# Predictive model for the horizontal displacement of a dam using autoregressive neural network

Gabriel OLTEAN*, Laura-Nicoleta IVANCIU*, Mihaela GORDAN*,

Ioan STOIAN** and Istvan KOVACS**

* Technical University of Cluj-Napoca, Cluj-Napoca, Romania

** Cluj-Napoca Subsidiary IPA R&D Institute for Automation, Cluj-Napoca, Romania

gabriel.oltean@bel.utcluj.ro, laura.ivanciu@bel.utcluj.ro, mihaela.gordan@com.utcluj.ro, stoian@automation.ro, kovacsi1000@yahoo.com

***Abstract*—The interpretation of data gathered from dam monitoring directly influences the detection of abnormal behaviors. Using previously recorded data, predictive models can be developed, so that the signs of a possible failure are detected as early as possible. The paper presents a multi-step ahead predictive model to generate the values for the horizontal displacement of a dam, using previous values of the displacement, water level and temperature. The model is based on an autoregressive neural network that was trained and tested using historical data. The results show a good prediction accuracy (maximum 2.63% relative errors), especially for up to 8 months ahead prediction).**

***Keywords—dam monitoring; horizontal displacement; time series, NARX, autoregressive neural network, multi-step ahead prediction***

## I. INTRODUCTION

Dam monitoring is crucial for safety reasons. Many cases of dam failures could have been avoided, had a monitoring solution been available [1]. A dam monitoring instrument should be able to keep track of the current behavior of the dam, record data and detect any abnormal activity in the functioning of the dam. The parameters that are being monitored and the frequency of the measurements should be sufficient to detect the abnormality, but not too many, as to become redundant or over-detailed Different measurement frequency is required for different parameters (e.g., in embankment dams, seepage measurements are taken more often than horizontal displacement) [2] [3].

The collected data can also be used to develop predictive models for estimating the future behavior of the dam and predict possible abnormalities. Since the collected data comes from various types of measurements, data preprocessing and fusion is required in order to correctly interpret the information [4]. Many data fusion algorithms employ the use of computational intelligence techniques. In [5], a fusion strategy based on fuzzy reasoning for the interpretation of visualization data is developed, in order to detect a safe landing site. Machine learning algorithms and computational intelligence techniques perform data fusion in [6], for medical image analysis and classification.

A dynamic system identification solution using nonlinear autoregressive exogenous (NARX) model structures with functional expansion of input patterns is proposed in [7]. A six months ahead prediction of the ionosphere parameters is achieved in [8], using neural networks and historical data (past values of the IR5 index). Bayes theory is employed in developing a mathematical model for establishing the abnormal behavior of a dam, using data from multiple monitoring points, in [9]. While single-point monitoring models provide local information, a multi-point model is able to provide an overall description of the dam behavior. Data collected from 7 monitoring points is used to build a model that achieves fusion analysis and diagnosis of abnormal behavior.

According with our knowledge, there are not systematic approaches of developing predictive models for dam displacement using different types of information.

This work describes the development of a predictive model for the horizontal displacement of a dam, using an auto-regressive neural network and historical data regarding the water level, temperature and horizontal displacement. The paper is organized as follows: Section II presents an overview of the proposed model, Section III describes the implementation details and discusses a series of results, while Section IV concludes the paper.

## II. OVERVIEW OF THE PROPOSED MODEL

The problem to be solved here falls in the domain of nonlinear system identification, to develop a predictive model which is useful for multi-step-ahead prediction (in the close loop form). The best candidate to successfully solve this challenge is a nonlinear autoregressive exogenous model (NARX). This model associates the output values (of a time series) with past values in the same time series, but also with current and past values of some input (exogenous) series. The algebraic equation that define this predictive model is:

$$\begin{aligned} y(t) = F(&y(t-1),...,y(t-d_y),\\ &u_1(t),\ u_1(t-1),...,u_1(t-d_1),\\ &u_2(t), u_2(t-1),...,u_2(t-d_2),\ ...,\\ &u_n(t),\ u_n(t-1),...,u_n(t-d_n)) \end{aligned} \tag{1}$$

where $y$ is the output, $u_1$, $u_2$, …, $u_n$ are the exogenous inputs that affect the output, and $d_y$, $d_1$, $d_2$, $d_n$ are the numbers of delays corresponding with each variable.

The most important aspect in developing such a model is to select the F function that can implement the nonlinear mapping between the variables as accurately as possible. Since a multilayer feed-forward neural network has the potential to be a universal approximator, we select such a solution here for the F function. The general block diagram for the nonlinear

autoregressive exogenous model based on an artificial neural network that implements relation (1) is presented in Fig. 1.

Multi-step long term predictions with dynamic modelling are suitable for complex system prognostic algorithms since they are faster and easier to calculate, compared to various other prognostics methods. The recurrent neural networks, therefore, have been widely employed as one of the most popular data-driven prognostics methods and a considerable number of studies across different disciplines have stated their merits, by introducing different methodologies [10].

For the NARX model, the output is fed back to the input of the feed-forward neural network. Because the true output (target) is available during the training of the network, one could create a series-parallel architecture, in which the true output is used as input, instead of feeding back the estimated output. This has two advantages. The first is that the input to the feed-forward network is more accurate. The second is that the resulting network has a purely feed-forward architecture, and static back-propagation can be used for training [11].

The flowchart of the process of building the predictive model is presented in Fig. 2. Developing the ANN requires a supervised learning procedure based on a numerical data set. First, the raw time series needs to be transformed and formatted in an appropriate manner, to be suitable for the training procedure.

For the training procedure, the full data set must be split into three data subsets: training, validation and testing subsets. The training subset will be directly used to adapt the neural network parameters (weights of the connections between neurons and neuron biases) in each training epoch. The validation subset supervises the training, to detect a possible over-fitting phenomenon. Finally, the testing subset measures the performance of the neural network, since it is not involved in the training process. Moreover, some real data should be kept as checking data, to verify the accuracy of the multistep ahead prediction, by using the model with the closed-loop ANN.

The architecture of the open-loop ANN is first decided upon: a multi-layer perceptron with an input layer, a hidden layer and an output layer. According with [12], where it is stated that, from a function approximation perspective, the single hidden layer is quite adequate as the basic topology and since a two-layer feed-forward network (one hidden layer and one output layer) with sigmoid hidden neurons and linear output neurons, can fit multi-dimensional mapping problems arbitrarily well [11], this is the solution for the neural network architecture adopted here. Also, in this point it is necessary to set the delays for both the exogenous inputs and the output (feedback). For the hidden layer, the number of neurons will be determined via a series of trial runs, in order to obtain an optimal network structure.

Because of possible different delays and the existence of the feedback, the data should be prepared to correctly define the feedback output's targets, layer delay states, and input data. The open-loop ANN is then trained and its performances are evaluated. Each time a neural network is trained, a different solution can result, due to different initial weight and bias values and different split of data into training, validation, and test sets. As a result, different neural networks trained on the same problem can give different outputs for the same input [11]. To ensure that an optimal neural network of good accuracy and acceptable complexity has been found, we must retrain several times, using the same network architecture, or modifying the data division, or acting on the network architecture (number of hidden neurons, numbers of delays).

Once the training and validation process is finished, the last step is to convert the open-loop neural network to the to the closed-loop configuration, which is further used for multi-step-ahead prediction – the predictive model.

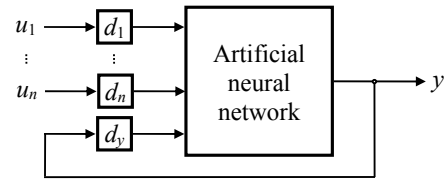


Figure 1. The block diagram of the nonlinear autoregressive exogenous model (NARX), based on a neural network

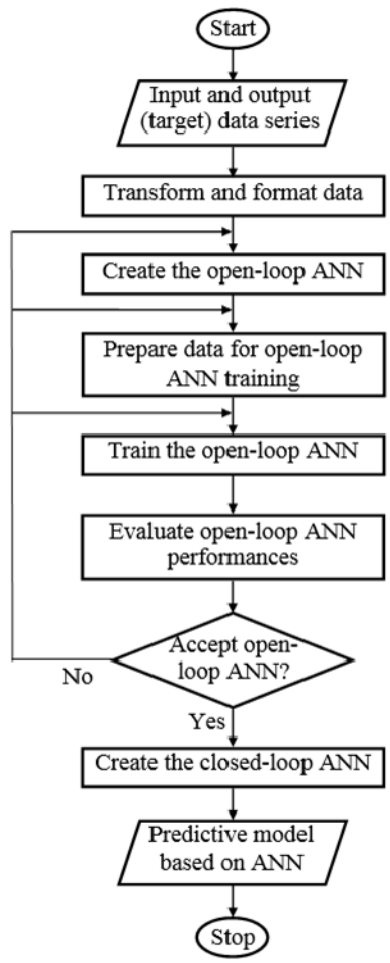


Figure 2. The flowchart of the process of developing the predictive model

## III. Implementation and Experimental Results

Predicting the horizontal displacement of a dam can be a quite difficult task, because the actual value depends on the previous values of the displacement and other exogenous parameters, such as water level and atmospheric temperature.

Furthermore, the dependence between these parameters is both nonlinear and delayed, meaning that a variation in the current values implies a change of the output value, but not instantaneously. The literature study shows that the best candidate to solve such a problem is a NARX model.

The development and implementation of the predictive model based on ANN was carried out in the MATLAB environment, using the built-in functions available in the Neural Network Toolbox and a series of custom functions and scripts written by the authors.

Our data refers to three correlated time series, as monthly average of measured values during the exploitation period of a hydrographic dam. The first time series contains the values of the water level at the upstream side of the dam. The second time series contains the air temperature, temperature that directly influences the ambient temperature of the dam through its diurnal and seasonal variations. The last time series refers to the horizontal displacement of the dam crest in its central zone, on the upstream-downstream direction. All original time series are normalized in a [*LB*; *HB*] range using the relation:

$$Xnorm = LB + (HB - LB)\frac{X - X\min}{X\max - X\min} \quad (2)$$

where *LB* and *HB* are the low bound and respectively the high bound of the normalized range; *Xnorm* is the normalized value; *X* - is the original value; *X*min and *X*max – are the minimum and respectively the maximum values in the original data. For all the time series involved here, the same normalized range is used, namely *LB* = 10 and *HB* =100. The normalized time series are represented in Fig. 3, where *Lnorm* stands for normalized value of water level, *Tnorm* stands for normalized value of temperature, while *Dnorm* stands for normalized value of horizontal displacement.

The length of the data series is 456, meaning 456 successive monthly data. The dependent variable, the one that should be predicted, is the horizontal displacement of the dam crest in its central zone. One can see that the displacement presents some yearly seasonality, as long as there is a pattern that tends to repeat (with some fluctuations) at every 12 months. By visual analysis of the data it is observed that the horizontal displacement seems to be well correlated with the temperature, with a delay of 2 to 3 months. Also, there is a dependence of horizontal displacement with the water level. This last dependence is more obvious for lowest values of water level, as it happens for example for the 273rd month where it is the minimum normalized water level (17.88), which leads to a minimum of normalized displacement (40.35).

The data series should be split into training, validation and testing subsets. Considering the idea of the dynamic system, where the current value of the output depends on some previous values of inputs but also of some previous values of output itself, the data split should be treated in a controlled manner, not purely random. Consequently, we select the first 384 data, from the 1st month to the 384th month as the training subset; the next 31 data, from the 385th month to the 415th month as the validation subset; the next 28 data, from the 416th month to the 443rd month as the testing subset. The remaining 13 data, from the 444th month to the 456rd month are stored as independent checking data, for the final predictive model.

The open-loop neural network is created by selecting one hidden layer and determining the number of neurons in the hidden layer and the number of time delays for each input. After a trial-and-error process, the accepted neural network presents 7 neurons in the hidden layer, with hyperbolic tangent sigmoid activation function and one neuron with linear activation function in the output layer. For the inputs, there are: one delay for the exogenous input *Lnorm*, three delays for the exogenous input *Tnorm* and three delays for the feedback input *Dnorm* (Fig. 4).

The performance validation graph is illustrated in Fig. 5, where the evolution of mean squared error (mse) within the data subsets is depicted. To deal with the large decrease of the mse (from 718 down to 3.3104 in the validation subset) improving the neural network performance during the training process, a logarithmic scale is used for the vertical axis. In the first 7 epochs, there is a steep improvement of the mse in all three data subsets. A "fine tuning" phase follows, up to the 93rd epoch, where the best validation performance is achieved. After the epoch 93, one ca notice that the over-fitting phenomenon is installed, the mse for the validation subset presents a slightly increasing trend during the next 20 epochs. The optimum trained ANN is then considered to be the one at 93 epoch that assures the tradeoff between the minimum error and generalization capability.

The trained neural network has appreciable fitting performances for all data subsets. This can be seen by analyzing the linear regression of the outputs of the ANN relative to the targets, as they are presented in Fig. 6. The regression value *R* that indicates the "goodness of fit" is very close to the maximum possible value (1 for perfect fit): 0.98391 in the training subset, 0.97994 in the validation subset, 0.96698 in the test subset, and 0.98349 across all data.

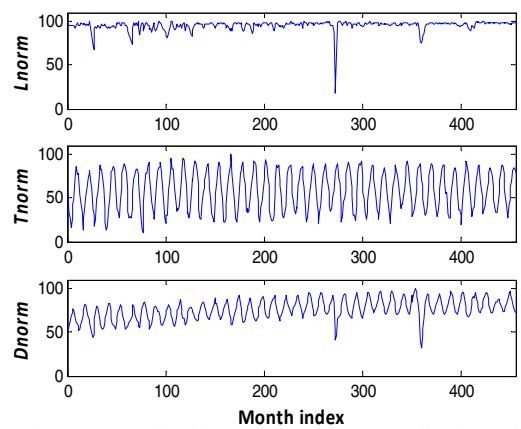


Figure 3. Normalised date series: *Lnorm* – normalized water level; *Tnorm* – normalized temperature; *Dnorm* – normalized horizontal displacement

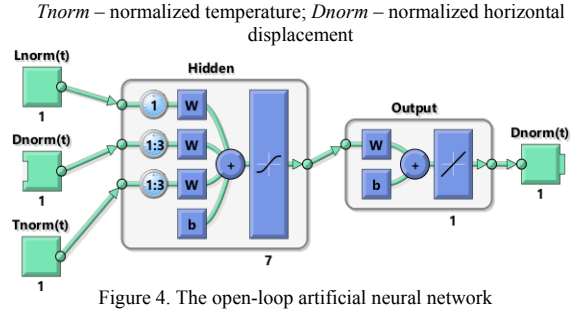


Figure 4. The open-loop artificial neural network

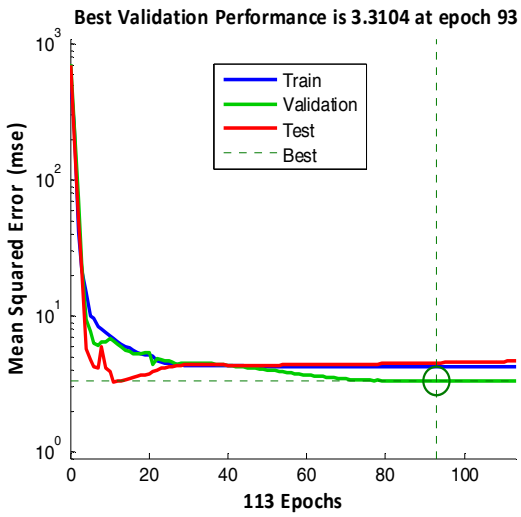


Figure 5. Performance validation graph

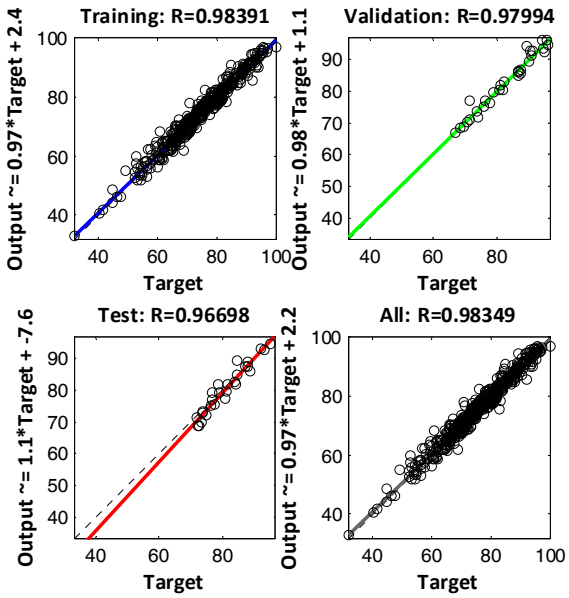


Figure 6. Linear regression for trained ANN

Fig. 6 also gives a qualitative appreciation of the prediction accuracy. Ideally, all data points should be placed exactly on the first bisector. Our results show that all data points are placed in the right places, very close to the bisector.

The time series response of the open-loop trained neural network is presented in Fig. 7. According with the initial data split, the first 384 data points (months) correspond to the training subset (blue points). All errors, computed as the difference between the reference value (targets) and predicted values (output) falls in a narrow range.

The largest positive error is 7.08 for month index 72, where the reference value is 69.79, while the predicted value is 61.71 (relative error of +11.58%). The largest negative error is -7.35 for month index 176, where the reference value is 60.91, while the predicted value is 68.26 (relative error of -12.07%).

For the validation subset – in green (month index from 385 to 415), the largest positive error, 3.06 appears for the month index 412, where the reference value is 95.38 and the predicted value is 92.32 (+3.21%); while the largest negative error is -5.44 for month index 407, where the reference value is 71.47 and the predicted value is 76.91 (relative error of -7.61%).

For the test subset (month index from 417 to 443 – red points), the behavior of the open-loop network is similar as for the two previous subsets. The largest positive error, 3.75 appears for the month index 417, where the reference value is 72.57 and the predicted value is 68.82 (+5.17%); while the largest negative error is -4.7 for month index 434, where the reference value is 84.51 and the predicted value is 89.22 (relative error of -5.28%).

As a general appreciation, the response of our NARX-based predictor (analyzing the regression values $R$ – see Fig. 6 in a correlation with individual errors – see Fig. 7) is a very good one. It is better to have a prediction model that present many small errors that the one that present only few big errors; this means that the model is not over fitted and it encapsulates the necessary knowledge extracted from training data to be a reliable one, with appreciable generalization capabilities.

To be used as it was intended as an independent multistep ahead prediction model, the NARX model should be operated in closed-loop configuration. For that, feedback input should be connected to the output of the neural network, as it is shown in Fig. 8.

To finally validate the prediction model, we used it to preform multistep ahead prediction for the normalized displacement, using independent data series that was not at all involved in the neural network training process, data series that also contain the reference normalized displacement.

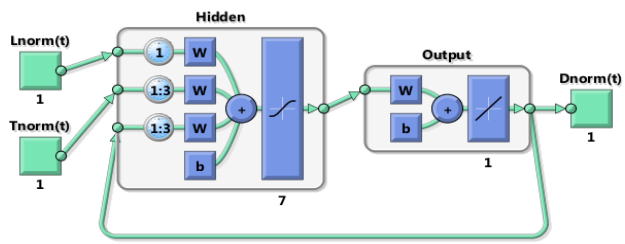


Figure 8. The closed-loop prediction model

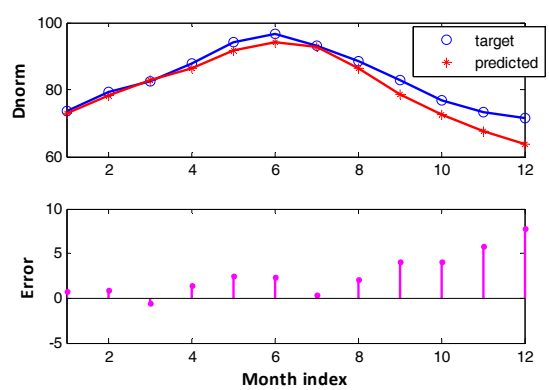


Figure 9. Multi-step ahead prediction for the normalized displacement. Comparison with reference values.

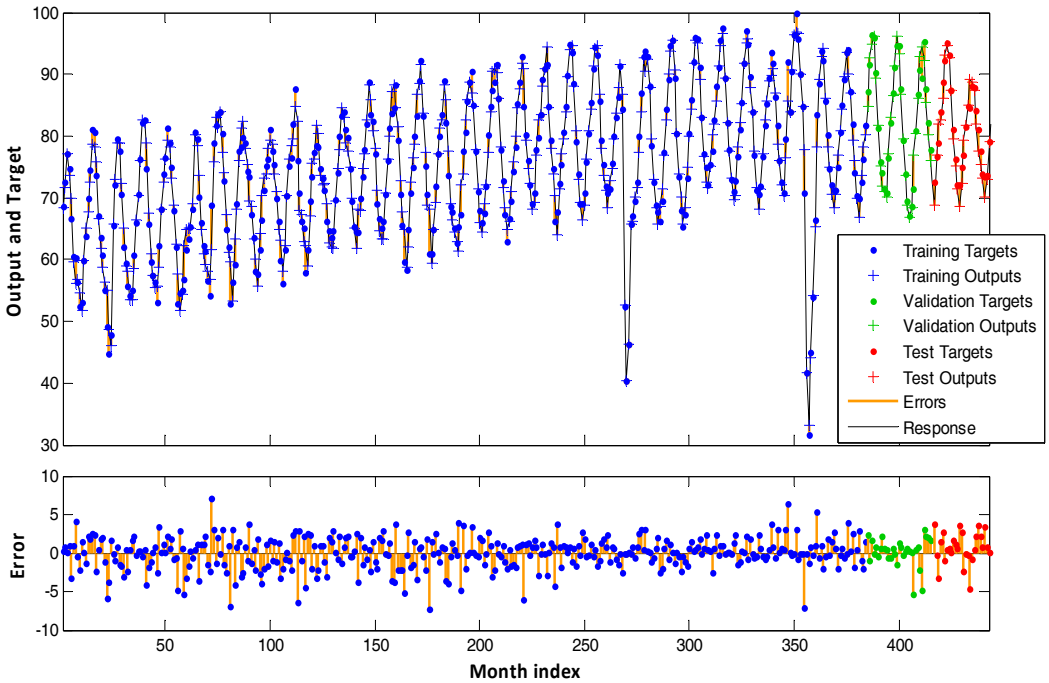


Figure 7. Normalized displacement: predicted values (Output) vs. reference values (Target)

The graphical representation of the predicted values for 12 month (one year) against the reference (target) values is illustrated in Fig. 9. The prediction accuracy is very high, especially in the first eight months, with a maximum error of 2.48 for month index 5, where the reference value is 94.24 while the predicted value is 91.76 (relative error of +2.63%). The errors for the last 4 month seems to increase progressively, that is a normal behavior due to the fact that these values are calculated using for the feedback input the values predicted in the previous step, values that already can differs from the reference values.

## IV. Conclusions

The paper presents a multi-step ahead predictive model to generate the values for the horizontal displacement of a dam. This model can be used as an efficient tool in the complex process of dam monitoring and surveillance as long as it is capable to predict with confidence the horizontal displacement of a dam using past values of horizontal displacement itself, and also current and past values for two exogenous inputs: water level and temperature. The results show a good prediction accuracy (maximum 2.63% relative errors), especially for up to 8 months ahead prediction).

The main advantage of this approach is that it systematically uses the current and previous values of the displacement, as well as of two exogenous inputs – water level and temperature, in order to obtain a reliable predicted value of the future horizontal displacement of the dam. Using the prediction as a model of the dam behavior, abnormalities can be detected and prevented. Another possible use of the model is to manage the water level, based on the weather reports and the predicted value of the dam displacement. Future developments include investigating whether the model can be transformed, by realizing a switch between the input and output - to estimate the appropriate water level (output, former input), based on atmospheric temperature and allowed horizontal dam displacement (input, former output).

## Acknowledgment

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI-UEFISCDI, Project DAMFU, PN-III-P2-2.1-PTE-2016-0134, 45PTE/2016, 2016-2018.

## References

[1] Dam Safety Monitoring 101: Monitoring the health of dams. Why & how we do it [online] https://www.ferc.gov/industries/hydropower/safety/initiatives/surveillance-monitoring/monitoring.pdf

[2] A.R. Chavan and S.S. Valunjkar, "A Study of instruments used for dam instrumentation in gravity and earthen gravity and dams", *Intl. Journal of Engineering and Technical Research IJETR*, vol. 3, issue 5, pp. 355 – 361, May 2015.

[3] Collection Analytics Reporting Storage Application (CARS), Dam Safety [online] http://trinitytg.com/wp-content/uploads/2014/11/CARS-Dam-Safety2.pdf

[4] Dam Safety Performance Monitoring Program, Ch. 14, 2017 [online] https://www.ferc.gov/industries/hydropower/safety/guidelines/eng-guide/chap14.pdf

[5] L.M. Goncalves Miranda, "Data Fusion with Computational Intelligence techniques: a case study of Fuzzy Inference for terrain assessment," Dissertation Thesis, 2014.

[6] B. Cheng, "Data fusion by using machine learning and computational intelligence techniques for medical image analysis and classification," Doctoral Thesis, 2012.

[7] H.K. Sahooa, P.K. Dash, and N.P. Rath, "NARX model based nonlinear dynamic system identification using low complexity neural networks and robust H∞ filter", *Applied Soft Computing*, vol. 13, pp. 3324-3334, 2013.

[8] F. Fessant, S. Bengio and D. Collobert, "On the prediction of solar activity using different neural network models", *Annales Geophysicae*, vol. 14, issue 1, pp. 20-26, January 1996.

[9] H. Jin-ping, T. Yuan-yuan and S. Yu-qun, "Fusion model of multi monitoring points on dam based on Bayes theory", *Procedia Engineering*, vol. 15, pp. 2133-2138, 2011.

[10] O. Bektas, J. A. Jones, "NARX time series model for remaining useful life estimation of gas turbine engines", *Third European Conference of the Prognostics and Health Management Society*, pp. 1-10, 2016.

[11] Neural Network Toolbox, The MathWorks, Inc. [online] http://www.mathworks.com/help/nnet/index.html

[12] J. Principe and B. Chen, "Universal approximation with convex optimization: gimmick or reality? [Discussion Forum]," *IEEE Comput IntelMag,* vol. 10(2), pp. 68-77, 2015.