Machine Learning based Electric Load Forecasting for Short and Long-term Period

Tomáš Vantuch[a], Aurora González Vidal[b], Alfonso P. Ramallo-González[b], Antonio F. Skarmeta[b], Stanislav Mišák[a]
[a]*Centre ENET at VŠB - Technical University of Ostrava, Czech Republic*
[b]*Computer Science Faculty, University of Murcia, Spain*
tomas.vantuch@vsb.cz, aurora.gonzalez2@um.es, alfonsop.ramallo@um.es, skarmeta@um.es, stanislav.misak@vsb.cz

***Abstract*—Electricity is currently the most important energy vector in the domestic sector and industry. Unlike fuels, electricity is hard and expensive to store. This creates the need of precise coupling between generation and demand. In addition, the transmission lines of electric power need to be sized for a given maximum power, and overloading them may result in blackout or electrical accidents. For these reasons, energy consumption forecasting is vital.**

The time scale for forecasting depends on who is interested in such prediction. Grid operators have to predict the electricity demand for the next day, to program the generation accordingly. Grid designers have to predict energy consumption at the scale of years, to ensure that the infrastructure is sufficient. On the other hand, smart grid controllers with almost instant response time may need a prediction on the order of minutes.

We have seen that changing the time scale in electricity load forecasting changes the approach, and that depending on the scale different methods should be used to ensure the highest accuracy with the smallest computational cost. We show here how forecasting accuracy decreases with the increase of time scale due to the impossibility of using of all variables. Several well established computational models were compared on three different regression based criteria and the results revealed that boosting model was able to outperform their competitors in most of the comparisons.

***Index Terms*—Electric load forecasting, artificial intelligence, machine learning, multi-objective optimization**

I. Introduction

Electric load forecasting as a supportive tool in controlling mechanisms is vital and has been described in many research projects. As one of its applications the Active Demand Side Management (ADSM) proposed by Misak [1] counts with this ability to support the maximal efficiency of energy use gathered from renewable energy sources. The increase of energy efficiency is one of the motivations also in the project ENTROPY, an European international project with the aim of quantifying energy reduction potential of behavioral change encouraged by ICT. There are several common features between ADSM and ENTROPY projects, both of them rely on the network of intelligent sensors, as an ecosystem within the internet of things (IoT) [2] and intelligent data processing with application of machine learning and artificial intelligence techniques. The ADSM deals with energy management on small scale of the single household or several residences at most, which differs from the ENTROPY. Its scale is much larger due to its focus and current deployment in several buildings of the University at Murcia.

The topic of electric load forecasting as a field of data science is well established, which is already proven by several successful competitions [3] and high quality review studies [4], [5]. The majority of the forecasts are based on previous electric load patterns and current weather conditions since they are considered as the most relevant variables affecting the buildings heat loses and therefore the conditioning demand that accounts for around 40% of the total energy demand (depending on the climate).

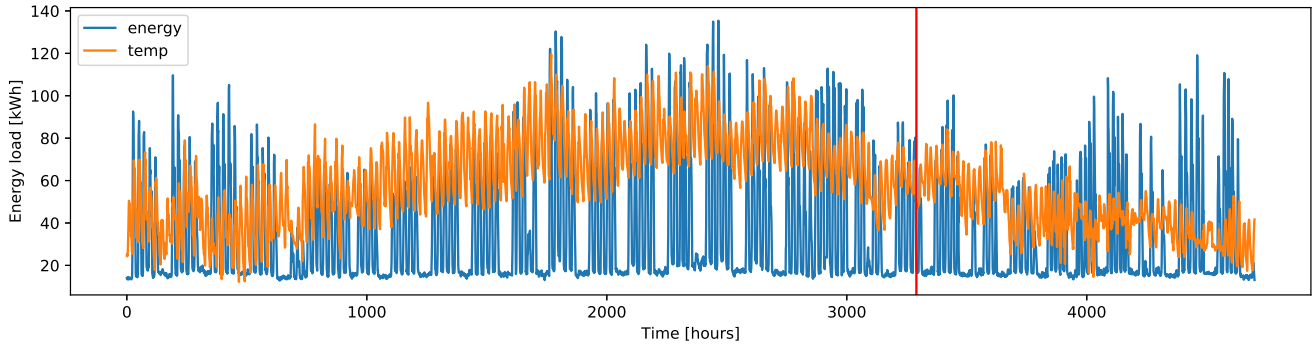
The motivation of this paper is to design and evaluate an electric load forecasting model making use the historical data of the forecast variable and the outside temperature. The adjusted time scales are a short-term one hour and a long-term one week ahead forecasting. These scales differ in extracted features as well as in algorithms adjustment. The energy load forecasting will serve for calculation of an expected energy consumption for a given week. In this purpose we will present a comparative metric as a summation of residuals divided by the total energy consumption in the observed week (cumulative error per week - CEPW). With the coefficient of variation of Root-mean-square error (CV-RMSD) and Mean absolute percentage error (MAPE) it will complete the tree criteria applied for the evaluations of forecasting models.

The sections of this paper are as follows. Section II describes the data and its processing with feature extraction. The algorithm are described in section III and their adjustments and results follows in section IV. Section V summarizes the findings of the work.

This paper has been also possible partially by the European Commission through the 516 H2020-ENTROPY-649849, the Spanish National Project CICYT EDISON (TIN2014-52099-R) granted by the 517 Ministry of Economy and Competitiveness of Spain (including ERDF support), the project TUCENET Sustainable Development of Centre ENET LO1404, the Project LTI17023 "Energy Research and Development Information Centre of the Czech Republic" and project CZ.1.05/2.1.00/19.0389.

Ramallo-González would like to thank the program Saavedra Fajardo (20035/SF/16) funded by Consejería de Educación y Universidades of CARM, via Fundación Séneca-Agencia de Ciencia y Tecnología de la Región de Murcia.

Fig. 1. Visualization of both time series applied in this experiment. The time set of temperature is multiplied by 3 to fit the scale of energy load. The red vertical line represents the split of dataset into training and testing subsets.



II. Experiment Design

The experiment aims to analyze and forecast an amount of electric load based on two input time series. The outside air temperature and the past values of electric load. The data are collected at the University of Murcia in Spain from February 18th to the December 30th within a 15 minutes of resolution. The goal of this experiment is to create a forecasting model for one or several hour ahead, therefore the data were averaged into one hour resolution and all further extracted features as well as the normalization was proceeded on these aggregated data.

To gain the maximal information value from the given time series, we need to employ a feature extraction work-flow to obtain a matrix of input features. These features forming the sample vectors will serve as input vector for further applied machine learning models.

A. Feature extraction

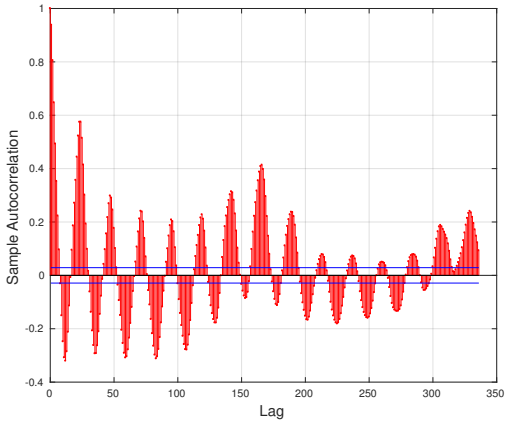
Several features were extracted from the given time series with respect to their estimated relevancy or the fundamental knowledge of the observed system. Those features were formed into vectors of input variables and they are described in following. The relevancy of features was estimated by calculation of Pearson correlation coefficient (correlation) [6] and Maximal Information coefficient (MIC) [7].

The first very important set of variables is based on the measured time. The information related to time is necessary due to the presence of pattern repetition in the data. Similarities of load patterns rather occur among working days than between any working day compared to the weekend when the electric load possess a different behavior. Based on these facts, the first variables were as follows

- month
- day
- hour
- weekday - the day of week [0 - monday,... 6 - sunday]
- weekend - [0 - working day, 1 - weekend] *the holiday during the working day was classified as weekend

The strong seasonality in electric load time set (see Fig. 1) and its autocorrelation (see Fig. 2) proves the necessity of

Fig. 2. Autocorrelation of energy consumption time set with number of lags equals to hours of two weeks.



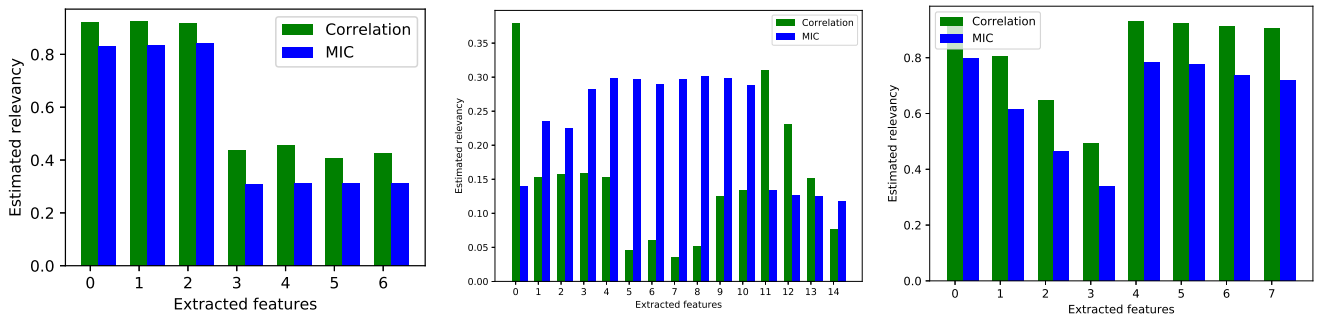
variables derived from this lagged-relevancy and they are as it follows:

1) energy load of the same hour and the same day but previous week
2) the average load from same kind of days (working day / weekend) and same hour but previous week
3) the average load from same day and hour but 4 previous weeks
4) maximal load of working days / weekends during previous week
5) average load of working days / weekends during previous week
6) maximal load of working days / weekends two weeks ago
7) average load of working days / weekends two weeks ago

The Fig. 3 depicts estimated relevancies for enumerated features above (each index from list matches the same from bar chart).

The second group of features was extracted from the time set of outside temperature ($temp$). It was supposed to reflect the past trends of this variable due to its impact on the inside building's temperature. The heating and cooling system of the building is one of the most significant loads of the

Fig. 3. Estimated relevancies (Maximal Information Coefficient and Pearson correlation coefficient) for features extracted from energy load (left), outside temperature (middle) and AEA and lagged values of energy load (right) time set.



predicted variable. The temperature inside the building is not changing instantaneously due to the character of the building itself, therefore it is necessary to pay attention on the outside temperature for several days in the past.

The features extracted from outside temperature consist of maximal and minimal temperature of current and previous three days as well as the current and past four hours. We were able to use the temperature of the predicted week because our final goal was to forecast the expected electric load from the current week. The Fig. 3 depicts the estimated relevancies for temperature related features ((0) - current $temp$, (1-4) - min/max $temp$ of today, (5-10) - min/max $temp$ of previous 1-3 days, (11-14) - past 4 values of $temp$). All of those features were kept for further application.

The additional variables extracted by an approach entitled as Analog Ensemble Application (AEA) completes the list of variables listed above. AEA is a forecasting model originally applied for meteorological ensemble forecasts [8] but since than, the other areas of study could benefit from its use [9], [10]. The default version of the algorithm consists of three major phases which are worthy to mention, because only some of them were applied in our modeling.

The first phase simply searches through the samples of the training set and selects the most similar observations based on the given similarity metric. Such metric depends on the kind of the features, the euclidean distance is applied the most frequently. In our case the similarity was calculated as an average between euclidean distance applied on continuous data and the dice coefficient [11] was computed in case of categorical data.

The second phase applies a filtering mechanism to set up the right number of analogs (most similar data samples) for the third phase. In some cases, the relevancy of past predicted variable against current one is compared. As the similarity of the feature vector decrease, the relevance of its past predictor decreases as well. Based on a given statistic or adjusted threshold value, the final number of analogs is taken. In our case, the number of analogs was experimentally adjusted to 4. This choice is reasoned by fact that these kind of features are highly inter-correlated, therefore it brings redundancy into dataset and the total performance was not increased by any higher number of them.

The last phase covers the process of the computing of the prediction based on a given ensemble of analogs. This can be driven by the application of some linear or evolutionary based technique. In our case, these analogs were used as another features of the input vector, because of their high relevancy.

In case of one-hour ahead forecasting, the four additional features were added which simply represents the lagged value of energy consumption (lag from x_{t-1} to x_{t-4}).

The Fig. 3 shows the estimated relevancies for lagged values of energy load (indexes 0 - 3 for laggs 1 - 4) and analogs found by AEA (indexes 4 - 7). The found analogs posses the highest relevancy, which underline their necessity for use.

III. Applied Algorithms

We chose four different ML algorithms to compare their performance based on the given metrics. All of them are widely known algorithms with applications in many previous studies and they are listed below

- Artificial Neural Network (ANN) [12]
- Support Vector Regression (SVR) [13]
- Random Forest Regression (RFR) [14]
- eXtreme Gradient Boosting (XGB) [15]
- Flexible Neural Tree (FNT) [16]

IV. Adjustments and Results

In order to weight all features equally, it is necessary to standardize the input data. We decide to apply normalization, so all input features will have zero mean and standard deviation equal to one. Than the entire dataset is divided into two major parts and it is the training and testing subset. The ratio 70 - 30 was applied in this case.

The training process involves the tuning of model's hyper-parameters and for this purpose the grid-search optimization is applied [17]. It simply iterates through all given combinations of parameters from defined ranges and tests their performance on an iterative cross-validation process. The ranges of hyper-parameters are listed in Table I.

ANN involves in its optimization the number of neurons, in our case it was for only one hidden layer (hidden layer sizes), because more layers were not improving its performance (in our experiment). Four options were given for the parameter of neuron's activation function (activation) and

TABLE I
RANGES AND OPTIONS FOR HYPER-PARAMETERS ADJUSTMENT WHICH IS DRIVEN BY GRID-SEARCH OPTIMIZATION.

Alg.	Parameter	Values
ANN	hidden layer sizes	20, 30, 40, 50, 60
	activation	identity, logistic, tanh, relu
	learning rate	constant, invscaling, adaptive
	solver	lbfgs, sgd, adam
SVR	kernel	linear, radial basis, polynomial
	C	1, 5, 10
	gamma rate	0.001,...,0.01
RF	number of trees	50,70,100,150,200
	max depth	[1..5]
	max features	[3..20]
	min samples split	[3..20]
	min samples leaf	[3..20]
XGB	number of trees	50,70,100,150,200
	max depth	[2..5]
	learning rate	[0.15...0.55]
	reg. type	linear
FNT	number of individuals	60, 120, 180
	number of iterations	[5000]
	MOO algorithm	nsga2
	optimization functions	CV-RMDS and CEPW
	size of individual	500
	crossover rate	0.8
	mutation rate	0.15

they are the identity function ($f(x) = x$), logistic function ($f(x) = 1/(1 + e^{-x})$), the hyperbolic tan function (tanh) ($f(x) = \tanh(x)$) and rectified linear unit function (relu) ($f(x) = max(0, x)$). The parameter entitled as solver drives the approach of weight optimization (lbfgs, sgd, adam). The variable progress of learning rate varied in three options, but it was applied only in case of stochastic gradient descent (sgd) weight optimization.

Three different kernel functions were an option for optimization in the case of SVR as well as penalization constant C and gamma rate is the parameter of kernel function (not applied in case of linear kernel function).

RFR parameters are related to adjustment of its trees. The maximal depth controls the vertical size of the tree, minimal samples per leaf and minimal samples split attributes controls over-fitting of a tree and maximal features stands for number of features from bagged subset that are applied of the trained tree. The number of trees stands for the total size of the ensemble.

Some of the parameters of XGB are similar to RFR because of the algorithm features that are shared among them. The only adjusted parameters were the maximal depth of trees and learning rate, which is a parameter of shrinking of new feature weights (each iteration produces a tree based estimation trained on ensemble's residuals and this process makes the learning slower - more conservative). The linear regression function that serves to compute the final value based on all boosted trees, was chosen experimentally as a function with highest performance.

The parameters of FNT were set by our best practice with respect to other previous studies. The algorithm required longer learning time due to its stochastic nature and also the selection of best predictor was driven differently comparing to the previous models. During each iteration, NSGA2 algorithm ensured the comparison of each synthesized solution with all dominant solutions from the Pareto-front set. The final solution was chosen from Pareto optimal solutions by fuzzy decision making process which takes into account all three involved performance criteria equally. The involved criteria as it was mentioned before are CV-RMSD, MAPE, CEPW and their are defined as follows

$$\text{CV-RMSD} = \frac{\sqrt{\sum_{i=0}^{n}(y_t - y_p)^2/n}}{\overline{y_t}} \tag{1}$$

$$\text{MAPE} = \frac{100}{n}\sum_{i=0}^{n}\left|\frac{y_t - y_p}{y_t}\right| \tag{2}$$

$$\text{CEPW} = \sum_{i=0}^{w}\left(\frac{\sum_{j=0}^{m}\left|y_t - y_p\right|}{\sum_{j=0}^{m} y_t}\right)/w \tag{3}$$

where y_t and y_p means target and predicted value of energy load, n is the number of samples in testing dataset, w is the number of weeks and m stands for the number of samples in examined week. The same options for hyper-parameters were applied in both cases of forecasting (one hour ahead and one week ahead). The models were optimized towards the lowest mean square error and three selected evaluative criteria were calculated afterwards. It was the CV-RMSD, MAPE and the summation of residuals divided by the total energy consumption in that week (CEPW). The difference between the predicted time horizons relies on application of lagged values in case of one hour ahead forecasting as it was mentioned before.

The series of optimized hyper-parameters as well as results of forecasting accuracy are listed in Tables II, III and IV. As we can see, the algorithms based on ensemble of regression models performed better in most of the evaluative criteria.

V. DISCUSSION AND CONCLUSIONS

The experiment presented in this paper deals with application of ML algorithms in the task of energy load forecasting. The applied data represents the time series of two observed events (electric energy consumption and outside temperature) measured at the selected department building at the university of Murcia. These time series served as a source data for feature extraction in order to obtain relevant input vectors for applied ML models. All of the extracted features were based on the statistical relevance (presence of repetitive patterns known as auto-correlation) and applied samples similarity estimation (AEA).

Although, its application brought several highly relevant features they did not cover the necessity of use of electric load lagged values in order to improve the forecasting accuracy. Very similar scenario has been observed in almost all available studies [18]. Comparing to other studies [18]–[20],

Fig. 4. Visualization of forecasting accuracy gained by RFR on 1 hour ahead time scale on the entire training dataset.

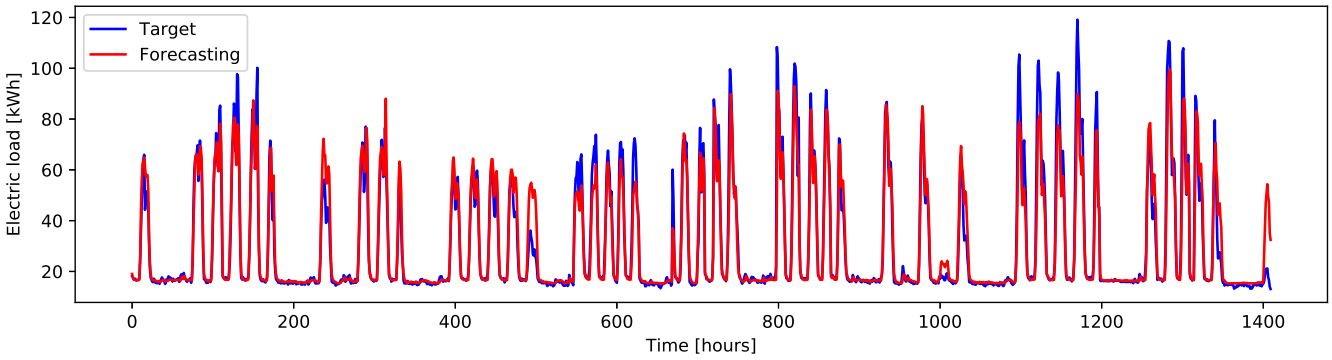


TABLE II
1 HOUR AHEAD FORECASTING ACCURACY WITH HYPER-PARAMETERS OF BEST MODELS FROM CROSS-VALIDATION PROCESS.

Model with optimized hyper-parameters	max CV-RMSD	avg. CV-RMSD	max CEPW	avg. CEPW
SVR 'kernel': 'linear', 'C': 10	23.57	20.19	13.82	12.14
ANN 'activation':'relu', 'solver':'lbfgs', 'hidden_neurons':30	35.86	26.87	22.92	16.98
RFR 'features': 18, 'trees': 250, 'bootstrap': True, 'depth': 5	34.67	21.80	16.85	11.40
XGB max_depth: 4, learning_rate: 0.25, n_trees: 100	24.39	18.38	11.69	9.81
FNT individuals: 60, iterations: 5000, crossover r.:0.8, mutation r.: 0.2	34.91	23.77	20.16	15.44

TABLE III
1 WEEK AHEAD FORECASTING ACCURACY WITH HYPER-PARAMETERS OF BEST MODELS FROM CROSS-VALIDATION PROCESS.

Model with optimized hyper-parameters	max CV-RMSD	avg. CV-RMSD	max CEPW	avg. CEPW
SVR 'kernel': 'linear', 'C': 1	36.18	28.16	28.30	18.92
ANN 'activation':'relu', 'solver':'lbfgs', 'hidden_neurons':40	42.76	33.79	29.12	19.19
RFR 'features': 9, 'trees': 300, 'bootstrap': True, 'depth': 5	41.60	27.28	21.39	13.78
XGB max_depth: 5, learning_rate: 0.2, n_trees: 100	32.05	31.57	54.61	15.23
FNT individuals: 180, iterations: 5000, crossover r.:0.8, mutation r.: 0.2	40.72	32.16	27.65	18.54

TABLE IV
1 WEEK AHEAD FORECASTING ACCURACY WITH HYPER-PARAMETERS OF BEST MODELS FROM CROSS-VALIDATION PROCESS.

Model	hour ahead		week ahead	
	max MAPE	total MAPE	max MAPE	total MAPE
SVR	16.66	11.88	30.85	19.45
ANN	23.26	15.61	22.69	15.30
RFR	16.41	9.19	20.43	11.02
XGB	11.84	8.34	17.35	12.33
FNT	24.17	14.89	26.38	18.07

significantly lower forecasting performance was achieved in our study due to several possible factors. Our original data may contain higher level of stochasticity which is always considered as a characteristic difference between residential, commercial and industrial load profiles and on the other hand, we focused on the application of the basic representatives of the machine learning approaches in order to obtain some referential results for our future studies, instead of testing of their advanced modifications which will be address in our future work. It will also focus on an advanced time series processing to obtain relevant features in order to increase its predictability.

It is necessary to pay attention also on the computational complexity of the applied models, not only during their prediction phase but also during their learning phase. It is because, their real deployment into the complex control system may require the repetitive re-learning on the growing database. It is easy to decide based on the already published analysis which algorithm is computationally the most expensive. Based on the applied adjustments and the learning algorithm, we may rank the applied algorithms into three groups. The ensemble based classifiers like RFR and XGB posses the lowest computational complexity of their training phase, the complexity of ANN and

SVR varies based on the given adjustment but it still can be considered as reasonable for this task. In case of FNT, which is evolutionary optimized ensemble of genetically programmed trees, the computational requirements are definitely the highest among the tested algorithms. Also with rather average predictive performance, the FNT does not appear as a winning candidate from our testing.

Although, nowadays applications of such unconventional computation models based on swarm intelligence, evolutionary optimization and fuzzy logic are gaining researchers attention due to their potential. In our case, the FNT trained by MOO was able to compete with the well established models like SVR and ANN. FNT in its complexity during prediction phase is very similar to ANN, the difference is in the number and way of connections between neurons as well as the applied activation function. For successful deployment of the final model, the forecasting accuracy has to be increased while its simplicity has to be kept or ideally reduced.

REFERENCES

[1] S. Mišák, J. Stuchlỳ, J. Platoš, and P. Krömer, "A heuristic approach to active demand side management in off-grid systems operated in a smart-grid environment," *Energy and buildings*, vol. 96, pp. 272–284, 2015.

[2] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of things," *International Journal of Communication Systems*, vol. 25, no. 9, p. 1101, 2012.

[3] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, "Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond," 2016.

[4] L. Suganthi and A. A. Samuel, "Energy models for demand forecastinga review," *Renewable and sustainable energy reviews*, vol. 16, no. 2, pp. 1223–1240, 2012.

[5] A. Ahmad, M. Hassan, M. Abdullah, H. Rahman, F. Hussin, H. Abdullah, and R. Saidur, "A review on applications of ann and svm for building electrical energy consumption forecasting," *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 102–109, 2014.

[6] S. M. Stigler, "Francis galton's account of the invention of correlation," *Statistical Science*, pp. 73–79, 1989.

[7] J. B. Kinney and G. S. Atwal, "Equitability, mutual information, and the maximal information coefficient," *Proceedings of the National Academy of Sciences*, vol. 111, no. 9, pp. 3354–3359, 2014.

[8] L. Delle Monache, F. A. Eckel, D. L. Rife, B. Nagarajan, and K. Searight, "Probabilistic weather prediction with an analog ensemble," *Monthly Weather Review*, vol. 141, no. 10, pp. 3498–3516, 2013.

[9] S. Alessandrini, L. Delle Monache, S. Sperati, and J. Nissen, "A novel application of an analog ensemble for short-term wind power forecasting," *Renewable Energy*, vol. 76, pp. 768–781, 2015.

[10] E. Vanvyve, L. Delle Monache, A. J. Monaghan, and J. O. Pinto, "Wind resource estimates with an analog ensemble approach," *Renewable Energy*, vol. 74, pp. 761–773, 2015.

[11] L. Rokach and O. Maimon, "Clustering methods," in *Data mining and knowledge discovery handbook*. Springer, 2005, pp. 321–352.

[12] T. Hastie, *Neural network*. Wiley Online Library, 1998.

[13] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in neural information processing systems*, 1997, pp. 155–161.

[14] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[15] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[16] Y. Chen, B. Yang, J. Dong, and A. Abraham, "Time-series forecasting using flexible neural tree model," *Information sciences*, vol. 174, no. 3, pp. 219–235, 2005.

[17] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing Systems*, 2011, pp. 2546–2554.

[18] G. M. U. Din and A. K. Marnerides, "Short term power load forecasting using deep neural networks," in *Computing, Networking and Communications (ICNC), 2017 International Conference on*. IEEE, 2017, pp. 594–598.

[19] W.-C. Hong, "Electric load forecasting by support vector model," *Applied Mathematical Modelling*, vol. 33, no. 5, pp. 2444–2454, 2009.

[20] F. M. Bianchi, E. De Santis, A. Rizzi, and A. Sadeghian, "Short-term electric load forecasting using echo state networks and pca decomposition," *Ieee Access*, vol. 3, pp. 1931–1943, 2015.