangalore port along west coast of india. *Aquatic Procedia*, 4:143 – 150, 2015. ISSN 2214-241X. doi: 10.1016/j.aqpro.2015.02.020.

G. J. Komen, L. Cavaleri, M. Donelan, K. Hasselmann, S. Hasselman, and P. A. E. M. Jansse. *Dynamics and Modelling of Ocean Waves*. Cambridge Univ. Press, 1996.

B. Krishna, Y. R. Satyaji Rao, and P. C. Nayak. Time series modeling of river flow using wavelet neural networks. *Journal of Water Resource and Protection*, 03(01):50–59, 2011. doi: 10.4236/jwarp.2011.31006.

M. S. Longuet-Higgins. Longshore currents generated by obliquely incident sea waves: 1. *Journal of Geophysical Research*, 75(33): 6778–6789, 1970. ISSN 2156-2202. doi: 10.1029/JC075i033p06778.

M. Lunani, A. Sudjianto, and P. L. Johnston. Generating efficient training samples for neural networks using latin hypercube sampling. In *Intelligent Engineering Systems Through Artificial Neural Networks*, volume 5, pages 209–214. ASME Press, 1995. ISBN 0791800482.

M. Macas, F. Moretti, A. Fonti, A. Giantomassi, G. Comodi, M. Annunziato, S. Pizzuti, and A. Capra. The role of data sample size and dimensionality in neural network based forecasting of building heating related variables. *Energy and Buildings*, 111:299 – 310, 2016. ISSN 0378-7788. doi: 10.1016/j.enbuild.2015.11.056.

R. J. May, H. R. Maier, and G. C. Dandy. Data splitting for artificial neural networks using som-based stratified sampling. *Neural Networks*, 23(2):283 – 294, 2010. ISSN 0893-6080. doi: 10.1016/j.neunet.2009.11.009.

A. P. Piotrowski, M. J. Napiorkowski, J. J. Napiorkowski, and M. Osuch. Comparing various artificial neural network types for water temperature prediction in rivers. *Journal of Hydrology*, 529, Part 1:302 – 315, 2015. ISSN 0022-1694. doi: 10.1016/j.jhydrol.2015.07.044.

L. Prechelt. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11(4):761 – 767, 1998. ISSN 0893-6080. doi: 10.1016/S0893-6080(98)00010-0.

M. Riedmiller. Rprop - description and implementation details. Technical report, Institut fr Logik, Komplexitt und Deduktionsysteme, University of Karlsruhe, W-76128,University of Karlsruhe, January 1994.

A. K. Sahoo, M. J. Zuo, and M.K. Tiwari. A data clustering algorithm for stratified data partitioning in artificial neural network. *Expert Systems with Applications*, 39(8):7004 – 7014, 2012. ISSN 0957-4174. doi: 10.1016/j.eswa.2012.01.047.

M. Tayyab, J. Zhou, X. Zeng, and R. Adnan. Discharge forecasting by applying artificial neural networks at the jinsha river basin, china. *European Scientific Journal*, 2016. doi: 10.19044/esj.2016.v12n9p108.

B. Tirozzi, S. Puca, S. Pittalis, A. Bruschi, S. Morucci, E. Ferraro, and Corsini S. *Neural Networks and Sea Time Series: Reconstruction and Extreme-Event Analysis*. Birkhuser, 2006.

F. Tong and X. Liu. Samples selection for artificial neural network training in preliminary structural design. *Tsinghua Science and Technology*, 10(2):233–239, 2005. ISSN 1007-0214. doi: 10.1016/S1007-0214(05)70060-2.

Lin Wang, Yi Zeng, and Tao Chen. Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. *Expert Systems with Applications*, 42(2):855 – 863, 2015. ISSN 0957-4174. doi: 10.1016/j.eswa.2014.08.018.