Long and Short Term Energy Demand Forecasting using XGBoost Models

Jose Robles, Undergraduate Student[1], Freedy Sotelo-Valer, Doctor[2], Johnny Nahui-Ortiz, Ph.D[3] and Jorge Lopez-Cordova, Maestro[4]
[1]Universidad Nacional de Ingeniería, Peru, jose.robles.l@uni.pe
[2]Universidad Nacional de Ingeniería, Peru, fresov@uni.edu.pe
[3]Universidad Nacional de Ingeniería, Peru, jnahui@meningenieros.com
[4]Universidad Ricardo Palma, Perú, jorgelopex024@gmail.com

***Abstract*– As part of the technical studies in energy demand required by regulatory entities in Peru, this paper proposes the use of XGBoost Linear and Decision Trees models based on econometric long and short term variables to forecast the energy demand. Considering that data of energy demand per year is only available since 1980, which means a small dataset, Leave-One-Out Cross-Validation method was used in order to measure the performance of the models with unseen data. After the training stage, in terms of econometric variables, models based on long term variables show to be more robust than models with the short term ones. In addition, Decision Trees shows a better performance than Linear Models with a noticeable difference in the coefficient of determination for both training and test datasets.**

***Keywords*-- Energy Demand Forecast, Machine Learning, XGBoost, Cross Validation, Decision Trees, Econometric Models, Artificial Intelligence.**

I. Introduction

Currently, due to the limitation in energy storage technologies, companies in the electrical sector needs to carry out forecast studies on energy demand in order to produce as much as is consumed to achieve a balance in the electrical grid. Moreover, these studies can bring an insight vision if the current grid needs technical improvements to reach the energy demand [1].

According to OSINERGMIN, the regulatory entity of the electrical sector in Peru, companies in charge of the power distribution needs to make investment plans every 4 years forecasting the energy demand for the next 30 years in order to fulfill the electrical requirements in the future. In this context, for energy demand, it is common to use econometric models and time series to understand the behavior of the future demand taking into account econometric variables such as PBI, fares, number of clients and the past behavior of the demand. However, considering that these approaches require an extensive knowledge of economy and statistics, some of the state-of-the-art techniques to model the behavior of the energy demand include the use of machine learning algorithms [2].

Many of the machine learning algorithms in this field are focus on the energy demand for buildings considering variables such as temperature, irradiation, wind speed and previous energy consumed data [3-5]. As a result of this, there is the well-known Demand Response Model which requires a bidirectional communication mechanism to have an online smart grid which uses not only machine learning techniques but other different algorithms to predict the energy demand in a short timeframe of seconds and provide it [2]. However, considering the lack of a bidirectional communication in a conventional electrical grid, up to date there are no model based on machine learning techniques focus on the prediction for long timeframes (months or years) and moreover based on econometric variables.

In this context, this paper solves the necessity to analyze the introduction of machine learning techniques in the energy demand forecast of a conventional electrical grid in contrast with the complex models proposed by the regulatory entities. Thus, considering a supervised machine learning approach, this paper proposes the use of two types of XGBoost Regression models to make the energy demand prediction. The first model is an XGBoost Linear model which simulates the behavior of econometric models. The second model used is an XGBoost Decision Trees model which is an interpretable machine learning model that can bring us a feature importance analysis. Moreover, as the econometric models can be based on long and short term econometric variables, both models are trained with each set of variables to analyze and compare their robustness in terms of the econometric variables and the machine learning model. Finally, taking into account that the small datasets available in the OSINERGMIN database [6] can bias or variance our results (underfitting or overfitting), Leave-One-Out Cross-Validation method was used to measure the mean absolute error of each model with unseen data.

The rest of the paper is organized as follows: Section 2 describes Machine Learning Methods that are used in the paper. Then, Section 3 includes a brief explanation of econometrics variables used as well as the database. Section 4 includes the evaluation scores used to measure the performance of the models. Finally, Section 5 and 6 describes the results obtained from all models and conclusions.

II. Machine Learning Methods

A. XGBoost Models

XGBoost (Extreme Gradient Boosting) is a scalable end-to-end tree boosting system developed by Chen in 2016 [7]. This algorithm, either for regression or classification, is one of the most popular supervised learning methods among data scientist and also it is well known to achieve state-of-the-art results on many machine learning challenges [8].

Digital Object Identifier: http://dx.doi.org/10.18687/LACCEI2021.1.1.219
ISBN: 978-958-52071-8-9 **ISSN**: 2414-6390
DO NOT REMOVE

The main idea of this ensemble method is to combine several weak learners into a strong learner training each learner with the residual error of previous learners. Thus, the strong learner can be expressed as the sum of all weak learners used in the model as shown in (1).

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \quad (1)$$

Where K is the number of weak learners, f_K is the function of each one and x_i are the input variables. At this point, XGBoost Decision Trees and Random Forest are the same model; however, the main difference between them comes from how they are trained. Thus, XGBoost Models are trained by a Gradient Tree Boosting algorithm in which the model is trained in an additive manner taking into account the previous model $\hat{y}_i^{t-1}$ to minimize the objective function (2) adding the new model f_k that most improves our strong learner model. Please refer to [7] for detailed information about the Gradient Tree Boosting algorithm.

$$\ell^t = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \Omega(f_t) \quad (2)$$

Where n is the number of training examples, $l()$ is the training loss term and $\Omega()$ is the regularization term.

B. *Leave-One-Out Cross-Validation*

Leave-One-Out Cross-Validation (LOOCV)is a type of Cross-Validation method in which the number of folds is equal to the size of the dataset. This method allows us to use all training examples as validation points using, for one iteration, data point as the validation set and leaves all other points as the training set.

The main reason to use a cross-validation method is to measure the presence of overfitting or underfitting in each model as well as their performance with unseen data. Meanwhile the main disadvantage of using LOOCV with big datasets is the computational cost, in the case of small datasets this method will help us to know if the model is overfitting or underfitting the training data. This last fact was the main reason to include this method in the study.

III. METHODOLOGY

As this paper was oriented to forecast the energy demand in the Peruvian Electrical Grid, a subset of data belonging to the South region of Lima was selected from the OSINERGMIN database. This dataset consists in 5 main econometric variables for Long Term models such as years, PBI, fares, number of clients and energy sales in GW.h as shown in Table 1.

Meanwhile long term econometric models are based on (3), in the case of short term models it is necessary to find the variations of a long term variable with respect to its previous value as shown in (4). Thus, it is necessary to generate this data from the long term dataset selected.

$$Y_t = \alpha + x_t \quad (3)$$

$$\Delta Y_t = a_0 - \sum_{k=1}^{k^*} b_k \Delta Y_{t-k} - \sum_{j=1}^{j^*} c_j \Delta Y_{t-j} \quad (4)$$

Where Y_t is the energy sales, x_t is the input variables (PBI, fares, clients), Δ is the first order differentiation operator, k* and j* are time delays for input and output variables respectively and finally α , a_0 , b_k and c_j are constants.

TABLE I. LONG TERM ECONOMETRIC DATASET.

Years	PBI	Fares	Clients	Energy Sales
1980	75114.3269	100	263320	619.4552
1981	79600.4952	105.66038	271286	663.3718
1982	78502.1468	118.8679	280470	699.0842
1983	68119.9777	101.8868	295042	711.1492
1984	69748.0665	124.5283	322249	739.1400
1985	70906.1205	132.0755	331903	744.4486
:	:	:	:	:
:	:	:	:	:
2013	200400.6910	64.8647	866901	2558.9500
2014	207997.9440	67.15167	894830	2630.8489
2015	214439.3380	72.5177	914531	2669.4559
2016	220209.5370	76.2006	935177	2731.1979
2017	224828.8320	75.2119	955359	2756.4910
2018	235320.8442	78.4106	983281	2798.8447

Then, considering that the number of examples in the long term dataset is 39, in the case of short term variable we worked with one-time delay to avoid overfitting in our models and the drastic reduction in the number of examples. Thus, Table 2 and Table 3 shows the short term dataset generated and the description of each variable respectively. Having defined the datasets that are used in this paper, it is important to consider that as econometric models are not using the time as a variable, this variable will be omitted for our models.

TABLE II. SHORT TERM ECONOMETRIC DATASET.

Years	DPBI	D1PBI	…	DEnergy
1982	-1098.3484	4486.1683	…	35.7124
1983	-10382.1691	-1098.3484	…	12.065
1984	1628.0888	-10382.1691	…	27.9908
1985	1158.0540	1628.0888	…	5.3086
1986	10176.2344	1158.0540	…	49.2252
1987	10126.6760	10176.2344	…	100.8634
:	:	:	:::	:
:	:	:	:::	:
2013	10803.3700	10854.4450		109.6500
2014	7597.2530	10803.3700		71.8989
2015	6441.3940	7597.2530	…	38.6070
2016	5770.1990	6441.3940	…	61.7421
2017	4619.2950	5770.1990	…	25.2931
2018	10492.0122	4619.2950	…	42.3537

TABLE III. SHORT TERM VARIABLES.

Variable	Description
DPBI	Variation of PBI
D1PBI	One-time delay Variation of PBI
DFARES	Variation of Fares
D1FARES	One-time delay Variation of Fares
DCLIENT	Variation of Clients
D1CLIENT	One-time delay Variation of Clients
D1ENERGY	One-time delay Variation of Energy Sales
DENERGY	Variation of Energy Sales

Thus, once data was processed in Python, the function describe() from pandas' library was used to verify if data needs to be normalized. In general, data normalization is used when input variables have a different range of values widely dispersed among themselves. By this normalization, it is possible to decrease the number of iterations that the learning algorithm needs to carry out and thus its convergence is faster than models without data normalization. This latter is due to the fact that if a variable has a standard deviation much higher than the others, this variable will have a greater predominance in the objective function and consequently the learning algorithm will make the model parameters oscillate inefficiently and the model will not be able to learn from the other variables as it should be expected [9]. In this context, Fig. 1 and 2 shows values as mean, standard deviation, minimum and maximum values of each variable for large and short term variables.

	PBI	FARES	CLIENTS
count	39.000000	39.000000	39.000000
mean	116414.873618	75.184732	578183.923077
std	53921.348046	23.137765	220375.493610
min	62084.724160	13.207547	263320.000000
25%	77957.842370	64.935510	374711.000000
50%	93423.028820	75.471698	587766.000000
75%	148663.059500	82.037439	738724.000000
max	235320.844200	132.075472	983281.000000

Fig. 1. Description of Long Term Variables.

	DPBI	D1PBI	DFARES	D1FARES	DCLIENT	D1CLIENT	D1ENERGY
count	37.000000	37.000000	37.000000	37.000000	37.000000	37.000000	37.000000
mean	4208.658082	4046.337976	-0.736482	-0.669948	19243.108108	18703.756757	57.757724
std	6927.929678	6846.501844	13.490320	13.516312	11391.033013	11441.019645	47.697311
min	-15589.425770	-15589.425770	-35.849057	-35.849057	-7682.000000	-7682.000000	-98.421307
25%	494.157000	494.157000	-4.723372	-4.723372	11248.000000	11173.000000	33.275000
50%	4619.295000	4486.168300	0.593052	0.593052	18583.000000	16645.000000	54.286517
75%	10158.775000	10126.676000	5.303032	5.366023	24348.000000	24332.000000	90.942000
max	15713.704000	15713.704000	28.301887	28.301887	58361.000000	58361.000000	141.125000

Fig. 2. Description of Short Term Variables.

It can be seen that variables, for both long and short term, have wide dispersed values of mean and standard deviation which means that they need a data normalization. From all data normalization methods available, we selected the data standardization which realize the normalization using the following equation:

$$z = \frac{x - u}{s} \tag{5}$$

Where x is the input variable, u its mean, s its standard deviation and z is the normalized variable. Applying (5) to all input variables, it can be seen from Fig. 3 and 4 that now all variables have a standard deviation value of 1 while their means are in closer values. With all variables normalized, the next step was to make the data split.

In the context of this study, as long and short term dataset consists each one in 39 and 37 examples respectively, we made the partition using 20% of the dataset in the test set for each set of variables resulting in 31 and 29 training examples respectively.

	PBI	FARES	CLIENTS
count	3.900000e+01	3.900000e+01	3.900000e+01
mean	-3.757678e-16	-9.109522e-17	-2.277381e-17
std	1.000000e+00	1.000000e+00	1.000000e+00
min	-1.007581e+00	-2.678616e+00	-1.428761e+00
25%	-7.132060e-01	-4.429651e-01	-9.233010e-01
50%	-4.263960e-01	1.240249e-02	4.348068e-02
75%	5.980597e-01	2.961698e-01	7.284843e-01
max	2.205174e+00	2.458783e+00	1.838213e+00

Fig. 3. Description of Normalized Long Term Variables.

	DPBI	D1PBI	DFARES	D1FARES	DCLIENT	D1CLIENT	D1ENERGY
count	3.700000e+01	3.700000e+01	3.700000e+01	3.700000e+01	3.700000e+01	37.000000	3.700000e+01
mean	-4.800964e-17	-9.339376e-17	3.300663e-17	1.894130e-17	1.440289e-16	0.000000	2.460494e-16
std	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000	1.000000e+00
min	-2.857720e+00	-2.867999e+00	-2.602798e+00	-2.602715e+00	-2.363711e+00	-2.306242	-3.274378e+00
25%	-5.361632e-01	-5.188315e-01	-2.955371e-01	-2.998912e-01	-7.018774e-01	-0.658224	-5.132936e-01
50%	5.927267e-02	6.424161e-02	9.855462e-02	9.344265e-02	-5.794980e-02	-0.179945	-7.277574e-02
75%	8.588593e-01	8.880941e-01	4.476924e-01	4.465694e-01	4.481500e-01	0.491935	6.957263e-01
max	1.660676e+00	1.704135e+00	2.152534e+00	2.143472e+00	3.434095e+00	3.466233	1.747840e+00

Fig. 4. Description of Normalized Short-Term Variables.

Thus, to split the training and test datasets and the consequently implementation of XGBoost models, libraries Scikit-learn and XGBoost were used. As mentioned before, for each set of variables, XGBoost Linear model and XGBoost Decision Trees model were created as follows:

```
# Models for Long Term variables
modelLT1 = XGBRegressor(random_state=0, booster='gblinear')
modelLT2 = XGBRegressor(random_state=0, booster='gbtree')

# Models for Short Term variables
modelST1 = XGBRegressor(random_state=0, booster='gblinear')
modelST2 = XGBRegressor(random_state=0, booster='gbtree')
```

Finally, with models created and considering the number of training examples, we proceeded to perform the LOOCV method in order to know if these models can be used to predict the energy demand without overfitting the training set.

IV. Evaluation

In order to measure the performance of each model, in the case of the cross-validation, the score function used was the Mean Absolute Error (MAE). MAE score is commonly used in cross-validation methods due to its simplicity to compare the performance of each model. Thus, the MAE score is defined formally as shown below:

$$MAE = \frac{\sum_{i=1}^{n} | y_i - \hat{y}_i |}{n} \quad (6)$$

Where n is the number of examples, y_i is the target variable and $\hat{y}_i$ is the predicted value of the target variable using the model. Once MAE score in LOOCV was evaluated and compared between each model, all models were trained and tested considering for this case the coefficient of determination R2 which is defined as follows:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \quad (7)$$

Where $\bar{y}$ is the mean of the target variable. Thus, for this score the best possible punctuation is 1.0 and it can be negative (because the model can be arbitrarily worse) while for a model which always predicts the expected value of y disregarding the input features would get a score of 0.0 [10]. The main reason to use R2 score is because it is a common metric used when econometric models are fitted.

V. Results

Results of MAE score obtained from LOOCV are shown below. As can be seen, Short Term Models (STM) have less MAE scores than Long Term Models (LTM); however, it is necessary to indicate that target variables for both group of models are in a different scale due to LTM has the energy sales as variable while STM has the variation of energy sales. For this reason, the comparison in LOOCV is made between machine learning models. In the case of LTM, XGBoost Decision Trees appears to be the best one over the XGBoost Linear while in the case of STM both models have almost the same MAE score.

After this first stage, having an insight vision of the performance of all models with LOOCV results, our models were trained and then their R2 scores were found for both training and test sets.

```
#LOOCV results for Long Term Models
>>> Average MAE score Linear: 228.5189
>>> Average MAE score Tree: 67.3560

#LOOCV results for Short Term Models
>>> Average MAE score Linear: 20.8648
>>> Average MAE score Tree: 23.2950
```

In the case of XGBoost Linear models, by comparing its parameters with the description of target variables, it can be seen that both intercepts are almost equal to the mean of the target variables. For this reason, it can be concluded that both linear models manage to capture the behavior of target variables.

```
# Long Term XGB Linear model parameters
modelLT1.coef_, modelLT1.intercept_
>>> (array([227.351, -41.0236, 233.643]), array([1471.69]))
yLT.mean()
>>> 1503.1095

# Short Term XGB Linear model parameters
modelST1.coef_, modelST1.intercept_
>>> (array([10.2884, 9.92311, -3.2221, -1.70612, -1.53666,
-2.62816, 7.04359]), array([53.0116]))
yST.mean()
>>> 57.7155
```

Finally, comparing the R2 score of the training sets, it can be seen that LTM has been fitted better than the STM. Meanwhile in the case of the test datasets, it can be seen that the STM is the worst model.

```
# R2 scores for Training Set
>>> LTM R2 score Linear: 0.8749
>>> STM R2 score Linear: 0.6927

# R2 scores for Test Set
>>> LTM R2 score Linear: 0.83017
>>> STM R2 score Linear: -0.0105
```

For this case, we handle the problem by trying a different set of delays. The best results were obtained with the set of variables shown in Table 4 and 5. It is necessary to mention that for this new dataset the number of examples was reduced to 34. Thus, for this case we made the partition of the dataset considering 10% for the test set in order to have 30 training examples.

TABLE IV. SHORT TERM ECONOMETRIC DATASET.

Years	DPBI	D4PBI	…	DEnergy
1985	1158.0540	4486.1683	…	5.3086
1986	10176.2344	-1098.3484	…	49.2252
1987	10126.6760	-10382.1691	…	100.8634
1988	-10573.0188	1628.0888	…	69.2792
1989	-15589.4258	1158.0540	…	-7.0995
1990	-2961.86224	10176.2344	…	-27.1133
:	:	:	:::	:
:	:	:	:::	:
2013	10803.3700	494.1570	…	109.6500
2014	7597.2530	15713.7040	…	71.8989
2015	6441.3940	14119.0340	…	38.6070
2016	5770.1990	10854.4450	…	61.7421
2017	4619.2950	10803.3700	…	25.2931
2018	10492.0122	7597.2530	…	42.3537

TABLE V. SHORT TERM VARIABLES.

Variable	Description
DPBI	Variation of PBI
D4PBI	Fourth-time delay Variation of PBI
DFARES	Variation of Fares
D2FARES	Second-time delay Variation of Fares
DCLIENT	Variation of Clients
D1ENERGY	One-time delay Variation of Energy Sales
D4ENERGY	Fourth-time delay Variation of Energy Sales
DENERGY	Variation of Energy Sales

With this new set of variables, the results below show the better performance of the STM in comparison with the first set of variables; however, the LTM remains to have the best performance.

```
# R2 scores for Training Set
>>> LTM R2 score Linear: 0.8749
>>> STM R2 score Linear: 0.4074

# R2 scores for Test Set
>>> LTM R2 score Linear: 0.83017
>>> STM R2 score Linear: 0.5787
```

The same analysis was realized for both long and short term XGBoost Decision Trees models. However, while for linear models we can analyze their behavior by their equation parameters, for XGBoost Decision Trees models we have a number of trees equal to 100 and a maximum depth of 3 as shown in Fig. 5 and 6.

Thus, as the number of trees per model is large, in order to obtained significant information from these latter models, the feature importance analysis was performed by counting the times a variable appears in a node of a Decision Tree. This latter was done using the plot_importance() function of XGBoost library.

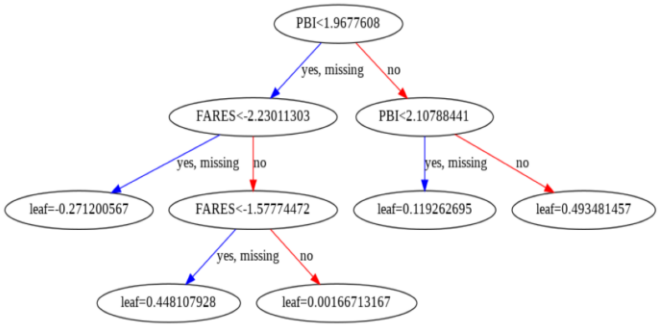

Fig. 5. Decision Tree #95 of LTM.

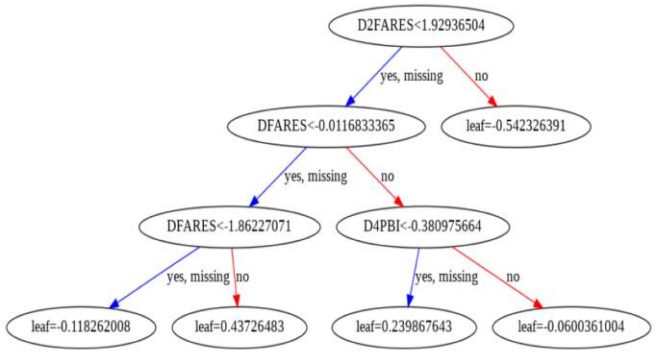

Fig. 6. Decision Tree #45 of STM.

In this context, Fig. 7 and 8 shows the feature importance for long and short term models respectively. It can be seen that in both cases the most important variables by far were the PBI and its variation followed by the number of clients and the variation of fares respectively.

On the other hand, analyzing the R2 score of these models, results bellow shows that both models have been fitted better to the training set than the linear models. In addition, in the case of the test set, the LTM have the best performance with a difference of 0.24 in the R2 score. From these last results, it can be concluded that XGBoost Decision Trees models are better than XGBoost Linear models as it can forecast unseen data with a better accuracy.

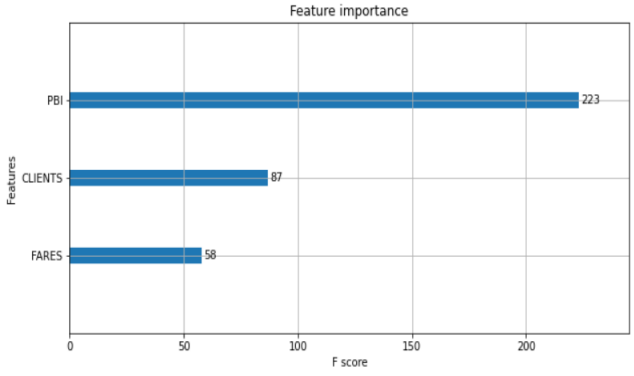

Fig. 7. Feature Importance plot of LTM.

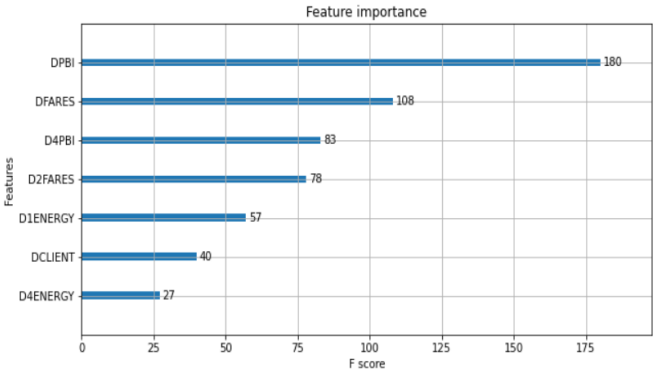

Fig. 8. Feature Importance plot of STM.

```
# R2 scores for Training Set
>>> LTM R2 score Linear: 0.9999
>>> STM R2 score Linear: 0.9998

# R2 scores for Test Set
>>> LTM R2 score Linear: 0.9774
>>> STM R2 score Linear: 0.7380
```

VI. Conclusions

Considering that the purpose of this paper was to introduce the use of XGBoost models in the energy demand forecast, results have demonstrated that this type of machine learning algorithm is a powerful technique not only due to being one of the fastest algorithms available today but also due to its robustness to predict the energy demand with a high accuracy. As XGBoost DT model is a new alternative to perform the predictions, it is necessary to consider that the XGBoost Linear model can be considered in the energy demand prediction as a simple parametric model which is the main disadvantage of the XGBoost DT model. However, XGBoost DT model can be interpretable through a Feature Importance analysis which is an important tool that cannot be used with other black box machine learning models.

In the case of the econometric variables, as this work considers the long and short term variables stablished by the regulatory entities, we concluded that the best set of variables was the long term variables due to its direct results and less complexity to calculate the energy demand for any year while in the case of the short term ones they require the calculation of previous years which makes this set of variables time dependent and the approximation error can be accumulated.

On the other hand, as the size of the dataset was not very large, the running time for learning algorithms was not considerable and thus not presented. Finally, a recommendation for future works in this field could be the introduction of time series with the econometric long term variables in the machine learning models.

References

[1] L. González, "Proyección de la Demanda de Energía Eléctrica a Corto Plazo, mediante Redes Neuronales Artificiales [Short-term Electric Power Demand Forecast through Artificial Neural Networks]," M.S. thesis, Faculty of Electricity and Computer Engineering, Littoral Polytechnic Higher School, Guayaquil, Ecuador, 2016. [Online]. Available: http://www.dspace.espol.edu.ec/xmlui/handle/123456789/38708

[2] I. Antonopoulos, V. Robu, B. Couraud, D. Kirli, S. Norbu, A. Kiprakis, D. Flynn, S. Elizondo-Gonzalez, and S. Wattam, "Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review," Renewable and Sustainable Energy Reviews, vol. 130, p. 109899, 2020.

[3] I. Crucianu, O. Bularca, and A.-M. Dumitrescu, "Modelling and forecasting of electrical consumption for demand response applications," 2019 IEEE Milan PowerTech, 2019.

[4] N. G. Paterakis, E. Mocanu, M. Gibescu, B. Stappers, and W. V. Alst, "Deep learning versus traditional machine learning methods for aggregated energy demand prediction," 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2017.

[5] A. Ioanes and R. Tirnovan, "Energy Demand Curve Modeling with Machine Learning Algorithms," 2019 8th International Conference on Modern Power Systems (MPS), 2019.

[6] *Procedimiento para fijación de Peajes y Compensaciones para SST y SCT [Procedure for setting Fees and Compensations of SST and SCT]*, Organismo Supervisor de la Inversión en Energía y Minería, 2020. [Online]. Available: https://www.osinergmin.gob.pe/seccion/institucional/regulacion-tarifaria/procesos-regulatorios/electricidad/fijacion-SST-SCT

[7] T. Chen and C. Guestrin, "XGBoost," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[8] *XGBoost – Machine Learning winning solutions (incomplete list).* GitHub, Aug. 2013. [Online]. Available: https://github.com/dmlc/xgboost/tree/master/demo#machine-learning-challenge-winning-solutions.

[9] "Preprocessing data," Scikit-learn. [Online]. Available: https://scikit-learn.org/stable/modules/preprocessing.html.

[10] "Metrics: R2 score." Scikit-learn. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html.