encit 2020

ABCM Associação Brasileira de Engenharia e Ciências Mecânicas

18th Brazilian Congress of Thermal Sciences and Engineering
November 16-20, 2020 (Online)

ENC-2020-0785
MACHINE LEARNING TECHNIQUES APPLIED TO ITAIPU STREAMFLOW FORECASTING

Jorge Gustavo Sandoval Simão
Industrial and Systems Engineering Graduate Program (PPGEPS), Pontifical Catholic University of Parana (PUCPR). Curitiba, Parana, Brazil.
jgssimao@gmail.com
Gabriel Trierweiler Ribeiro
Department of Electrical Engineering, Federal University of Parana (UFPR). Curitiba, Parana, Brazil.
gabrielribeiro.ee@gmail.com
Viviana Cocco Mariani
Mechanical Engineering Graduate Program (PPGEM), Pontifical Catholic University of Parana (PUCPR). Curitiba, Parana, Brazil.
viviana.mariani@pucpr.br
Leandro dos Santos Coelho
Industrial and Systems Engineering Graduate Program (PPGEPS), Pontifical Catholic University of Parana (PUCPR). Curitiba, Parana, Brazil.
leandro.coelho@pucpr.br

Abstract. *Time series forecasting has gained lots of attention recently. This is because many real-world phenomena can be modeled as time series. On the other hand, applications of machine learning models to the forecasting problem are gaining interest among researchers as well as the industry and energy systems. This work evaluates three machine models including Random Forest, Support Vector Regression and k-Nearest Neighbor applied to time series forecasting for the Itaipu's streamflow case study. The results are presented in terms of different performance metrics (RMSE, RMSLE and MAE), where the Random Forest and k-Nearest Neighbor models outperform the Support Vector Regression to the Itaipu's streamflow forecasting.*

Keywords: *Random Forest, Support Vectors Regression, k-Nearest Neighbor, Time Series, Forecasting*

1. INTRODUCTION

Itaipu is a binational hydroelectric power plant, located on the border between Brazil and Paraguay, on the Parana River, close to the Brazilian city of Foz do Iguaçu and the Paraguayan city of Hernandarias. Its administration is carried out by Itaipu Binacional, with co-participation between the companies Eletrobras (Brazil) and *Administración Nacional de Eletricidad*, ANDE (Paraguay). According to Pimenta *et al.* (2018), it was set up in 1974, with works beginning in 1975 and production in 1984. Itaipu is the second largest hydroelectric plant in the world, with 20 generating units, each with a power of 700 MW, totalizing an installed power of 14,000 MW.

According to official records (Itaipu Binacional, 2020a), in 2016 the plant reached a new record in its annual production, reaching 103.098,366 MWh (103 million MWh), surpassing the previous mark of 98.630,035 MWh, of 2013 In 2019, one of the driest years in the plant's history, Itaipu produced 79.444,510 MWh (79.4 million MWh). As of the beginning of its operations, it has already generated more than 2.5 billion megawatt-hours and is currently responsible for supplying approximately 11.3% of the electricity consumed in Brazil and 88.1% in Paraguay.

The amounts of energy destined for both countries are defined on the day before delivery, during the process of preparing the Itaipu Daily Operation Program (PDO), discretized at intervals of thirty minutes. The powerhouse of the plant is divided into two sectors, one operating at 50 Hz and the other at 60 Hz, each with 10 generating units. The energy generated by the 60 Hz sector is transmitted to the Brazilian Electric System, using the Furnas 765 kV System and Copel's 525 kV System. The energy produced in the 50 Hz sector is drained to the Paraguayan Electric System, using four 220 kV lines and a 500 kV line, belonging to ANDE. The surplus of energy destined for Paraguay is sent to the Brazilian Electricity System through the Foz do Iguaçu substation, in Parana, using the Furnas Direct Current System.

At this point, it is important to note that the total generation of the 50 Hz sector is the result of the amounts of energy needed to serve ANDE and Eletrobras, using direct current link, limited to the availability of that sector. As a result, deviations in the forecast of interchange with ANDE can compromise the service to the Brazilian Electric System and significantly reduce the efficiency of the plant's energy production (Itaipu Binacional, 2020b).

Streamflow forecasting is a crucial part of water resource planning and management Yasseen *et al.* (2015). In recent decades, there have been an increasing number of publications focusing on improving the accuracy of streamflow forecasting . In its research, Shoaib *et al.* (2018) categorized the existing streamflow forecasting models into two main categories: theory-driven models and data-driven models. Compared with theory-driven models, data-driven models have attracted considerable interest in recent years because they do not in- volve physical mechanisms and can be accessibly applied. To establish an appropriate robust mapping relationship, abundant regression models have been applied for streamflow forecasting (Tongal and Booij, 2018), aiming to find a model that can effectively make an accurate prediction. Streamflow changes have intricate patterns and are difficult to accurately forecast due to influences from the climate, the geographical environment, social development, and human activities. So, how to find a model that can accurately predict streamflow rates, as well as adapt to data trends?

In response to the research question, the solution applied in this study was the application of different predictive modeling techniques on flow data. Thus, through different loss functions it is possible to measure which model can find the best result, thus indicating which model is the best for the specified data set.

These characteristics make the load forecast assume an extremely important role in the energy generation processes, since expressive variations, occurring in real time, may result in less use of available resources, difficulties in controlling the downstream level and even, affect the safety of the operation of the interconnected systems.

Thus, the contribution of this study is to measure, among three regression techniques, which has the best performance in creating models for forecasting time series, based on the flow data of the Itaipu Powerplant. As a way of evaluating machine learning models, in a limited data sample, *k*-fold cross validation was used, by measuring RMSE (Root Mean Square Error), RMSLE (Root Mean Squared Logarithmic Error) and MAE (Mean Absolute Error).

The remainder of this paper is organized as follows. Section 2 presents the methods used, as well as the data set. Section 3 discusses the results obtained, and Section 4 presents conclusions remarks and future research directions.

2. RELATED WORKS

Several data-driven regression models have been used to predict flow, but in some publications, techniques that are more used stand out. Extreme Gradient Boosting were used in studies like Ni *et al.* (2020), but Random Forest, Artificial Neural Networks and various types of regression techniques have been widely used. In the latest studies, it is also possible to see the use of deep learning as a predictive regression model as presented in Liu *et al.* (2020).

In the paper published by Trierweiler Ribeiro *et al.* (2020) the topic is addressed using Echo State Networks, a recurrent neural network for the processing of temporal dependencies. According to the author the low computational cost and powerful performance of ESN make it widely used in a range of applications including forecasting tasks and nonlinear modeling. His paper presents a Bayesian optimization algorithm (BOA) of ESN hyperparameters in load forecasting with its main contributions including helping the selection of optimization algorithms for tuning ESN to solve real-world forecasting problems, as well as the evaluation of the performance of Bayesian optimization with different acquisition function settings. Forecasting and various optimization methods are used too in Ribeiro *et al.* (2020b) where a hybrid learning framework is developed to forecast multi-step-ahead (one, two, and three-month-ahead) meningitis cases in four states of Brazil. First, the proposed approach applies an ensemble empirical mode decomposition (EEMD) to decompose the data into intrinsic mode functions and residual components. Then, each component is used as the input of five different forecasting models, and, from there, forecasted results are obtained. Finally, all combinations of models and components are developed, and for each case, the forecasted results are weighted integrated (WI) to formulate a heterogeneous ensemble forecaster for the monthly meningitis cases.

Forecasting techniques were used even in the COVID-19 pandemic, according to da Silva *et al.* (2020) and Ribeiro *et al.* (2020a). At the first paper cited, Bayesian regression neural network, cubist regression, k-nearest neighbors, quantile random forest, and support vector regression are used stand-alone, and coupled with the recent pre-processing variational mode decomposition (VMD) employed to decompose the time series into several intrinsic mode functions. All techniques are evaluated in the task of time-series forecasting with one, three, and six-days-ahead the cumulative COVID-19 cases in five Brazilian and American states, with a high number of cases up to April 28th, 2020. In the second, the stacking-ensemble learning approach, also the CUBIST regression, RF, RIDGE, and SVR models are adopted as base-learners and Gaussian process (GP) as meta-learner. The models' effectiveness is evaluated based on the improvement index, mean absolute error, and symmetric mean absolute percentage error criteria.

For energy prediction purposes, multi-step wind speed forecasting is used in Rodrigues Moreno *et al.* (2020), as a combination of two signal decomposition strategies, known as variational mode decomposition (VMD) and singular spectral analysis (SSA), with modulation signal theory. The proposed decomposition approach is further coupled with a long short-term memory neural network (LSTM), the adaptive neuro-fuzzy system (ANFIS), echo state network (ESN), support vector regression (SVR) and Gaussian regression process (GRP) models resulting in new ensemble learning approaches.

Specifically for streamflow forecasting (using several techniques), there's other related papers like this ones from Cheng *et al.* (2020), Wyatt *et al.* (2020) and Darbandsari and Coulibaly (2020). In the first one, an artificial neural network

(ANN) and a long short term memory (LSTM) have been adopted to forecast streamflow at daily and monthly scales for a long lead-time period. For long lead-time streamflow forecasting, a recursive forecasting procedure, which takes the last one-step-ahead forecast as a new input for the next-step-ahead forecast, is used in the ANN and LSTM forecasting systems. Two models are trained and validated for streamflow forecasting using the rainfall and runoff datasets collected from the Nan River Basin and Ping River Basin, Thailand, covering the period 1974 to 2014. The second paper evaluate the potential improvements gained by including in-situ soil moisture data in seasonal streamflow forecasting models in rainfall-dominated watersheds. Precipitation and soil moisture data from four watersheds in the U.S. were incorporated into a modified principal components analysis and regression method to predict seasonal (4-month) streamflow totals at 0-, 1-, 2-, and 3-month lead times. At the third one, his study proposes an entropy-based selection procedure to construct an ensemble of streamflow forecasts by better addressing the aforementioned contradictory criteria prior to performing the Bayesian Model Averaging.

3. MATERIAL AND METHODS

The dataset used for this research was obtained in the ONS (Operador Nacional do Sistema Elétrico) official website, through the link: http://www.ons.org.br/Paginas/resultados-da-operacao/historico-da-operacao/dados_hidrologicos_vazoes.aspx. This organization keep track from the streamflow's operations, divided by category, year and month. For the purpose of this forecasting study, data from ten years of energy production (2010-2020) were used, with each year subdivided into the total monthly amount. A short-term forecast (1 year) was made, and the data were divided into training (70%), testing (15%) and validation (15%), with an interval of one month between measurements, as shown in Table 1.

Descriptive Statistics	**Original**	**Normalized**
Number of Samples	182	182
Sample Interval	1 mo	1 mo
Mean	5094.5	0.5214
Standard Deviation	2227.56	0.2280
Minimum	0	0
First Quartile	3812	0.3894
Median	5094.5	0.5214
Third Quartile	6240.75	0.6387
Maximum	9771	1.0

Table 1: Original and normalized dataset statistics

The normal distribution was not suitable for the data sets, so Box-Cox Transformation with a λ of 2.0 was applied to force its transformation, resulting in the data shown in Figure 1.

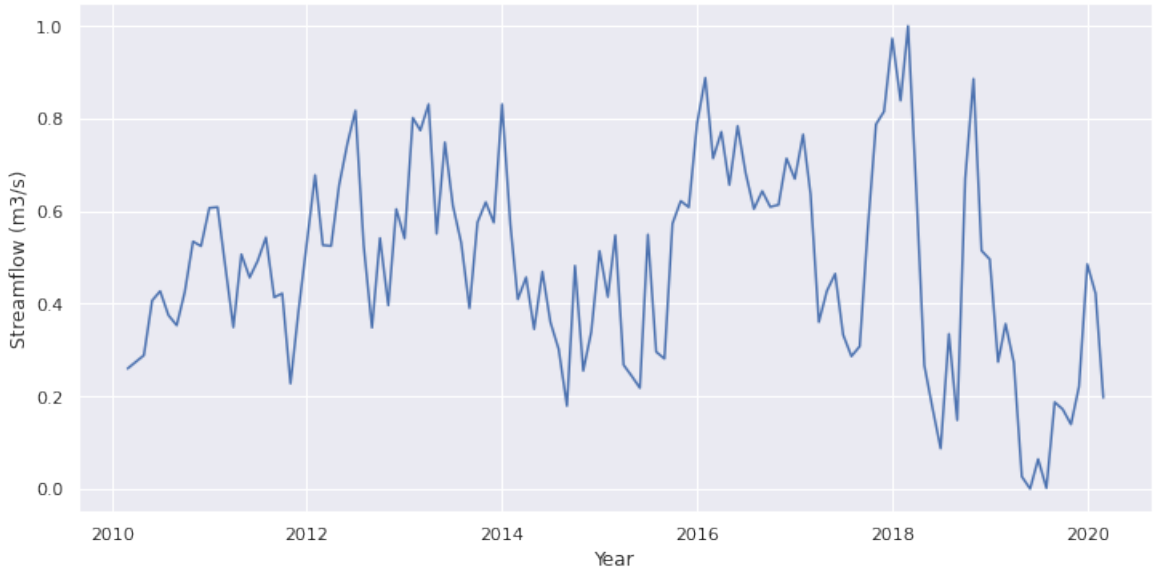

Figure 1. Ten years of monthly normalized streamflow, from May 2010 to May 2020

According to Figure 1, data were selected from May 2010 until May 2020. Only two fields existed in the original data set: Date and Flow. for better analysis, an index field (sequential integer) was added, and the streamflow values

normalized in range [0,1], for use with the three proposed methods.The streamflow data demonstrates big variations in it's measures between the years, which indicates that some patterns are maintained only for short periods, demonstrating that probably an analysis with a shorter data interval may prove to be more accurate than a long-term one.

3.1 Random Forest

Random forests are a combination of tree predictors $\mathrm{h}\left(\mathbf{x};\theta_{\mathrm{k}}\right), \mathrm{k}=1,\ldots,\mathrm{K}$ such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest, according to its original paper from Breiman (2001). In this equation, *x* represents the observed input (covariate) vector of length p with associated random vector *X* and the θ_{k} are independent and identically distributed *w* random vectors. The generalization error for forests converges to a limit as the number of trees in the forest becomes large, and the generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them.

In this paper, this method is used for regression setting, for which there's a numerical outcome, *Y*, but make some points of contact with classification (categorical outcome) problems. The observed (training) datais assumed to be independently drawn from the joint distribution of (X,Y) and comprises $\mathrm{n}(\mathrm{p}+1)$ - tuples $\left(\mathbf{x}_1,\mathrm{y}_1\right),\ldots,\left(\mathbf{x}_{\mathrm{n}},\mathrm{y}_{\mathrm{n}}\right)$. Used for regression, the random forest prediction is the unweighted average over the collection:

$$h(\mathbf{x}) = (1/K)\sum_{k=1}^{K} h\left(\mathbf{x};\theta_k\right) \tag{1}$$

In this technique, $k \to \infty$ so the Law of Large Numbers ensures that

$$E_{\mathbf{X},Y}(Y-\bar{h}(\mathbf{X}))^2 \to E_{\mathbf{X},Y}\left(Y-E_{\theta}h(\mathbf{X};\theta)\right)^2 \tag{2}$$

The quantity on the right is the prediction (or generalization) error for the random forest, designated PE_f^*. The convergence in (2) implies that random forests do not overfit (Segal, 2004). Define the average prediction error for an individual tree $h(X;\theta)$ as

$$\mathrm{PE}_{\mathrm{f}}^{*} = E_{\theta}E_{X,Y}[Y-h(X;\theta)]^2 \tag{3}$$

Assume that, for all θ, the tree is unbiased, i.e. $EY = E_X h(X;\Theta)$.
Then,

$$\mathrm{PE}_{\mathrm{f}}^{*} \leq \rho \mathrm{PE}_{\mathrm{t}}^{*} \tag{4}$$

where ρ is the weighted correlation between residual $Y-h(X;\Theta)$ and $Y-h\left(X;\Theta'\right)$ for independent Θ and Θ'. This inequality points out what is required for accurate RF regression: (1) a low correlation between residuals of differing tree members of the forest, and (2) a low prediction error for the residual trees. Furthermore, the RF will decrease the individual tree error $\mathrm{PE}_{\mathrm{t}}^{*}$ by factor ρ. The strategy employed to achieve these ends is as follows: (1) keep individual errors low by growing trees to their maximum depth; and (2) keep residual correlations low via randomization, which is accomplished by (a) growing each tree on a bootstrap sample from the training data and (b) specifying the number of covariables $p \gg m$ at each node of every tree and picking the best split of that node based on those covariables (Wong and Zhou, 2008).

3.2 Support Vector Regression

Support Vector Regression (SVR) is a method based on the concept of Support Vector Machines proposed by Cortes and Vapnik (1995), adapted for regression. SVR have advantages in high dimensionality space because SVR optimization does not depend on the dimensionality of the input space (Drucker *et al.*, 1997). Its main equations (loss function and minimization) are expressed in Eqs. (3) and (4). SVR (Vapnik, 1995) is trained from a collection of observed data, where $\left(X_1,Y_1\right),\left(X_2,Y_2\right),\ldots\left(X_m,Y_m\right)$ are the training data samples, *M* is the number of samples, X_m with *m* = 1,2, ..., *M* is the *m*th input vector, and Y_m is the corresponding output. SVR aims to find a mapping $g(X) = a\cdot\phi(X)+b$ to fit the training data samples, where ϕ is the function projecting the input vector $X=[X_1,X_2,\ldots,X_M]$ to a high-dimensional feature space such that the sample can be linearly modeled in the higher-dimensional space, *a* is the slope vector, *b* is the intercept, and . is an inner product operator. The following operators are adopted:

$$l\left(g\left(X_m\right),Y_m\right) = \begin{cases} \left|g\left(X_m\right)-Y_m\right|-\varepsilon, if\left|g\left(X_m\right)-Y_m\right|>\varepsilon \\ 0, else \end{cases}, m=1,2,\ldots,M \tag{5}$$

$$Min\frac{1}{2}a\cdot a+\frac{\beta}{M}\sum_{m=1}^{M} l\left(g\left(X_m\right),Y_m\right) \tag{6}$$

Rather than simply minimizing the summed mapping error of the training data samples, SVR introduces a generalized loss function as expressed in Eq. (3) and minimizes Eq. (4) that incorporates the loss function in order to avoid overfitting behavior. Overfitting means that the mapping is too closely related to the particular set of training data samples and may thus fail to fit additional data reliably (Yu *et al.*, 2020).

In Eq. (5), ε is a nonnegative error threshold; $|g(X_m) - Y_m|$ is the mapping error of the m *m*th training data sample and is ignored if it is equivalent to or less than ε; and $l(g(X_m), Y_m)$ is the loss of *m*th training sample. In Eq. (6),β is a nonnegative penalty factor. The term $a \cdot a$ represents the model's complexity, while the term $\sum_{m=1}^{M} l(g(X_m), Y_m)$ stands for the empirical acummulative loss; adjusting the value of β makes a tradeoff between the model's complexity and the empirical accumulative loss.

3.3 *k*-Nearest Neighbor Regression

In pattern recognition, the *k*-nearest neighbors algorithm (*k*NN) is a non-parametric method proposed by Thomas Cover used for classification and regression tasks (Altman, 1992). In the *k*NN algorithm, given x_q, take vote among its k nearest neighbors (if discrete-valued target function) take mean of f values of k nearest neighbors (if real-valued). The *k*NN algorithm approximates a numeric-valued target function f:$R^n \rightarrow R$ returns the value of $\hat{f}(x_q)$ as its estimate of $f(x_q)$, which is just the mean value of f among the *k* training instances nearest to the test instance x_q (Yang and Webb, 2006), as displayed in Eq. (5):

$$\hat{f}(x_q) \frac{\sum_{i=1}^{k} f(x_i)}{k} \tag{7}$$

In his study, Cai *et al.* (2020) assumed that the conventional *k*NN regression for traffic flow forecasting contains two stages. First, after constructing the traffic flow state vector, *k* nearest neighbor traffic flow state vector is selected by a certain metric to measure the similarity between the current traffic flow state and the historical traffic flow state vector. Second, we can predict the traffic flow by fusing the weighted results of the selected neighbor traffic flow state vectors according to the similarity between the current traffic flow state vector and the historical traffic flow state vectors.

3.4 *k*-Fold Cross Validation

The Cross Validation (CV) is a statistical methodology for comparing and evaluating training algorithms by separating data into two parts. One is utilized to train a model and the other is utilized to test it (Stone M., 1974). *k*-fold CV assesses the generalization of algorithms in machine learning approach. Based on the category of *k*-fold CV, the data is first separated into k equally measured parts. Subsequently, *k* iterations of training and testing phases are performed such that a different part of the data is held-out for testing phase within each iteration, while the remaining (*k*-1) parts are used for training phase (Zhao *et al.*, 2018).

Suppose we have a model with one or more unknown parameters $f(\alpha)$, and a data set $\hat{y}$ to which the model can be fitted. The primary method to estimate the tuning parameter α using *k*-fold CV divides the data into rougher parts. Since the model computes the mean squared error (MSE) for each $i = 1, 2, \ldots, k$, the cross validation error (CVE) can be calculated by Eq. (6).

$$CVE(\alpha) = \frac{1}{K} \sum_{i=1}^{k} MSE_i(\alpha) \tag{8}$$

where CVE defines the cross validation error.

In this paper, the number of cross validation partitions, *k*, was defined as 10. For statistical confidence, the training and testing process is repeated 10 independent runs with the dataset randomly permuted in each run prior to splitting in training and testing subsets. The accuracy of trained model is verified through three loss functions: RMSE, RMSLE, MAE.

3.5 General View of the Proposed Forecasting Approach

Figure 2 presents a detailed flowchart of the proposed system. Suppose we have an initialized machine learning architecture (RF, SVM, *k*NN) and an initial hyperparameter set. After learning with training inputs and outputs, a trained model is achieved. Then, its predictions, given the validation set inputs, are compared with true validation outputs using the RMSE, RMSLE and MAE. The optimization algorithms, intended to minimize the validation RMSE, RMSLE and MAE, iteratively adjusts the hyperparameters until an optimal set of hyperparameters is found. Next, a model is tuned with the optimal hyperparameters and produces predictions, given the test inputs. Those predictions are compared with true test set outputs using (again) the three metrics used before for the test set.

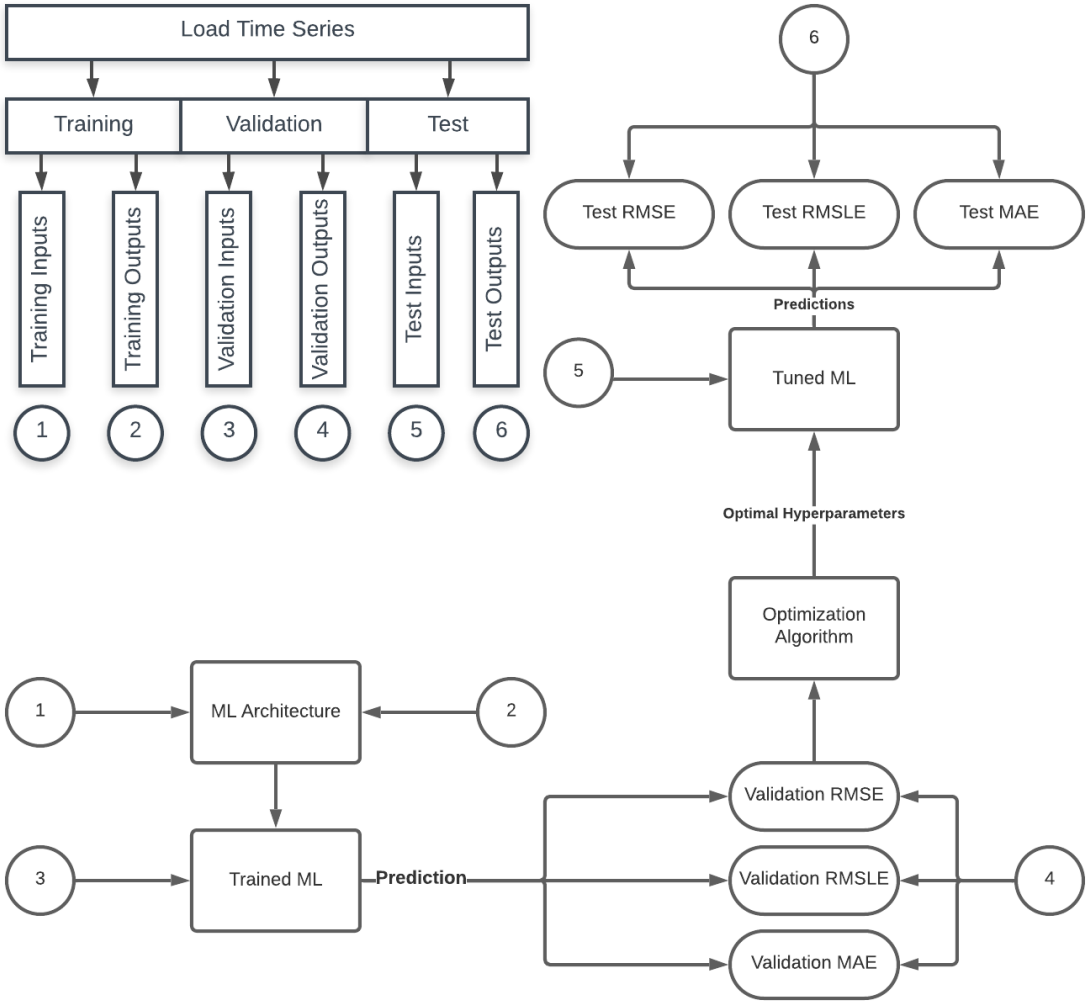

Figure 2. The general framework of the proposed forecasting approach for RF, SVM and KNN with optimizations

4. NUMERIC RESULTS

This section presents the results of the hyperparametrization changes at RF, SVR and KNN for regression, comparing them through RMSE, MSE and MAE. The parameters were submitted to GridSearch, trying to find the best values for the dataset.

4.1 Random Forest

Using Random Forest, the first parameter to be found was the max depth of the tree. It defines the number of nodes along the longest path from the start of the tree to the farthest leaf node. Higher values will make the model more complex and can lead to overfitting. This parameter was chosen as the first because this would save some computational time when the other parameters were tuned. This parameter was set between 1 and 35 leaf nodes, leading to the results shown at Table 2

Max Depth	RMSE
1	0.1580
3	0.1494
6	0.1374
13	0.1363
15	0.1363

Table 2: Random Forest maximum depth hyperparametrization best results

The Table 2 presents the results of a cartesian search of values for max depth ordered by RMSE. For this dataset, is

established that the best value is found with max depth is 1, resulting in a RMSE equals to 0.1580. With maximum depth available, is possible analize the sample rate using its top five values (1,3,6,13,15,20).

Max Depth	Sample Rate	RMSE
1	0.1580	0.1638
3	0.1494	0.1632
6	0.1374	0.1626
13	0.1363	0.1623
15	0.1363	0.1596

Table 3: Random Forest sample rate hyperparametrization best results

Table 3 presents sample rate results, showing a slightly performance increase using max depth equals 3 and sample rate equals to 0.99 (RMSE: 0.1638). However, the best parameter indicated by maximum depth suggests that using one leaf node improves performance. To find the best ntrees value, were used maximum depth at 1 and 3, leading to a best solution at 1 leaf node (keeping RMSE at 0.1580). Comparisons between the default and tuned models are presented in Table 4

	Default	Tuned
RMSE	0.1580	0.1638
RMSLE	0.1494	0.1632
MAE	0.1374	0.1626

Table 4: Random Forest default and tuned models best results

Accordinly to the Table 4, hyperparametrizations adjustments brought an improvement of 2.19% to the RMSE score (13.61% to 15.80%).This result stabilized at 15.80% most likely because of the low amount of records available for testing and validation.

4.2 Support Vector Regression

The SVR algorithm used in this study was modeled using the following hyperparametrization: C and γ. For each one of the three kernels tested (RBF, Poly and Sigmoid) various values for C and Gamma were tested until the best metrics were found. The Table 5 presents the best results of C and γ to each one of the kernels

	RBF	Poly	Sigmoid
Target	-0.8949	-1009.85	-0.8993
C	6406.37	689.35	9960.09
γ	2293.12	-	10.72

Table 5: C and γ values for best targets in mutiple SVR kernels

In Table 5, the polynomial kernel adjusts γ value accordinly to C, that's why its not showing up. Each kernel in its default configurations were tested, and then we tried to find the best parameters to each one, using bayesian optimization. The best values were presented at Table 6

	Default RBF	Tuned RBF	Default Poly	Tuned Poly	Default Sigmoid	Tuned Sigmoid
RMSE	0.1566	0.0828	0.1781	0.1777	0.1786	0.0828
RMSLE	0.1037	0.0550	0.1199	0.1197	0.1200	0.0550
MAE	0.1248	0.0763	0.1382	0.1377	0.3191	0.0763

Table 6: Support Vector Regression best kernel results

As shown in Table 6, the polynomial kernel presented the best results overall, indicating that the feature space combination influences the final result of the metrics. Using Bayesian Optimization, C and γ hyperparameters were modified to find the best combinations. unfortunately the default values presented a best performance, except for the polynomial kernel, which obtained similar results.

4.3 k-Nearest Neighbor

According to the results found for the KNN for regression, the metrics of RMSE, RMSLE and MAE present better results when the algorithm is in its twelfth iteration (k = 11), as defined in Figure 3

In Figure 3, the x-axis represents the number of iterations tested and the y-axis the metric value (RMSE, RMSLE and MAE). In this three results, it is possible to identify that the best performance occurs when the number of neighborhoods

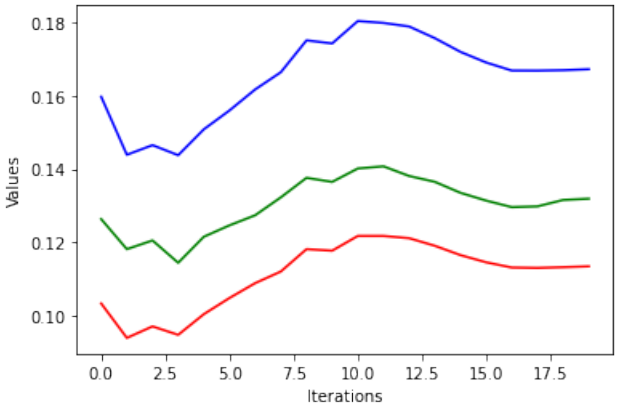

Figure 3. RMSE, RMSLE and MAE of the eleventh iteration

is equal to 11, which is then considered the ideal value for the hyperparameter. To reach this results, k were tested between 2 and 20 (number of neighborhoods), resenting the results of the top of the curvature (k between 9 and 13) described at Table 7

k	**9**	**10**	**11**	**12**	**13**
RMSE	0.1751	0.1743	0.1804	0.1799	0.1789
RMSLE	0.1181	0.1177	0.1217	0.1217	0.1211
MAE	0.1376	0.1365	0.1401	0.1407	0.1381

Table 7: Curve results between neighborhoods 9 and 13

In Table 7 is possible to verify the ascension and descent of the performance curve of k, where $k = 11$ maintains the best values of RMSE, RMSLE and MAE. Once the *k*NN, RF and SVR values were calculated, the comparison of results between the techniques is shown in Figure 4.

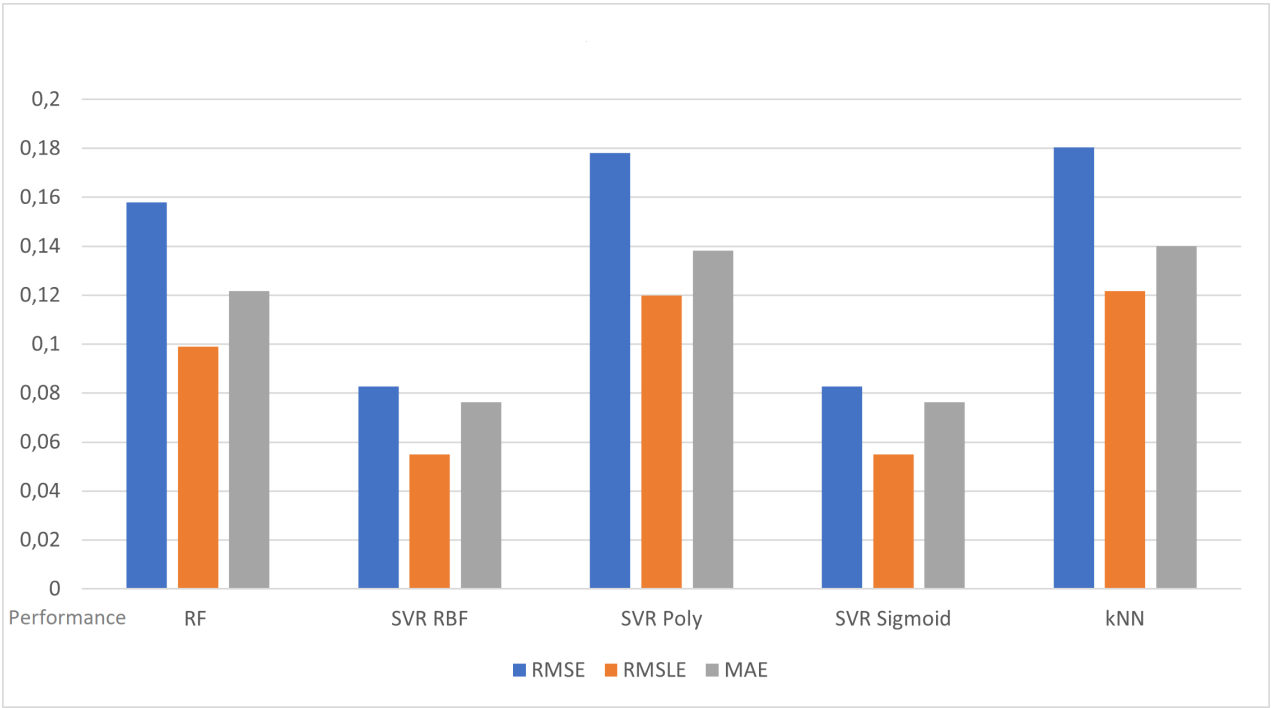

Figure 4. Comparison between RF, SVR and kNN results

The comparison graph shows that the technique that obtained the best overall performance for the Itaipu data set was *k*NN (RMSE about 18%, RMSLE 12% and MAE 14%). This better performance occurs even among other techniques that use parameterization that use several different kernels. The values presented for kNN are close to those of SVR with the polynomial kernel (RMSE about 17%, RMSLE 11% and MAE 14%). The RF technique showed a performance close

to the first two, but with a greater difference in results(RMSE about 15%, RMSLE 9% and MAE 12%). The SVR with kernel RBF and Sigmoid showed the worst results (which are very similar - RMSE about 8%, RMSLE 5% and MAE 7%).

5. CONCLUSIONS

In this paper, the results of the comparison of three predictive regression models for the prediction of streamflow time series for power generation were presented: Random Forest, Support Vector Regression and *k*NN. The results showed a higher efficiency of *k*NN, SVR (Polynomial) and RF, in this order.

However, among all the tested models, the SVR is the one with the most adjustable hyperparametrization possibility. Even though its parameters have been configured with an estimation function, it is still possible to try to make changes that make the model more efficient in relation to the others presented in this study.

Considering that intuitively, the polynomial kernel looks not only at the resources provided from input samples to determine their similarity, but also combinations of these, it is possible to conclude that this characteristic of the SVR polynomial kernel comparatively represented a difference in the SVR in relation to the other kernels tested (RBF and Sigmoid). This means that the analysis of this data set is influenced by the interaction features.

It is also concluded that, within the scope covered by this article and in the pre-processing of streamflow data performed, there was a difference between nonlinear techniques (Random Forest, *k*NN and polynomial SVR) and linear ones (SVR rbf and SVR sigmoid). This leads to the conclusion that for the erratic space of this data set, possibly a non-linear analysis will obtain more accurate results.

6. REFERENCES

Altman, N.S., 1992. "An introduction to kernel and nearest-neighbor Nonparametric Regression". *American Statistician*, Vol. 46, No. 3, pp. 175–185. ISSN 15372731. doi:10.1080/00031305.1992.10475879.

Breiman, L., 2001. "random forests". *Machine Learning*, Vol. 45, pp. 5–32. doi:10.1023/A:1010933404324. URL https://www.taylorfrancis.com/books/9781000730197/chapters/10.1201/9780367816377-11.

Cai, L., Yu, Y., Zhang, S., Song, Y., Xiong, Z. and Zhou, T., 2020. "A sample-rebalanced outlier-rejected k-nearest neighbor regression model for short-term traffic flow forecasting". *IEEE Access*, Vol. 8, pp. 22686–22696.

Cheng, M., Fang, F., Kinouchi, T., Navon, I. and Pain, C., 2020. "Long lead-time daily and monthly streamflow forecasting using machine learning methods". *Journal of Hydrology*, Vol. 590, p. 125376. ISSN 0022-1694. doi:https://doi.org/10.1016/j.jhydrol.2020.125376. URL http://www.sciencedirect.com/science/article/pii/S0022169420308362.

Cortes, C. and Vapnik, V., 1995. "support-vector networks". *Machine Learning*, Vol. 20, No. 3, pp. 273–297. ISSN 08856125. doi:10.1007/BF00994018.

da Silva, R.G., Ribeiro, M.H.D.M., Mariani, V.C. and dos Santos Coelho, L., 2020. "Forecasting brazilian and american covid-19 cases based on artificial intelligence coupled with climatic exogenous variables". *Chaos, Solitons Fractals*, Vol. 139, p. 110027. ISSN 0960-0779. doi:https://doi.org/10.1016/j.chaos.2020.110027. URL http://www.sciencedirect.com/science/article/pii/S0960077920304252.

Darbandsari, P. and Coulibaly, P., 2020. "Introducing entropy-based bayesian model averaging for streamflow forecast". *Journal of Hydrology*, Vol. 591, p. 125577. ISSN 0022-1694. doi:https://doi.org/10.1016/j.jhydrol.2020.125577. URL http://www.sciencedirect.com/science/article/pii/S0022169420310374.

Drucker, H., Surges, C.J., Kaufman, L., Smola, A. and Vapnik, V., 1997. "support vector regression machines". *Advances in Neural Information Processing Systems*, Vol. 1, pp. 155–161. ISSN 10495258.

Itaipu Binacional, 2020a. "Geração". URL https://www.itaipu.gov.br/energia/geracao (In Portuguese).

Itaipu Binacional, 2020b. "recordes operativos". URL https://www.itaipu.gov.br/energia/recordes-operativos.

Liu, D., Jiang, W., Mu, L. and Wang, S., 2020. "streamflow prediction using deep learning neural network: case study of yangtze river". *IEEE Access*, Vol. 8, pp. 90069–90086. ISSN 21693536. doi:10.1109/ACCESS.2020.2993874.

Ni, L., Wang, D., Wu, J., Wang, Y., Tao, Y., Zhang, J. and Liu, J., 2020. "streamflow forecasting using extreme gradient boosting model coupled with gaussian mixture model". *Journal of Hydrology*, Vol. 586, No. March, p. 124901. ISSN 00221694. doi:10.1016/j.jhydrol.2020.124901. URL https://doi.org/10.1016/j.jhydrol.2020.124901.

Pimenta, R.G., Gaio, G., Franco, E.M. and Muller, M.R., 2018. "Short Term Load Forecasting for Power Exchange between Brasil and Paraguay". *SBSE 2018 - 7th Brazilian Electrical Systems Symposium*, pp. 1–6. doi:10.1109/SBSE.2018.8395739.

Ribeiro, M.H.D.M., da Silva, R.G., Mariani, V.C. and dos Santos Coelho, L., 2020a. "Short-term forecasting covid-19 cumulative confirmed cases: Perspectives for brazil". *Chaos, Solitons Fractals*, Vol. 135, p. 109853. ISSN 0960-0779. doi:https://doi.org/10.1016/j.chaos.2020.109853. URL http://www.sciencedirect.com/science/article/pii/S0960077920302538.

Ribeiro, M.H.D.M., Mariani, V.C. and dos Santos Coelho, L., 2020b. "Multi-step ahead meningitis case forecasting based on decomposition and multi-objective optimization methods". *Journal of Biomedical In-*

formatics, Vol. 111, p. 103575. ISSN 1532-0464. doi:https://doi.org/10.1016/j.jbi.2020.103575. URL http://www.sciencedirect.com/science/article/pii/S1532046420302033.

Rodrigues Moreno, S., Gomes da Silva, R., Cocco Mariani, V. and dos Santos Coelho, L., 2020. "Multi-step wind speed forecasting based on hybrid multi-stage decomposition model and long short-term memory neural network". *Energy Conversion and Management*, Vol. 213, p. 112869. ISSN 0196-8904. doi:https://doi.org/10.1016/j.enconman.2020.112869. URL http://www.sciencedirect.com/science/article/pii/S0196890420304076.

Segal, M.R., 2004. "machine learning benchmarks and random forest Regression". URL https://escholarship.org/uc/item/35x3v9t4.

Shoaib, M., Shamseldin, A.Y., Khan, S., Khan, M.M., Khan, Z.M. and Melville, B.W., 2018. "a wavelet based approach for combining the outputs of different rainfall–runoff models". *Stochastic Environmental Research and Risk Assessment*, Vol. 32, No. 1, pp. 155–168. ISSN 14363259. doi:10.1007/s00477-016-1364-x.

Stone M., 1974. "cross-validatory choice and assessment of statistical Predictions". *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 36, No. 2, pp. 111–147.

Tongal, H. and Booij, M., 2018. "simulation and forecasting of streamflows using machine learning models coupled with base flow separation". *Journal of Hydrology*, Vol. 564. doi:10.1016/j.jhydrol.2018.07.004.

Trierweiler Ribeiro, G., Guilherme Sauer, J., Fraccanabbia, N., Cocco Mariani, V. and dos Santos Coelho, L., 2020. "Bayesian optimized echo state network applied to short-term load forecasting". *Energies*, Vol. 13, No. 9. ISSN 1996-1073. doi:10.3390/en13092390. URL https://www.mdpi.com/1996-1073/13/9/2390.

Vapnik, V.N., 1995. *the nature of statistical learning theory*. Springer-Verlag, Berlin, Germany. ISBN 0387945598.

Wong, S. and Zhou, X., 2008. *computational systems bioinformatics - methods and biomedical applications*. World Scientific Publishing Company. ISBN 9789813106994. URL https://books.google.com.br/books?id=WBI8DQAAQBAJ.

Wyatt, B.M., Ochsner, T.E., Krueger, E.S. and Jones, E.T., 2020. "In-situ soil moisture data improve seasonal streamflow forecast accuracy in rainfall-dominated watersheds". *Journal of Hydrology*, Vol. 590, p. 125404. ISSN 0022-1694. doi:https://doi.org/10.1016/j.jhydrol.2020.125404. URL http://www.sciencedirect.com/science/article/pii/S0022169420308647.

Yang, Q. and Webb, G., 2006. *pricai 2006: trends in artificial intelligence: 9th pacific rim international conference on artificial intelligence, guilin, china, August 7-11, 2006, proceedings*. Lecture Notes in Artificial Intelligence. Springer. ISBN 9783540366676. URL https://books.google.com.bn/books?id=CSz1tORiKIAC.

Yasseen, Z., El-Shafie, A., Jaafar, O., Afan, H. and Sayl, A., 2015. "artificial intelligence based models for streamflow forecasting: 2000-2015". *Journal of Hydrology*, Vol. 530, pp. 829–844. doi:10.1016/j.jhydrol.2015.10.038.

Yu, X., Wang, Y., Wu, L., Chen, G., Wang, L. and Qin, H., 2020. "comparison of support vector regression and extreme gradient boosting for decomposition-based data-driven 10-Day streamflow forecasting". *Journal of Hydrology*, Vol. 582, p. 124293. ISSN 00221694. doi:10.1016/j.jhydrol.2019.124293. URL https://doi.org/10.1016/j.jhydrol.2019.124293.

Zhao, X., Guo, X., Luo, J. and Tan, X., 2018. "efficient detection method for foreign fibers in cotton". *Information Processing in Agriculture*, Vol. 5, No. 3, pp. 320–328. ISSN 22143173. doi:10.1016/j.inpa.2018.04.002. URL https://doi.org/10.1016/j.inpa.2018.04.002.