Forecasting Drought Indices Using Machine Learning Algorithm

Dr. Jay Lee P.E., California Baptist University

Dr. Lee's research interest is in information technology and strategic decision-making practices in various engineering management fields. His current research topics include spatial analysis utilizing a commercial Geographic Information System (GIS) applications and machine learning-based forecasting in engineering practices.

Dr. Yeonsang Hwang P.E., Arkansas State University

Associate Professor of Civil Engineering

Dr. Tae-Hoon Kim, Purdue University Northwest

Dr. Tae-Hoon Kim, Ph.D in Information Science, associate professor of Computer Information Technology and Graphics. His teaching areas are computer networking, network security, network design, parallel computing, and data science. His research interests are reliable wireless sensor and ad hoc network, network anomaly detection, cyber-physical system, and applied data science.

Forecasting Drought Indices Using Machine Learning Algorithm

Jay Lee, Tae-Hoon Kim, and Yeonsang Hwang

Abstract

According to the existing studies, the historical climate record and seasonal temperature and precipitation records offer useful input for making short-term drought predictions. In the last few decades, numerous studies have been conducted to explore these data in a way to predict upcoming drought events. Despite the efforts, few studies have succeeded in quantifying uncertainties in the process of predicting drought index values due mainly to technical challenges and implications in computation. This paper proposes a new approach utilizing an artificial intelligence model for forecasting drought indices. This study uses a regression analysis model in machine learning, Lasso, which is normalized to improve the prediction accuracy. Lasso model will be implemented in Python using scikit-learn, and 10-fold cross-validation will be used to ensure the prediction accuracy. The proposed model uses the National Oceanic and Atmospheric Administration (NOAA) Climate Prediction Center (CPC) seasonal data to compute the Palmer Drought Severity Index (PDSI). The accuracy of the model is validated using the historical records of drought indices and available seasonal temperature and precipitation data provided by the NOAA CPC. The results of the forecasts produced by this model will be compared with the observed drought indices and validated. The mean error rate and root mean square error (RMSE) methods are used to measure the accuracy of the forecast at stations for validation. The validated model can be used in classroom and laboratory settings for general engineering studies.

1. Introduction

Drought is a part of the natural variability consists of various hydrologic interactions such as precipitation, evaporation, soil moisture, and groundwater level that are challenging to predict in advance. The major importance in the mitigation of drought impact is the effective methods for forecasting key features of drought events. Agriculture, environmental, and societal impact of drought has long been discussed within the climate and hydrologic communities [1], [2]. Intensified climate variability and increasing frequency of extreme climate events make the drought mitigation even more challenging [3].

While various drought mitigation strategies are explored and discussed, the first step is to utilize a good objective measure to understand the drought onset and progression for water resource decision-makers. Therefore, the analysis and forecast of the drought indices are possible. Among many other drought monitoring methods, Palmer Drought Severity Index (PDSI), Crop Moisture Index (CMI), and Standard Precipitation Index (SPI) are among many popularly used measures as stand-alone drought indices. US Drought Monitor is another effective and comprehensive approach to provide quantitative and visual information to all water users (https://droughtmonitor.unl.edu/). PDSI is the first comprehensive drought index developed to explain drought stages that impact agriculture using precipitation and temperature to estimate moisture supply and demand. The original idea was developed by Palmer [4], and some related indices have been derived. As a monthly time series, PDSI calculates the regionally standardized drought severity by comparing recent climate conditions and long-term averaged conditions and

is arguably the most important drought monitoring index for the spatial and temporal completeness.

Drought forecasts rely on a variety of methods. Yevjevich [5] introduced run theory that led to subsequent research developments until autoregressive (AR) model studies have become also very popular, e.g., Rao and Padmanabhan [6] studied PDSI forecasting. Stochastic model predictions, such as the Markov chain model, has been extensively studied [7]. For instance, nonparametric techniques [8] improved computing capabilities, which have many other hybrid applications. Data-driven techniques and machine learning approaches are among the recent developments in drought forecasts. Other researches have been pursued to provide probabilistic forecasts to explain uncertainties in the natural climate processes. Ensemble forecasts have been tried based on Monte Carlo, Bayesian technique, and the AR model. The simple conditional re-sampling technique has been tested by Hwang and Carbone [9] to properly explain natural variability with the help of the Climate Prediction Center (CPC) outlook and current temperature-precipitation condition.

2. Machine Learning Algorithms

Machine Learning (ML) is recently adopted in many different fields of application, such as finance, medical and healthcare, life science, security, automobile, etc. These wide applications are due mainly to its continuous evolution from experience. In ML, there are two subsets, supervised learning and unsupervised learning. Supervised learning uses the set of labeled data to train the model for the prediction, while unsupervised learning uses the set of non-labeled data. Due to this nature, supervised learning is being used to estimate or predict the value from known data patterns and mainly used for regression or classification in many applications, such as weather forecast, email filtering, Intrusion Detection System (IDS), etc. [10] - [13]. Unsupervised Learning is good for data with unknown patterns and mainly used for clustering or grouping the data in applications of market segmentation, social network analysis, organizing computer clusters, etc. [14], [15]. While ML has been widely adapted and being applied in many areas of study, it has not been actively adapted and implemented in the biomass energy field of study. Ozbas and et al. [16] compared four ML models, Linear Regression (LR), K Nearest Neighbors (KNN) Regression, Support Vector Regression (SVR), and Decision Tree Regression (DTR), to predict hydrogen production from biomass gasification. In their study, coefficient of determination (R^2), Mean Absolute Error (MAE), and Root Mean Square Error (MSE) are used to compare those models, and LR outperformed. Monroy and et al. [17] implemented Support Vector Machine (SVM) to predict light intensity using both experimental and simulated data for batch hydrogen production. Whiteman and Kana [18] proposed to use Artificial Neural Network (ANN) to find the relationships between process inputs for fermentative biohydrogen production, and their results show high accuracy in modeling the relationships.

Linear Regression (LR)

Among the various methods above, Linear Regression (LR) is the simplest and most common statistical technique for prediction modeling. LR utilizes a given dataset of independent variables with corresponding dependent variables to provide the linear regression equation. LR finds out the coefficient for each independent variable, which induces the least residual error (i.e., the

difference between dependent variable value and predicted value). The hypothesis function of ML LR with multiple variables is shown in Eq. 1 and the cost function for the parameter vector θ in R^{n+1} is in Eq. 2 where $\bar{\theta}$ is the coefficient vector, x_i are the variables, m is the number of samples or data, and y is a target value.

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots + \theta_n x_n \quad \text{Eq. 1}$$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 \quad \text{Eq. 2}$$

Linear Regression in ML uses a gradient descent technique to find the coefficient vector, $\bar{\theta}$, which minimizes cost function, $J(\theta)$. Ordinary Least Squares (OLS) LR model is a type of linear least-squares method to estimate unknown parameters, coefficient vector, of linear regression equation in Eq. 1, which minimizes the cost function, $J(\theta)$, in Eq. 2. In multiple variable problems in LR, multicollinearity has to be considered, which indicates the existence of near-linear relationships among the independent variables. When multicollinearity exists among the independent variable dataset, the prediction of OLS LR may be unbiased, but prediction variance would be large, and the predicted value would not be a true value. To reduce this type of error, a degree of bias should be added to the regression estimates. Ridge and Lasso are two common regularized LR by adding a penalty to the cost function. Ridge LR adds penalty equivalent to the sum of the square of coefficients and Lasso LR adds sum of the absolute value of coefficients to the cost function. Eq. 3 and Eq. 4 show the cost functions of Ridge and Lasso, respectively, where p is the number of independent variables and a is a regularization parameter. When a is zero, the cost function is equivalent to the OLS LR model. With larger value, the model penalizes the coefficients more and reduces the complexity and multicollinearity.

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \alpha\sum_{j=1}^{p}\theta_j^2 \quad \text{Eq. 3}$$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \alpha\sum_{j=1}^{p}\left|\theta_j\right| \quad \text{Eq. 4}$$

Lasso is selected in this study since it identifies and utilizes the more related features for the prediction.

Support Vector Regression (SVR)

Support Vector Machine (SVM) can also be used for regression and Support Vector Regression (SVR) uses the same basic principles with few differences where it finds the regression function as flat as possible with low prediction error [19]. For a given dataset of independent variables, SVR finds the linear function of Eq. 5 that all values of the dependent variable are within a given tolerance.

$$y = wx + b \quad \text{Eq. 5}$$

Given a function, SVR seeks for small w to make Eq. 5 flat. In other words, SVR tries to find the minimum w, which satisfies all data points that are close to the leaner function with a variation of ε. This problem can be formulated as Eq. 6 and Eq. 7 for I = 1, …, n.

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \qquad \text{Eq. 6}$$

$$subject\ to \quad \begin{cases} y_i - wx_i - b \le \varepsilon + \xi_i \\ wx_i + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* > 0 \end{cases} \qquad \text{Eq. 7}$$

In the formulation, SVR utilizes C for the trade-off between the flatness of function and the number of deviations larger than tolerance, ε. However, SVR described above only applies to linear dataset. For the non-linear dataset, SVR uses a kernel function that transforms the data into a higher dimensional feature space, which transforms non-linear dataset to the linear form of the dataset in order to apply linear separation. The optimization model for non-linear SVR can be formulated in Eq. 8 where α_i and α_i^* are Lagrange multipliers and $K(x_i, x)$ is the Kernel.

$$y = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) \cdot K(x_i, x) + b \qquad \text{Eq. 8}$$

One of the most commonly used kernel functions is Radial Basis Function (RBF) and it is shown in Eq. 9, where γ is a free parameter.

$$K(x, x') = exp(\gamma\|x - x'\|^2) \qquad \text{Eq. 9}$$

The optimization parameters of SVR (RBF) are C and γ, which should be selected carefully.

K-Nearest Neighbor Regression (KNN)

KNN (K-Nearest Neighbor) is also a supervised learning algorithm. Unlike other methods, KNN simply uses a training dataset to predict the result based on the outputs of nearest neighbors without generalizing the model. As KNN is known as instance-based or lazy learning, it finds the average of the values from k nearest neighbors. The weights are given based on the distance, Euclidean or Hamming. Euclidean distance is generally used for continuous values, while Hamming distance is used for discrete values. The nearer neighbor contributes more (more weight) than the others. The k value should be determined carefully. The smaller k results in higher variance or less stable results, while the larger k results in higher bias or less precise results. In general, adaptive method, heuristics, or cross-validation is used to select proper k value.

3. Modeling and results

Three ML models are used in this study to forecast the drought index, and the results are compared in this section. Before modeling the selected ML methods, data needs to be cleaned. The following discusses the data preparation followed by the modeling process with the results.

Evaluation of model

The data used in this study includes a set of seasonal weather data to compute the Palmer Drought Severity Index (PDSI), which includes precipitation, maximum temperature, and

minimum temperature collected by the National Oceanic and Atmospheric Administration (NOAA) Climate Prediction Center (CPC) for 122 years from 1895 to 2017. Since there is no direct correlation between the data collected from different locations, this study classifies the data by area for modeling. For the best result, this study normalizes the PDSI into the range of 1 to 3 for wet, normal, and drought. The study also compares the models with different input features. The basic input feature includes precipitation, maximum temperature, and minimum temperature. The study then adds month as the fourth input feature and year as the fifth input feature, respectively.

In modeling, Lasso Linear Regression (Lasso), SVR (RBF), and KNN algorithms are implemented in Python to predict the PDSI and compared it to measure its performance. Hyper-parameters should be carefully selected for the best fitting in each model. This study uses a grid search to find those hyper-parameters. Table 1 shows the selected hyper-parameters of each model. Alpha value of 10-6 in Lasso, Cost of 1 and gamma of 8 in SVR, and the number of neighbors of 8 in KNN for all three different numbers of feature testing.

Table 1. Hyper-parameters

	Lass	SVR	KNN
Hyper-parameters	$\alpha = 10^{-6}$	cost = 1 gamma = 8	k = 8

Once the hyper-parameters are selected, each model is validated to avoid the under and overfitting problem. The validation process splits the data into three folds of the dataset, training, testing, and validation datasets. The training dataset is used to train the model, and the test dataset is to test the trained model. The validation dataset is to apply the trained model to validate the testing procedure result. The purpose of validation ensures to resolve the overfitting problem. However, a single validation process does not guarantee to resolve such a problem. Therefore, K-fold cross-validation is generally utilized for the ML model validation. K-fold cross-validation randomly splits the dataset into K mutually exclusive groups and evaluates the model for K times. Each evaluation, K-1 group of datasets are used for the training, and one group is used for testing. A different group is chosen for the testing dataset for each evaluation. The evaluation score is calculated for each validation and averaged to evaluate the performance of the model. 10-fold cross-validation is used in this study. Root Mean Square Error (RMSE) is then used to compare the performance of each model in this study, which is determined by the square root of the sum of the difference between target and prediction values as shown in Eq. 10, where y_i is a target and p_i is a prediction value.

$$RMSE = \sqrt{\sum(y_i - p_i)^2} \qquad \text{Eq. 10}$$

Once the model is trained, this study computes RMSE using both training and testing dataset for each K. Then mean RMSE from 10-Fold cross-validation is shown in Table 2.

Table 2. Root Mean Square Errors

	3 features		4 features		5 features	
	Training	**Testing**	**Training**	**Testing**	**Training**	**Testing**
Lasso	0.3133	0.3153	0.3122	0.3127	0.3110	0.3117
SVR	0.2970	0.3077	0.2747	0.2990	0.2322	0.3145
KNN	0.2818	0.3194	0.2718	0.3111	0.2768	0.3149

The RMSE from the training dataset indicates how close the model fits the training dataset, while the one from testing shows how good the model predicts. High training RMSE is an indication of underfitting, while low training RMSE with high testing RMSE is happening for overfitting. Therefore, the lower in both training and testing RMSE, the better is fitting model. Both RMSE values are compared with the number of input features in Figures 1 and 2.

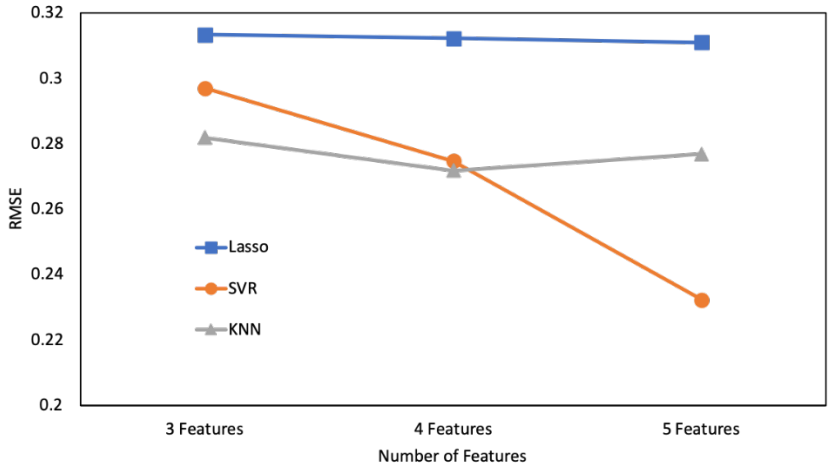

Figure 1. RMSE of training dataset

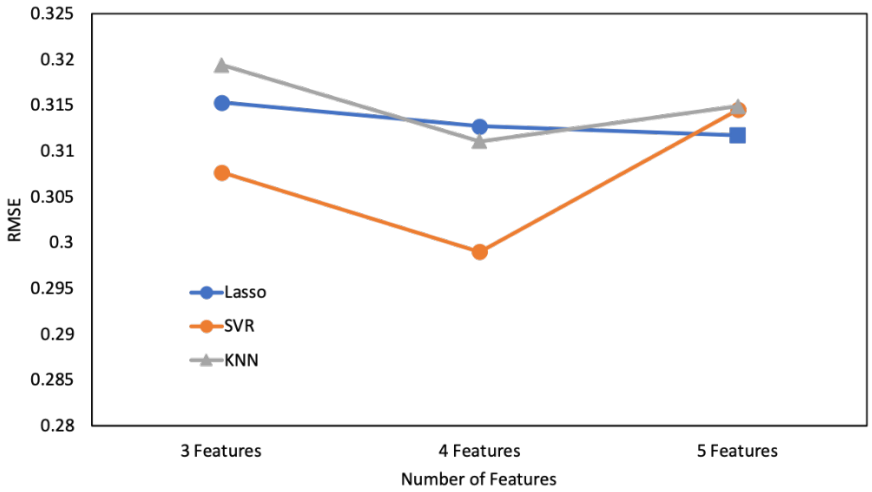

Figure 2. RMSE of testing dataset

Figure 1 indicates that the model fitting is getting better as the number of input feature increases for both Lasso and SVR algorithms. There is no significant difference in the KNN algorithm. The SVR algorithm shows significantly better fitting to the training dataset with the increment of the number of input features. However, the RMSE of the testing dataset, as in Figure 2, shows somewhat different results. The Lasso algorithm shows gradual prediction improvement as the number of input features increases. For both the KNN and SVR algorithms, the prediction error is lower with the 4-input feature option. The SVR algorithm shows the highest prediction error with a 5-input feature option. Since the Lasso algorithm has the capability of removing non-related features, continuously both RMSEs are decreasing. However, for the KNN and SVR algorithms, it appears that overfitting occurs when the number of input features increases. Therefore, the number of input features does not harm the Lasso model, but it is the matter with the SVR or KNN model. Therefore, the number of input features should be carefully selected for both the KNN and SVR algorithms to avoid overfitting. Overall, the SVR algorithm shows the best prediction result with the 4-input feature option.

4. Conclusions

This study proposes a Machine Learning (ML) model for forecasting drought indices. This study used a regression analysis model in Lasso, which is normalized to improve prediction accuracy. The proposed model used the National Oceanic and Atmospheric Administration (NOAA) Climate Prediction Center (CPC) seasonal data to compute the Palmer Drought Severity Index (PDSI). The accuracy of the model was validated using the historical records of drought indices and available seasonal temperature and precipitation data provided by the NOAA CPC. The results of the forecasts produced by this model were validated by the observed drought indices. The mean error rate and root mean square error (RMSE) indicated that the accuracy of the forecast at stations is best when the SVR algorithm was used with the 4-input feature option.

5. Implementation in classroom

The proposed artificial intelligence (AI) model can be used in various classroom and laboratory settings with machine learning software packages for general engineering studies thanks mainly to its versatility in modeling the procedures of forecasting any future events based on big data. It would be particularly beneficial to a student who attempts to learn how to predict future weather events for planning purposes in the AI approach. In a traditional approach, Ordinary Least Square (OLS) regression is one of the methods generally used for the prediction. By using the machine learning approach proposed in this paper, a student will learn the state-of-art machine learning approach and compare it with the traditional regression method. Additionally, students will learn how to manage the data set for better prediction as well as the key factors that may affect the overall forecasts.

As far as the application of the proposed model in a classroom setting, one can use the model for either two 4-hr labs or a one-semester project, which includes data collection, modeling, and validation. For the lab instruction, the instructor can guide through the data collection and managing procedures as well as the primary data set for the region of interest during the first lab. The instructor can then teach how to use open-source functions and their functionality for modeling. In the second lab, the instructor can introduce how to train and validate the machine learning models, followed by the instruction of how to run the model for forecasting future events. For a semester project, one can use the model for a group project with 3-4 students. The instructor can provide minimum information only to help with the setup, available data, and validation procedures. It is ideal then to let the student group perform modeling and forecasting on their own and compare the results with other groups.

For a specific classroom example, one can use the proposed model in undergraduate civil engineering courses where climate variability is an important factor. In the undergraduate civil engineering curriculum, courses such as Engineering Hydrology and Water and Waste Systems cover water resources subject areas that discuss basic rainfall-watershed response principles and appropriate response system design, respectively. Accreditation Board for Engineering and Technology, Inc (ABET) requires Student Learning Outcome 2 relevant to the authors' presented work. The Student Learning Outcome 2 reads an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors (https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2019-2020/#GC3, visited Apr. 27, 2020). The application of the proposed model will serve to strengthen the student learning outcome by promoting a new thought process considering the climate variability in water resources systems.

6. References

[1] R. R. Heim, Jr., "A review of twentieth-century drought indices used in the united states," *Bulletin of the American Meteorological Society*, Vol 83(8), pp. 1149-1165, 2002

[2] A. Steinemann, "Using climate forecasts for drought management," J*ournal of Applied Meteorology and Climatology*, Vol 45, pp. 1353-1361, 2006

[3] Ove Hoegh-Guldberg, Daniela Jacob, Michael Taylor, Marco Bindi, Sally Brown, Ines Camilloni, Arona Diedhiou, Riyanti Djalante, Kristie L. Ebi, Francois Engelbrecht, Joel Guiot, Yasuaki Hijioka, Shagun Mehrotra, Antony Payne, Sonia I. Seneviratne, Adelle Thomas, Rachel F. Warren, Guangsheng Zhou, and Petra Tschakert, "Impacts of 1.5ºc global warming on natural and human systems" In Global Warming of 1.5° C: An IPCC Special Report on the impacts of global warming of 1.5° C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty. IPCC, 2018.

[4] W.C. Palmer, "Meteorological drought," Research Paper No. 45, 1965

[5] V Yevjevich, "Objective approach to definitions and investigations of continental hydrologic droughts" *Hydrology papers (Colorado State University)*, No. 23, pp 4-18, 1967

[6] A.R. Rao and G. Padmanabhan, "Analysis and modeling of palmer drought index series," *Journal of hydrology* , Vol 68(1), pp. 211-229, 1984

[7] Z Sen, "Critical drought analysis by second-order markov chain, Journal of hydrology," Vol 120, pp. 183-202, 1990

[8] T.W. Kim, J.B. Valdés, and C. Yoo, "Nonparametric approach for estimating return periods of droughts in arid regions," *Journal of Hydrologic Engineering*, Vol *8*(5), pp. 237-246, 2003

[9] Yeonsang Hwang and Gregory J. Carbone, "Ensemble forecasts of drought indices using a conditional residual resampling technique," *Journal of Applied Meteorology and Climatology*, Vol 48(7), pp. 1289-1301, 2009

[10] Al-Obeidat, Feras, Bruce Spencer, and Omar Alfandi. "Consistently accurate forecasts of temperature within buildings from sensor data using ridge and lasso regression." *Future Generation Computer Systems* 2018.

[11] Manogaran, Gunasekaran, and Daphne Lopez. "A survey of big data architectures and machine learning algorithms in healthcare." *International Journal of Biomedical Engineering and Technology*, Vol 25.2-4, pp. 182-211, 2017

[12] Aybar-Ruiz, A., et al. "A novel grouping genetic algorithm–extreme learning machine approach for global solar radiation prediction from numerical weather models inputs." *Solar Energy*, Vol 132, pp. 129-142, 2016

[13] Yin, Chuanlong, et al. "A deep learning approach for intrusion detection using recurrent neural networks." *IEEE, Access* 5, 2017: 21954-21961.

[14] Borthakur, Debanjan, et al. "Smart fog: Fog computing framework for unsupervised clustering analytics in wearable internet of things." *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, 2017.

[15] Di Capua, Michele, Emanuel Di Nardo, and Alfredo Petrosino. "Unsupervised cyber bullying detection in social networks." *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016.

[16] Ozbas, Emine Elmaslar, et al. "Hydrogen production via biomass gasification, and modeling by supervised machine learning algorithms." *International Journal of Hydrogen Energy,* 44.32 2019: 17260-17268.

[17] Monroy, Isaac, Eliane Guevara-López, and Germán Buitrón. "A mechanistic model supported by data-based classification models for batch hydrogen production with an immobilized photo-bacteria consortium." *International Journal of Hydrogen Energy,* 41.48, 2016: 22802-22811.

[18] Whiteman, J. K., and EB Gueguim Kana. "Comparative assessment of the artificial neural network and response surface modelling efficiencies for biohydrogen production on sugar cane molasses." *BioEnergy Research,* 7.1, 2014: 295-305.

[19] Smola, Alex J., and Bernhard Schölkopf. "A tutorial on support vector regression." *Statistics and computing,* 14.3, 2004: 199-222.