Computing (2023) 105:1623–1645
https://doi.org/10.1007/s00607-023-01164-y

REGULAR PAPER

Energy generation forecasting: elevating performance with machine and deep learning

Aristeidis Mystakidis[1,2] · **Evangelia Ntozi**[1] · **Konstantinos Afentoulis**[1] · **Paraskevas Koukaras**[1,2] · **Paschalis Gkaidatzis**[1] · **Dimosthenis Ioannidis**[1] · **Christos Tjortjis**[1,2] · **Dimitrios Tzovaras**[1]

Received: 28 July 2022 / Accepted: 4 February 2023 / Published online: 25 February 2023

Abstract

Distribution System Operators (DSOs) and Aggregators benefit from novel Energy Generation Forecasting (EGF) approaches. Improved forecasting accuracy may make it easier to deal with energy imbalances between production and consumption. It also aids operations such as Demand Response (DR) management in Smart Grid architecture. This work aims to develop and test a new solution for EGF. It combines various methodologies running EGF tests on historical data from buildings. The experimentation yields different data resolutions (15 min, one hour, one day, etc.) while reporting accuracy errors. The optimal forecasting technique should be relevant to a variety of forecasting applications in a trial-and-error manner, while utilizing different forecasting strategies, ensemble approaches, and algorithms. The final forecasting evaluation incorporates performance metrics such as coefficient of determination (R^2), Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), presenting a comparative analysis of results.

Keywords Forecasting · Time series analysis · Energy generation · Machine learning · Deep learning · Artificial neural networks

Mathematics Subject Classification 62J02 · 37M10 · 68T99 · 82C32

Evangelia Ntozi and Konstantinos Afentoulis have contributed equally to this work.

✉ Christos Tjortjis
c.tjortjis@ihu.edu.gr

Extended author information available on the last page of the article

1 Introduction

Secured sustainability requires higher effective energy management with minimal energy losses. As a result, the future power grid should give exceptional levels of flexibility in energy management. To that end, intelligent decision making requires accurate future energy demand and load forecasts. EGF is essential in the development and management of power systems. It enables energy suppliers to estimate electricity usage and plan for future power demands. It also enables power distributors to optimal manage and match future electricity production with demand. As a result, EGF has recently gained considerable attention, sometimes proven to be a challenging subject.

In the context of EGF, this article presents time-series data analysis, a commonly used approach for evaluating a succession of values associated with unique timestamps. This includes elaborating on common evaluation metrics, such as coefficient of determination (R^2), Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), mostly derived from statistics to assess the predictive ability of each implemented model. The forecasting methods and algorithms that were utilized in this study can be divided into Machine Learning (ML) and Deep Learning (DL) techniques and lastly, an ensemble method, key part of this work's contribution.

Data quality is one of the most common concerns when performing accurate predictions. Data should be complete, up to date and accessible in its entirety. Its quantity, on the other hand, while it also depends on the type of algorithm used, might constitute a balancing challenge. In other words, although large data sets can be rather beneficial when it comes to model training, sometimes time and space complexity can pose as a significant constraint. Other related difficulties concern model over-fitting and handling outliers. According to the literature, highest granularity on data may be used to make more accurate energy load projections [1].

Practical implications yield the cooperation of energy stakeholders, such as DSOs and Aggregators resulting in the development of more efficient DR strategies [2]. Also, financial planning, tariff design, power system operation and electrical grid maintenance, load switching, and infrastructure construction are all important applications of EGF.

The remainder of this article is structured as follows: Sect. 2 reviews the state of the art and provides the necessary context. Section 3 analyzes the developed concepts/methodology of the proposed approach for time-series EGF. Section 4 presents the results of experiments conducted on pilots. The paper concludes with Sect. 5, discussing final thoughts, implications and future prospects.

2 Background

This section discusses the state of the art in time-series EGF. It also reports on the algorithms, ensemble methods and evaluation metrics utilized for experimentation.

2.1 State of the art

This sub-section discusses the state of the art regarding EGF focusing on Photovoltaic (PV) use cases utilizing ML, DL algorithms and ensemble methods.

2.1.1 PV use cases

PV power forecasting has received a lot of attention. The efforts to improve the accuracy of Electrical Energy Generation (EED) forecasts, include the utilization of various computational and statistical techniques [3]. Forecasting models are broadly classified into two categories: indirect and direct forecasting models. Indirect models predict the solar irradiance of the area that the PVs are installed. Various methods were utilized to predict the production of the PVs on different time scales, including numerical weather prediction, image, statistical and hybrid Artificial Neural Networks (ANNs) [4]. In direct forecasting models, power generation of the PVs is forecasted directly using historical data samples, such as PV power output and associated meteorological data. Thus, [5] implemented direct and indirect methods to predict power generation of a PV system, and concluded that the direct method is superior.

2.1.2 Energy generation forecasting types

Furthermore, EGF was classified similarly with the common categorization of Energy Load Forecasting (ELF) [6–8]. In this paper we customize and employ ELF notation and terminology to fit the EGF domain.

Therefore, four main groups were distinguished: Very Short-Term Generation Forecasting (VSTGF), Short-Term Generation Forecasting (STGF), Medium-Term Generation Forecasting (MTGF) and Long-Term Generation Forecasting (LTGF) [9, 10]. In real-time control, the VSTGF is appropriate, since its predicting period ranges between a few minutes and one hour ahead. The STGF is utilized for forecasting within one hour to one week or month ahead [11].

Authors in [9] compared 45 academic papers on Energy Efficiency Directive (EEDi) forecasting based on time frame, inputs, outputs, the scale of the project, and value. This study revealed that despite the simplicity of regression models, they are mostly useful for long-term load forecasting compared with AI-based models such as ANN, Fuzzy logic, and SVM, which are appropriate for STGF.

Achieving STGF accuracy is challenging. Solar radiation/PV power forecasting is a non-linear problem, which depends on several weather parameters. As a result, finding the proper parameter estimation method for an nonlinear problem is difficult. Several methods have been proposed for forecasting in the literature, as the choice of forecasting model depends on forecasting horizon and selected location. Data-driven models are based on the extraction of useful information from the input training data, and based on this information, these models predict the output. The performance of these methods is susceptible to the quality of the training data. ML techniques need some historical data for the training of the model, so, for direct PV power forecasting, the availability of such historical PV power data is an essential requirement.

ANN and Support Vector Machine (SVM) [12–14] are the two most often used ML algorithms in the field of PV forecasting. Several studies exist in the area of ANN that validate the better non-linear fitting ability of neural networks compared to time-series models. Wide research in neural networks, from an early simple architecture to a late deep configuration, results to performance of these networks. Authors in [15] presented a new DL model Bi-directional Long Short-Term Memory (Bi-LSTM), for PV power forecasting. After comparing the results of various structures of Neural Networks (NNs), and time-series models (ARMA, ARIMA, and SARIMA), the prediction accuracy of NNs was reported to be higher with less computational time. Unlike classic Recurrent Neural Network (RNN), LSTM contains a memory unit that helps to keep the long spans data and can also solve the gradient descent problem. This, enables LSTM to extract temporal information from the time-series data. Similarly, a Deep Belief Network (DBN) was presented in [16] to learn the non-linear features from the previous PV power time-series data. Additionally, authors in [17] provided a broad overview on optimization based hybrid models developed on ANN models.

The selection of the suitable hyperparameters values are pivotal for ML algorithms and have a major impact on forecasting accuracy. For example, C and gamma are the two important parameters of SVM, whose inappropriate values are responsible for overfitting and underfitting issues. As a result, many researchers adopt intelligent optimisers for hyperparameter tuning of ML algorithms. The research of [18] reported the improvement in R^2 score (coefficient of determination) from 0.991 to 0.997 by utilising an improved ACO to optimise SVM parameters. Similarly, [19] adopted a genetic algorithm, while an improved chicken swarm optimisation was adopted from [20] to tune the hyperparameters of Extreme Learning Machine (ELM), and both reported a better forecast accuracy with the incorporation of optimisation algorithms. Additionally, [21] and [22] presented DL based models for PV power forecasting by optimising the parameters with Particle Swarm Optimization (PSO) and Randomly Occurring Distributedly Delayed PSO (RODDPSO) techniques, respectively.

The work of [23] developed 12 data driven models of shallow ML and DL. The findings are that Extreme Gradient Boosting (XGBoost) and Long Short-Term Memory (LSTM) are the most accurate shallow and DL model, respectively. Thus, they concluded that LSTM performs well for short-term prediction (1-hour ahead), but not for long term prediction (24 h ahead), because the sequential information becomes less relevant and useful when the prediction horizon is long. Secondly, the presence of weather forecast uncertainty deteriorates XGBoost's accuracy and favors LSTM, because the sequential information makes the model more robust to input uncertainty. Gaussian Processes (GPs) appear to be one of the promising methods for providing probabilistic forecasts. In this paper, the Log-normal Process (LP) is newly introduced and compared to the conventional GP. The LP is especially designed for positive data like residential load forecasting and little regard was taken to address this issue previously [24].

2.1.3 Time-series forecasting techniques

There are several techniques to utilize time-series forecasting with ML and DL models. In most cases, the sliding window approach, a commonly used technique for time-series forecasting, is utilized [25]. This technique along with several ML or DL algorithms is being used in many fields [26–28].

For multiple steps ahead forecasting, there are three popular strategies, direct, rolling (or recursive) [29] and sequence to sequence [30].

2.1.4 Ensemble methods

The intention is to utilize, tune and combine the best algorithms. A comparison between the three best algorithms has been made, in order to identify how each model behaves on specific parts of the target parameter. Our investigation is focused on regression and averaging models. Therefore, we utilize ensemble models which are hybrid component combination based models. They improve the accuracy and reduce the variance. Recently, different ensemble models have been proposed and used widely in numerous practical fields [31–33]. For instance, [34] presented an ensemble framework composed of three models, including Random Forest (RF), Decision Tree (DT), and Gradient Boosted Trees (GBT) for big data time-series. Also, it is worth mentioning that some ensemble models could help to reduce overfitting [35–38].

Several works on weighted ensembles can be found in the literature. The proposed techniques are classified as either constant or dynamic weighting, with [39] being the first to mention using different models for one method, introducing the ensemble learning concept. In the neural network field, Perrone and Cooper [38] introduced two ensemble strategies. By averaging the estimates of numerous regression base learners, the Basic Ensemble Method (BEM) integrates them. They show that BEM can lower the squared error of forecasts by a factor of N (estimators' quantity). Furthermore, the Generalized Ensemble Technique (GEM) was introduced as a linear combination of regression base learners, with the premise that this ensemble method will avoid overfitting the data. The researchers employed cross-validation to build the ensemble estimation methods using all of the training data. Their methodology was utilized for image character classification (NIST OCR).

Since then, many techniques have been proposed, such as Bagging and Boosting [40] or Stacking and Voting methods that rely on weights for each model [41, 42]. Whilst Bagging and Boosting are primarily concerned with minimizing standard deviation/variance and bias, Stacking techniques are concerned by determining the best strategy to mix basic learners. These ensembles are built by stacking the weighted average using the weighted average result of different basic learners. In the research of [36], an optimization-based nesting method that discovers the optimum weights to merge basic learners. This was accomplished by employing Bayesian search to produce basic learners and a heuristic model to construct such learners with a specific amount of variety and performance.

Regarding dynamic weighted average time-series ensembles for energy forecast, there are also several researchers utilizing variations of these techniques. An ensemble method based on LSTMs, Support Vector Regression Machine (SVRM), and Extremal Optimization (EO) algorithm is studied by [43], with LSTM forecasts aggregated into a nonlinear-learning regression top-layer composed of SVRM, and the EO is introduced to optimize the top-layer parameters. Finally, fine-tuning the top-layer provides the final ensemble forecast for wind speed. Two case studies are used to test the proposed EnsemLSTM. In the work of [44], the authors rely on dynamic weighted average on seasonality parameters for day-ahead PV power generation prediction using time-series ensemble models.

2.2 Algorithms

This section reports on ML and DL algorithms used.

2.2.1 Machine learning

Decision Tree Regression (DTR) Starting simple, a Decision Tree model was used. This is a supervised learning method that can be used for classification and regression problems in which a decision tree structure is formulated through the repetitive segmentation of the data set. The features of the said data lead to a set of binary decision rules to be followed towards mapping and eventually predicting the value of the target variable. Although, decision trees generally perform more poorly compared to neural networks for nonlinear data, they are easier to understand, interpret and even visualise, and require no normalization of data. For its implementation the Python library scikit-learn was used.

Random Forest (RF) RFs, also known as random decision forests, are an ensemble learning approach for classification, regression, and other applications that operates by training a large number of decision trees. Regarding classification, the RF output is the class determined by the majority of trees. The mean or average forecast of the individual trees is returned for regression tasks [45]. Random decision forests address the issue of decision trees overfitting their training set [46].

Extreme Gradient Boosting (XGBoost) Both capable for regression and classification Extreme Gradient Boosting is a scalable end-to-end tree boosting system (also called XGBoost), which is used widely by data scientists to achieve state of the art results on many ML challenges [47]. It is a novel sparsity-aware algorithm for sparse data and weighted quantile sketch for approximate tree learning. More importantly, its creators provided insights on cache access patterns, data compression and sharding to build a scalable tree boosting system. By combining these insights, XGBoost scales beyond billions of examples using far fewer resources than existing systems.

2.2.2 Deep learning

Simple multi-layer perceptron (MLP) An ANN is composed from a network of linked units or nodes known as artificial neurons, which are generally modeled after the neurons in the human brain. Each link, like synapses in a human brain, has the ability to send a signal to other neurons. An artificial neuron receives a signal, analyses it, and can signal neurons to which it is linked [48–50].

Long short-term memory recurrent neural network (RNN-LSTM) LSTM is an artificial RNN architecture [51] used in the field of DL. Unlike standard feed-forward neural networks, LSTM has feedback connections. It cannot only process single data points (such as images), but also entire sequences of data (such as time-series, speech or video).

Gated recurrent unit recurrent neural network (RNN-GRUs) Gated recurrent units (GRUs) are a gating method in recurrent neural networks first proposed by Kyunghyun Cho et al. in 2014 [52]. The GRU behaves similarly to a LSTM with a forget gate, [53], but with fewer parameters, because it lacks an output gate. GRU outperformed LSTM on specific tasks, such as polyphonic music modeling, speech signal modeling, and natural language processing. GRUs have been demonstrated to perform better on smaller data sets [54].

2.3 Evaluation metrics

In order to compare the aforementioned algorithms and techniques, their results were evaluated using a series of metrics. Their mathematical formulations and brief descriptions follow.

2.3.1 R-squared (R^2)

The coefficient of determination (R^2) constitutes the comparison of the variance of the errors to the variance of the data which is to be modeled. In other words, it describes the proportion of variance 'explained' by the forecasting model and, therefore, unlike the following error-based metrics, the higher its value, the better the fit. It can be calculated as follows (Eq. 1):

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum(y_i - f_i)^2}{\sum(y_i - \overline{y})^2} \tag{1}$$

where SS_{res} is the sum of squares of residuals (errors) and SS_{tot} is the total sum of squares (proportional to the variance of the data), y_i is the actual power load value, $\overline{y}$ is the mean of the actual values and f_i is the forecasted value for the power load.

2.3.2 Mean absolute error (MAE)

The calculation of MAE is relatively simple (Eq. 2), since it only involves summing the absolute values of the errors (which is the difference between the actual and the predicted value) and then dividing the total error by the amount of observations. Unlike other statistic methods, the MAE considers the same weight for all errors.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - f_i| \tag{2}$$

where y_i is the actual and f_i is the forecasted value for the power load and N is the amount of values.

2.3.3 Mean squared error (MSE)

The MSE indicates how good a fit is by calculating the squared difference between the i^{th} observed value and the corresponding model prediction and then finding the average of the set of errors (Eq. 3). The squaring, apart from removing any negative signs, also gives more weight to larger differences. It is clear that the lower the MSE, the better the forecast.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - f_i)^2 \tag{3}$$

where y_i is the actual and f_i is the forecasted value for the power load and N is the amount of values.

2.3.4 Root mean squared error (RMSE)

The RMSE is defined as the square root of the average squared difference of actual value and prediction value—in other words, the square root of MSE. While these two metrics have very similar formulas (Eqs. 3 and 4), the RMSE is more widely used, since it is measured in the same units as the variable in question. According to its mathematical definition, the RMSE applies more weight on larger errors, given that the impact of a single error to the total is proportional to its square and not its magnitude.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - f_i)^2} \tag{4}$$

where y_i is the actual and f_i is the forecasted value for the power load and N is the amount of values.

3 Research design

The goal of this study was to address EGF issues. Our approach was to create ML and DL techniques utilizing and comparing several algorithms. Our final goal was to tune the three best algorithms and use an ensemble method for predictions. There are many examples of tree based models' behaviour near zero values [55, 56] while other cases demonstrate their differences with ANN based models in energy data sets [57]. The implementation was utilized with Python programming language, and libraries such as Pandas [58], Numpy [59], Matplotlib [60], Sklearn [61] and Tensorflow [62].

According to our novel ensemble approach, while each of the three best models has their strengths and weaknesses the best option would be to exploit the advantages and create an ensemble method employing all of them. Analyzing each part of the target data set, the best algorithm or a combination of weighed averages predictions

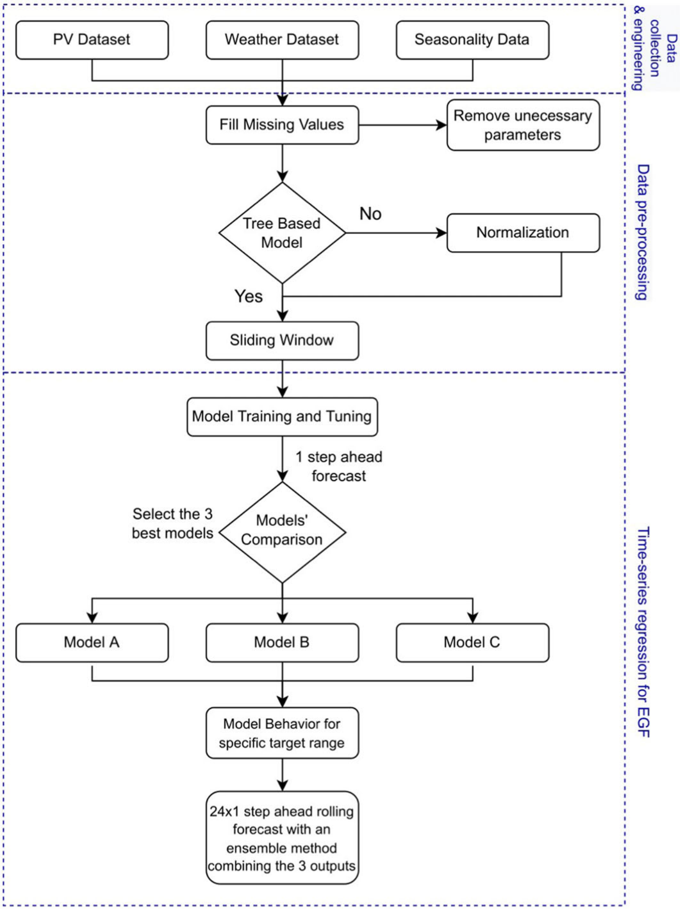

Fig. 1 Overview of research design

was used for creating an ensemble method to boost the performance. For this reason we measured MAE, Standard Deviation (STD) and average values for the predictions of each algorithm for specific ranges of the target parameter and compared them with the real values.

The proposed methodology comprises three phases: (i) Data collection and engineering, (ii) Data pre-processing and (iii) Time-series regression for EGF (Fig. 1).

3.1 Data collection and engineering

We used three different sources for data collection, in order to upgrade our problem from uni-variable to multi-variable.

Table 1 Missing values for weather parameters

Feature	Status
Date time	False
Maximum temperature	True
Minimum temperature	True
Temperature	False
Wind chill	True
Heat index	True
Precipitation	False
Snow	True
Snow depth	True
Wind speed	False
Wind direction	True
Wind gust	True
Visibility	True
Cloud cover	False
Relative humidity	False
Conditions	True

- PV data set: The original forecast target of this research is the power production from a PV Park. The data represent a two-year period, from the beginning of 2019 until the end of 2020. The PV Park is located in the premises of the University of Cyprus (UCY), in Nicosia, Cyprus, which constitutes one of the pilot sites of the project DRIMPAC (see Acknowledgements). The power capacity of the park is around 21 KWp.
- Weather condition data: Weather condition data.[1] The data were gathered for the same period as PV data and represent the relative humidity, temperature, wind speed, cloud cover, precipitation and timestamp per 1 h gap (among other values that were not used).
- Seasonality data. Information produced by the timestamp were utilized, like day of the week, month of the year, day of year, hour of day etc.

3.2 Data pre-processing

The first part of pre-processing was to combine the three data sets to one, by joining them on the timestamp. After, a process has been made for each parameter (specially for the weather data set) for filling the missing values. For the parameters with low percentage of missing values (below 5%) and not more than eight continuous missing timestamps, a forward linear interpolation process was made to fill the gaps. This procedure filled several parameters, while others remained at the same state. Table 1 shows the variables with missing values having 'True', while the filled having 'False'. At this point, the data set was slightly altered, replacing outlier values outside 1–99%

[1] www.visualcrossing.com.

for the total data set with the similar minimum and maximum values that were within this range.

For the forecasting part, the filled parameters (relative humidity, temperature, wind speed, cloud cover, precipitation) were used, while the remaining ones were excluded from the data set. After filling the missing values and before diving to time-series regression techniques, there was a slightly different approach between the tree based and not tree based models, i.e. that the former do not require normalization according to their definition. For the normalization of the latter, the MinMaxScaler function of Python's sklearn module [61] was used, ranging the data set to [0,1].

The populated parameters (relative humidity, temperature, wind speed, cloud cover, precipitation) were utilized for forecasting, whereas the other ones were dropped from the dataset.

3.3 Time-series regression for EGF

For the implementation of the time-series regression, the Sliding Window technique was utilized, involving a 24-step ahead shifting of data, and its outcome constitutes the final data set. For the RNN based models (LSTM, GRU) a three dimensional sliding window (parameters, rows, time-steps) was utilized. Finally for the other tree based and ANN models, a two-dimensional sliding window (parameters + time-steps, rows) was used.

This was then split into training and testing parts (mostly 80–20% respectively—i.e. 19 months for training and 5 months for testing) and the training part was used as an input to all models that were presented in Sect. 2.2. Each of these models was optimized during the training phase by hyperparameter tuning via existing methods, such as grid search and random search, as well as single runs with variations of each model's parameters and/or additions to its structure (for the case of neural network models). The prediction that came from the implementation of each model, was compared with the testing part of the data set and evaluated under the same metrics, also described in Sect. 2.3. The detailed results of the said comparison are presented in the following Section. The analysis continues with the selection of the three best models according to the metrics, and most importantly MAE, R^2 and RMSE, and the determination of the target range when each shows the most accurate results. In this way, the best model for each range was utilized for a rolling forecast that uses the last 24 h and predicts the production for one hour ahead.

Figure 2 shows the utilized regression models including implementations machine and DL approaches. The output of the forecast is used as an input for the next hour, while the next 24 h' weather parameters (relative humidity, temperature, wind speed, cloud cover, precipitation) are predicted by meteorological stations and seasonality parameters (day of the week, month of the year, day of year, hour of day) were easily extracted by analyzing the timestamp.

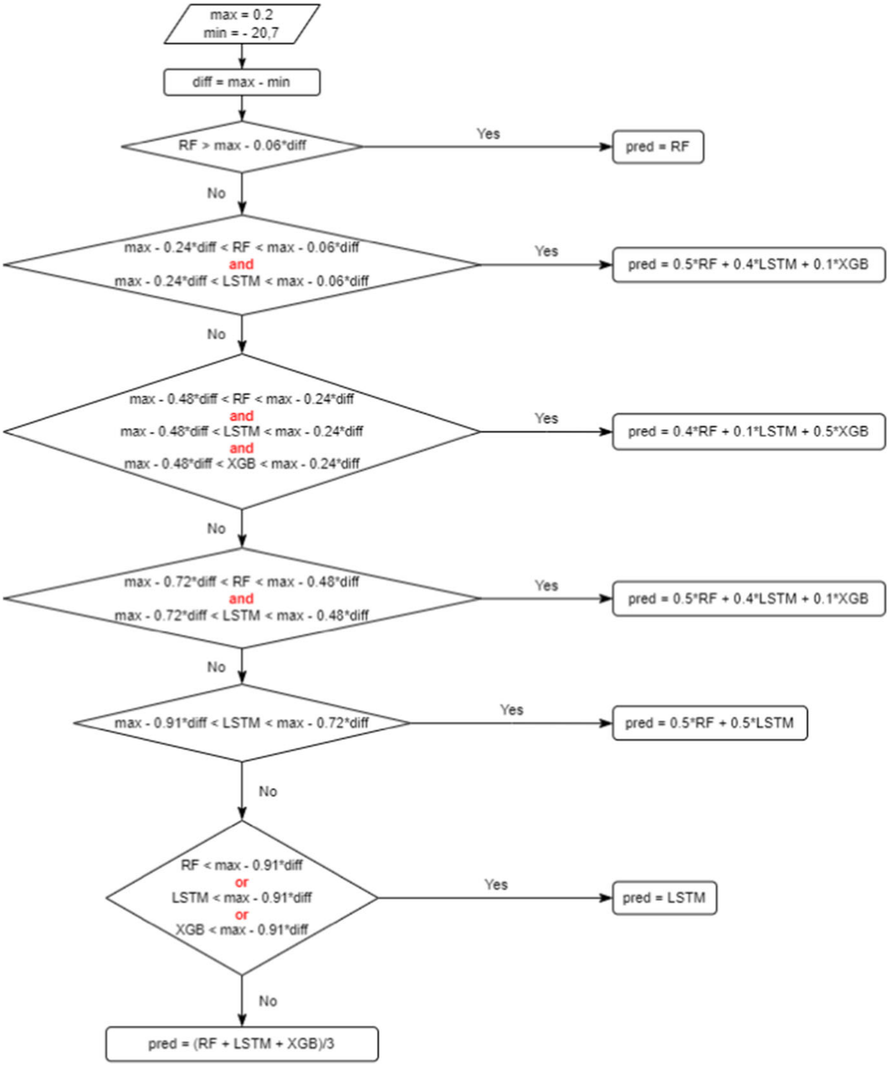

Fig. 2 Ensemble approach

3.4 Algorithmic tuning

This sub-section contains the detailed architecture of each model used for the EGF in the case of PV Park. The best version of each algorithm is presented in Table 2 including the model's configuration and parameters, as well as whether the data have been normalized with Min-Max Normalization prior to each model's implementation.

Furthermore, several attempts to avoid overfitting were utilized, besides the utilization of an ensemble method, that could alleviate this issue (as already mentioned). As far as LSTM and GRU are concerned, an increased 30% Dropout layer was used, while

Table 2 Models' Architecture for PV Park

Model	Configuration	Normalised
Decision Tree	criterion = 'absolute_error', max_depth = 5 max_features ='auto'	No
RF	n_estimators = 400, max_depth = 10, max_features = 'auto' min_samples_split = 2, min_samples_leaf = 4	No
XGBoost	n_estimators = 100, objective = 'reg:squarederror' scale_pos_weight = 1, max_depth=6, verbosity = 0	No
DNN	model.add(tf.keras.layers.Dense(units = 9, activation = 'relu')) model.add(tf.keras.layers.Dense(units = 9, activation = 'relu')) model.add(tf.keras.layers.Dense(units = 1))	Yes
RNN - LSTM	model.add(LSTM(240, activation='relu', input_shape= (trainX.shape[1], trainX.shape[2]), return_sequences=True)) model.add(LSTM(240, activation='relu', return_sequences=False)) model.add(Dropout(0.3)) model.add(Dense(trainY.shape[1])) model.compile(optimizer='adam', loss='mae') model.fit(trainX, trainY, epochs=10, batch_size=16 validation_split=0.1, verbose=1)	Yes
RNN-GRUs	model.add(GRU(60, activation='relu', input_shape= (trainX.shape[1], trainX.shape[2]), return_sequences=True)) model.add(GRU(30, activation='relu', return_sequences=False)) model.add(Dropout(0.3)) model.add(Dense(trainY.shape[1])) model.compile(optimizer='adam', loss='mae') model.fit(trainX, trainY, epochs=10, batch_size=16 validation_split=0.1, verbose=1)	Yes
Ensemble Method	Details in Fig. 2	Depending on the Model

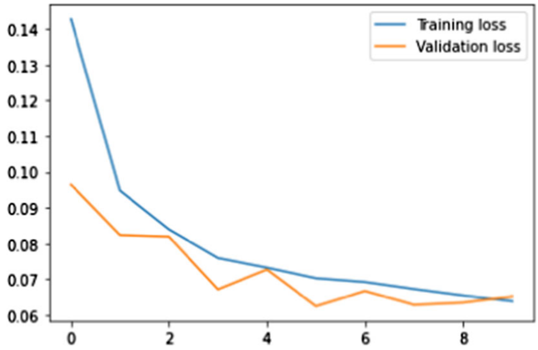

Fig. 3 PV Park Power LSTM train vs validation loss

Table 3 PV Park results

Model	R^2	MAE	MSE	RMSE
Implemented models and results of each metric				
DT	0.8689	0.9471	4.7987	2.1906
RF	0.8955	0.8660	3.8205	1.9546
XGBoost	0.8735	1.043	4.6318	2.1522
DNN	0.8710	1.0491	4.6564	2.1872
RNN-LSTM	0.8894	0.9419	4.0653	2.0163
RNN-GRUs	0.8798	1.0617	4.4192	2.1022
Ensemble Method	0.8913	0.8306	3.9972	1.9993

several numbers of neurons were tested for both high performance and reduced chance for overfitting. Moreover, the comparison of Training Loss versus Validation Loss was almost equal, strengthening the low overfitting argument Fig. 3. For XGBoost, RF and DT, several hyperparameters were tested and utilized like max_depth and n_estimators.

4 Results

This section reports on the EGF results. It presents and elaborates on values of common evaluation metrics R^2, MAE, MSE, RMSE to evaluate the predictive performance of each implemented model. Table 3 reports on metric values for the all fine-tuned forecasting models along with their final parameter tuning.

Table 4 reports on MAE, Standard Deviation (STD) and Average (AVG) values for the all fine-tuned forecasting models along with their final parameter tuning for six different range parts of the data set. Based on the STD PV Park power target parameter range (Fig. 4), a statistical analysis was conducted, separating range of the parameter into six segments. Besides the upper bound (Very High—values near zero and higher than −1) and the lower bound (Very Low—values lower than − 18), the other segments were High [−1, −5), Medium High [−5, −10), Medium Low [−10, −15), Low [−15, −18].

Table 4 Best results on different ranges of the PV Park Power (P) in kW

	$P > -1$	$-1 \geq P > -5$	$-5 \geq P > -10$	$-10 \geq P > -15$	$-15 \geq P > -18$	$P < -18$
Best implemented models and their statistics						
MAE_RF	0.13	1.74	2.11	1.49	1.46	2.56
MAE_XGB	0.25	2.09	2.00	1.72	2.04	3.54
MAE_LSTM	0.34	1.90	2.40	1.51	1.47	2.42
AVG_RF	−0.16	−3.93	−7.95	−12.31	−15.16	−15.85
AVG_XGB	−0.18	−4.31	−8.17	−11.97	−15.07	−14.94
AVG_LSTM	−0.39	−3.84	−7.94	−12.41	−15.48	−16.00
AVG_Real	−0.06	−2.71	−7.41	−12.77	−16.46	−18.41
STD_RF	0.79	3.10	3.24	2.89	2.45	2.37
STD_XGB	1.02	2.76	2.92	2.29	2.82	3.01
STD_LSTM	0.89	3.23	3.46	2.69	2.24	2.35

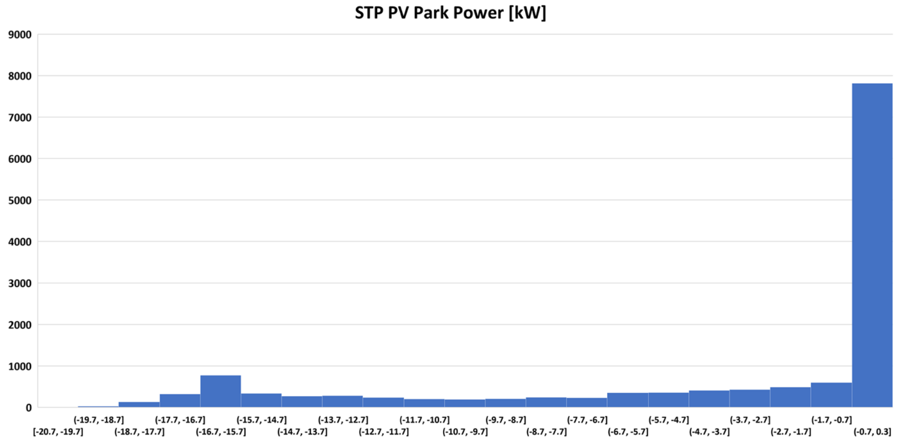

Fig. 4 PV Park Power histogram

As far as Very High segment is concerned, the MAE of RF was by far the lower, followed by XGBoost and LSTM respectively. For High segment, RF was the best algorithm again, but the LSTM was close, leaving the XGBoost behind. On the other, Medium High segment's MAE showed that XGBoost was the best in this part of the data set, followed by RF and LSTM. Regarding the Medium Low and Low parts, RF and LSTM were the best as they scored almost the same MAE. Finally, for the Very Low segment, LSTM was the best algorithm by far, followed by RF and XGBoost.

Based on these results (Table 4), for each part of the data set the best algorithm or a combination of weighed average predictions are used for creating an ensemble method to boost the performance. Furthermore, comparing the real vs predicted average's and standard deviation's values, several assumptions can be highlighted. More specifically, the goal was not only to choose the best MAE for each segment, but also to understand each algorithm's behaviour by average and STD, while identifying the closest to real average and lowest standard deviation on each segment.

Furthermore, utilizing a trial and error approach on each segment, the goal was to identify the best combinations and weights for the multiple weighed average method. Finally, Table 5 and Fig. 5 present the final results of our approach using ensemble with rolling forecast.

5 Conclusion

This paper investigates time-series data analysis in the context of EGF. It examines a sequence of variables with distinct timestamps for improving the performance of EGF. Also, it incorporates R^2, MAE, MSE and RMSE metrics to evaluate each model's predicting capabilities. The conceived forecasting approach includes the integration of multiple ML, DL algorithms and an ensemble method.

The results showed that the most accurate models for our data set were Random Forest, LSTM and XGBoost. However, after investigating each model's individual

Table 5 Ensemble method's results for 24 h ahead with Rolling Forecast

Time-steps	Metrics			
Ahead	R^2	MAE	MSE	RMSE
1	0.894	0.819	3.920	1.977
2	0.849	1.020	5.555	2.350
3	0.814	1.153	6.839	2.606
4	0.795	1.209	7.545	2.737
5	0.781	1.248	8.057	2.827
6	0.773	1.272	8.358	2.880
7	0.769	1.287	8.511	2.907
8	0.768	1.288	8.554	2.913
9	0.768	1.290	8.571	2.915
10	0.767	1.295	8.607	2.920
11	0.765	1.306	8.677	2.933
12	0.763	1.313	8.732	2.943
13	0.761	1.322	8.804	2.956
14	0.760	1.329	8.845	2.963
15	0.760	1.332	8.862	2.966
16	0.760	1.335	8.895	2.971
17	0.759	1.338	8.929	2.977
18	0.759	1.340	8.943	2.979
19	0.758	1.340	8.975	2.984
20	0.757	1.339	9.015	2.990
21	0.756	1.337	9.037	2.994
22	0.754	1.345	9.141	3.012
23	0.751	1.352	9.239	3.030
24	0.750	1.358	9.261	3.035

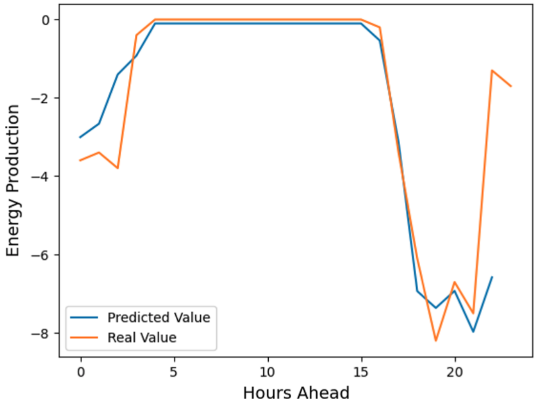

Fig. 5 Rolling forecast 24-h ahead energy production example

behaviour, we identified that each performed better for specific ranges of the target parameter. This observation led us to create a dynamic weighted average ensemble model that in detailed comparison, was more accurate than every other standalone utilized model.

5.1 Implications

Energy and climate policies around the world, are becoming increasingly ambitious over the years since the vital need for a sustainable future has driven the countries worldwide into a series of initiatives to drastically increase energy efficiency and reduce pollutant emissions. These policies aim to reduce Green House Gas (GHG) emissions, moving away from fossil fuels towards cleaner energy, and more specifically, to increase the share of the Renewable Energy Resources (RES). Hence, the penetration of more intermittent distributed generation, would increase both the financial and technical challenges on the existing electrical networks and markets. Briefly, these challenges include unacceptable voltage fluctuations, transformer overloading, energy shortages, electric vehicle charging [63] and increased energy losses in distribution networks.

A very promising solution to overcome these challenges, is the exploitation of the flexibility that can be obtained from small, distributed loads, such as buildings, that may offer Demand Response Services (DRSs). These services, offered by Smart Buildings, enable consumers to monetize their flexibility and use power in a highly efficient and remunerative way, while decarbonizing. DRSs may be classified into two groups: (i) Explicit DR i.e. “incentive based”, refers to consumers choosing to receive direct payments to change their consumption upon request, which is typically triggered by activation of balancing services [64], differences in electricity prices or a constraint on the network. (ii) Implicit DR schemes i.e. “price based”, refers to consumers choosing to be exposed to time-varying electricity prices or/and time-varying network tariffs and react to those price differences depending on their own possibilities and constraints.

5.2 Limitations-challenges

The first obstacle was locating all the necessary data to perform this study. Even though the presented data set illustrated several weather and seasonality parameters, the original set included only the PV Park Power parameter. As a result, the data collection and pre-processing step of finding and combining other parameters proved challenging. The objective was to acquire enough data to represent EGF conditions.

Furthermore, as far as the modeling part was concerned, several other issues existed, like the sliding window timestamp size both for the tree based and the network based models. In addition, many challenges must be tackled to exploit flexibility. For instance, when a building is integrated with generation units, such as PVs, micro-sized Combined Heat and Power (mCHPs) units, or Wind Turbines (WTs) it is critical to additionally predict the total expected generation for various time-frames ahead (day ahead, intra-day, near real time) to determine the total amount of flexibility that the building may offer.

5.3 Future work

This study could be improved by investigating the following aspects. Initially, as for the time-series part, an implementation with the same models and the utilization of the ensemble model with direct forecast strategy and a comprehensive comparison with the existing rolling (recursive) strategy could provide more insights. Another similar idea would be to utilize sequence to sequence RNN models, with the same data set.

Besides the time-series strategies, a zero inflated regression strategy could be modeled. The data set's target parameter (PV Park Power) comprises of high amount of zeros (we used higher weights for Tree based models on that range), and this strategy could be quite helpful for these types of data [65, 66].

Moreover, the suggested framework could be improved by automating the process to the point that it can be used as stand-alone software using only the relevant input data sets. Also, by examining economical and environmental Key Performance Indicators (KPIs) such as energy bills, water bills, purchase records, emissions to the air, emissions to the water, emissions to the land, and resource consumption. Finally, the suggested predictive approach's commercial viewpoint could be enhanced by addressing more practical applications. As an example, financial planning, tariff design, power system operation and electrical grid maintenance, load switching, and infrastructure construction.

Acknowledgements This work is supported by the project DRIMPAC—A unified DR interoperability framework enabling market participation of active energy consumers" funded by the EU's Horizon 2020 research and innovation programme under the Energy Efficiency Innovation Action, Grant Agreement No. 768559.

Funding Open access funding provided by HEAL-Link Greece.

Declarations

Conflict of interest There are no known conflicts of interest other than with staff working at the Information Technologies Institute, Centre for Research & Technology and the School of Science and Technology, International Hellenic University, Greece.

References

1. Lusis P, Khalilpour KR, Andrew L, Liebman A (2017) Short-term residential load forecasting: impact of calendar effects and forecast granularity. Appl Energy 205:654–669. https://doi.org/10.1016/j.apenergy.2017.07.114

2. Koukaras P, Gkaidatzis P, Bezas N, Bragatto T, Carere F, Santori F, Antal M, Tjortjis C, Tzovaras D (2021) A tri-layer optimization framework for day-ahead energy scheduling based on cost and discomfort minimization. Energies 14(12):3599. https://doi.org/10.3390/en14123599
3. Nti I, Asafo-Adjei S, Agyemang M (2019) Predicting monthly electricity demand using soft-computing technique. Int Res J Eng Technol 6:1967–1973
4. Tanaka K, Uchida K, Ogimi K, Goya T, Yona A, Senjyu T, Funabashi T, Kim C-H (2011) Optimal operation by controllable loads based on smart grid topology considering insolation forecasted error. IEEE Trans Smart Grid 2(3):438–444
5. Kudo M, Takeuchi A, Nozaki Y, Endo H, Sumita J (2009) Forecasting electric power generation in a photovoltaic power system for an energy network. Electr Eng Jpn 167(4):16–23
6. Kumar CJ, Veerakumari M (2012) Load forecasting of Andhra Pradesh grid using PSO, DE algorithms. Int J Adv Res Comput Eng Technol
7. Koukaras P, Bezas N, Gkaidatzis P, Ioannidis D, Tzovaras D, Tjortjis C (2021) Introducing a novel approach in one-step ahead energy load forecasting. Sustain Comput Inform Syst 32:100616. https://doi.org/10.1016/j.suscom.2021.100616
8. Yang Y, Wang Z, Gao Y, Wu J, Zhao S, Ding Z (2022) An effective dimensionality reduction approach for short-term load forecasting. Electric Power Syst Res 210:108150. https://doi.org/10.1016/j.epsr.2022.108150
9. Hammad MA, Jereb B, Rosi B, Dragan D et al (2020) Methods and models for electric load forecasting: a comprehensive review. Logist Sustain Transp 11(1):51–76
10. Subbiah SS, Chinnappan J (2022) Deep learning based short term load forecasting with hybrid feature selection. Electric Power Syst Res 210:108065. https://doi.org/10.1016/j.epsr.2022.108065
11. Khwaja AS, Anpalagan A, Naeem M, Venkatesh B (2020) Joint bagged-boosted artificial neural networks: Using ensemble machine learning to improve short-term electricity load forecasting. Electric Power Syst Res 179:106080
12. Gigoni L, Betti A, Crisostomi E, Franco A, Tucci M, Bizzarri F, Mucci D (2017) Day-ahead hourly forecasting of power generation from photovoltaic plants. IEEE Trans Sustain Energy 9(2):831–842
13. Liu L, Zhan M, Bai Y (2019) A recursive ensemble model for forecasting the power output of photovoltaic systems. Solar Energy 189:291–298
14. Visser L, AlSkaif T, Van Sark W (2019) Benchmark analysis of day-ahead solar power forecasting techniques using weather predictions. In: 2019 IEEE 46th photovoltaic specialists conference (PVSC), pp 2111–2116. IEEE
15. Sharadga H, Hajimirza S, Balog RS (2020) Time series forecasting of solar power generation for large-scale photovoltaic plants. Renew Energy 150:797–807
16. Chang GW, Lu H-J (2018) Integrating gray data preprocessor and deep belief network for day-ahead PV power output forecast. IEEE Trans Sustain Energy 11(1):185–194
17. Ojha VK, Abraham A, Snášel V (2017) Metaheuristic design of feedforward neural networks: a review of two decades of research. Eng Appl Artif Intell 60:97–116
18. Pan M, Li C, Gao R, Huang Y, You H, Gu T, Qin F (2020) Photovoltaic power forecasting based on a support vector machine with improved ant colony optimization. J Clean Prod 277:123948
19. Zhou Y, Zhou N, Gong L, Jiang M (2020) Prediction of photovoltaic power output based on similar day analysis, genetic algorithm and extreme learning machine. Energy 204:117894
20. Liu Z-F, Li L-L, Tseng M-L, Lim MK (2020) Prediction short-term photovoltaic power using improved chicken swarm optimizer-extreme learning machine model. J Clean Prod 248:119272
21. Zheng J, Zhang H, Dai Y, Wang B, Zheng T, Liao Q, Liang Y, Zhang F, Song X (2020) Time series prediction for output of multi-region solar power plants. Appl Energy 257:114001
22. Jallal MA, Chabaa S, Zeroual A (2020) A novel deep neural network based on randomly occurring distributed delayed PSO algorithm for monitoring the energy produced by four dual-axis solar trackers. Renew Energy 149:1182–1196
23. Wang Z, Hong T, Piette MA (2020) Building thermal load prediction through shallow machine learning and deep learning. Appl Energy 263:114683
24. Shepero M, Van Der Meer D, Munkhammar J, Widén J (2018) Residential probabilistic load forecasting: a method using gaussian process designed for electric load data. Appl Energy 218:159–172
25. Chu C-SJ (1995) Time series segmentation: a sliding window approach. Inf Sci 85(1):147–173. https://doi.org/10.1016/0020-0255(95)00021-G

26. Vafaeipour M, Rahbari O, Rosen MA, Fazelpour F, Ansarirad P (2014) Application of sliding window technique for prediction of wind velocity time series. Int J Energy Environ Eng 5(2):105. https://doi.org/10.1007/s40095-014-0105-5
27. Mozaffari L, Mozaffari A, Azad NL (2015) Vehicle speed prediction via a sliding-window time series analysis and an evolutionary least learning machine: a case study on san francisco urban roads. Eng Sci Technol Int J 18(2):150–162. https://doi.org/10.1016/j.jestch.2014.11.002
28. Mystakidis A, Stasinos N, Kousis A, Sarlis V, Koukaras P, Rousidis D, Kotsiopoulos I, Tjortjis C (2021) Predicting covid-19 ICU needs using deep learning, XGBoost and random forest regression with the sliding window technique. IEEE Smart Cities, July 2021 Newsletter
29. Xue P, Jiang Y, Zhou Z, Chen X, Fang X, Liu J (2019) Multi-step ahead forecasting of heat load in district heating systems using machine learning algorithms. Energy 188:116085. https://doi.org/10.1016/j.energy.2019.116085
30. Mariet Z, Kuznetsov V (2019) Foundations of sequence-to-sequence modeling for time series. In: Chaudhuri K, Sugiyama M (eds), Proceedings of the 22nd international conference on artificial intelligence and statistics. Proceedings of machine learning research, vol 89, pp 408–417. PMLR, New York. https://proceedings.mlr.press/v89/mariet19a.html
31. Zameer A, Arshad J, Khan A, Raja MAZ (2017) Intelligent and robust prediction of short term wind power using genetic programming based ensemble of neural networks. Energy Convers Manage 134:361–372
32. Chen J, Zeng G-Q, Zhou W, Du W, Lu K-D (2018) Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization. Energy Convers Manag 165:681–695
33. Song G, Dai Q (2017) A novel double deep elms ensemble system for time series forecasting. Knowl Based Syst 134:31–49
34. Ribeiro GT, Mariani VC, dos Santos Coelho L (2019) Enhanced ensemble structures using wavelet neural networks applied to short-term load forecasting. Eng Appl Artif Intell 82:272–281
35. Sollich P, Krogh A (1995) Learning with ensembles: how overfitting can be useful. In: Touretzky D, Mozer MC, Hasselmo M (eds) Advances in neural information processing systems, vol 8, pp 190–196. MIT Press, Cambridge.https://proceedings.neurips.cc/paper/1995/file/1019c8091693ef5c5f55970346633f92-Paper.pdf
36. Shahhosseini M, Hu G, Pham H (2022) Optimizing ensemble weights and hyperparameters of machine learning models for regression problems. Mach Learn Appl 7:100251. https://doi.org/10.1016/j.mlwa.2022.100251
37. Elder J (2018) Chapter 16—the apparent paradox of complexity in ensemble modeling*. In: Nisbet R, Miner G, Yale K (eds) Handbook of statistical analysis and data mining applications (Second Edition), 2nd edn, pp 705–718. Academic Press, Boston. https://doi.org/10.1016/B978-0-12-416632-5.00016-5. https://www.sciencedirect.com/science/article/pii/B9780124166325000165
38. PERRONE MP, COOPER LN (1995) When networks disagree: ensemble methods for hybrid neural networks. Technical report, Providence Ri Inst for Brain and Neural Systems. https://doi.org/10.1142/9789812795885_0025
39. Tukey JW (1977) Some thoughts on clinical trials, especially problems of multiplicity. Science 198(4318):679–684. https://doi.org/10.1126/science.333584
40. Bühlmann P (2012) Bagging, boosting and ensemble methods. Springer, Berlin, pp 985–1022. https://doi.org/10.1007/978-3-642-21551-3_33
41. Ahuja R, Sharma SC (2021) Stacking and voting ensemble methods fusion to evaluate instructor performance in higher education. Int J Inf Technol 13(5):1721–1731. https://doi.org/10.1007/s41870-021-00729-4
42. Sarajcev P, Kunac A, Petrovic G, Despalatovic M (2021) Power system transient stability assessment using stacked autoencoder and voting ensemble. Energies. https://doi.org/10.3390/en14113148
43. Chen J, Zeng G-Q, Zhou W, Du W, Lu K-D (2018) Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization. Energy Convers Manag 165:681–695. https://doi.org/10.1016/j.enconman.2018.03.098
44. Yang D, Dong Z (2018) Operational photovoltaics power forecasting using seasonal time series ensemble. Solar Energy 166:529–541. https://doi.org/10.1016/j.solener.2018.02.011
45. Ho TK (1998) The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844. https://doi.org/10.1109/34.709601

46. Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning: data mining, inference and prediction, 2nd edn. Springer, Berlin. http://www-stat.stanford.edu/~tibs/ElemStatLearn/
47. Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. CoRR arXiv:1603.02754
48. Bengio Y, Lee D, Bornschein, J, Lin Z (2015) Towards biologically plausible deep learning. CoRR arXiv:1502.04156
49. Marblestone AH, Wayne G, Kording KP (2016) Toward an integration of deep learning and neuroscience. Front Comput Neurosci. https://doi.org/10.3389/fncom.2016.00094
50. Olshausen BA, Field DJ (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381(6583):607–609. https://doi.org/10.1038/381607a0
51. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
52. Cho K, van Merrienboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder-decoder approaches
53. Gers FA, Schmidhuber J, Cummins F (1999) Learning to forget: continual prediction with LSTM. In: 1999 9th international conference on artificial neural networks ICANN 99. (Conf. Publ. No. 470), vol 2, pp 850–8552 . https://doi.org/10.1049/cp:19991218
54. Gruber N, Jockisch A (2020) Are GRU cells more specific and LSTM cells more sensitive in motive classification of text? Front Artif Intell. https://doi.org/10.3389/frai.2020.00040
55. Lee S-K, Jin S (2006) Decision tree approaches for zero-inflated count data. J Appl Stat 33(8):853–865. https://doi.org/10.1080/02664760600743613
56. Guikema SD, Quiring SM (2012) Hybrid data mining-regression for infrastructure risk assessment based on zero-inflated data. Reliab Eng Syst Saf 99:178–182. https://doi.org/10.1016/j.ress.2011.10.012
57. Ahmad MW, Mourshed M, Rezgui Y (2017) Trees versus neurons: comparison between random forest and ANN for high-resolution prediction of building energy consumption. Energy Build 147:77–89. https://doi.org/10.1016/j.enbuild.2017.04.038
58. Wes McKinney: data structures for statistical computing in Python. In: van der Walt S, Millman J (eds) Proceedings of the 9th Python in science conference, pp 56–61 (2010). https://doi.org/10.25080/Majora-92bf1922-00a
59. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, del Río JF, Wiebe M, Peterson P, Gérard-Marchant P, Sheppard K, Reddy T, Weckesser W, Abbasi H, Gohlke C, Oliphant TE (2020) Array programming with NumPy. Nature 585(7825):357–362. https://doi.org/10.1038/s41586-020-2649-2
60. Hunter JD (2007) Matplotlib: a 2d graphics environment. Comput Sci Eng 9(3):90–95. https://doi.org/10.1109/MCSE.2007.55
61. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
62. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: large-scale machine learning on heterogeneous systems. Software available from tensorflow.org . http://tensorflow.org/
63. Lamedica R, Maccioni M, Ruvio A, Timar TG, Carere F, Sammartino E, Ferrazza D (2022) A methodology to reach high power factor during multiple EVS charging. Electric Power Syst Res 210:108063. https://doi.org/10.1016/j.epsr.2022.108063
64. Koukaras P, Tjortjis C, Gkaidatzis P, Bezas N, Ioannidis D, Tzovaras D (2021) An interdisciplinary approach on efficient virtual microgrid to virtual microgrid energy balancing incorporating data preprocessing techniques. Computing. https://doi.org/10.1007/s00607-021-00929-7
65. Lambert D (1992) Zero-inflated Poisson regression, with an application to defects in manufacturing. Technometrics 34(1):1–14. https://doi.org/10.1080/00401706.1992.10485228
66. Cheung YB (2002) Zero-inflated models for regression analysis of count data: a study of growth and development. Stat Med 21(10):1461–1469. https://doi.org/10.1002/sim.1088

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Aristeidis Mystakidis[1,2] · Evangelia Ntozi[1] · Konstantinos Afentoulis[1] · Paraskevas Koukaras[1,2] · Paschalis Gkaidatzis[1] · Dimosthenis Ioannidis[1] · Christos Tjortjis[1,2] · Dimitrios Tzovaras[1]

Aristeidis Mystakidis
a.mystakidis@iti.gr; a.mystakidis@ihu.edu.gr

Evangelia Ntozi
ntozevan@iti.gr

Konstantinos Afentoulis
afentoul@iti.gr

Paraskevas Koukaras
p.koukaras@iti.gr; p.koukaras@ihu.edu.gr

Paschalis Gkaidatzis
pgkaidat@iti.gr

Dimosthenis Ioannidis
djoannid@iti.gr

Dimitrios Tzovaras
dimitrios.tzovaras@iti.gr

1 Information Technologies Institute, Centre for Research and Technology, 6th km Charilaou-Thermi, 57001 Thessaloniki, Macedonia, Greece

2 School of Science and Technology, International Hellenic University, 14th km Thessaloniki-Moudania, 57001 Thessaloniki, Macedonia, Greece