

Nowcasting of the energy production of wind power plants through spatially-aware model trees

(Discussion Paper)

Annunziata D'Aversa^{1,2,*}, Gianvito Pio^{1,2}

¹Dept. of Computer Science, University of Bari "Aldo Moro", Via E. Orabona, 4, 70125 Bari, Italy

²Data Science Lab, National Interuniversity Consortium for Informatics (CINI), Via Volturno, 58, 00185 Roma, Italy

Abstract

The accurate prediction of the energy production from renewable power plants in short-term intervals is of paramount importance in smart grids, to ensure an efficient distribution of energy within the network. Existing predictive approaches are mainly based on autoregressive models, machine learning methods and, more recently, on neural network architectures that also exploit spatio-temporal information. However, most of them are not able to capture spatial information at different degrees of locality, and tend to impose the presence of linear (or non-linear) dependencies among data.

In this paper, we discuss a novel approach that is based on linear model trees, to simultaneously model linear and non-linear dependencies, properly extended to capture the spatial dimension at different degrees of locality. The proposed approach is able to work in the multi-step predictive setting, that means that it can simultaneously provide predictions for multiple time intervals in the future.

Our experiments on a real dataset about the energy produced by wind power plants demonstrate the effectiveness of our method also in comparison with state-of-the-art neural network architectures.

Keywords

Time series nowcasting, Spatio-temporal autocorrelation, Multi-step prediction

1. Introduction

Smart grids are networks that distribute electricity with the support of sensors, advanced communication technologies, and predictive components. Within the latter, models able to forecast the energy consumption and production play a fundamental role. Indeed, in long-term scenarios, they can support planning interventions on the network, aiming not only to decrease production costs but also to contribute to the reduction of greenhouse gas emissions. On the other hand, in short-term scenarios, the forecasting (usually called *nowcasting*, in the case of very short-term timeframes) of energy production and consumption can be useful for performing real-time load balancing actions, that may include powering on backup plants or drawing energy from customers' accumulators.

In general, predictive models can be built by relying on machine learning methods by exploiting historical data and spatial information of nodes. Indeed, the spatial dimension may introduce

SEBD 2024: 32nd Symposium on Advanced Database Systems, June 23-26, 2024, Villasimius, Sardinia, Italy

*Corresponding author.

✉ annunziata.daversa@uniba.it (A. D'Aversa); gianvito.pio@uniba.it (G. Pio)

🆔 0000-0003-1791-5998 (A. D'Aversa); 0000-0003-2520-3616 (G. Pio)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

spatial autocorrelation phenomena, which refer to dependencies that may exist among observations at nearby geographical locations. In this context, the spatial proximity among power plants or among customers can influence measurements due to similar climatic conditions.

Another important aspect is that real-world time-series coming from sensor measurements often exhibit a combination of linear and non-linear trends. This is very common when measurements depend on weather conditions, which may easily show non-linear phenomena, e.g., possibly due to storms or other extreme events. Non-linear phenomena may also emerge in the case of power grid failures. Therefore, capturing both linear and non-linear trends and relationships, along with the exploitation of historical data and spatial information, could improve the model performance and lead to more accurate predictions.

In the literature, several nowcasting approaches have been proposed leveraging on autoregressive models [1, 2], machine learning models [3, 4] and hybrid models [5]. However, in the literature, we can find only a few works that also take into account the spatial dimension [6, 7, 8, 9]. For instance, in [6] the authors propose a method for 5-minute ahead wind power forecasting. The authors capture spatio-temporal dependencies using a method based on sparse parametrization of VAR models, which selects coefficients that link sites with a spatial co-dependence, discarding those exhibiting weak dependencies. Another relevant example is [8], where the authors proposed a spatio-temporal graph convolution neural network for the short-term prediction of the energy produced by wind power plants. The authors consider a multi-step setting, where 16 future values (at a 15-minutes interval) are predicted simultaneously.

The contribution of the temporal and spatial dimensions has also been considered in the context of the more classical forecasting scenarios to predict the hourly energy production of photovoltaic power plants 24 hours ahead [10], or to predict the monthly energy consumption of customers one year ahead [11]. Also these works consider a multi-step setting, where the 24 hourly predictions (in [10]) and the 12 monthly predictions (in [11]) are returned simultaneously by the model, possibly exploiting dependencies among them. The spatial dimension is considered by resorting to two well known techniques in spatial statistics: the Local Indicator of Spatial Association (LISA), that represents a local measure of spatial autocorrelation [12], and the Principal Coordinates of Neighbour Matrices (PCNM), that represent the spatial structure in the data [13]. Such indicators are used to augment the feature space of training instances.

Recently, several neural network architectures that consider both temporal and spatial dimensions have been proposed, but they were applied in different application domains. A relevant example is MTGNN [14], that is a graph convolutional network applied to multiple domains, including energy and traffic speed forecasting. MTGNN employs multiple temporal convolutional networks (TCNs) with various kernel sizes, for learning temporal dependencies at different scales, and a self-adaptive adjacency matrix to capture spatial correlations.

It is noteworthy that, although some of the mentioned approaches are able to represent and exploit the spatial information, they cannot capture spatial dependencies at different degrees of locality. A first attempt to capture local spatial information can be found in [15], where the authors proposed the method D^2 STGNN applied to the traffic speed forecasting. D^2 STGNN identifies both diffusion signals, representing how traffic conditions spread through the network, and inherent patterns, such as recurring traffic patterns or daily / seasonal variations. The model adopts a spatio-temporal localized convolution to capture hidden diffusion time series, while a combination of GRU (for short-term dependencies) and multi-head self-attention mechanism

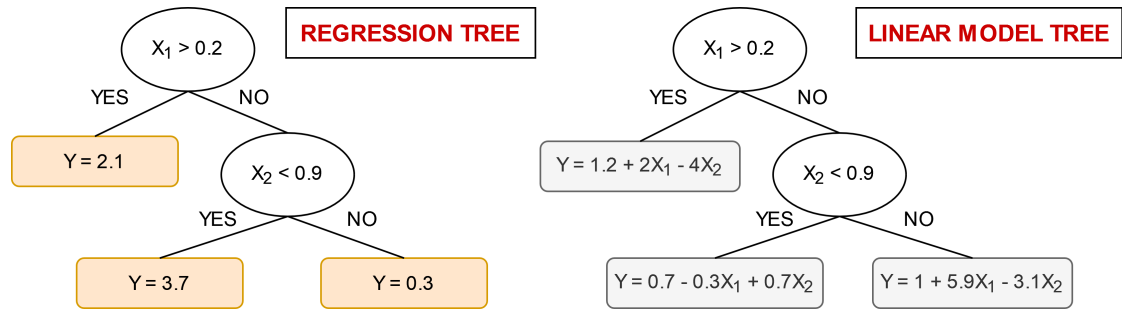


Figure 1: An example showing the difference between a classical regression tree and a linear model tree.

(for long-term dependencies) is employed to model hidden inherent time series.

In this paper, we discuss an approach to solve nowcasting tasks in the context of the prediction of the energy produced by wind power plants, in a multi-step setting. Specifically, we aim at learning a nowcasting model capable of predicting the energy production for 12 time-steps, at a 15-minutes granularity. Methodologically, contrary to most existing approaches, we capture both linear and non-linear phenomena through linear model trees. Moreover, we extend them to effectively capture and model the spatial information at different levels of locality.

2. Spatially-aware linear model trees

As introduced in Section 1, we aim at adopting an approach that is able to capture both linear and non-linear dependencies. In this respect, we argue that linear model trees [16] can represent a possible solution, since they combine the ability to model non-linear dependencies of regression trees with that of linear models. Existing methods for the construction of model trees employ a learning process characterized by a top-down induction procedure that recursively partitions the training set, which is analogous to that adopted by conventional tree-based algorithms.

In linear model trees, in the leaf nodes we find linear models instead of constant approximations of classical regression trees. More formally, given a set of independent variables and a dependent variable y , a standard regression tree returns, for each leaf node k , a constant value c_k , namely, $y = c_k$ for all the instances falling in the leaf node k . Such constant value is usually an aggregation (mean, median, etc.) of the value y of the training instances falling in the leaf node. On the other hand, in model trees, each leaf node of the tree *contains* a linear regression model that predicts the target variable based on the data points that reach that leaf. An example to illustrate the difference between a regression tree and a linear model tree is shown in Fig. 1.

The quality of a split is usually measured using a criterion that quantifies how well the split separates data with respect to the target variable. For example, in CART [17], the quality of a split is evaluated by the Mean Squared Error (MSE). When a node is split, the MSE is computed for each resulting child node, and the weighted sum (according to the number of instances) of these MSE values represents the quality of the split. The best split is defined as the one that minimizes the MSE. In the case of linear model trees, the behavior is similar: the only difference is that the MSE on the child nodes is computed after fitting a linear model on them.

In our approach, we considered the multi-step (MS) setting proposed in [11] that consists in predicting multiple future values of the target variable simultaneously. In particular, our approach falls in the Multi-Input Multi-Output (MIMO) category [18], which goal is learning a global predictive model that returns the whole vector of predictions, also taking into account the possible dependencies between future values, that in principle may be beneficial in terms of forecasting accuracy. More formally, we consider as input features w historical values of the target variable $y_{t-w}, y_{t-w+1}, \dots, y_{t-1}$, in order to predict the value of the target variable for h future timesteps $y_t, y_{t+1}, \dots, y_{t+h}$, simultaneously. Note that, in this case, the reduction of the MSE of a split is evaluated as the average reduction of MSE over all the future timesteps.

In the literature, we can find several implementations of linear model trees [16, 19, 20]. In this work, we consider the simplest implementation, where internal nodes are simple tests involving descriptive variables, while leaf nodes are linear models, as shown in the right part of Fig. 1. This choice makes our extension towards the consideration of the spatial dimension more straightforward. Specifically, as introduced in Section 1, we aim at extending linear model trees to effectively capture and model the spatial dimensions at different levels of locality.

Methodologically, we introduce the consideration of the spatial dimension as a post-processing step of the tree construction: we aim at capturing spatial relationships within each subset implicitly defined by a leaf node of the model tree, potentially capturing spatial relationships at different levels of locality. Considering to have multiple different positions (e.g., production plants or consumers), each represented through several w -dimensional training instances, we act as follows: for each instance $x_{\alpha,t}$ fallen into a leaf node l , related to the time step t and to the geographic position α , we compute a set of additional features $S_{\alpha,t,l}$. These features are computed as the weighted average of the w -dimensional historical observations at the same time step t from other positions¹ in l , where the weights are determined by the spatial closeness between α and the other positions (see Figure 2). More formally, $S_{\alpha,t,l}$ is defined as follows:

$$S_{\alpha,t,l} = \frac{1}{\sum_{\beta \in P_l, \beta \neq \alpha} C[\alpha, \beta]} \cdot \sum_{\beta \in P_l, \beta \neq \alpha} C[\alpha, \beta] \cdot x_{\beta,t} \quad (1)$$

where P_l is the set of distinct positions of the training instances fallen into the leaf node l ; $x_{\beta,t}$ is the vector of w historical observations of the location β at the time step t ; $C[\alpha, \beta]$ is the spatial closeness between the positions α and β computed as follows:

$$C[\alpha, \beta] = 1 - \frac{D[\alpha, \beta]}{\max(D)} \quad (2)$$

where D is the distance matrix among locations computed according to the geodesic distance.

The additional features are computed and added to all the training instances falling into the leaf node. Finally, a new linear model is trained and the contribution of the added features is assessed using a validation set. Therefore, we compare two distinct linear models, as depicted in Figure 3. The first model is exclusively trained on the original features (during the construction of the tree), while the second model incorporates both the original features and the additional ones computed according to the spatial closeness. We selectively retain the model that demonstrates

¹Note that, if a given leaf node contains training instances associated with only one position, this step is skipped.

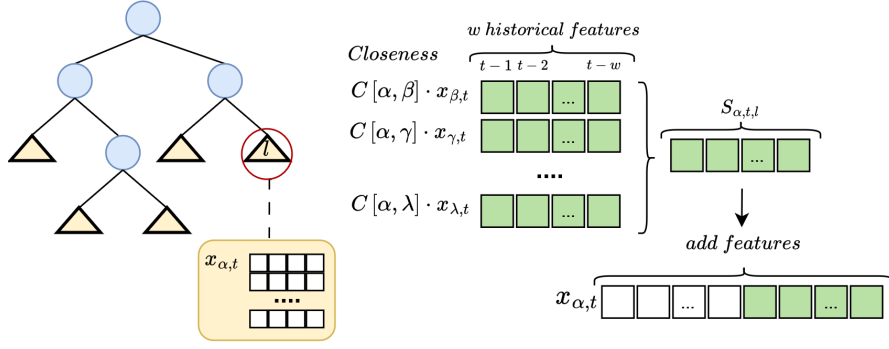


Figure 2: An example of computation of additional features $S_{\alpha, t, l}$ for the instance $x_{\alpha, t}$ fallen in the leaf node l , given the presence of other instances in l belonging to the geographic positions β, γ , and λ .

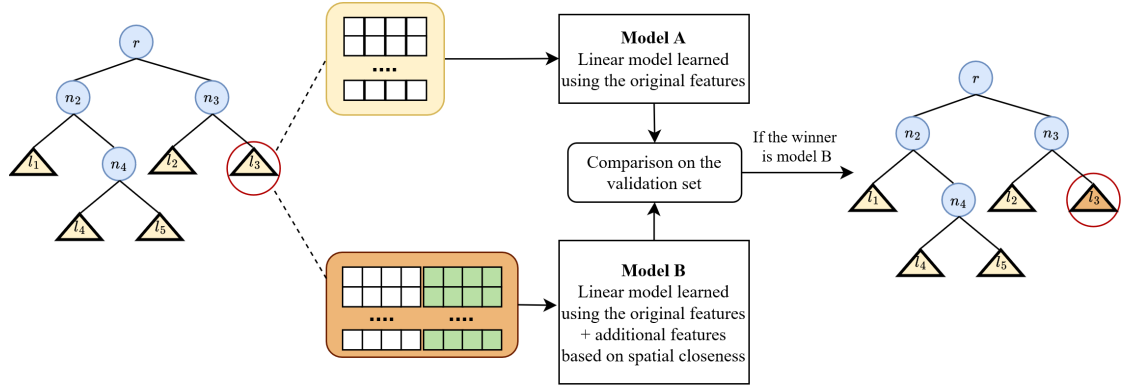


Figure 3: Comparison of the predictive performance of two linear models on the training instances of a leaf node: the first model is learned from the original features, while the second is learned from the original features expanded with the features computed according to the spatial closeness.

the lowest validation error within each leaf node. This selection process ensures that we tailor our modeling approach to the specific peculiarities of each subset of data falling into leaf nodes. Consequently, within this tree, some leaf nodes may employ models that incorporate spatial features, while others may rely only on the original features (i.e., when the additional features based on spatial closeness appear to provide no advantage).

After performing this process on all the leaf nodes, we apply a pruning step to prevent overfitting and possibly capture *more global* (i.e., less local) spatial dependencies. In particular, we propose an extended version of the Reduced Error Pruning (REP) algorithm [21]: starting from the bottom of the tree and working backward, for each internal node, it compares the error made by the unpruned tree with that made simulating that the subtree rooted on the node is pruned. The subtree is actually pruned only if the resulting tree performs no worse than the unpruned one over the validation set. In our extended version, we also consider the possible contribution coming from the features based on the spatial closeness. In particular, we compare the unpruned tree with the pruned tree and with the pruned tree that also considers the features based on the spatial closeness. Considering the example reported in Figure 4, given

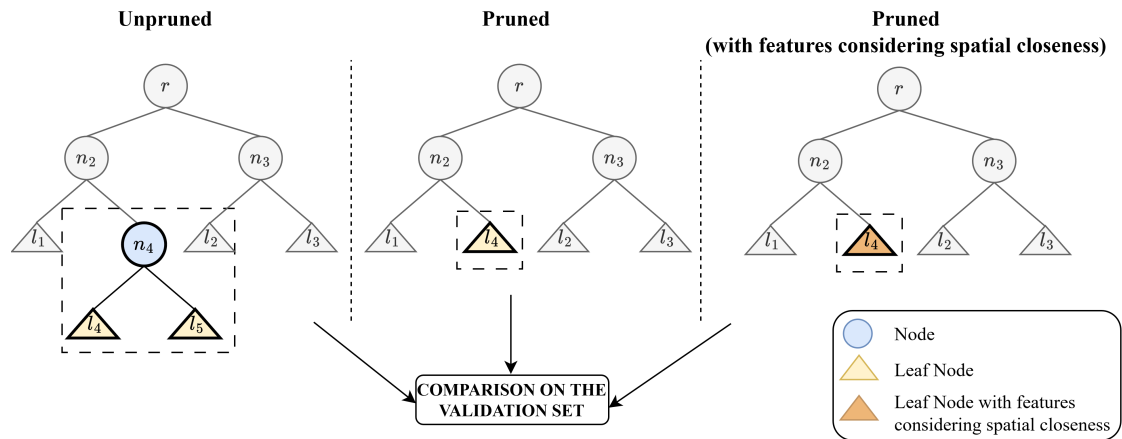


Figure 4: Extended version of the Reduced Error Pruning strategy that also takes into account the contribution of additional features that consider the spatial closeness.

the internal node n_4 , we compare the errors made on the validation set by three models: *i*) the model represented by its two children nodes l_4 and l_5 (see the left part of Figure 4); *ii*) the model obtained after pruning the subtree rooted in n_4 and learning a new linear model from the instances falling into it (see the middle part of Figure 4); *iii*) the model obtained after pruning the subtree rooted in n_4 and learning a new linear model from the instances falling into it, expanded with the features considering the spatial closeness (see the right part of Figure 4). If the model *ii*) or the model *iii*) leads to an improvement on the validation set, the tree is pruned accordingly. This process continues in a bottom-up fashion until no improvement is obtained.

3. Experiments

In order to assess the effectiveness of the proposed approach, we performed our experiments on a real-world wind power plants dataset, provided by a lead company in the energy distribution field. The dataset consists of measurements of the energy production of 60 wind plants, collected every 15 minutes for a period of 1 year. Together with the geographic position (latitude and longitude), the plants are described by some technical characteristics, namely, avg_wind_turbine_height, rotor_diameter, and number_of_wind_turbines.

Following a cross-validation setting for time series, we consider a sliding window approach where the training set consists of 4 months of data, the validation set corresponds to the last month of the training set, and the test set is the subsequent month. We performed the experiments considering a multi-step setting, where the goal is to predict the energy production of 12 target time-steps ahead simultaneously. As historical measurements associated with each instance, we consider 12 previous values of energy production, i.e., $w = 12$. It is noteworthy that, in real-world production scenarios, actual measurements are often made available after a certain amount of time. Therefore, we evaluated the performance of all the models considering different delays from the last observed measurement and the first target time-step to predict. The considered delays are 0 hours, 2 hours and 4 hours.

To learn the initial model tree, we considered the implementation available in the *linear-tree* python library². For all the experiments, we investigated two different configurations of its parameters, namely: $min_samples_leaf = 0.1$, $max_depth = 5$ and $min_samples_leaf = 0.05$, $max_depth = 20$. The original version of this system (henceforth denoted with **LT**), that ignores the spatial information, has been considered as the closest competitor to our approach. As additional competitor systems, we considered three different regressors that are able to work in the multi-step setting, namely, Linear Regression (henceforth denoted with **LR**), Random Forests (henceforth denoted with **RF**) and XGBoost Regressor (henceforth denoted with **XGB**). For all these competitors, we also assessed the performance achieved when the spatial information is considered by injecting PCNM variables [13]. This allows us to specifically evaluate the contribution of the novel strategy that we proposed to model the spatial dimension. Finally, we considered two state-of-the-art neural network architectures that can work in the multi-step setting and capture spatio-temporal phenomena, i.e., **MTGNN** [14] and **D²STGNN** [15].

As evaluation measure, we collected the Relative Squared Error (RSE) for LT, and the percentage of improvement with respect to the best configuration of such a model for the proposed method and for all the considered competitor systems. The RSE is formally defined as $RSE = \frac{\sum_t (y_t - \tilde{y}_t)^2}{\sum_t (y_t - \bar{y})^2}$, where y_t and \tilde{y}_t are the true and the predicted values, respectively, for the t -th time-step, while \bar{y} is the average value of a given target time-step in the training set.

The adoption of the RSE, instead of more commonly adopted measures like MAE/MSE/RMSE, allows us to evaluate the actual usefulness of the predictive models in real scenarios, with respect to adopting a baseline predictor that always returns the mean of the measurements: an RSE value close to 0.0 means that the model returns perfect predictions; an RSE value close to 1.0 corresponds to a model that performs analogously to the baseline that always returns the mean; an RSE value higher than 1.0 means that the model performs worse than such a baseline.

In Table 1, we report the RSE results, averaged over all the target time-steps and over all the folds of the cross-validation. As expected, all the considered methods perform worse with higher delays. Nevertheless, all the RSE values remains under 1.0, which means that they can still provide more useful indications than those provided by the baseline predictor based on the average. Looking at the results obtained by our approach, it clearly provides advantages over LT with all the values of delay and in both configurations of its parameters. On the contrary, all the other competitors perform worse than (or equal to) LT, except for few specific cases, where the improvement is no more than 0.6%. These results confirm the adequacy of adopting model trees in this application domain, due to the co-presence of linear and non-linear phenomena.

Looking at the contribution provided by the PCNM variables to the competitors, we can observe no evident differences with respect to the same methods with no PCNM features, with some peculiar cases in which the error also increases (see, for example, RF+PCNM vs RF). This is possibly due to the fact that PCNM variables do not take historical factors into account. On the other hand, our approach incorporates additional historical features, taking into account the spatial closeness at different degrees of locality. This approach clearly performs better than injecting *static* features dependent on the positions as seen by the approaches relying on PCNM.

In general, we can observe that our approach outperforms all the considered competitors, including those based on recent neural network architectures. Surprisingly, they obtained the

²<https://github.com/cerlymarco/linear-tree>

	Model	min_samples_leaf	max_depth	0 hours delay	2 hours delay	4 hours delay
RSE	LT	0.1	5	0.261	0.499	0.645
	LT	0.05	20	0.260	0.498	0.644
% of Improvement	LT+PCNM	0.1	5	0.00%	0.60%	0.30%
	LT+PCNM	0.05	20	-2.70%	-9.00%	-4.00%
	RF	default	default	-3.40%	-3.60%	-4.70%
	RF+PCNM	default	default	-3.40%	-4.00%	-4.80%
	XGB	default	default	0.00%	0.20%	0.00%
	XGB+PCNM	default	default	-0.40%	0.20%	0.00%
	LR	-	-	-0.80%	-0.20%	0.00%
	LR+PCNM	-	-	-0.80%	-0.20%	0.20%
	MTGNN	-	-	-181.89%	-63.73%	-35.19%
	D ² STGNN	-	-	-123.92%	-60.26%	-30.96%
	Our approach	0.1	5	6.10%	4.40%	2.30%
Our approach	0.05	20	6.50%	5.20%	2.80%	

Table 1

Average RSE obtained by LT and percentage of improvement with respect to such a method (with `min_samples_leaf = 0.1` and `max_depth = 5`), obtained by our approach and by the competitor systems. Positive and negative percentages are emphasized with green and orange backgrounds, respectively.

worst results among the considered systems. This is possibly due to the complexity of their architecture that requires a huge amount of training data (possibly much higher than those available in this context) to properly learn an accurate model.

4. Conclusion

In this paper, we presented an approach for nowcasting the energy produced by wind power plants in a multi-step predictive setting. We enabled linear model trees to capture spatial phenomena at different degrees of locality. Specifically, we incorporate additional features that represent historical observations of other plants, taking into account their spatial closeness. Moreover, we also extended the REP pruning strategy to consider the spatial dimension.

Our experiments, performed on a real-world dataset, proved the effectiveness of the proposed approach, in comparison with standard linear trees and other state-of-the-art competitors that are also able to model the spatial dimension.

For future work, we intend to evaluate the effectiveness of the proposed method in other domains, and to perform a deep evaluation of the difference in terms of (theoretical and empirical) model complexity with respect to unpruned linear trees and complex neural networks.

Acknowledgments

This work was partially supported by the project FAIR - Future AI Research (PE00000013), Spoke 6 - Symbiotic AI, under the NRRP MUR program funded by the NextGenerationEU. The research of Annunziata D'Aversa is funded by a PhD fellowship within the framework of the Italian "POR Puglia FSE 2014-2020" – Axis X - Action 10.4 "Interventions to promote research and for university education - PhD Project n. 1004.121 (CUP n. H99J21006620008).

References

- [1] Aasim, S. Singh, A. Mohapatra, Repeated wavelet transform based arima model for very short-term wind speed forecasting, *Renewable Energy* 136 (2019) 758–768.
- [2] P. Bacher, H. Madsen, H. A. Nielsen, Online short-term solar power forecasting, *Solar Energy* 83 (2009) 1772–1783.
- [3] Q. Hu, S. Zhang, M. Yu, Z. Xie, Short-term wind speed or power forecasting with heteroscedastic support vector regression, *IEEE Transactions on Sustainable Energy* 7 (2016) 241–249.
- [4] C. Li, G. Tang, X. Xue, A. Saeed, X. Hu, Short-term wind speed interval prediction based on ensemble gru model, *IEEE Transactions on Sustainable Energy* 11 (2020) 1370–1380.
- [5] P. Jiang, Y. Wang, J. Wang, Short-term wind speed forecasting using a hybrid model, *Energy* 119 (2017) 561–577.
- [6] J. Dowell, P. Pinson, Very-short-term probabilistic wind power forecasts by sparse vector autoregression, *IEEE Transactions on Smart Grid* 7 (2015) 763–770.
- [7] X. G. Agoua, R. Girard, G. Kariniotakis, Short-term spatio-temporal forecasting of photovoltaic power production, *IEEE Transactions on Sustainable Energy* 9 (2018) 538–546.
- [8] Z. Li, L. Ye, Y. Zhao, M. Pei, P. Lu, Y. Li, B. Dai, A spatiotemporal directed graph convolution network for ultra-short-term wind power prediction, *IEEE Transactions on Sustainable Energy* 14 (2023) 39–54.
- [9] M. Khodayar, J. Wang, Spatio-temporal graph deep neural network for short-term wind speed forecasting, *IEEE Transactions on Sustainable Energy* 10 (2019) 670–681.
- [10] M. Ceci, R. Corizzo, F. Fumarola, D. Malerba, A. Rashkovska, Predictive modeling of pv energy production: How to set up the learning task for a better prediction?, *IEEE Transactions on Industrial Informatics* 13 (2017) 956–966.
- [11] A. D’Aversa, S. Polimena, G. Pio, M. Ceci, Leveraging spatio-temporal autocorrelation to improve the forecasting of the energy consumption in smart grids, in: P. Pascal, D. Ienco (Eds.), *Discovery Science*, Springer Nature Switzerland, Cham, 2022, pp. 141–156.
- [12] L. Anselin, Local indicators of spatial association – LISA, *Geographical analysis* 27 (1995) 93–115.
- [13] S. Dray, P. Legendre, P. R. Peres-Neto, Spatial modelling: a comprehensive framework for principal coordinate analysis of neighbour matrices (PCNM), *Ecological modelling* 196 (2006) 483–493.
- [14] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, C. Zhang, Connecting the dots: Multivariate time series forecasting with graph neural networks, in: *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 753–763.
- [15] Z. Shao, Z. Zhang, W. Wei, F. Wang, Y. Xu, X. Cao, C. Jensen, Decoupled dynamic spatial-temporal graph neural network for traffic forecasting., volume 15, *VLDB Endowment*, 2022, pp. 2733–2746.
- [16] J. R. Quinlan, et al., Learning with continuous classes, in: *5th Australian joint conference on artificial intelligence*, volume 92, World Scientific, 1992, pp. 343–348.
- [17] L. Breiman, J. Friedman, C. Stone, R. Olshen, *Classification and Regression Trees*, Routledge, 2017.
- [18] S. B. Taieb, G. Bontempi, A. F. Atiya, A. Sorjamaa, A review and comparison of strategies

for multi-step ahead time series forecasting based on the NN5 forecasting competition, *Expert systems with applications* 39 (2012) 7067–7083.

- [19] Y. Wang, I. Witten, Induction of model trees for predicting continuous classes, *Induction of Model Trees for Predicting Continuous Classes* (1997).
- [20] D. Malerba, F. Esposito, M. Ceci, A. Appice, Top-down induction of model trees with regression and splitting nodes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (2004) 612–625. doi:10.1109/TPAMI.2004.1273937.
- [21] J. Quinlan, Simplifying decision trees, *International Journal of Man-Machine Studies* 27 (1987) 221–234.