

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Correlated time-series in multi-day-ahead streamflow forecasting using convolutional networks

Felipe O. Barino¹, Vinicius N. H. Silva², Andrés P. López-Barbero², Leonardo de Mello Honório¹, and Alexandre Bessa dos Santos¹

¹ Federal University of Juiz de Fora (UFJF), Juiz de Fora, 36036-900, Brazil

² Federal Fluminense University (UFF), Niterói, 24220-900, Brazil.

Corresponding author: Felipe O. Barino (e-mail: felipe.barino@engenharia.ufjf.br).

This work was supported in part by the Programa de Bolsas de Pós-graduação da Universidade Federal de Juiz de Fora (PBPG-UFJF) through the Programa de Pós-graduação em Engenharia Elétrica (PPEE-UFJF), Coordenação de Aperfeiçoamento Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Instituto Nacional de Energia Elétrica (Inerge-UFJF), and Santo Antônio Energia.

ABSTRACT Information about future streamflow is important for hydropower production planning, especially for damless hydro-power plants. The river flow is a reflection of various hydrological, hydrogeological, and meteorological factors, which increases the direct modeling difficulty, and favors the use of data-driven methods. In this paper, we propose the use of one-dimensional convolutional neural networks (1d-CNN) for multi-day ahead river flow forecasting and we present a multi-input model using correlated-input time-series. The proposed model was applied at the Madeira River, the Amazon's largest and most important tributary, near the Santo Antônio damless hydro-power plant. We compared the proposed correlated-input 1d-CNN to a single-input 1d-CNN model and some baseline models. Furthermore, we conclude that 1d-CNN performed better than all baseline models and that the correlated-input forecasting model is 5 times smaller than the single-input equivalent with accuracy improvements.

INDEX TERMS time series forecasting, convolutional neural networks, Madeira River, hydropower scheduling

TABLE I
NOTATIONS

Abbreviation	Description
CNN	Convolutional neural network
1d-CNN	CNN using 1d input data
ANN	Artificial neural network
DFT	Discrete Fourier transform
<i>sp</i>	Seasonal persistent model
<i>t2d</i>	Turbidity to discharge linear model
MLP	Multilayer perceptron
CNN-Q	1d-CNN with discharge input
CNN-QT	1d-CNN with discharge and turbidity input
ReLU	Rectified linear unit function
MSE	Mean squared error
NMSE	Normalized mean squared error
MAPE	Mean absolute percentage error

I. INTRODUCTION

Streamflow forecasting is an important activity that can help in hydro-power scheduling, reservoir planning, and flood management. Therefore, streamflow forecasting plays an important role in the environmental and economic sciences. In what concerns the Brazilian electric system, river flow forecasting is extremely relevant, considering 66.6% of domestic electrical energy was supplied by water sources in 2018 [1]. Future streamflow data could be used to model the amount of energy that would be possible to generate from water sources and use this model to optimize the use of resources, especially in systems that use damless hydroelectric plants. Indeed, the runoff is the most important input for the hydropower system and one of the most decisive parameters in hydropower production planning and decision making [2], [3].

However, the river flow forecast depends on a wide number of variables such as precipitation, evapotranspiration, ambient temperature, soil characteristics of the catchment,

and urbanization [4]. Moreover, modeling river dynamics based on these variables requires a large number of hydrological, hydrogeological, and meteorological parameters [5]. In this context, the use of data-driven techniques has been growing for river flow forecasting [6]–[13].

Traditional data-driven techniques are based on statistical models that use past time-series data to fit a function and forecast its future values. Examples of these techniques are the linear autoregressive (AR), nonlinear autoregressive (NAR), autoregressive moving average (ARMA), and nonlinear autoregressive moving average (NARMA); and they have been successfully used in river flow forecasting [6]–[8].

In the last decades, machine learning has been extensively employed in data-driven forecasting models for river flow [10]–[19] and other environmental parameters [20]–[24], substituting the function parametric fitting of traditional data-driven techniques by a machine learning model. Among the vast number of machine learning algorithms, the artificial neural network (ANN) has been widely used and proved to be equivalent to the well-accepted statistical models, such as ARMA, NAR, and NARMA [25]–[27]. Although ANNs are equivalent to these statistical models, ANN models offer the designer a flexible way of implementation; model complexity can be varied by changing the hidden layer size or the neuron's transfer function and the number of input/output parameters can also be easily changed. Moreover, ANN algorithms showed better performance for natural resources forecasting than statistical models [28], [29]. Hence, due to these advantages, ANN-based forecasting algorithms have been extensively used in the past years [30]–[36].

These data-driven models gained more power, flexibility, and ease of implementation as the state-of-the-art machine learning algorithms have moved towards deep learning, a type of machine learning algorithm that uses several layers to analyze data. This multilayer characteristic of deep learning structures allows multiple layers of data abstraction and feature extraction [37], [38]. Therefore, the task of feature extraction that once was performed by the machine learning scientist could now be incorporated into the machine learning model.

Although the rise of deep learning was mainly due to its two-dimensional feature extraction abilities, deep learning has recently been applied to unidimensional data such as time series, in different fields, such as sensor-based human activity recognition [39], [40], trading tendencies identification [41], [42], and medicine [43].

In this work, we present a unidimensional convolutional neural network (1d-CNN) river flow forecasting model using correlated-input time series. We investigated the use of turbidity and river flow as the input of a 1d-CNN to forecast the river discharge multiple days ahead. To the best of our knowledge, the use of turbidity and discharge as the input of

a 1d-CNN for river flow forecasting has never been investigated before.

The proposed correlated-input CNN was applied at the Madeira River at the Santo Antônio power plant, an important Amazon basin damless hydropower plant. Furthermore, we compared the proposed correlated-input CNN model to a single-input CNN model and baseline models: linear model between the correlated time series, seasonal persistent model, and multilayer perceptron (MLP).

II. METHODS

A. Data description and analysis

The study presented in this paper was conducted using discharge and water turbidity data collected at the Madeira River from 2011 to 2018 at the Santo Antônio hydroelectric power plant. A section of the northern region of Brazil can be seen in Fig. 1, the main rivers of this region are shown in blue and the Madeira River is highlighted in red, the site where data was collected is also highlighted by a cross near the city of Porto Velho.

Both time series had 24 hours sampling rate and can be seen in Fig. 2, where $t[n]$ represents the river turbidity time series and $d[n]$ the river discharge time series. Note that the power plant staff collects this data daily at the same site, therefore we can ensure that both time series are synced in both location and time. Moreover, the model implementation does not require more data acquisition, only the implementation of the processing algorithm.

This data was sequentially separated into three time-dependent subsets: train, validation, and test. The first 45% data points were assigned to the train set, the next 25% to the validation set, and the remaining 30% (most recent data) to the test set. We used the first two subsets to train and select the forecasting models and the last subset to evaluate and compare the models. This sequential separation of data was made to ensure that all data used to train the forecasting models were older than the data used to evaluate them, therefore avoiding biased evaluation due to time dependence.

Before model selection, we analyzed the data to choose the

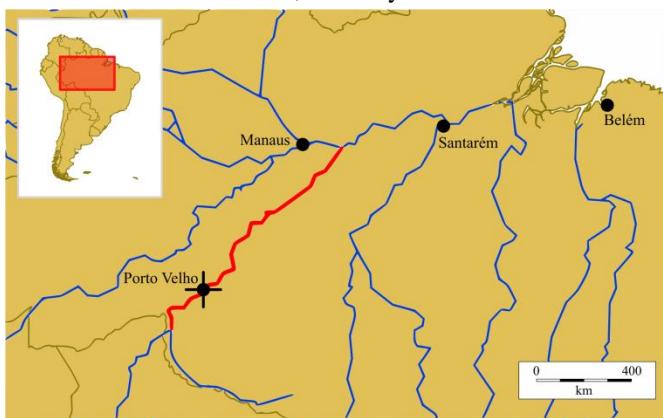


FIGURE 1. Map showing the North Region of Brazil and its main rivers. The Madeira River is highlighted in red and the cross represents the Santo Antônio power plant location.

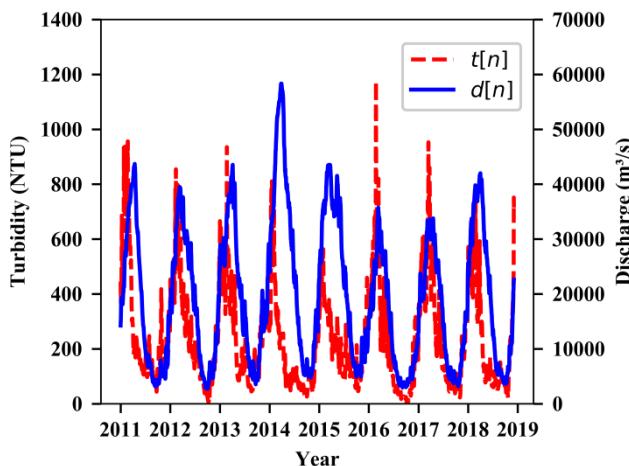


FIGURE 2. Turbidity (dashed red line) and discharge (solid blue line) times series collected daily at the Madeira River.

number of steps for model input and output. Firstly we obtained the time series frequency content to measure its periodicity. Therefore, we calculated the turbidity and discharge time series discrete Fourier transform (DFT) and measured their fundamental frequency. It is known that during hydrological events, river flow and sediment concentration are related, in a way that peaks of discharge could occur synchronously with sediment concentration peak, before or after the correspondent discharge one [44]. And by visual inspection of Fig. 2, we estimated that both time series have the same fundamental frequency, but with turbidity peaks occurring before the discharge peaks.

Moreover, we searched for the mean lag that the discharge time series has concerning the turbidity time series. We lagged the turbidity time-series $t[n]$ by k steps and calculated the correlation between $t[n - k]$ and the discharge time series $d[n]$ ($\text{corr}(t[n - k], d[n])$), this process was repeated for several k values. The maximum value of $\text{corr}(t[n - k], d[n])$ occurs when k is the number of days that the turbidity time series is ahead of the discharge time series, we denoted this value by k_{lag} .

Given the periodic characteristic of both $t[n]$ and $d[n]$, we expected the $\text{corr}(t[n - k], d[n])$ to increase to a certain peak and start to decrease, passing to the point where the correlation was lower than $k = 0$ correlation value. We used this point k_{max} where $\text{corr}(t[n - k_{max}], d[n]) = \text{corr}(t[n], d[n])$ to determine the forecasting model input size. This choice was made to take advantage of the time steps in which the signals were more correlated. The number of steps ahead to forecast the river flow was chosen as k_{lag} , to take advantage of the high correlation between today's turbidity and river discharge k_{lag} days ahead.

B. Baseline models

To evaluate and compare the 1-d CNN forecasting model, we propose some simple baseline models using the discharge periodic characteristic and the correlation between the studied time series, the Seasonal persistent model (sp), and

the turbidity-to-discharge linear model ($t2d$), respectively. We also used a multilayer perceptron (MLP) as a machine learning baseline model.

1) SEASONAL PERSISTENT MODEL

The first baseline model was based on the naive repetition of the last season value, i.e. we took advantage of the periodic character of the river discharge and assumed that the discharge at a given time will be equal to the discharge of the last season equivalent. Let T_d be the river discharge period and y_{sp} be the k_{lag} days-ahead river discharge estimated by the sp model, y_{sp} , is given by:

$$y_{sp} = d[n - T_d] \quad (1)$$

2) TURBIDITY TO DISCHARGE LINEAR MODEL

The next baseline model used the correlation between the $t[n]$ and $d[n + k_{lag}]$ to estimate $d[n + k_{lag}]$. We used the train and validation data to fit a first-order polynomial that relates the turbidity to the future discharge, therefore the turbidity to discharge linear model is given by:

$$y_{t2d} = a \cdot x_{t[n]} + b \quad (2)$$

where y_{t2d} is the k_{lag} days-ahead river discharge estimated by the $t2d$ model, $x_{t[n]}$ is today's known turbidity, a and b are, respectively, the angular and linear coefficients calculated by the first-order polynomial fitting of the training and validation data.

3) MLP FORECASTING MODEL

The most complex baseline model we used was the MLP-based forecasting model. It is a one hidden layer multilayer perceptron that receives the k_{max} last discharge values and estimates the k_{lag} days ahead of river discharge. We choose the MLP as a machine learning baseline model because neural networks were widely used for forecasting river parameters and the MLP is an important neural network structure due to its simplicity and universal approximation capability [45].

We trained several MLP topologies using the RMSprop training algorithm with a 0.001 learning rate and using early stopping with 10 epochs patience. We searched for the hidden layer size testing this parameter from 10 to 90 neurons, in 5 neuron steps. Both input and output data were normalized to the $[0, 1]$ interval and rescaled to perform the model evaluation. The best configuration, and our MLP baseline model, was a 70 neuron MLP with a sigmoid activation function, trained until epoch 28.

C. CNNs in time series processing

Backpropagation convolutional neural networks were first introduced to solve the automatic handwritten digits identification problem [46]. While the traditional fully connected layer used by the MLP learns patterns scattered over its coordinates, convolutional layers were designed to find local patterns, such as objects in a scene, regardless of its position [37], [38]. Another advantage of convolutional layer architecture is that the filter coefficients are shared for all

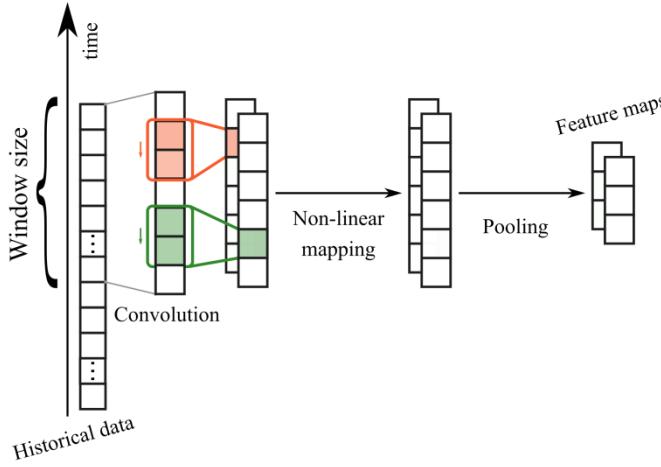


FIGURE 3. Scheme of a 1D convolutional layer for time series processing. input positions, while dense layers (another name for fully connected layers) have connections between every neuron in adjacent layers.

These same characteristics can be extended to 1d data, such as time series. Therefore, a pattern learned in a given time can be recognized in another time frame, regardless of when the pattern happened [47]. This approach to time series forecasting has been explored to air pollution and electric load estimation [47]–[49], solar power generation forecast [50], rainfall [51], and multi-location streamflow forecast using stacked time series [52].

The scheme of a 1d convolutional input layer used for time series feature extraction can be seen in Fig. 3; in this illustration, a convolutional layer with two kernels of size two is shown. In this work, we used a single convolutional layer followed by two dense layers. The first dense layer receives the flattened feature maps and the last one outputs the multi-day ahead discharge estimation. Therefore, the last dense layer has one neuron and the hidden dense layer a configurable hidden layer size. We used the ReLU (rectified linear unit) activation function in both the convolutional and hidden layers.

At the CNN input, part of the historical data is windowed and used as the convolutional layer input. This layer convolves the input time series with M convolutional kernels. After this convolution, a bias b_j was added and the result fed into a ReLU function that outputs:

$$v_j^{x,i} = \max\{0, b_j + \sum_{m=0}^{M-1} \sum_{p=0}^{P-1} w_{jm}^p s_m^{x+p,i}\} \quad (3)$$

where $v_j^{x,i}$ is the value of the j^{th} feature map at the position x concerning the i^{th} input time series, w_{jm}^p is the p^{th} value for the P sized convolutional kernel, and $s_m^{x,i}$ is the i^{th} input sequence at position x . The $v_j^{x,i}$ values were fed into a max-pooling layer of size Q , described by:

$$a_j^{x,i} = \max_{1 \leq q \leq Q} (v_j^{x+q,i}) \quad (4)$$

The $a_j^{x,i}$ values were then flattened into a vector of α_k components, where k represents each component of the

feature maps and is fed into a fully connected layer, the hidden layer. The output of this layer i^{th} neuron is:

$$H_i = \max\{0, b_i + \sum_k \alpha_k w_{i,k}\} \quad (5)$$

where $w_{i,k}$ is the synaptic weight connecting α_k to the H_i neuron. Finally, the predicted value in terms of H_i and the weights connecting the hidden layer to the output layer is:

$$y_{CNN} = \sum_{i=0}^{N_h} w_{1,i} H_i \quad (6)$$

D. Proposed CNN models

We propose the use of correlated time series as the input of the 1d CNN forecasting model equated in section II.C. The use of correlated time series as input to a convolutional layer was previously studied in [53], where the authors showed using correlated time series as the input of a forecasting model using a convolutional input layer followed by a recurrent layer increased the forecasting performance.

As discussed in section II.A., the river turbidity and discharge are highly correlated; therefore we used these two parameters in the correlated-input 1d CNN. Hence, two forecasting models based on one-dimensional convolutional layer were designed and evaluated: the baseline CNN model named CNN-Q, which used only the river discharge of k_{\max} days to forecast the k_{lag} days ahead river discharge; and the correlated-input CNN model (named CNN-QT), which used last k_{\max} days of turbidity and discharge to forecast the river discharge k_{lag} days ahead.

The designing procedure for both models was the same, we searched for the best hyper-parameter combination in the subset of the CNNs hyper-parameter space: $M \in \{6, 8, 12\}$ kernels with size $P \in \{2, 4, 8\}$, pooling window of size $Q \in \{1, 2, 4\}$, and hidden layer with $N_h \in \{10, 30, 50\}$ neurons. We trained all 324 models five times using Adam [54]. The training aimed at the reduction of the forecasting mean squared error (MSE) and we selected the smallest model that had a good validation mean absolute percentage error (MAPE) over the five tests.

E. Evaluation metrics

All baseline and CNN-based models were evaluated using the test set data. The metrics used to evaluate and compare these models were the MAPE, the normalized root mean squared error (NRMSE), the coefficient of determination (R^2), and the Nash–Sutcliffe model efficiency coefficient (NSE). We also compared the models visually by plotting a graph with the real time-series alongside the forecasted time-series and a scatterplot comparing the observed and estimated river discharge.

III. RESULTS

A. Time series analysis

The frequency-domain characteristic of the turbidity and discharge time series is shown in Fig. 4 by the discharge

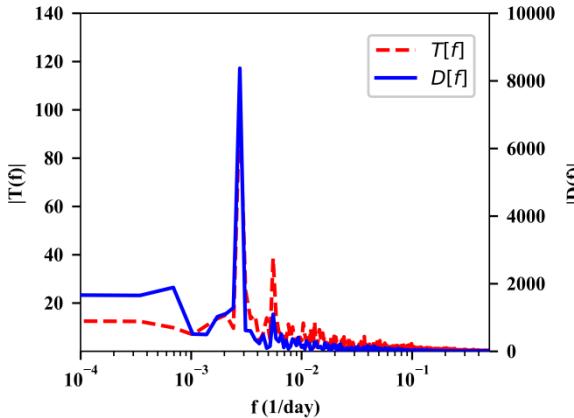


FIGURE 4. DFT amplitude of the discharge ($|D(f)|$) and turbidity ($|T(f)|$) time series.

($|D(f)|$) and turbidity ($|T(f)|$) time series DFT amplitude. Note that the river turbidity and discharge DFT amplitude peaks together at $f_0 = 0.0028$ day $^{-1}$. Therefore, the turbidity and discharge from 2011 to 2018 at the Madeira River has the same period $T \approx 362$ days. We also calculated the correlation between $|D(f)|$ and $|T(f)|$ as 0.96, indicating that the minor peaks also occurred at the same frequencies, with highly correlated amplitudes; therefore the main difference between both time series was each frequency component phase.

The correlation between $t[n - k]$ and $d[n]$ with $k \in 0, 1, \dots, 90$ days, used to design the input and output size of the forecasting models can be seen in Fig. 5. The correlation increases to a maximum value of 0.74 at $k_{lag} = 38$ days and decreases to the initial value at $k_{max} = 75$ days. Therefore, our forecasting models were configured to an input window size of 75 days to predict the river flow 38 days ahead. This input window selects the last 75 data points of the historical data as model input; see the example illustrated by Fig. 3

B. CNN models configuration

The model selection results for the proposed correlated-input and single-input 1d CNN models are shown in Table II. For both CNN-Q and CNN-QT, we chose the hyper-parameters combination that resulted in the best mean MAPE with minor variation over the five tests. The validation performance was 22.6% and 25.6% MAPE for the CNN-Q and CNN-QT models, respectively. But note that the single-input model has more filters and hidden neurons, and therefore, more trainable parameters. Note also that a unit pooling size is equivalent to no pooling at all, since the convolution outputs are grouped one by one, and this also increases the number of connections and the parameter counting.

On the other hand, fewer hidden neurons were used by the CNN-QT model, which indicates that the convolutional layer could perform better feature extraction than the CNN-Q convolutional layer, using fewer filters. The similar validation MAPE observed during the model selection and the fewer trainable parameters indicated that the use of

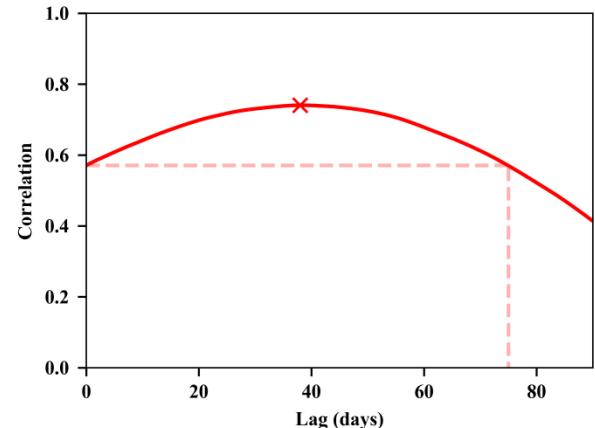


FIGURE 5. Correlation between $t[n - k]$ and $d[n]$ for $k \in 0, 1, \dots, 90$.

correlated time series in 1d-CNN could indeed facilitate the forecasting task and the use of smaller 1d-CNN models for the River Madeira discharge forecasting.

C. Baseline models performance

The metrics calculated by the proposed baseline models are summarized in Table III. The most simple model (Seasonal persistent or *sp*) obtained a good performance due to the discharge fundamental frequency high influence on the signal power, note in Fig. 4 the amplitude difference between the discharge fundamental frequency peak and higher frequency peaks.

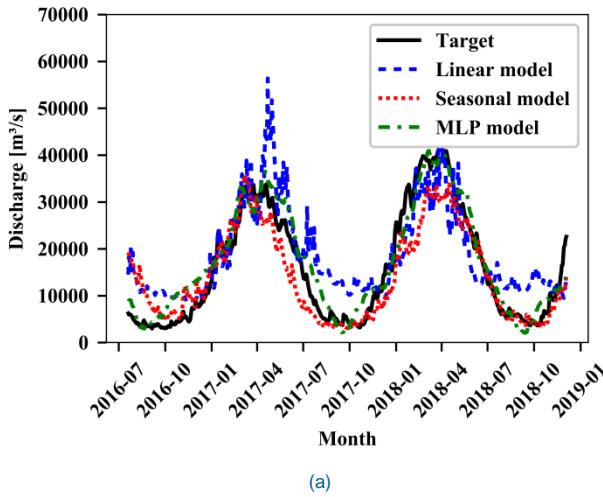
The turbidity to discharge linear model (*t2d*), on the other hand, performed worse than expected. Due to the high correlation between $t[n - 38]$ and $d[n]$ seen in Fig. 5, we expected the *t2d* model to perform better than the *sp* model, but we estimated the *t2d* MAPE as two times worse than *sp* model. We could attribute the high error and low R^2 obtained by the *t2d* model to the turbidity high-frequency components than can be seen in Fig. 4.

Among the baseline models, the MLP performed better, with low MAPE, NRMSE, and R^2 closer to unity. Moreover, for 38 days ahead forecasting, the 33.60% error is a good approximation, but a poor accuracy improvement to the seasonal persistent model given the MLP complexity.

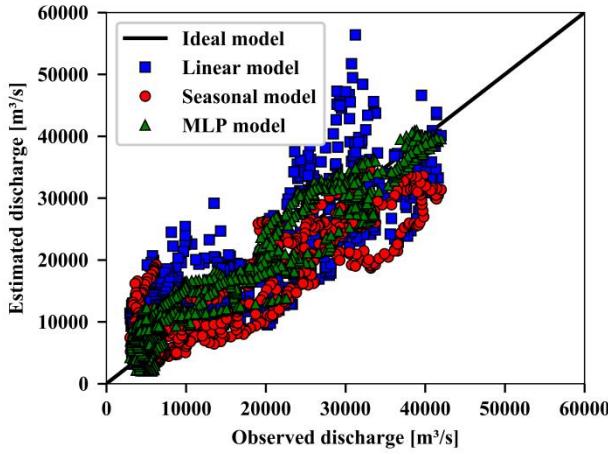
A visual comparison between all baseline models can be seen in Fig. 6. In Fig. 6a the seasonal persistence could be easily seen as the first peak was repeated by the second one. The scatter plot in Fig. 6b showed the *sp* model outputs followed the ideal model curve slightly below; therefore, this

TABLE II
SELECTED HYPER-PARAMETERS

Model	1d convolutional layer			Dense layer
	Filters	Kernel size	Pooling size	Neurons
CNN-Q	12	2	1	50
CNN-QT	8	4	2	30



(a)



(b)

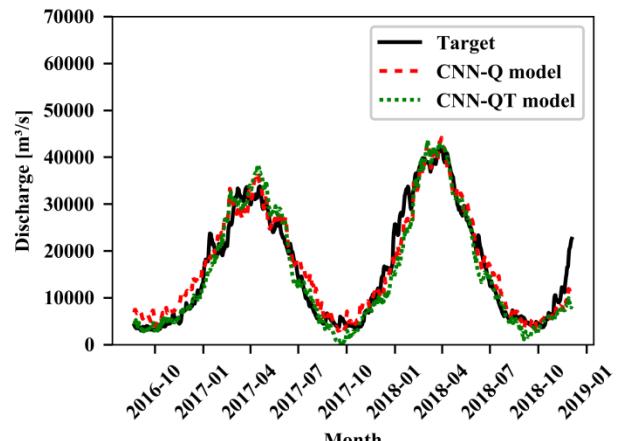
FIGURE 6. Baseline models comparison, (a) forecasted time series and (b) observed vs estimated scatter plot.

model tended to sub-estimate the discharge value on the tested data, especially for higher values.

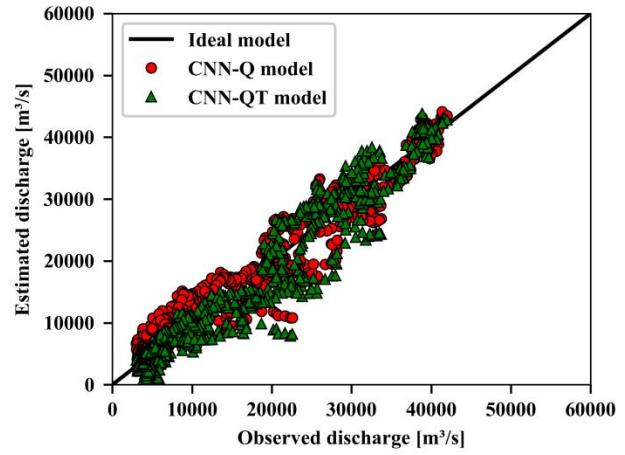
On the other hand, the linear model that relates the turbidity to the 38-day-ahead discharge tended to be less accurate for smaller values as can be seen in both Fig. 6a and Fig. 6b, and to over-estimate the discharge. In Fig. 6a the high-frequency characteristics of the turbidity could be observed by the fast-changing discharge sub-peaks estimated by this model. Moreover, Fig. 6b showed that the estimated values did not follow a straight line concerning the observed data, as seen by the $R^2=0.66$; furthermore, the model saturates at around $10,000 \text{ m}^3/\text{s}$ and provided a poor estimative of lower discharge values.

TABLE III
BASELINE MODELS METRICS

Model	MAPE	NRMSE	R^2
<i>sp</i>	35.61 %	0.325	0.81
<i>t2d</i>	76.70 %	0.429	0.66
MLP	33.60%	0.213	0.91



(a)



(b)

FIGURE 7. CNN models comparison, (a) forecasted time series and (b) observed vs estimated scatter plot.

D. CNN models performance and comparison

The performance metrics estimated using the test set data for the two CNN-based models designed in this work are shown in Table IV. The traditional approach to forecasting using 1d-CNN (model CNN-Q), obtained great improvement on the accuracy when compared to the baseline models, with almost 30% improvement on the MLP accuracy.

The proposed correlated-input CNN model proposed in this work obtained an accuracy improvement concerning the CNN-Q model that represents a 20% accuracy improvement, with similar R^2 , at the cost of a higher NRMSE. This accuracy improvement represents more than 40% more accuracy than in the baseline models.

Fig. 7 shows the CNN models forecasting results and the scatter plots of these models. Note that both models are visually similar as can be seen in the time plots shown in Fig. 7a. Data dispersion was also similar, as showed in Fig. 7b, the estimated values are near the ideal model line with a similar slope, as seen numerically by the $R^2=0.93$ and $R^2=0.92$.

TABLE IV
CNN-BASED MODELS METRICS

Model	MAPE	NRMSE	R ²
CNN-Q	25.21%	0.186	0.93
CNN-QT	19.67%	0.202	0.92

Comparing the CNN models with the MLP model, the best baseline model presented in this work, we could see that the CNN models presented better accuracy, an improvement on slope-change detection, and are easier to implement than MLPs (no feature extraction is needed). Furthermore, although the CNN models could provide better estimative of the target time series behavior compared to the CNN-Q model, the CNN-QT model had better accuracy, especially for low discharge values.

Table V summarizes the performance of all models including the NSE and confidence interval. We estimated the 68% C.I. by the standard deviation of each model error. We could see that although the *sp* and MLP models presented relatively close MAPE, the C.I. of the *sp* model was roughly five times bigger than the C.I. estimated for the MLP model, indeed note the *sp* and *t2d* models data spread in Fig. 6(b). Additionally, a negative NSE was obtained for these two models. This indicates that the mean value of the discharge time series is a better estimation than the one provided by the *sp* and *t2d* models.

From this table, we could also see that the CNN-based models also presented lower uncertainties, as expected by the R² and the low data spread seen in Fig. 7(b). Comparison to the well-accepted machine learning forecasting model (MLP) tested in this work, showed the accuracy improvement observed by the CNN-based models didn't compromise the forecast confidence, nor the NSE.

Therefore, results showed the use of CNNs in time series could improve the forecast quality and that the use of correlated-input indeed increases accuracy, as shown in [53]. Although the CNN-QT model had more input data, leading to more features to extract and more patterns to learn, the training time was 11.5 ± 6.6 seconds for CNN-Q and 10.7 ± 6.0 seconds for the CNN-QT model, showing no mean training time difference. Because despite the greater amount of data incorporated into CNN-QT, the number of trainable parameters of CNN-QT was substantially smaller, which could have favored the training time.

Indeed, as discussed in section III.B., the main advantage

TABLE V
COMPARISON BETWEEN MODELS

Model	MAPE	NRMSE	NSE	R ²	C.I. (m ³ /s)
<i>sp</i>	35.61 %	0.325	< 0	0.81	15,147
<i>t2d</i>	76.70 %	0.429	< 0	0.66	15,088
MLP	33.60%	0.213	0.925	0.91	3,157
CNN-Q	25.21%	0.186	0.932	0.93	2,985
CNN-QT	19.67%	0.202	0.921	0.92	3,211

of the CNN-QT model over the CNN-Q model is its size. The CNN-QT was almost five times smaller than the CNN-Q model, concerning the trainable parameters. CNN-QT had a smaller hidden layer, fewer filters, and fewer connections from the convolutional layer to the hidden layer due to the bigger pooling size. And despite the smaller size, the CNN-QT showed better forecasting performance. Hence, incorporating the turbidity time series to the 1d-CNN forecasting model improved the model simplicity and the resulted accuracy.

Furthermore, the hidden layer size of the CNN-QT model (30 neurons) was smaller than the CNN-Q model (50 neurons) and the MLP model (70 neurons). Therefore, if we apply transfer learning to the convolutional layer of the CNN-QT model, among the machine learning based models presented in this paper, the CNN-QT is computationally easier and faster to train.

Moreover, the performance of the proposed CNN-based model met the performance showed by other studies found in the literature for data-driven techniques towards multi-day ahead streamflow forecasting. In [55] the authors showed a comparison between ANN, WNN (wavelet neural networks), ANFIS (adaptive neuro-fuzzy inference system), and WNF (Wavelet neural networks model) forecasting models for 5 days ahead river flow forecasting and obtained a peak detection error as low as 3% for the WNN and as high as 72% for WNF and 98% for the ANN and ANFIS models, respectively.

The work presented in [56], on the other hand, showed an ANFIS forecasting model with good performance, measured by an NSE of 0.823 for 8 days ahead streamflow forecasting. The authors of [57] showed a comparison of several multi-step forecasting models and obtained a MAPE ranging from 14.41% to 43.99% for 7 days ahead river flow forecasting. And in [58] the authors showed a comparison between several regression models for river flow forecasting and the results showed great variability in the peak detection error, this error varied from 4% to 89% absolute discharge peak error. And the best model evaluated in this study obtained an NSE of 0.819 for 3-day ahead river flow forecasting.

As one can see the performance obtained by the proposed forecasting method agrees with the performance found in the literature for multi-day ahead streamflow forecasting. Although the results may vary a lot due to the different number of steps ahead or river studied the ~20% MAPE and ~0.9 NSE obtained in this work showed that the use of CNN in river flow forecasting is promising and could increase the performance of future multi-day ahead streamflow forecasting models.

IV. CONCLUSIONS

This study presented the use of correlated time series as the input of a unidimensional convolutional neural network for streamflow forecasting at the River Madeira. Analyzing Madeira's time series we concluded that the river discharge

periodicity was 362 days and that it was 38 days lagged concerning turbidity. We used this information to design and evaluate several forecasting models.

Comparison between the proposed correlated-input 1d-CNN and a single-input 1d-CNN showed the use of turbidity and discharge could decrease the Madeira's river flow forecasting model by a factor of five. Furthermore, the CNN-based models presented better estimation accuracy than all baseline models tested, and the CNN-QT obtained slightly better performance, despite its reduced size.

Since our results showed turbidity was ahead of the river discharge and that the use of these two parameters improved the forecasting model, we believe that the approach we took on this paper should work for any river that the discharge is lagged from the turbidity. Hence in future works is interesting to evaluate the proposed method for different rivers.

Therefore, the use of turbidity and discharge in CNN-based Madeira's river flow forecasting model could represent a step forward in deep learning for river flow forecasting. A smaller model could be trained with fewer data and still be reliable, and when training with large datasets the training is faster and computationally cheaper.

Moreover, the results obtained in this work may encourage machine learning scientists working on CNN-based forecasting models to study the use of correlated time series. Although more investigation on the forecasting of different time series and parameters is needed, the results showed that the use of correlated-input is promising in what concerns the model reduction.

REFERENCES

- [1] EPE, "Balanco Energético Nacional 2019 - ano base 2018," Rio de Janeiro, 2019. [Online]. Available: <http://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-377/topicos-494/BEN>.
- [2] S.-C. Kao *et al.*, "Projecting changes in annual hydropower generation using regional runoff data: An assessment of the United States federal hydropower plants," *Energy*, vol. 80, pp. 239–250, 2015.
- [3] C. L. Wu, K. W. Chau, and Y. S. Li, "Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques," *Water Resour. Res.*, vol. 45, no. 8, 2009.
- [4] C. Sudheer, R. Maheswaran, B. K. Panigrahi, and S. Mathur, "A hybrid SVM-PSO model for forecasting monthly streamflow," *Neural Comput. Appl.*, vol. 24, no. 6, pp. 1381–1389, 2014.
- [5] K. Sene, *Hydrometeorology*. Springer, 2010.
- [6] S. W. Fleming and G. K. C. Clarke, "Autoregressive Noise, Deserialization, and Trend Detection and Quantification in Annual River Discharge Time Series," *Can. Water Resour. J. / Rev. Can. des ressources hydriques*, vol. 27, no. 3, pp. 335–354, 2002, doi: 10.4296/cwrj2703335.
- [7] K. Mohammadi, H. R. Eslami, and R. Kahawita, "Parameter estimation of an ARMA model for river flow forecasting using goal programming," *J. Hydrol.*, vol. 331, no. 1–2, pp. 293–299, 2006.
- [8] Y. G. Tesfaye, M. M. Meerschaert, and P. L. Anderson, "Identification of periodic autoregressive moving average models and their application to the modeling of river flows," *Water Resour. Res.*, vol. 42, no. 1, 2006.
- [9] E. Kentel, "Estimation of river flow by artificial neural networks and identification of input vectors susceptible to producing unreliable flow estimates," *J. Hydrol.*, vol. 375, no. 3–4, pp. 481–488, 2009.
- [10] L. E. Besaw, D. M. Rizzo, P. R. Bierman, and W. R. Hackett, "Advances in ungauged streamflow prediction using artificial neural networks," *J. Hydrol.*, vol. 386, no. 1–4, pp. 27–37, 2010.
- [11] C. L. Wu and K.-W. Chau, "Data-driven models for monthly streamflow time series prediction," *Eng. Appl. Artif. Intell.*, vol. 23, no. 8, pp. 1350–1367, 2010.
- [12] Z. M. Yaseen, A. El-Shafie, H. A. Afan, M. Hameed, W. H. M. W. Mohtar, and A. Hussain, "RBFNN versus FFNN for daily river flow forecasting at Johor River, Malaysia," *Neural Comput. Appl.*, vol. 27, no. 6, pp. 1533–1542, 2016.
- [13] J. Zhou, T. Peng, C. Zhang, and N. Sun, "Data pre-analysis and ensemble of various artificial neural networks for monthly streamflow forecasting," *Water*, vol. 10, no. 5, p. 628, 2018.
- [14] A. F. Atiya, S. M. El-Shoura, S. I. Shaheen, and M. S. El-Sherif, "A comparison between neural-network forecasting techniques-case study: river flow forecasting," *IEEE Trans. neural networks*, vol. 10, no. 2, pp. 402–409, 1999.
- [15] Y. B. Dibike and D. P. Solomatine, "River flow forecasting using artificial neural networks," *Phys. Chem. Earth, Part B Hydrol. Ocean. Atmos.*, vol. 26, no. 1, pp. 1–7, 2001, doi: [https://doi.org/10.1016/S1464-1909\(01\)85005-X](https://doi.org/10.1016/S1464-1909(01)85005-X).
- [16] S. Alvisi and M. Franchini, "Fuzzy neural networks for water level and discharge forecasting with uncertainty," *Environ. Model. Softw.*, vol. 26, no. 4, pp. 523–537, 2011.
- [17] V. Jothiprakash and R. B. Magar, "Multi-time-step ahead daily and hourly intermittent reservoir inflow prediction by artificial intelligent techniques using lumped and distributed data," *J. Hydrol.*, vol. 450, pp. 293–307, 2012.
- [18] F.-J. Chang, P.-A. Chen, Y.-R. Lu, E. Huang, and K.-Y. Chang, "Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control," *J. Hydrol.*, vol. 517, pp. 836–846, 2014.
- [19] H. Tongal and M. J. Booij, "Simulation and forecasting of streamflows using machine learning models coupled with base flow separation," *J. Hydrol.*, vol. 564, pp. 266–282, 2018.
- [20] H. Samet, M. Reisi, and F. Marzbani, "Evaluation of neural network-based methodologies for wind speed forecasting," *Comput. Electr. Eng.*, vol. 78, pp. 356–372, Sep. 2019, doi: 10.1016/j.compeleceng.2019.07.024.
- [21] Najeebulah, A. Zameer, A. Khan, and S. G. Javed, "Machine Learning based short term wind power prediction using a hybrid learning model," *Comput. Electr. Eng.*, vol. 45, pp. 122–133, Jul. 2015, doi: 10.1016/j.compeleceng.2014.07.009.
- [22] S. Liu, L. Xu, and D. Li, "Multi-scale prediction of water temperature using empirical mode decomposition with back-propagation neural networks," *Comput. Electr. Eng.*, vol. 49, pp. 1–8, Jan. 2016, doi: 10.1016/j.compeleceng.2015.10.003.
- [23] N. Shabbir, R. AhmadiAhangar, L. Kutt, M. N. Iqbal, and A. Rosin, "Forecasting Short Term Wind Energy Generation using Machine Learning," in *2019 IEEE 60th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)*, Oct. 2019, pp. 1–4, doi: 10.1109/RTUCON48111.2019.8982365.
- [24] A. Sharma and A. Kakkar, "Forecasting daily global solar irradiance generation using machine learning," *Renew. Sustain. Energy Rev.*, vol. 82, pp. 2254–2269, Feb. 2018, doi: 10.1016/j.rser.2017.08.066.

- [25] T. Hill, L. Marquez, M. O'Connor, and W. Remus, "Artificial neural network models for forecasting and decision making," *Int. J. Forecast.*, vol. 10, no. 1, pp. 5–15, 1994.
- [26] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 240–254, 1994.
- [27] K. H. Chon and R. J. Cohen, "Linear and nonlinear ARMA model parameter estimation using an artificial neural network," *IEEE Trans. Biomed. Eng.*, vol. 44, no. 3, pp. 168–174, 1997.
- [28] J. J. Flores, M. Graff, and H. Rodriguez, "Evolutive design of ARMA and ANN models for time series forecasting," *Renew. Energy*, vol. 44, pp. 225–230, Aug. 2012, doi: 10.1016/j.renene.2012.01.084.
- [29] M. Valipour, M. E. Banihabib, and S. M. R. Behbahani, "Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir," *J. Hydrol.*, vol. 476, pp. 433–441, Jan. 2013, doi: 10.1016/j.jhydrol.2012.11.017.
- [30] F. Kaytez, M. C. Taplamacioglu, E. Cam, and F. Hardalac, "Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines," *Int. J. Electr. Power Energy Syst.*, vol. 67, pp. 431–438, May 2015, doi: 10.1016/j.ijepes.2014.12.036.
- [31] L. Wang, Z. Wang, H. Qu, and S. Liu, "Optimal Forecast Combination Based on Neural Networks for Time Series Forecasting," *Appl. Soft Comput.*, vol. 66, pp. 1–17, May 2018, doi: 10.1016/j.asoc.2018.02.004.
- [32] J. Szoplik, "Forecasting of natural gas consumption with artificial neural networks," *Energy*, vol. 85, pp. 208–220, Jun. 2015, doi: 10.1016/j.energy.2015.03.084.
- [33] J. Buitrago and S. Asfour, "Short-Term Forecasting of Electric Loads Using Nonlinear Autoregressive Artificial Neural Networks with Exogenous Vector Inputs," *Energies*, vol. 10, no. 1, p. 40, Jan. 2017, doi: 10.3390/en10010040.
- [34] Y. Yao *et al.*, "A paired neural network model for tourist arrival forecasting," *Expert Syst. Appl.*, vol. 114, pp. 588–614, Dec. 2018, doi: 10.1016/j.eswa.2018.08.025.
- [35] N. H. Zainuddin *et al.*, "Improvement of time forecasting models using a novel hybridization of bootstrap and double bootstrap artificial neural networks," *Appl. Soft Comput.*, vol. 84, p. 105676, Nov. 2019, doi: 10.1016/j.asoc.2019.105676.
- [36] D. B. Alencar, C. M. Affonso, R. C. L. Oliveira, and J. C. R. Filho, "Hybrid Approach Combining SARIMA and Neural Networks for Multi-Step Ahead Wind Speed Forecasting in Brazil," *IEEE Access*, vol. 6, pp. 55986–55994, 2018, doi: 10.1109/ACCESS.2018.2872720.
- [37] Y. Bengio, "Learning deep architectures for AI," *Found. trends® in Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
- [38] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," *APSIPA Trans. Signal Inf. Process.*, vol. 3, p. e2, 2014, doi: 10.1017/atsip.2013.9.
- [39] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," 2015.
- [40] A. Ignatov, "Real-time human activity recognition from accelerometer data using Convolutional Neural Networks," *Appl. Soft Comput.*, vol. 62, pp. 915–922, 2018.
- [41] J.-F. Chen, W.-L. Chen, C.-P. Huang, S.-H. Huang, and A.-P. Chen, "Financial time-series data analysis using deep convolutional neural networks," in *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, 2016, pp. 87–92.
- [42] S. Alonso-Monsalve, A. L. Suárez-Cetrulo, A. Cervantes, and D. Quintana, "Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators," *Expert Syst. Appl.*, vol. 149, p. 113250, 2020.
- [43] A. Y. Hannun *et al.*, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nat. Med.*, vol. 25, no. 1, p. 65, 2019.
- [44] G. P. Williams and others, "Sediment concentration versus water discharge during single hydrologic events in rivers," *J. Hydrol.*, vol. 111, no. 1, pp. 89–106, 1989.
- [45] K. Hornik, M. Stinchcombe, H. White, and others, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [46] Y. LeCun *et al.*, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, 1990, pp. 396–404.
- [47] Q. Tao, F. Liu, Y. Li, and D. Sidorov, "Air pollution forecasting using a deep learning model based on 1D convnets and bidirectional GRU," *IEEE Access*, vol. 7, pp. 76690–76698, 2019.
- [48] K. Amarasinghe, D. L. Marino, and M. Manic, "Deep neural networks for energy load forecasting," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, 2017, pp. 1483–1488.
- [49] W. He, "Load forecasting via deep neural networks," *Procedia Comput. Sci.*, vol. 122, pp. 308–314, 2017.
- [50] W. Lee, K. Kim, J. Park, J. Kim, and Y. Kim, "Forecasting solar power using long-short term memory and convolutional neural networks," *IEEE Access*, vol. 6, pp. 73068–73080, 2018.
- [51] A. Haidar and B. Verma, "Monthly Rainfall Forecasting Using One-Dimensional Deep Convolutional Neural Network," *IEEE Access*, vol. 6, pp. 69053–69063, 2018.
- [52] C. Huang *et al.*, "Robust Forecasting of River-flow Based on Convolutional Neural Network," *IEEE Trans. Sustain. Comput.*, 2020.
- [53] R.-G. Cirstea, D.-V. Micu, G.-M. Muresan, C. Guo, and B. Yang, "Correlated Time Series Forecasting Using Multi-Task Deep Neural Networks," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1527–1530, doi: 10.1145/3269206.3269310.
- [54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv Prepr. arXiv1412.6980*, 2014.
- [55] H. Badrzadeh, R. Sarukkalige, and A. W. Jayawardena, "Impact of multi-resolution analysis of artificial intelligence models inputs on multi-step ahead river flow forecasting," *J. Hydrol.*, vol. 507, pp. 75–85, Dec. 2013, doi: 10.1016/j.jhydrol.2013.10.017.
- [56] Y. Zhou, S. Guo, and F.-J. Chang, "Explore an evolutionary recurrent ANFIS for modelling multi-step-ahead flood forecasts," *J. Hydrol.*, vol. 570, pp. 343–355, Mar. 2019, doi: 10.1016/j.jhydrol.2018.12.040.
- [57] H. Tongal and R. Berndtsson, "Impact of complexity on daily and multi-step forecasting of streamflow with chaotic, stochastic, and black-box models," *Stoch. Environ. Res. Risk Assess.*, vol. 31, no. 3, pp. 661–682, Mar. 2017, doi: 10.1007/s00477-016-1236-4.
- [58] M. Rezaie-Balf, S. Kim, H. Fallah, and S. Alaghmand, "Daily river flow forecasting using ensemble empirical mode decomposition based heuristic regression models: Application on the perennial rivers in Iran and South Korea," *J. Hydrol.*, vol. 572, pp. 470–485, May 2019, doi: 10.1016/j.jhydrol.2019.03.046.



Felipe O. Barino was born in Juiz de Fora, Brazil, in 1996. He received his B.Sc. degree in electronics from the Federal University of Juiz de Fora (UFJF), Minas Gerais, Brazil, in 2019 and is now pursuing his M.S. degree in electrical engineering at the Programa de Pós-Graduação em Engenharia Elétrica - PPEE/UFJF. His current research interests are related to instrumentation, optical fiber sensors, and machine learning.



Leonardo de Mello Honório was born in Juiz de Fora, Brazil, in 1971. He received the B.Sc. degree from the Federal University of Juiz de Fora (UFJF), in 1993, and the M.Sc. and Ph.D. degrees from EFEI, Brazil, in 1999 and 2002, respectively, all in electrical engineering. He was a Visiting Researcher with Porto and Irvine California University, in 2006 and 2012, respectively. He is currently an Associate Professor with UFJF. His current research interests include evolutionary algorithms, probabilistic methods, optimal power flow, robotics, autonomous vehicles, fuzzy logic, pattern recognition, and optimization.



Vinicius N. H. Silva was born in Niterói, Brazil, in 1979. He received the M.Sc. degree in telecommunication engineering from Federal Fluminense University (UFF), Niterói, in 2009, and the Ph.D. degree in telecommunication engineering from Télécom Bretagne, France, in 2013. Since 2014, he has been an Associate Professor with the Telecommunication Engineering Department, Federal Fluminense University (UFF). His research interests include turbulence effects in free-space optics, optical fiber sensors, and optical communications.



Alexandre Bessa dos Santos was born in Juiz de Fora, Brazil, in 1975. He received the M.Sc. and Ph.D. degrees in electrical engineering from Pontifical Catholic University (PUC), Brazil, in 2001 and 2005, respectively. Since 2010, he has been an Associate Professor with the Department of Circuits at Federal University of Juiz de Fora (UFJF), Juiz de Fora. His research interests include applied electromagnetism, optical sensors, computational methods, instrumentation, and metrology.



Andrés P. López-Barbero was born in Montevideo, Uruguay, in 1966. He received the M.Sc. degree in optical engineering from the Technological Institute of Aeronautics (ITA), São Paulo, Brazil, in 1994, and the Ph.D. degree in electrical engineering from Campinas State University (UNICAMP), São Paulo, in 2000. Since 1989, he has been an Associate Professor with the Telecommunication Engineering Department, Federal Fluminense University (UFF), Niterói, Brazil. His research interests include optical fiber sensors, optical communications, and sensors for electric power networks.