

## One Day Ahead Stream Flow Forecasting

Joao P. Carvalho<sup>1,2</sup> Filipe V. Camelo<sup>2</sup>

<sup>1</sup>INESC-ID

<sup>2</sup>Instituto Superior Técnico, Universidade de Lisboa, Portugal

joao.carvalho@inesc-id.pt

fvcamelo@gmail.com

### Abstract

Short-term stream flow forecasts are required for simulation, optimization, and decision-making purposes in applications ranging from hydropower planning to flood prevention. The particular case of one-day ahead stream flow forecasting is an important but difficult problem that has been increasingly studied using hybrid computational intelligence and machine learning techniques. However, these studies present several limitations. In this work we attempt to address those limitations by (1) replicating and validating previous works; (2) using more objective evaluation criteria; (3) applying several computational intelligence techniques to datasets representative of diverse geographic areas; (4) preprocessing data and performing an extensive parameter optimization in order to improve previous results.

**Keywords:** Stream flow forecasting; One step-ahead forecasting; ANFIS; Artificial Neural Networks; Support Vector Machines.

### 1. Introduction and related work

Fresh water is both essential for human life support and a crucial resource for multiple industries. The planning and operation activities concerning fresh water resources, especially of fresh water streams, are becoming increasingly important due to human population growth and environmental awareness. As such, short-term stream flow forecasts are required for simulation, optimization, and decision-making purposes. The particular case of one-day ahead stream flow forecasting consists on the prediction of the next day flow of a river given current and past stream flow observations on a single location. It is a difficult problem for several reasons:

- Stream flows are highly non-linear and dependent on geographic location, climate and topography of the region of the watershed, reservoir or hydrologic basin;
- Stream flows are often of non-stationary nature due to wet and dry periods over the year;

- Streams often have sudden floods that result in very steep flow slopes. These flow peaks often last a single day and can be several orders of magnitude larger than the regular flow, making them a huge forecasting challenge.

Both hybrid computational intelligence and machine learning techniques have proved to be rather successful when approaching one-step ahead forecasting of non-linear time-series (e.g. [8][9][16][0]). It is therefore natural that such black-box techniques have been previously implemented with the purpose of stream flow forecasting. The existing literature uses techniques such as Artificial Neural Networks (ANN), Adaptive Neuro Fuzzy Inference Systems (ANFIS), and also Support Vector Machines (SVM) [7][22][24][12]. Other approaches, such as [14] use predictive fuzzy clustering but only for average monthly flow predictions, which present a smoothed behavior and therefore address a different problem. Despite the many studies, most, if not all, have several limitations:

1. They only address a particular stream/river, therefore not taking into account the diversity of climates, topographies, basins, etc. Therefore, their conclusions cannot be extrapolated/generalized to other stream flows – even if they often do it.
2. They do not present an objective measure of the quality of the forecast. They might compare several techniques, but no ground truth (or baseline) is usually presented. Therefore, even if one technique is better than the others, it can still be inadequate for one-day ahead stream flow forecast.
3. High errors when dealing with sudden flow changes: none of the presented studies can forecast sudden floods.

Overall, the presented results are often misleading: the resulting forecasting graphs might seem impressive, but this is usually only due to the used representation scale, since most forecasts exhibit a constant displacement between the forecast and the observed flow value. If the graphs are zoomed in, it is possible to observe at least one day delay between the observed flow and the forecast, especially when there is a sudden flow change (see section 2).

In this work we attempt to address the above mentioned limitations by (1) replicating and validating the previous works; (2) creating baselines for each dataset in

---

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) under project PTDC/IVC-ESCT/4919/2012 and funds with reference UID/CEC/50021/2013.

order to have a more objective evaluation criteria; (3) applying the several techniques on several datasets representative of diverse geographic areas; (4) preprocessing the datasets and performing an extensive parameter optimization for all black box techniques in order to improve previous results.

The paper is organized as follows: Section 1 describes the addressed problem; In section 2 we described the implemented techniques; In section 3 we defend the need for a ground truth and propose the use of a simple baseline for result evaluation; In section 4 we describe the used datasets; Section 5 focus on data preprocessing; Section 6 describes the implemented models and their optimization; Section 7 shows and discusses the obtained results; and finally, section 7 presents the conclusions.

## 2. Methods

We used three different computational intelligence forecasting techniques to address this problem: Artificial Neural Networks (ANN), Adaptive Neuro Fuzzy Inference Systems (ANFIS), and Support Vector Machines (SVM).

Artificial neural networks (ANNs) are a family of statistical learning algorithms originally inspired by biological neural networks that can be used to estimate or approximate non-linear functions with an arbitrary number of inputs. Artificial neural networks are generally presented as systems of interconnected "neurons" which can compute values from inputs, and are capable of learning and of pattern recognition. The ANN model used in this study was implemented using Matlab Neural Network toolbox (version 7.0.1) [15] and all the simulations were performed under this environment.

ANFIS is a kind of artificial neural network based on a Takagi–Sugeno fuzzy inference system introduced by Jang [11]. Since it integrates both neural networks and fuzzy logic principles, it has the potential to capture the benefits of both in a single framework. Its inference system corresponds to a set of fuzzy IF–THEN rules that have learning capability to approximate nonlinear functions. For this study the ANFIS model was implemented in the Matlab environment using the functions available from the fuzzy logic toolbox.

Support vector machines (SVMs) [3] are supervised learning models with associated learning algorithms that analyze data and recognize patterns, and can be used for classification and, as in our case, adapted for regression analysis. The SVM was implemented using LibSVM [1] under the Matlab environment.

## 3. Baseline

Previous works usually use error metrics such as the Root Mean Square Error – RMSE (see section 5.4) as indicators for the performance of the forecast. Usually these metrics are good enough to compare different approaches,

but are not very relevant in terms of absolute value in the absence of a ground truth, i.e., of some comparison baseline.

Autoregressive–moving-average (ARMA) and other Box & Jenkins models 0 are the most used techniques for monthly stream flow forecast in hydic energy planning, and could be considered as a baseline. However, such methods are not well adapted to non stationary time series, and even though they can give acceptable results for accumulated monthly forecasts, they simply are not adequate to the one day ahead forecast problem, where flow variations have a much more complex dynamics. In [14] it can be seen that these methods provide acceptable errors in "well behaved" months, but the error escalates much more than when using non-linear models in months with high flow variations.

A very simple baseline consists in using the assumption that next day flow will be the same as the current day observation. One of the major pitfalls of the existing studies is ignoring the fact that their results might even be worse than this simple baseline, and even recent studies [24] present techniques that simply fail to beat it. This denotes the difficulty of predicting stream flows. It is important to present the results with this metric in mind and set objectives, not only surpassing previous studies metrics, but also certifying that the baseline metric is also improved.

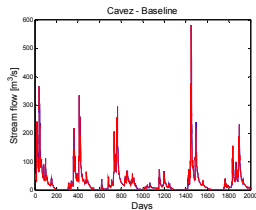


Fig. 1: Baseline line (in red) and original stream flow signal (blue) comparison. The baseline is the assumption  $t + 1 = t$  from one of the datasets used in this study (Cavez).

The chosen baseline can also be used to demystify the commonly used graphs showing the forecast vs. the data series. Fact is that such graphs are not useful due to the size of the datasets (covering several years and containing thousands of data points – see section 3). Let us see the example presented in Fig.1, where is represented the forecast given by the proposed baseline in one of the studied datasets. In blue we can see the dataset, and in red the forecast. In print hardly any differences between the two signals are visible, hinting an excellent performance. However, if instead of showing the forecast for 2000

days, we show a much smaller period, e.g. from day 700 to 800 (Fig. 2), it is clear that we are simply using the proposed baseline, i.e., assuming that the next day flow will be identical to current day observation. Considering that in most previous work the difference between the forecast and the dataset is clearly visible when showing the complete dataset, it is easy to show that these graphs are not good performance indicators unless a proper scale is used.

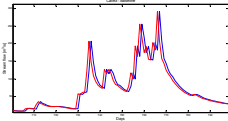


Fig. 2: Baseline (in red) and original stream flow signal (blue) comparison for the Cavez dataset from day 700 to day 800. The one day delay becomes clearly visible.

#### 4. Datasets

The datasets used in this work are representative of regions with different climates. Table 1, presents some relevant flow information regarding these datasets. The Castanheiro dataset has 30 years of training data and 8 for the testing set. The Cavez has 7 years of training and 3 years of testing data. The Pailugou dataset has 3 years of training data and 3 for testing. The Trés Marias has 10 years of training and 3 years of testing.

TABLE I  
TRAINING SETS (MAX, MIN AND MEAN FLOWS IN  $m^3/s$ )

Dataset	Max	Min	Mean	Std dev	Points
Castanheiro	1384	0	45,78	98,28	10958
Cavez	582	0,47	24,0	41,7	2737
Pailugou	9604	4,00	617,7	1046	1095
Trés Marias	4275	48	642,1	678,7	3653

TESTING SETS

Dataset	Max	Min	Mean	Std dev	Points
Castanheiro	1124	0,05	33,2	75,8	2922
Cavez	453	0,98	27,0	44,1	1095
Pailugou	8841	7,29	559,7	1082	1095
Trés Marias	4707	80	702,5	697,0	1461

Table 1 also includes some relevant statistical parameters of the used datasets. It is possible to observe that the maximum flows are highly skewed from the minimum flows, and that the mean is only a small fraction of the maximum flows. The Pailugou dataset has a higher mean due to the high peaks and the dry seasons that have close to zero stream flows. The mean of the Trés Marias dataset is an order of 15% of the maximum, while the others are around 3% to 6%. This is due to a higher and more constant flow throughout the year.

#### 5. Data Preprocessing

In any model development process, familiarity with the available data and its preprocessing is of utmost importance since it usually has a positive effect on the model performance [5][22]. The characteristics of the used datasets (especially the skew toward minimum flows) and the used techniques, led us to test the application of logarithmic transformations and to normalize all input values:

$$q_i^{\log} = \log_{10}(1 + q_i) \quad (13)$$

$$q_i = 10^{q_i^{\log}} - 1 \quad (14)$$

$$Q_{\max, \min} = q_{\max, \min} \times (1 \pm b) \quad (15)$$

$$q_i^* = \frac{q_i - Q_{\min}}{Q_{\max} - Q_{\min}} \quad (16)$$

$$q_i^r = q_i^* \times (Q_{\max} - Q_{\min}) + Q_{\min} \quad (17)$$

(13) represents the logarithmic transformation:  $q_i^{\log}$  is the logarithmic transformed stream flow and  $q_i$  is the real stream flow value; the unitary value is added to all the stream flows in the logarithmic function to avoid fractions. (14) represents the inverse transformation used to transform the logarithmic stream flows into real world values. Before the inputs are introduced into the models they are normalized in order to have values ranging from 0 to 1 (a buffer is used to prevent limit conditions for all used techniques). (16) is used for input normalization, and (17) is used to transform the outputs of the models to real values.

#### 6. Models' Development

All models will take as inputs the observed current day stream flow ( $Q_t$ ), expressed in cubic meter per second ( $m^3/s$ ), and a given number of past  $Q_{t-d}$  observations, with  $d = 1, \dots, j$  being the previous  $j$  day stream flow. The flow variation between day  $k$  and day  $l$ ,  $\delta_{kl} = Q_{t-k} - Q_{t-k-l}$  was also tested as a possible input.

The choice for the range of tested inputs was based on [7] and [24], where the best results are obtained with no more than 3 delays as inputs. The differentiated inputs are also found in the literature as an option, hence its inclusion in the current study. All different combinations for these inputs were tested when optimizing the models (e.g. 1, use  $Q_{t-1}$  and  $Q_t$ ; e.g. 2, use  $Q_{t-2}$ ,  $Q_{t-1}$ ,  $Q_t$  and  $\delta_{01}$ ; etc.).

##### 6.1. ANN Model

When developing an ANN model the selection of the architecture is the first step. In the literature there is no consensus on number of layers and neurons for each layer. Therefore we started by following the procedures described in [24] in order to replicate the described results.

Then we proceeded to the model optimization in order to try to improve the obtained results.

A simple three layer feedforward network architecture was selected. For the selection of the number of neurons on the hidden layer, various simulations were performed. The simulations varied the number of neurons on the hidden layer from 2 to 15. The training function selected was the `trainlm`, which implements the Levenberg-Marquardt training algorithm [13]. The training function has various parameters that need to be set:

- 1) Maximum number of epochs to train: 1000;
- 2) Performance goal (MSE): 0;
- 3) Minimum performance gradient:  $10^{-10}$ ;
- 4) Initial  $\mu$ : 0.001;
- 5)  $\mu$  decrease factor: 0.1;
- 6)  $\mu$  increase factor: 10;
- 7) Maximum  $\mu$ :  $10^{10}$ ;
- 8) Maximum time to train in seconds: inf;

These are the default values for these parameters and stopping conditions. No guidance was found on the literature towards recommendations or optimal parameters, therefore the default values were used. The tests performed were done with logarithmic transformed data and normalization and also only normalization. A series of tests with different hidden neurons were tried for each set of inputs and the model with the lowest RMSE are selected to be the best models for further analysis.

## 6.2. ANFIS Model

We started by replicated the model described in [24]. However it was not possible to obtain the details for the training parameters. Therefore, the following default parameters for the stopping conditions were used:

- 1) Training epoch number: 50;
- 2) Training MSE goal: 0;
- 3) Initial step size: 0,01;
- 4) Step size decrease rate: 0,9;
- 5) Step size increase rate: 1,1.

Triangular functions were selected to be associated with the input parameters following the recommendation in [24], where this function was found to give better results for the prediction of stream flow time series (even if the differences were not much relevant). The number of membership functions for each input was found to be very influential, so simulations were performed for all possible combinations from 2 to 4 membership functions per input variable.

The simulations were performed using both normalized inputs and logarithmic transformed inputs. 50 epochs were selected to be the stopping criteria. The model with the lowest RMSE was selected as the best models and then further analyzed.

## 6.3. SVM model

The optimization of the SVM model focused on the C and  $\gamma$  parameters. The Radial Basis Function (RBF) was chosen as kernel function (following the recommendations and procedures from [24]).

The grid search method from [2] and [24] was implemented trying both  $\epsilon$ -SVM and  $\nu$ -SVM. The results were not satisfactory and very different from the ones in the literature. Hence, multiple combinations of C and  $\gamma$  were trained and tested using cross-validation. Ranges for C and  $\gamma$  were as follows:

- 1) C = 2n with n = -5,-4,...,5;
- 2)  $\gamma$  = 2n with n = -5,-4,...,3.

The  $\gamma$  obtained in previous studies were small numbers, in the order of  $2^{-3}$ . Since the search space is almost infinite for these parameters, a start point had to be defined. Between all the simulations the RMSE was compared and the model with the lowest error was selected for further analysis.  $\nu$ -SVM yielded better results than  $\epsilon$ -SVM, with  $\nu = 0.5$ .

## 6.4. Performance metrics

The following metrics were used to evaluate the performance of each model:

$$MSE = \frac{1}{N} \sum_{i=1}^N (d_i - y_i)^2 \quad (15)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |d_i - y_i| \quad (16)$$

$$MARE = \frac{1}{N} \sum_{i=1}^N \frac{|d_i - y_i|}{y_i} \quad (17)$$

$$NS = 1 - \frac{\sum_{i=1}^N (d_i - \bar{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y}_i)^2} \quad (18)$$

where N is the number of points,  $d_i$  is the predicted value from the model and  $y_i$  and  $\bar{y}_i$  are the observed flow and the mean of the observed flows, respectively.

The Mean Square Error (MSE) measures the mean squared error of the prediction, bringing emphasis to high prediction errors in detriment of the errors with decimal values.

The Mean Absolute Error (MAE) measures the mean deviation each prediction has.

The Mean Absolute Percentage Error (MAPE) measures the percentage deviation from the prediction to the measured value.

Finally the Nash-Sutcliffe (NS), also called Coefficient of Efficiency [10], that is used in measuring hydrology forecasting models' performance. This metric also measures the ability of a model to predict values that are different from the mean. A 0.9 value is considered a very good model and values between 0.8 and 0.9 represent a fairly good model and any value below 0.8 is considered unsatisfactory [23].

## 7. Results and Discussion

Table 2 resumes the best results obtained in each dataset for each optimized model (and the baseline). It also includes the results obtained on previous works (where applicable). The best performance for each metric is highlighted in bold.

When replicating previous works' models, many discrepancies were found between the obtained results and the published ones when implementing the same architecture, especially in what concerns the ANN models. In some occasions we obtained better results, in others we could not approach them. This might be partially explained by the lack of information on the training algorithm stopping criteria and parameters. This fact ended up not being of much relevance since the performed optimizations allowed us to improve all the previous published results for each technique (all RMSE were lower). It should be noted that in the case of the Pailogou dataset the previously published results did not beat the baseline [24].

One exception should be noted: due to the lack of information (the authors did not reply to our requests), we were not able to replicate the input preprocessing method used in [7]. The method, which uses ANN and some sort of input convolution, arguably allowed the authors to obtain a very low forecast RMSE in the Castanheiro dataset (an improvement of almost 30% over our baseline). Unfortunately we were not able to verify the results and/or generalize the efficiency of the input convolution method to other methods and datasets.

Overall it is noticeable that, after optimization, all techniques obtain very similar results on each dataset, allowing for an improvement between 5% and 10% over the baseline.

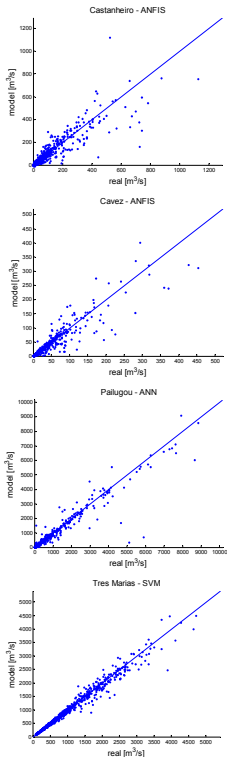


Fig. 3 Scatter plots of the best techniques for the 4 datasets. a) Castanheiro (ANFIS); b) Cavez (ANFIS); c) Pailogou (ANN); d) Tres Marias (SVM).

On average ANFIS gave the best results, but no technique was better than the others in all the datasets. Moreover, the error metrics yielded different techniques as “winners” for different datasets. This further concludes that none of the implemented techniques is more suitable to stream flow forecasting than the other. One should also mention each technique used different inputs and different architectures to achieve the best results. Also, on different datasets each technique reached the best results with different inputs. This hints that it is necessary to perform extensive parameter optimization whenever addressing stream flow forecasting in new datasets.

According to the Coefficient of Efficiency (NS), the optimized forecasts were considered very good in the Trés Marias (0.97) and Pailougou datasets (0.91), good in the Cavez dataset (0.84) and fairly good in the Castanheiro dataset (0.80).

An unusual result is the fact that training errors are higher than the errors in the test sets. This is probably due to the fact that the training sets have a higher flow peaks to total number of points ratio, implying that the test sets were easier for the models to predict than the training sets.

Regarding the effect of input preprocessing, both ANN and ANFIS models produced the best results when their inputs were preprocessed using the logarithm function (the only exception was with ANN for criteria RMSE in the Castanheiro dataset). The SVM did not show much improvement with the logarithm preprocessing.

Figure 3 shows the flow scatterplots (real vs. forecasted) for the best technique in each dataset. As expected, the largest deviations occur with the larger flows.

Figure 4 shows how the best technique in each dataset behaves in the most difficult section of the dataset. Even if the results seem very good, further zoom-in shows that no method is able to forecast sudden flow variations, almost always exhibiting a one-day delay period of adaptation in the most extreme cases.

## 8. Conclusions

In this paper we addressed the problem of one day ahead stream flow forecasting by comparing and optimizing hybrid and computational intelligence techniques.

One of the first remarks from the discussed results is the applicability of the logarithm transformation for stream flow forecasting. After implementing this preprocessing step, the ANN and ANFIS models decreased their MAE considerably. This transformation function can help these models predicting stream flows.

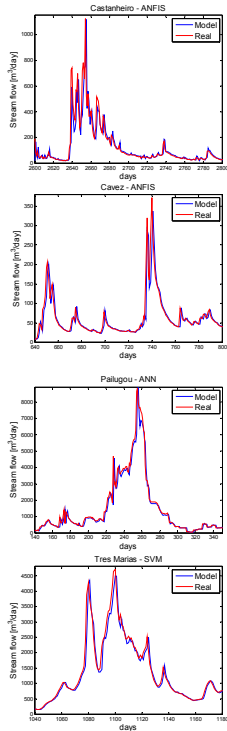


Fig. 4 Highest peak close-up of the best forecast in each dataset. a) Castanheiro (ANFIS); b) Cavez (ANFIS); c) Pailougou (ANN); d) Tres Marias (SVM).

Even if it was possible to obtain improvements over the baseline and good NS values, it is important to mention that none of the techniques produced good results in what concerns the peak flows forecast. The scatter plots demonstrated that the high flows have the highest errors, and all the techniques in all datasets displayed a constant displacement in relation to the expected output in such situations. Therefore, we can conclude that this is a (relevant) problem these models were not able to resolve.

The optimized ANN, ANFIS and SVM gave similar results for all datasets. One cannot consider one technique to be more suitable than the other for this application. It was also possible to conclude that there is no specific parameter selection or architecture that is more favorable for the stream flow forecasting problem. Parameter and input

optimization is necessary when approaching new datasets since the results from the models cannot be generalized.

Finally we can add that, in case it is not possible to experiment with several techniques, ANFIS would be the most appropriate choice, logarithm preprocessed inputs should be used, and the flow variation between current day and previous day ( $\delta_{10}$ ) should be included in the input set.

The presented work is obviously not final. Other available hybrid techniques could be implemented and compared to the existing results, and an even more extensive study to select the best combination of inputs should be made. The convolution technique applied to the inputs presented in [7] should also be validated, implemented and tested using other techniques and datasets.

TABLE 2  
RESULTS WITH BEST RMSE FROM THE SIMULATIONS

Castanheiro	Inputs	Arch.	Test					Training						
			RMSE	MAE	MARE	NS	R	RMSE	MAE	MARE	NS	R		
Baseline	t	t+1=t	32,83	7,45										
ANN <sup>(*)</sup>	t-2, t-1, t	3-4-4-1	<b>21,98</b>	<b>5,19</b>	n/a	<b>0,92</b>								
ANN	t-3, t-2, t-1, t	3-5-1	30,06	6,81	19,18	0,8034	<b>0,9186</b>	35,22	10,75	29,99	0,8217	0,9208		
ANFIS	$\delta_{01}$ , t-2, t-1, t	4,2,3,2	29,99	6,71	17,88	0,7995	<b>0,9194</b>	35,98	9,10	18,55	0,7907	0,9187		
SVM	t-2, t-1, t	2 <sup>2</sup> , 4	31,11	6,44	<b>17,02</b>	0,7511	0,9179	38,63	9,01	19,45	0,7138	0,9122		

Cavez	Inputs	Arch.	Test					Training						
			RMSE	MAE	MARE	NS	R	RMSE	MAE	MARE	NS	R		
Baseline	t	t+1=t	15,66	4,38										
ANN	t-3, t-2, t-1, t	4-4-1	14,37	3,96	13,70	0,8307	<b>0,9334</b>	15,95	4,32	13,68	0,8600	0,9376		
<b>ANFIS</b>	$\delta_{01}$ , t	<b>3,2</b>	<b>14,33</b>	4,06	13,98	<b>0,8452</b>	<b>0,9323</b>	16,19	4,15	13,27	0,8565	0,9356		
SVM	$\delta_{01}$ , t-2, t-1, t	32, 2-5	14,76	<b>3,80</b>	<b>13,33</b>	0,8175	0,9306	16,83	4,10	13,35	0,8317	0,9317		

Pailugou	Inputs	Arch.	Test					Training						
			RMSE	MAE	MARE	NS	R	RMSE	MAE	MARE	NS	R		
Baseline	t	t+1=t	337,69	73,18										
ANN <sup>(*)</sup>	t-2, t-1, t	3-5-1	388,26	n/a	12,80	0,871	0,938	444,67	n/a	16,12	0,820	0,906		
<b>ANN</b>	<b>t-3, t-2, t-1, t</b>	<b>4-6-1</b>	<b>304,54</b>	<b>73,88</b>	12,51	<b>0,9076</b>	<b>0,9605</b>	412,37	115,23	17,59	0,8119	0,9198		
ANFIS <sup>(*)</sup>	t	3	368,26	n/a	12,03	0,884	0,947	442,05	n/a	15,84	0,822	0,907		
ANFIS	t-2, t-1, t	1, 2-5	307,50	78,15	12,47	0,9041	0,9602	399,66	118,69	21,63	0,8294	0,9243		
SVM <sup>(*)</sup>	t-2, t-1, t	16, 0,25	364,56	n/a	<b>11,71</b>	0,887	0,947	399,64	n/a	14,44	0,854	0,925		
SVM	t-2, t-1, t	1, 2-5	307,50	78,15	12,47	0,9041	0,9602	456,65	125,18	18,16	0,7666	0,9005		

Tres Marias	Inputs	Arch.	Test					Training						
			RMSE	MAE	MARE	NS	R	RMSE	MAE	MARE	NS	R		
Baseline	t	t+1=t	112,46	49,30										
ANN	$\delta_{01}$ , t	2-10-1	101,72	<b>39,61</b>	4,43	0,9772	0,9895	168,10	66,84	9,14	0,9346	0,9688		
ANFIS	$\delta_{01}$ , t	3,4	103,01	39,69	<b>4,38</b>	0,9770	0,9890	162,92	61,50	7,69	0,9371	0,9709		
<b>SVM</b>	<b><math>\delta</math>, t-2, t-1, t</b>	<b>16, 2-5</b>	<b>100,51</b>	40,96	4,58	<b>0,9775</b>	<b>0,9897</b>	184,93	66,29	8,08	0,9226	0,9624		

(\*) – models from [24]  
(\*\*) – model from [7]

## References

[1] C.-C. C. Lin and Chih-Jen, "LIBSVM : a library for support vector machines," in *ACM Transactions on Intelligent Systems and Technology*, 2011.

[2] C.-W. Hsu, C.-C. Chang and C.-J. Lin, "A Practical Guide to Support Vector Classification," Department of Computer Science, National Taiwan University, Taipei, 2003.  
[3] Cortes, C., Vapnik, V., "Support-vector networks". *Machine Learning* 20 (3): 273, 1995.

- [4] G. Box, G. Jenkins, and G.C. Reinsel, "Time Series Analysis, Forecasting and Control", 3rd ed. Oakland, California: Holden Day, 1994.
- [5] H. R. Maier and G. C. Dandy, "Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications," *Environmental Modelling & Software*, no. 15, pp. 101-124, 2000.
- [6] H. Yu and B. M. Wilamowski, "Levenberg-Marquardt Training," in *Industrial Electronics Handbook*, vol. 5 – Intelligent Systems, 2nd Edition, CRC Press, 2011, ch 12, pp. 12-1 to 12-15.
- [7] I. Pulido-Calvo and M. M. Portela, "Application of neural approaches to one-step daily flow forecasting in Portuguese watersheds," *Journal of Hydrology*, pp. 332, 1-15, 2007.
- [8] J. A. Tomé and J. P. Carvalho, "One Step Ahead Prediction Using Fuzzy Boolean Networks," in *EUSFLAT Conf.*, 2005.
- [9] J. A. Tomé and J. P. Carvalho, Decision Validation and Emotional Layers in Fuzzy Boolean Networks, NAFIPS2004 - 23rd International Conference of the North American Fuzzy Information Processing Society, Jun. 2004, pp. 136-139
- [10] J. Nash and J. Stutcliffe, "River flow forecasting through conceptual models. I. A discussion of principles," *Journal of Hydrology*, no. 10, pp. 282-290, 1970.
- [11] J.-S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," in *IEEE Transactions on Systems Man and Cybernetics*, 1993.
- [12] M. C. Demirl, et al., "Flow forecast by SWAT model and ANN in Pracana basin, Portugal," *Advances in Engineering Software*, pp. 467-473, 2009.
- [13] M. Hagan and M. Menhaj, "Training feed-forward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, p. 989-993, 1999.
- [14] M. Magalhaes et al, "Predictive fuzzy clustering model for natural streamflow forecasting", in Proc. of the 2004 IEEE International Conference on Fuzzy Systems, pp. 1349-1354 vol.3, Budapest, Hungary, 2004
- [15] MathWorks, "Neural Network Toolbox Bibliography - MATLAB & Simulink," MathWorks, [Online]. Available: <http://www.mathworks.com/help/nnet/ug/neural-network-toolbox-bibliography.html>.
- [16] Pindoriya, N.M., Singh, S.N., Singh, S.K. "One-step-ahead hourly Load Forecasting using artificial Neural Network", Proc. of the ICPS '09. International Conference on Power Systems, 2009.
- [17] R.-E. Fan, P.-H. Chen and C.-J. Lin, "Working Set Selection Using Second Order Information for Training Support Vector Machines," *Journal of Machine Learning Research*, no. 6, pp. 1889-1918, 2005.
- [18] S. Ch, N. Anand, B. Panigrahi and S. Mathur, "Streamflow forecasting by SVM with quantum behaved particle swarm optimization," *Neurocomputing*, pp. 18-23, 2013
- [19] S. R. N and P. C. Deka, (Feb, 2014) "Support vector machine applications in the field of hydrology: A review," *Appl. Soft Comput. J.* [Online]. Available: <http://dx.doi.org/10.1016/j.asoc.2014.02.002>.
- [20]
- [21] Vuković, P., "One-step ahead predictive fuzzy controller", *Fuzzy Sets and Systems*, 122 (1), pp.107-115, 2001.
- [22] W. Wang, P. H. v. Gelder, J. Vrijling and J. Ma, "Forecasting daily streamflow using hybrid ANN models," *Journal of Hydrology*, pp. 383-399, 2006.
- [23] W. Wang, P. H. v. Gelder, J. Vrijling and J. Ma, "Forecasting daily streamflow using hybrid ANN models," *Journal of Hydrology*, pp. 383-399, 2006.
- [24] Z. He, X. Wen, H. Liu and J. Du, "A comparative study of artificial neural network, adaptive neuro fuzzy inference system and support vector machine for forecasting river flow in the semiarid mountain region," *Journal of Hydrology*, pp. 379-386, 2014.