A Comparison of Automated Machine Learning tools for Predicting Energy Building Consumption in Smart Cities

Daniela Soares[1], Pedro José Pereira[1,2], Paulo Cortez[2], and Carlos Gonçalves[3]

[1] EPMQ - IT Engineering Maturity and Quality Lab, CCG ZGDV Institute,
GuimarÃčes, Portugal
{daniela.soares,pedro.pereira}@ccg.pt
[2] ALGORITMI Center/LASI, Dep. Information Systems, University of Minho,
pcortez@dsi.uminho.pt
[3] ISEP, Polytechnic of Porto, rua Dr. António Bernardino de Almeida, 4249-015
Porto, Portugal
cag@isep.ipp.pt

Abstract. In this paper, we explore and compare three recently proposed Automated Machine Learning (AutoML) tools (AutoGluon, H2O, Oracle AutoMLx) to create a single regression model that is capable of predicting smart city energy building consumption values. Using a recently collected one year hourly energy consumption dataset, related with 29 buildings from a Portuguese city, we perform several Machine Learning (ML) computational experiments, assuming two sets of input features (with and without lagged data) and a realistic rolling window evaluation. Furthermore, the obtained results are compared with a univariate Time Series Forecasting (TSF) approach, based on the automated FEDOT tool, which requires generating a predictive model for each building. Overall, competitive results, in terms of both predictive and computational effort performances, were obtained by the input lagged AutoGluon single regression modeling approach.

Keywords: Automated Machine Learning · Smart Cities · Regression.

1 Introduction

Due to advances in Information Technology (IT) and Artificial Intelligence (AI), nowadays it is easy to collect, store and process data that reflect relevant phenomena within the context of smart cities [13]. In particular, the efficient and environmentally responsible use of energy resources has become an important concern of smart cities decision makers, which aim to create ecological and sustainable environments for its citizens. Following this need, several works have been proposed regarding the usage of ML to predict energy consumption and demand, aiming to improve energy efficiency and sustainability [15,16,11,14,20,7].

In this paper, as a real-world demonstration use case, we address the prediction energy consumption of several buildings from a Portuguese city. Following a typical smart city context, energy consumption data is collected on a regular basis with an associated timestamp, thus its prediction can be addressed as a univariate Time Series Forecasting (TSF) task [12]. However, this TSF approach implies modeling each building separately, which can lead to a vast amount of forecasting models (one for each building), increasing the difficulty of ML model maintenance and monitoring. A different approach is to use a single regression model capable of predicting the energy consumption of any building, increasing the learning task difficulty but simplifying the ML deployment phase, since only one model is maintained.

Regarding the related works, the usage of a single regression building energy prediction model approach is a recent trend (e.g., [14,17,7]). Nevertheless, the majority of these related works do not adopt an Automated ML (AutoML) model selection and tuning. AutoML is particularly valuable for smart cities, allowing non-experts to more easily create and maintain ML predictive models [13]. In effect, the related works typically adopt a manual tuning of ML algorithms, performing some trial-and-error comparison performances on a particular dataset. Within our knowledge, there are only two studies that have employed AutoML tools for building energy consumption prediction, addressing this goal as a regression [19] or TSF [12] tasks. Yet, in these two studies, the authors have modeled each building separately, resulting in one ML model for each building. In contrast, in this work we target a single energy consumption prediction ML model for all buildings. In particular, we empirically compare 3 distinct and recent AutoML tools (AutoGluon, H2O, Oracle AutoMLx), using two sets of input features. The comparison assumes both predictive and computational effort performances measures, under a realistic and robust rolling window scheme [18]. Moreover, we also compare the single AutoML regression approach with an individual building ML modeling, obtained by using the FEDOT Automated TSF (AutoTSF) tool [12].

The remainder of the paper is organized as follows. Section 2 presents the related work. Then, Section 3 describes the datasets used, the AutoML and AutoTSF tools and the adopted evaluation procedure. Next, Section 4 presents and discusses the obtained empirical results. Finally, Section 5 presents the main conclusions and future work directions.

2 Related Work

Most related studies address the energy consumption prediction using one of the following approaches: 1) build a ML model to each building for which they want to predict future consumption [12,20,19]; or 2) model all public building using a single, and consequently more complex, ML model [3,15,4,14,17,7]. In most cases, the former approach involves having dozens or hundreds of models, increasing the complexity of ML deployment and monitoring tasks. Nevertheless, modeling the pattern from a specific building can simplify the training process and potentially lead to better predictions. On the other hand, the latter single model approach simplifies the deployment phase. For the multiple learning models approach, the previous works assumed an univariate TSF [12] or a regression task [20,19]. Regarding the single model approach, most works address assume a regression task (e.g., [17,7]).

In terms of the modeling phase, most of related works perform the ML model selection process manually, by implementing and comparing several ML algorithms [20,15]. Given the extensive set of models used in previous studies, in [16], the authors proposed a framework for selecting the most appropriate ML model for energy consumption prediction, based on multiple factors (e.g., size of dataset, number of attributes). Regarding the ML algorithms, the relevant recent studies that assumed regression tasks used: Artificial Neural Networks (ANN) [17,20,15,1,3], including Long Short-Terms Memory (LSTM) ANN [7]; Decision Trees (DT) [7,3]; Support Vector Machines (SVM) [7,17,20,15,3]; K-Nearest Neighbours (kNN) [17,3]; Linear Regression (LR) [20,15]; Ridge [3]; and Ensemble methods [3], including Random Forest (RF) [20,14], eXtreme Gradient Boosting (XGBoost) and Adaptative Boosting (Adaboost) [15]. It should be noted that all these research studies performed a ML algorithm selection without tuning its hyperparameters, thus no hyperparameter selection was executed.

In a different approach, in [12,19] the authors applied AutoML algorithms for data preprocessing, model selection and tuning. AutoML compares several algorithms with different hyperparameters values, returning the configuration with the best predictive performance. Recent studies have shown the value of these algorithms in the smart cities domain [13]. In particular, Wang et. al [19] apply two AutoML algorithms, namely auto-sklearn and TPOT, for electric load forecasting, comparing them with LR, RF, SVM, XGBoost and Gradient Boost Machines (GBM) using two different datasets. The data was modeled as a regression task and the TPOT algorithm presented the best overall results. In another study [12], the authors tested several Automated TSF (AutoTSF) algorithms under 9 time-series smart cities open datasets, 3 of which were related to energy consumption. The best overall results were achieved by FEDOT framework. Yet, both these studies using automated approaches modeled each building separately.

In this paper, we address the building energy consumption prediction task by using automated modeling tools, implementing and comparing two different approaches: 1) modeling all buildings with a single regression model, comparing 3 different and popular AutoML tools: AutoGluon, H2O and Oracle AutoMLx; and 2) modeling each building separately, using the AutoTSF FEDOT tool that provided the best overall results in [12]. Within our knowledge, this is the first study comparing these different approaches based on automated tools for model selection and tuning. To evaluate our models, we used a real-world dataset of energy consumption from 29 buildings from a Portuguese city. Furthermore, we employ a robust Rolling Window evaluation procedure [18] with statistical tests to validate our results.

3 Materials and Methods

3.1 Data

In this work, we explore energy consumption data that was provided by a Portuguese company. Due to commercial privacy concerns, several data details (e.g., geographic location, specific time stamps or individual energy consumption values) are not disclosed. The data were recently collected, corresponding to around one year of records. In total, we had access to 29 distinct energy consumption files, each including the consumption values from a public building that were recorded every 15 minutes during the collected time period. The data records contained several attributes, including the building unique delivery point identifier, the consumption time and date and the average energy consumed in kW. It is worth noting that the data correspond to the energy consumption of special low voltage connection, that is, buildings with contracted power equal to or less than 45 kVA, which are associated with small buildings or residential customers. Additionally, a supplementary dataset was provided, containing specific information about the buildings, namely the address, typology (e.g., school, administrative, cultural, parking lot, civil protection), useful area in m^2, year of construction and the delivery point identifier.

In terms of data preprocessing, the 29 energy consumption files were firstly integrated into a single Comma-Separated Values (CSV) file. Then, we merged the additional building attributes (provided in the supplementary dataset) into the same CSV file. Next, following a feedback that was obtained from the private company experts, the 15 minute values were aggregated into an hourly scale, by summing four consecutive consumption values for each building. Then, aiming to improve the predictive performance of the ML models, we generated three new lagged inputs, related to energy consumption values for the same building, as recorded in three distinct time periods: the previous hour; the previous day at the same hour; and the previous week at the same hour. The rationale of the lagged inputs is to allow an autoregressive ML modeling for the regression approach, as typically performed by univariate TSF models [12]. To verify the impact of the lagged attributes, we designed two input feature scenarios, described on Table 1. Scenario A includes only features related to the time and hour of the targeted energy consumption and contextual building information, while scenario B complements these inputs with the three additional lagged energy consumption values. It should be noted that Scenario B does not require a large storage of historical data to perform predictions on new buildings (only one week of data is needed).

3.2 AutoML Methods

As previously explained, in this work we explore two main ML approaches to predict energy consumption data: **regression**, assuming a single prediction model that is capable of predicting the hourly energy consumption for all buildings; and pure univariate **TSF**, generating an individual prediction model for each

Table 1. List of analyzed data attributes.

Context	Name	Description	Scenarios
Time	Hour Weekday	Consumption time. Day of the week of consumption.	A,B
Building	Typology Useful area	Building type, one of the 6 levels {administrative, cultural, school, parking lot, civil protection, others}). Building floor area (in m^2).	A,B
Energy	Previous hour Previous day Previous weekday	Consumption in the previous hour. Consumption in the previous day and same hour. Consumption in the previous weekday and same hour.	B
Target	Consumption	Energy consumption(in kW).	–

building (total of 29 models). For both approaches, we selected automated tools for the model selection and hyperparameter tuning.

The tool selection for the regression approach, is based on our previous work, published in 2023 [13], in which we developed an AutoML platform for smart cities (AI4CITY) that automatically creates full ML pipelines, choosing data preprocessing steps and AutoML tools based on the data characteristics and the ML task. The AI4CITY plaform makes use of H2O and AutoGluon AutoML search engines. In [13], it was shown that the automatic AI4CITY ML pipelines produced better results for 15 of the 26 analyzed smart city datasets when compared with the direct usage of the H2O and AutoGluon tools. Following this result, in this paper we adopt the AI4CITY tool under two variants, where the AutoML is executed using the H2O or AutoGluon algorithms. Additionally, by requirement of the private company that provided the data, we also explore the recently proposed Oracle AutoMLx tool [21] for the regression ML approach and that works under a computational cloud infrastructure. Regarding the univariate TSF approach, a recent study compared 8 distinct AutoTSF open-source tools under 9 smart city time series, 3 of which were related to energy consumption [12]. The FEDOT tool presented the best overall results and, therefore, we selected it for our TSF experimentation.

In terms of ML models tested by each AutoML tool, we used the default settings (in order to achieve an unbiased comparison). When available in the tool documentation, we detail the searched ML algorithms and the number of hyperparameters ($\mathcal{H}$) tuned by the AutoML:

- **H2O** – Generalized Linear Model (GLM) ($\mathcal{H} = 1$), RF ($\mathcal{H} = 0$), Extremely Randomized Trees (XRT) ($\mathcal{H} = 0$), GBM ($\mathcal{H} = 8$), XGBoost ($\mathcal{H} = 9$), Deep Learning Neural Network (DLNN) ($\mathcal{H} = 7$) and two Stacked Ensembles (all

– uses all base learners; and best – uses the best model per searched ML algorithm, total of 6 individual models). The Stacked Ensembles adopt a second-level GLM learner that weights the individual base learner predictions.

- **AutoGluon** – assumes the tabular prediction feature, which first executes several ML algorithms under a default hyperparameter grid search: GBM ($\mathcal{H} = 2$), CatBoost Boosted Trees ($\mathcal{H} = 0$), RF ($\mathcal{H} = 1$), Extra Trees ($\mathcal{H} = 1$), kNN ($\mathcal{H} = 1$), and a DLNN ($\mathcal{H} = 0$). Then, a Stacked Ensemble is created and returned as the global prediction model. The Stacked Ensemble works as special DLNN, in which the individual ML models are stacked in multiple layers and trained in a layer-wise manner, such as detailed in [6].
- **Oracle AutoML**x – AdaBoost ($\mathcal{H} = 2$), DT ($\mathcal{H} = 4$), Extra Trees ($\mathcal{H} = 6$), ANN, kNN ($\mathcal{H} = 2$), Light GBM ($\mathcal{H} = 9$), Linear SVM ($\mathcal{H} = 2$), LR ($\mathcal{H} = 3$), RF ($\mathcal{H} = 4$), SVM ($\mathcal{H} = 3$), and XGBoost ($\mathcal{H} = 8$).

As for the automatic preprocessing, both AI4CITY and Oracle AutoMLx use standard z-score scaling for numeric features and one-hot encoding for the categorical attributes (generation of binary value for each categorical level). However, the Oracle tool uses a maximum limit of 5 unique levels, after which a label encoding is applied, instead of one hot encoding. On the other hand, AI4CITY sets this limit as 50, after which the Inverse Document Frequency (IDF) transformation is applied. In the analyzed building energy consumption dataset, the typology attribute is the only categorical feature, with 6 unique values. Consequently, AI4CITY applies one hot encoding transformation to this column, while Oracle AutoML applies a label encoding. Lastly, for the univariate TSF approach, no preprocessing is applied to the dataset by the AI4CITY tool, thus it directly feeds the time series data into the FEDOT AutoTSF tool. Furthermore, to ensure a fair comparison between these tools, we used all AutoML search default values, except for two aspects. Firstly, the maximum execution time was set to 60 minutes, in order to prevent an excessive computational execution time. Secondly, all default internal cross validation procedures were disabled, being replaced by an internal time ordered holdout split scheme, since time matters in this domain, i.e., all prediction models should be trained using older data and validate or tested using more recent data records [18].

3.3 Evaluation

In order to develop a realistic and robust comparison, a Rolling Window (RW) procedure was developed to simulate several training and testing iterations over time [18,5]. The RW iteratively trains the ML models, using a window W of the oldest data observations, to perform H-ahead predictions. Then, for the next iteration, the window rolls a step of S, by discarding the S oldest data observations from the training data and updating it with the S newest records, during a total of U iterations. In this work, aiming to achieve a reasonable amount of $U = 20$ iterations, we have set $W = 5,760$ hours (approximately 8 months) and $H = 24$ hours (corresponding to one day), while $S = 144$ hours

(6 days) was calculated according to the formula $S = D - (W + H)/U$, where $D = 8,760$ hours is the length of the full dataset. For a particular RW iteration, for the AutoML tools (Gluon, H2O and Oracle), the available training data is further split by using a time ordered holdout split method, producing validation (with V =12 most recent records, corresponding to half a day) and and fitting (with the remainder older examples). The fitting set is used to train the ML models, while the validation set is used to rank the ML algorithms and guide the AutoML search. As for the FEDOT tool, we used its default internal time series validation scheme, similarly to the setup adopted in [12].

Both the predicted performance and the computing effort required by the AutoML tools were taken into consideration during the evaluation process. To measure the accuracy of a predictive model, we use two popular regression measures, namely the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). To evaluate the computational effort, two temporal measures were used, namely the time to train the model (in minutes) and the test time (in milliseconds). The results that are shown in the next section correspond to the median values of the $U = 20$ RW iteration executions. We used the median aggregation function since it is less sensitive to outliers when compared with the average function. Regarding the specific FEDOT tool, it performs a pure TSF modeling for each of the 29 buildings, thus generating 29 distinct forecasting models. The FEDOT results were thus aggregated by first computing the median of the $U = 20$ RW iterations for each individual TSF model and then computing the overall median for all buildings. Statistical significance is measured by using the nonparametric Wilcoxon test [9].

4 Results

All experiments were executed using code written using the Python programming language. The FEDOT, H2O, and AutoGluon computational experiments were executed using a Linux virtual machine with 12 cores and 64 Gb of RAM. As for the Oracle AutoMLx, the tool license we had access only runs in a cloud system with 16 Gb of RAM and a `VM.Standard.E4.Flex` computational environment which includes one GPU.

The overall results are presented in Table 2, in terms of the median values of the $U = 20$ RW iterations. From the table, it becomes clear that scenario B performance values are much improved, demonstrating the importance of time-dependent (lagged) factors. For the B scenario, the Gluon model obtained the most accurate prediction measures, exhibiting MAE and RMSE values of 3.38 kW and 7.57 kW, followed by H2O and Oracle. As for the computational effort, the Oracle tool provided the most computationally efficient results. Yet, it should be noted that the Oracle results were obtained by using a different (and better) computational cloud infrastructure. Thus, the computational measures cannot be directly compared with the ones obtained by Gluon and H2O. When using the same computational environment, Gluon required less computational effort, in terms of both training and prediciton times, when compared with the

H2O tool and for both scenarios (A and B). Regarding FEDOT, which performs the pure TSF modeling approach, the obtained predictive performance values are significantly worse (for both MAE and RMSE) when compared with all scenario B AutoML executions. As for the computational effort, FEDOT requires a median computational effort (when modeling just one building) that is higher when compared with the AutoML tools. This is a relevant issue, since the actual FEDOT running time for all buildings 29 times higher (e.g., the total FEDOT RW median training time for all 29 buildings is around 2,175 minutes). Thus, considering the obtained results, it becomes clear that the usage of a single regression model for scenario B is the best energy consumption modeling approach, since it produces a single ML model that requires less computation and provides better predictive performances.

Table 2. Comparison of results (median RW values; best values in **bold**).

Scenario	**ML Model**	**MAE (kW)**	**RMSE (kW)**	**Train Time (min)**	**Prediction Time (ms)**
A	Gluon	14.33	25.56	27.0	0.39
	H2O	15.56	27.63	60.0	1.23
	Oracle	14.11	25.76	**1.0**	**0.02**
B	Gluon	**3.38***	**7.57**	42.0	0.83
	H2O	3.51†	7.81	60.0	1.15
	Oracle	3.60†	9.26	9.0	0.11
TSF	FEDOT	6.89	15.96	75.0	12.99

⋆ – Statistically significant under a paired comparison with all other methods.
† – Statistically significant under a paired comparison with FEDOT.

Further scenario B result comparison details are provided in Table 3, which shows the median values of the predictive performance measures for each tool and building typology type, comparing the AutoML results with the FEDOT tool. The last row shows the average values, when considering all six building types. Overall, Table 3 favors the Gluon tool, which obtains the best median MAE values for 4 building types (Schools, Parking lots, Civil Protection, and Others) and the best median RMSE values for 3 types (Administrative, Parking lots and Civil protection), while also presenting the lowest average (last row) MAE (5.68 kW) and RMSE (9.83 kW) values. In terms of the distinct Gluon performances, the predictive errors are substantially higher for the parking lots (median MAE of 12.66 kW), civil protection (9.13) and administrative buildings (6.58 kW). In contrast, high quality predictions were obtained for the cultural buildings (median MAE of 0.90 kW), schools (2.44 kW) and others (2.38 kW).

For demonstration purposes, the left of Fig. 1 plots the median of the Regression Error Characteristic (REC) curves [2] for all RW interactions with their

Table 3. Comparison of results by building typology (median rolling window values; best values in **bold**).

Typology	**Gluon**		**H2O**		**Oracle**		**FEDOT**	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Administrative	6.58	**15.64**	6.97	18.20	**6.53**	15.89	18.61	35.48
Cultural	0.90	1.29	**0.83**	**1.25**	0.96	1.40	1.26	1.96
Schools	**2.44**	4.49	2.45	**4.24**	2.46	4.27	5.10	9.21
Parking lots	**12.66**	**20.37**	14.61	22.95	13.32	20.68	20.79	27.52
Civil protection	**9.13**	**13.63**	10.39	15.11	13.17	17.58	16.50	20.35
Others	**2.38**	3.54	2.71	3.78	2.44	**3.51**	4.68	5.88
Average	**5.68**	**9.83**	6.33	10.92	6.48	10.56	11.16	16.73

respective Wilcoxon 95% confidence intervals. For all absolute tolerance values (x-axis) the Gluon results are significantly better when compared with FEDOT. For instance, when assuming a tolerance of 5 kW, Gluon correctly predicts 80% of the records while FEDOT only predicts 60%. The middle of Fig. 1 exemplifies the quality of the obtained Gluon forecasts for the last RW $U = 20$ iteration and one administrative building. Finally, the right of Fig. 1 plots the full RW Gluon results in terms of a regression scatter plot ($x-$axis denotes the target values and $y-$axis represents the predictions). The predictions, shown in terms of blue colored points, are close to the perfect prediction diagonal line.

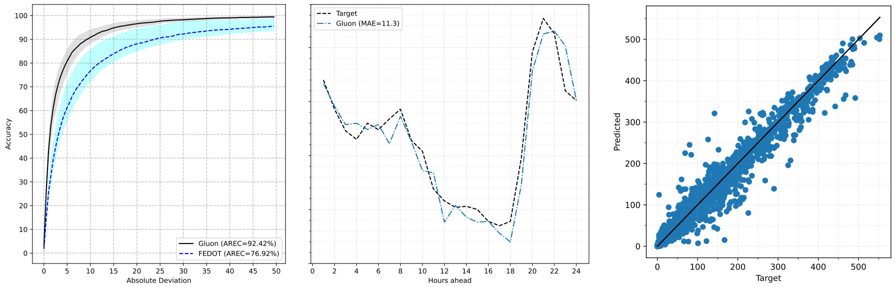

Fig. 1. RW median REC curve with Wilcoxon 95% confidence intervals (left), example of Gluon predictions for an Administrative building (middle) and Gluon RW regression scatter plot results (right).

Additionally, we performed an explainable AI (XAI) analysis by applying the SHapley Additive exPlanation (SHAP) method [10] (as implemented in the `shap` Python package) on the best ML model obtained by Gluon in the last RW iteration, which was a Weighted Ensemble (see Section 3.2). Fig. 2 presents the top 5 features in terms of their importance (left) and the SHAP values (right).

Regarding the feature importance, the consumption from the previous hour is the predominant attribute, with a relative importance of 55%, followed by the consumption in the previous day and same hour (8%), the hour of consumption (8%), the consumption in the previous weekday and same hour (7%) and the building useful area (2%). These results prove the relevance of newly lagged engineered features, since they correspond to 3 of the top 5 predominant attributes, demonstrating the relevance of an autoregressive modeling. The right of Fig. 2 shows the overall impact of each input in the predicted result. As an example, a decrease on the first input (consumption of the previous hour) reflects a decrease on the predicted consumption (denoted by the blue dots).

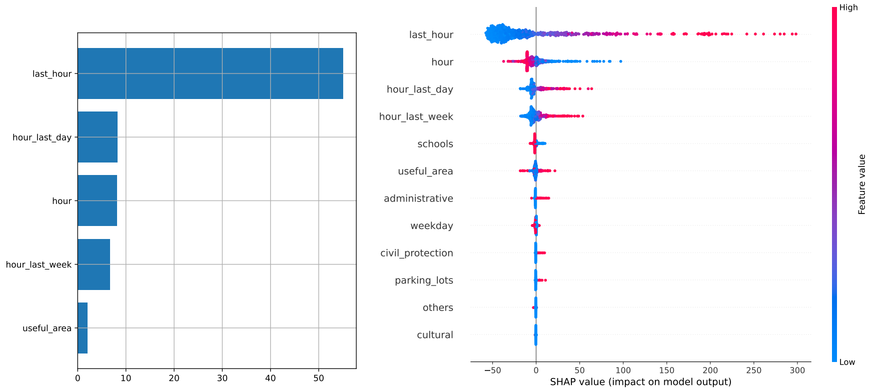

Fig. 2. Input importance for Gluon model (for iteration $U = 20$, left) and the impact of its inputs in the predicted responses (right).

5 Conclusions

Set within the context of Smart Cities, this paper presents an AutoML comparison study for predicting the hourly energy consumption of public buildings. Using a single regression modeling approach (for all buildings), we explore three recently proposed AutoML tools: AutoGluon, H2O and Oracle AutoMLx. The analyzed data was related with a one year collection period, related with 29 buildings from a Portuguese city. Several computational experiments were held, assuming a realistic rolling window evaluation, two input sets (with and without lagged attributes) and both predictive and computational effort measures. Furthermore, we also tested a univariate TSF approach that generates a predictive model for each building, using the FEDOT AutoTSF tool. The best overall results were obtained by the single regression AutoGluon lagged predictive method, returning a median MAE value of 3.38 kilowatts and requiring a reasonable computational effort. In future work, we intend to incorporate more building specific attributes, aiming to further improve the predictive results. We also plan to explore other AutoML tools (e.g., TPOT, TransmogrifAI) [8].

Acknowledgments

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

References

1. Bagnasco, A., Fresi, F., Saviozzi, M., Silvestro, F., Vinci, A.: Electrical consumption forecasting in hospital facilities: An application case. Energy and Buildings **103**, 261–270 (2015)
2. Bi, J., Bennett, K.P.: Regression error characteristic curves. In: Fawcett, T., Mishra, N. (eds.) Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA. pp. 43–50. AAAI Press (2003)
3. Burger, E.M., Moura, S.J.: Gated ensemble learning method for demand-side electricity load forecasting. Energy and Buildings **109**, 23–34 (2015). https://doi.org/10.1016/j.enbuild.2015.10.019
4. Chou, J.S., Tran, D.S.: Forecasting energy consumption time series using machine learning techniques based on usage patterns of residential householders. Energy **165**, 709–726 (2018). https://doi.org/10.1016/j.energy.2018.09.144
5. Cortez, P., Matos, L.M., Pereira, P.J., Santos, N., Duque, D.: Forecasting store foot traffic using facial recognition, time series and support vector machines. In: Graña, M., López-Guede, J.M., Etxaniz, O., Herrero, Á., Quintián, H., Corchado, E. (eds.) International Joint Conference SOCO'16-CISIS'16-ICEUTE'16 - San Sebastián, Spain, October 19th-21st, 2016, Proceedings. Advances in Intelligent Systems and Computing, vol. 527, pp. 267–276 (2016). https://doi.org/10.1007/978-3-319-47364-2_26
6. Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., Smola, A.J.: Autogluon-tabular: Robust and accurate automl for structured data. CoRR **abs/2003.06505** (2020)
7. Faiq, M., Tan, K.G., Liew, C.P., Hossain, F., Tso, C.P., Lim, L.L., Wong, A.Y.K., Shah, Z.M.: Prediction of energy consumption in campus buildings using long short-term memory. Alexandria Engineering Journal **67**, 65–76 (2023)
8. Ferreira, L., Pilastri, A.L., Martins, C.M., Pires, P.M., Cortez, P.: A comparison of automl tools for machine learning, deep learning and xgboost. In: International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021. pp. 1–8. IEEE (2021). https://doi.org/10.1109/IJCNN52387.2021.9534091
9. Hollander, M., Wolfe, D.A., Chicken, E.: Nonparametric statistical methods. John Wiley & Sons, NJ, USA (2013)
10. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 4765–4774 (2017)

11. Mosavi, A., Salimi, M., Faizollahzadeh Ardabili, S., Rabczuk, T., Shamshirband, S., Varkonyi-Koczy, A.R.: State of the art of machine learning models in energy systems, a systematic review. Energies **12**(7) (2019). https://doi.org/10.3390/en12071301
12. Pereira, P.J., Costa, N., Barros, M., Cortez, P., Durães, D., Silva, A., Machado, J.: A comparison of automated time series forecasting tools for smart cities. In: Marreiros, G., Martins, B., Paiva, A., Ribeiro, B., Sardinha, A. (eds.) Progress in Artificial Intelligence - 21st EPIA Conference on Artificial Intelligence, EPIA 2022, Lisbon, Portugal, August 31 - September 2, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13566, pp. 551–562. Springer (2022). https://doi.org/10.1007/978-3-031-16474-3_45
13. Pereira, P.J., Gonçalves, C., Nunes, L.L., Cortez, P., Pilastri, A.: AI4CITY - An Automated Machine Learning Platform for Smart Cities. In: SAC '23: The 38th ACM/SIGAPP Symposium on Applied Computing, Tallinn, Estonia, March 27 - 31, 2023. pp. 886–889. ACM (2023). https://doi.org/10.1145/3555776.3578740
14. Pham, A.D., Ngo, N.T., Truong, T.T.H., Huynh, N.T., Truong, N.S.: Predicting energy consumption in multiple buildings using machine learning for improving energy efficiency and sustainability. Journal of Cleaner Production **260**, 121082 (2020)
15. Robinson, C., Dilkina, B., Hubbs, J., Zhang, W., Guhathakurta, S., Brown, M.A., Pendyala, R.M.: Machine learning approaches for estimating commercial building energy consumption. Applied energy **208**, 889–904 (2017)
16. Seyedzadeh, S., Rahimian, F.P., Glesk, I., Roper, M.: Machine learning for estimation of building energy consumption and performance: a review. Visualization in Engineering **6**, 1–20 (2018)
17. Shapi, M.K.M., Ramli, N.A., Awalin, L.J.: Energy consumption prediction by using machine learning for smart building: Case study in malaysia. Developments in the Built Environment **5**, 100037 (2021)
18. Tashman, L.J.: Out-of-sample tests of forecasting accuracy: an analysis and review. International journal of forecasting **16**(4), 437–450 (2000)
19. Wang, C., Bäck, T., Hoos, H.H., Baratchi, M., Limmer, S., Olhofer, M.: Automated machine learning for short-term electric load forecasting. In: IEEE Symposium Series on Computational Intelligence, SSCI 2019, Xiamen, China, December 6-9, 2019. pp. 314–321. IEEE (2019). https://doi.org/10.1109/SSCI44817.2019.9002839
20. Wu, Z., Chu, W.: Sampling strategy analysis of machine learning models for energy consumption prediction. In: 2021 IEEE 9th International Conference on Smart Energy Grid Engineering (SEGE). pp. 77–81. IEEE (2021)
21. Yakovlev, A., Moghadam, H.F., Moharrer, A., Cai, J., Chavoshi, N., Varadarajan, V., Agrawal, S.R., Karnagel, T., Idicula, S., Jinturkar, S., Agarwal, N.: Oracle automl: A fast and predictive automl pipeline. Proc. VLDB Endow. **13**(12), 3166–3180 (2020). https://doi.org/10.14778/3415478.3415542