Energy Informatics

RESEARCH Open Access

Short-term forecasting of German generation-based CO_2 emission factors using parametric and non-parametric time series models

Adrian Ostermann[1,2*], Arian Bajrami[2] and Alexander Bogensperger[2]

*Correspondence:
adrian.ostermann@tum.de

[1] School of Engineering and Design, Technische Universität München (TUM), Munich, Germany
[2] FfE (Forschungsstelle Für Energiewirtschaft E.V.), Munich, Germany

Abstract

This study focuses on forecasting German generation-based CO_2 emission factors to develop accurate prediction models, which help to shift flexible loads in time with low emissions. While most existing research relies on point forecasts to predict CO_2 emission factors, the presented methods are utilized to perform interval forecasts. In addition, compared to other studies, recent data that extends over a long period is used. The study describes the used data and discusses the concept of walk-forward validation. Further, various models are employed and tuned to forecast the emission factors, including benchmark, parametric (e.g., SARIMAX), and non-parametric (bagging, random forest, gradient boosting, CNN, LSTM, MLP) models. The study reveals that all applied parametric and non-parametric models yield better results than the benchmark models, while the gradient boosting model has the lowest mean absolute error with 40.66 gCO_2/kWh, the lowest mean absolute percentage error 8.17%, and the random forest has the lowest root mean square error with 57.61 gCO_2/kWh. However, the potential of the deep learning models was not fully exploited. In a live application, the implementation effort should be evaluated against the benefit of better prediction.

Keywords: CO_2 emission factor, Forecasting, Short-term, Time series model, Parametric, Non-parametric, Machine learning, Deep learning

Introduction

One substantial risk to humanity is global warming, which already impacts our society, the economy, and the environment by causing environmental disasters, including escalating sea levels, floods, and droughts (Hosseini et al. 2019). Carbon dioxide (CO_2) released into the atmosphere is one of global warming's primary drivers (Hosseini et al. 2019). From 1880 to 2020, the earth's average surface temperature grew by 1.2 degrees Celsius, and by the end of the twenty-first century, it is expected to rise by 5.7 degrees Celsius (Umweltbundesamt. 2023). Thus, the only way to stop the temperature from increasing between 1.4 and 2.4 degrees Celsius is to minimize CO_2 and other greenhouse

gas (GHG) emissions through ambitious climate protection policies (Umweltbundesamt. 2023). The Paris Agreement aspires to restrict global warming below 2 degrees Celsius, ideally to 1.5 degrees Celsius, relative to pre-industrial levels. The European Union (EU) intends to achieve climate neutrality by 2050 (United Nations 2015). Therefore, the EU must invest in green technologies while improving energy efficiency. Compared to 1990, the European electrical sector's emissions decreased by 39% in 2019 (European Environment Agency 2023). Yet, generating power only from renewable sources would not be enough to meet the objective of climate neutrality. Due to their intermittency, location-specific output, uncertainty, and constraints in prediction, a significant percentage of variable renewable energy (VRE: wind and photovoltaic—PV) presents several issues for the energy system (Denholm and Hand 2011; Bistline 2017; Bird et al. 2016; Guerra et al. 2022). The energy system must be flexible to move demand into periods of high renewable energy generation to guarantee supply security, jeopardized by integrating significant shares of renewable energies. By combining the energy-consuming sectors of industry, buildings (heating and cooling), and transportation with the energy-producing sector, sector coupling is required to achieve flexibility, significant GHG emission reduction, and climate neutrality (Duscha et al. 2019; European Environment Agency 2023; Bieker 2023). It is important to note that CO_2 emissions in the following refer to CO_2 equivalent emissions and thus include all greenhouse gas emissions. To minimize CO_2 emissions, flexible consumers (e.g., fuel cells, batteries, heat pumps, electric vehicles, industrial processes) should schedule in times with low CO_2 emission factors of the electricity mix. Therefore, implementing minimal CO_2 emissions flexibility demand schedules requires precise short-term estimates and forecasts of CO_2 emissions factors.

Related research

Many prediction techniques have been developed, as time series forecasting has been a popular study topic. Forecasting methods are often described as statistical or machine learning-based. However, most machine learning algorithms are also statistical since they are based on maximum likelihood estimators (Srivastava et al. 2014). Barker writes that both terms still need to be defined and introduces the terms structured and unstructured (Barker 2020). Further, Athiyarath et al. suggest four distinct forecasting methodologies (Athiyarath et al. 2020): Regression techniques, stochastic approaches, soft computing strategies, and fuzzy logic forecasting. Stochastic methods demand prior knowledge of the forecast's target characteristics. Regression approaches, such as Linear Regression (LR), soft computing methods, such as support vector regression (SVR) and neural networks (NN), and fuzzy logic forecasting, on the other hand, do not call on previous knowledge of the time series and are more data-driven. With increasing data availability and cheaper computing power, these time series forecasting models have become more widespread (Lim and Zohren 2021). Souza et al. categorize the various forecasting techniques as follows: They refer to models that need prior knowledge of the times series as parametric, whereas models that do not require previous knowledge of the data distribution are non-parametric (Parmezan et al. 2019). To distinguish between the various models, we follow the definition of Souza et al. (Parmezan et al. 2019). Statistical approaches like exponential smoothing (ES) and autoregressive integrated moving average (ARIMA) are examples of parametric models. Examples of non-parametric

models are machine learning techniques like a random forest (RF) or a deep learning multi-layer perceptron (MLP). Deep learning models can learn complex data structures without manually engineering features and designing the model (Lim and Zohren 2021; Bengio et al. 2013). Numerous studies applied these methods to forecast CO_2 emissions.

Regarding long-term forecasting, Amarpuri et al. predicted long-term CO_2 emissions in India by applying a deep learning hybrid model of convolution NN and extended short-term memory network (CNN-LSTM) and comparing the results with ES (Twelfth International Conference 2019). Additionally, the proposed approach can predict other pollutant levels, although the model performance can be further increased by considering more parameters in the training set. Hosseini et al. used multiple linear regression (MLR) and multiple polynomial regression (MPR) to predict Iran's CO_2 emissions in 2030 under the assumptions of two scenarios. Both models achieve a coefficient of determination above 0.99, while the residual sum of squares of the MPR is lower. Therefore, the model is more accurate compared to the MLR. Their findings suggest Iran most likely misses its commitment to the Paris Agreement under their business-as-usual assumptions. Ameyaw and Yao forecasted Africa's total CO_2 emissions and West African states with a non-assumption-driven bidirectional LSTM (BiLSTM) model (Ameyaw and Yao 2018). Their proposed technique significantly improved compared to previous International Energy Outlook projections, resulting in a MAPE accuracy for the West African countries above 90%.

The subsequent studies focused on short-term CO_2 emission factors forecasting. Lowry employed a feed-forward NN and a seasonal autoregressive moving average (SARMA) model to forecast day-ahead CO_2 emission factors of the UK power grid (Lowry 2018). The models used by Lowry have the advantage that they are not dependent on collecting multiple exogenous data sets. He points out that either linear autoregressive or non-linear NN models can predict half-hour periods of high carbon intensity. However, in his study, the daily seasonal autoregressive model provided a 20% improvement in carbon reduction. A recent survey from Bodke et al. proposed two decomposition approaches to predict the CO_2 emission of electricity (Bodke et al. 2021). The forecast of the next 48 h enables the scheduling of flexible electricity consumption to minimize CO_2 emissions. They forecasted each time series component separately using either statistical or machine learning models, then merged the predictions for the overall projection. The authors' forecast approach was applied to several European states. The composition of the time series by statistical means into three components leads to the most accurate results for most countries. For France, their novel technique had a 25% lower MAPE than the compared top-performing state-of-the-art model. Leerbeck et al. employed a probabilistic day-ahead model to predict the Danish power grid's average and marginal CO_2 emission factors (Huber et al. 2021). First, they used both a forward selection algorithm and a penalized LR analysis to reduce more than 400 explanatory variables of their dataset to less than 30. The authors combined three LR models into a final model using Softmax weighted average and used an ARIMA for residual correction. Their final ARIMAX model resulted in forecast errors between 0.095 and 0.183 normalized root mean squared error (NRMSE) for the average emissions and 0.029–0.160 for the marginals depending on the forecast horizon (1–24 h). Further, their compound model results in an RMSE of 52.0 for the 24-h forecast on the average emission factor.

A recent study by Huber et al. first derived and forecasted marginal emission factors in Germany for 2017 (Huber et al. 2021). They followed the approach of Hong by employing a three-layer feed-forward MLP to predict the short-term marginal emission factors (Hong xxxx). Since the authors focused on carbon-efficient smart charging, they decided to use an eighth-hour forecast horizon, which covers the average duration of a parking event. The MLP outperformed the naive benchmark models and had a MAPE of 3.23% for a forecast horizon of 2 h on the test set. However, the performance of the MLP approached the naïve models with increasing forecast horizon and resulted in a MAPE of 4.57% for a forecast horizon of 8 h (Hong xxxx).

Motivation and objectives

This section points out some of the problems encountered in the extant research. Previous studies on forecasting CO_2 emission factors have almost exclusively focused on point forecasts. Probabilistic or quantile-based studies on CO_2 emission factors forecasting remain limited. Further, most studies do not include hyperparameter tuning based on walk-forward validation and compare only a few models. Another identified shortcoming is that most studies often use time-limited older data sets. E.g., Huber et al. only cover 1 year of data from 2017 (Huber et al. 2021). However, German energy generation is subjected to a significant transformation towards fluctuating renewable energy, directly affecting the CO_2 emission factors. While Huber et al. point out the advantages of using marginal-based CO_2 emission factors, we use freely available generation-based ones to encourage other researchers to validate and build on our results. To fill this literature gap, we use 3 years of recent and freely available data for Germany in this paper, perform correlation analysis, apply various parametric and non-parametric probabilistic forecast models, and compare the results based on diverse metrics.

Methods

This chapter describes the methodology of how to obtain the ex-ante hourly generation-based emission factors and perform point and quantile forecasts to predict the next 24 h via various models. The historic generation-based emissions factors are calculated according to "CO2 emission factor" section. Real-time market data for different production kinds, day-ahead estimates for specific product types, and other market data are accessible for all European Union nations and bidding zones at the ENTSO-E Transparency Platform (ENTSO-E. 2023). Further, the calculated ex-post emission factors can be accessed on the FfE open data platform (Munich 2023). Furthermore, data analysis and feature correlation analysis are performed to select features. The results are described in "Results" section. Afterward, the data is divided into 70% training, 20% validation, and 10% test sets. The train and validation sets are used for a walk-forward grid search to perform hyperparameter tuning. "Walk-forward validation and testing" section describes this process in detail. Figure 1 shows the approach to obtaining ex-ante generation-based CO_2 emission factors.

Table 1 lists the data to calculate Germany's ex-post generation-based emission factors and perform the forecasts. For this study, we used data starting from 01.01.2019 until 31.12.2022. The final features used for the models are presented in "Results" section.

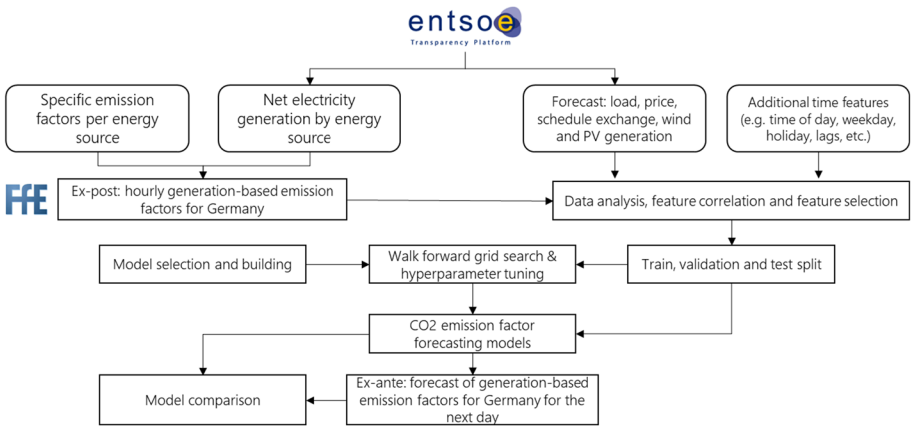

Fig. 1 Methodology for ex-post calculation and ex-ante forecast of generation-based emission factors based on (Fattler 2021)

Table 1 Overview of data and their source

Data	Availability	Source	Resolution
Generation-based emission factors	Real-time; uploaded ex-post to FfE-Platform	FfE	Hourly
Generation per production type	Real-time	ENTSO-E	15 min
Day-ahead aggregated generation	One day ahead, at 6 pm	ENTSO-E	Hourly
Day-ahead price	One day ahead, at 6 pm	ENTSO-E	Hourly
Day-ahead load	One day ahead, at 6 pm	ENTSO-E	15 min
Day-ahead wind and solar generation	One day ahead, at 6 pm	ENTSO-E	15 min
Schedule exchange imports and exports	One day ahead, at 6 pm	ENTSO-E	Hourly

Since the generation-based emission factors are hourly-based, the remaining data is adjusted to the hourly resolution.

CO_2 emission factor

Energy systems without 100% renewable energy produce CO_2 emissions while generating electricity. CO_2 emission factors express the number of emissions in gCO_2/kWh emitted during electricity generation or consumption of an energy unit (Zheng et al. 2015). Fattler describes several ways to calculate CO_2 emissions factors. Besides the marginal emission factors used by Huber et al., two main methods are generation-based and consumption-based (Huber et al. 2021; Fattler 2021). Since we use generation-based emission factors, the following focuses on this method. Marginal emissions are calculated by coupling the day-ahead price with the marginal pricing of all German power plants, allowing the marginal power plant to be selected. This procedure frequently yields a so-called merit order curve. After identifying the marginal power plant, the marginal emission factor may be determined as the ratio of its stoichiometric emission factor to its electric efficiency (Fattler 2021). The consumption-based emission factor considers the demand for power consumption and energy generation (Tranberg et al. 2018). The objective is to factor in national power generation, electricity flow between countries, and age in adjacent countries. A linear system of equations is developed

here, considering electricity output by energy carriers, storage charge and discharge of pumped hydro storage plants, electric load, and energy exports and imports. The linear equation system yields a consumption-based proportion of each production type, resulting in an emission factor based on average consumption (Munich 2023). The generation-based emission factor encompasses the national gross power generation by energy carrier W_{ec} and the related fuel-specific emission factor emf_{ecd} (Fattler 2021). Therefore, the share of each product type of the overall electricity generation $q_{gen,ec}$ is calculated for each point in time by Fattler 2021:

$$q_{gen,ec}\,(t) \;=\; \frac{W_{ec}\,(t)}{\sum_{ec=1}^{m} W_{ec}\,(t)} \tag{1}$$

According to Eq. (2) the average generation-based emission factor $emf_{el,ge,avg}$ can be calculated on an hourly basis by multiplying the share of each production per m energy carriers $q_{gen,ec}$ by its specific emission factor emf_{ec}:

$$emf_{el,gen,avg}\,(t) \;=\; \sum_{ec=1}^{m} q_{gen,ec}\,(t) \;*\; emf_{ec} \tag{2}$$

One limitation is that generation-based emission factors ignore the imported emissions from and exported emissions to neighboring countries resulting from cross-border trades. Additionally, grid losses are not accounted for.

Walk-forward validation and testing

The capacity of a model to generalize or perform effectively on anonymous data is critical in the model selection and validation processes. To evaluate the performance of a model, data is usually split into train and test sets. This allows for preparing the parameter estimation for the model with the train data and comparing the forecasting results against unseen or out-of-sample data based on the test set (Hyndman and Athanasopoulos 2021). Therefore, these predictions will be a reliable proxy for the model's performance in real-world applications. However, Hyndman and Fan outline several remaining issues with this methodology (Ngoc and Phuc 2021). On the test data, for instance, overfitting might still happen. Overfitting may occur if the model is modified to work well on a particular test set but fails to make reliable predictions on additional unobserved data. A third data split, the so-called validation set, is often included to address this problem. In this case, the validation set is evaluated after the models have already been trained on the training set. The validation phase evaluates the generalization capabilities of the various forecasting models in preparation for model optimization and hyperparameter adjustment. A final assessment of the test set is performed following the appearance of success in the experiments. E.g., k-fold cross-validation is well-known and frequently used for validating regression models (Fushiki 2011). K-fold cross-validation divides the available training data into k subsets of about equal size (Geisser 1975). Each set is a validation set for a model trained on the remaining k-1 subsets. Each group is given a chance to be held out. This validation approach is called random cross-validation because the observations in these subgroups are picked randomly without replacements. With time series forecasting, where future timestamp information cannot be used to

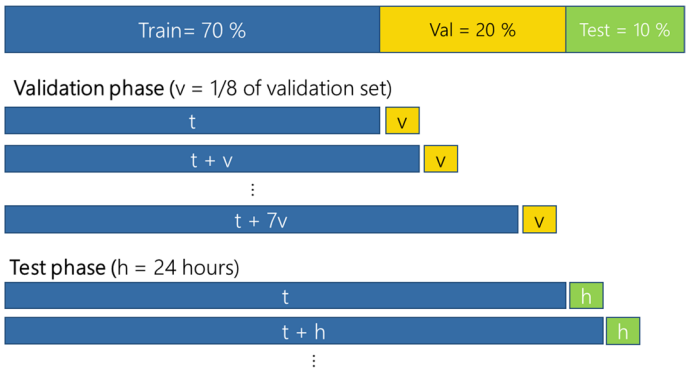

Fig. 2 Modified Walk-forward training, validation, and test phase

predict the past, the split cannot happen randomly. Therefore, Snijders suggests using continuous sections of time series as validation sets (Snijders xxxx). When applied to cross-validation, this implies that instead of randomly picking any k subsets of identical size, the data is divided into k time-continuous blocks of observations (Schnaubelt 2019). Further, two approaches can be distinguished in walk-forward cross-validation: sliding and expanding windows. One uses a fixed size for sliding windows as training and validation data. Training is done on n-data points and prediction validation on the following n-data points, moving or sliding forward the 2n training and validation window in time for the next step. At the same time, expanding windows adds an observation from the future validation set to the training set. It retrains the model after each transfer until all the validation data is part of the training set (Schnaubelt 2019). However, since we want to include all available data in our model and the validation is iterated many times for various models to obtain optimal hyperparameter settings, retraining and validating for every 24-h block would result in high computational costs (Schnaubelt 2019). Therefore, the validation data is split into eight consecutive sets of the validation data size during the validation phase. Afterward, according to the walk-forward methodology, the model is refitted after each prediction. On the other hand, the evaluation on the test set is intended to be utilized just once for each model once it has been tuned, reducing computation costs. Using a forecast horizon of 24 h for the test sets allows for an evaluation of the actual model behavior on unseen data, replicating how the model would behave in a practical application and providing a good indicator of the model's accuracy and uncertainty behavior. Figure 2 displays the walk-forward validation, testing method, and the corresponding split ratios.

Forecasting models

The following section describes the various forecasting models for CO_2 emission factors. First, the benchmark models are briefly described, following the parametric and non-parametric models.

Benchmark models

It is common to compare the performance of forecasting models against baseline or benchmark models (Hyndman and Athanasopoulos 2021; Ahmed et al. 2010). Hyndman and Athanasopoulos recommend several simple models as a baseline (Hyndman and

Athanasopoulos 2021). We use average, simple moving average, naive, and MLR for our analysis. The average model takes the average of the training as its forecast; therefore, the average forecast $y_{avg,t+i}$ for a forecast horizon of 24 h with $i = 1,2,...,24$ is defined as (Hyndman and Athanasopoulos 2021):

$$y_{avg,t+i} = \bar{y} = \frac{1}{n}\sum_{j=1}^{n} y_i \tag{3}$$

with n is equal to the number of training data observations. The simple moving average is obtained by the mean of the previous m data points. Iterative testing derives optimal m, which resulted in $m = 12$. Equation 4 defines the simple moving average $y_{sma,t+i}$ with $i = 1,2,...,24$ *as:*

$$y_{sma,t+i} = \frac{\sum_{j=0}^{m-1} y_{t-j}}{m} \tag{4}$$

The naïve model $y_{naive,t+i}$ with $i = 1,2,...,24$ is equal to the last 24 observation:

$$y_{naive,t+i} = y_{t-h+1} \tag{5}$$

whereas $h = 24$.

Parametric models

Souza et al. state that parametric methods oblige prior information about the data to develop these models (Parmezan et al. 2019). The individual models depend on a set of parameters that, in turn, rely on the characteristics of a given time series. Methods that employ this strategy utilize statistical techniques. Exponential smoothing techniques and linear and non-linear Autoregressive (AR) models are standard statistical forecasting models. Typical examples of statistical forecasting models include linear and non-linear Autoregressive (AR) models and exponential smoothing methods.

Exponential smoothing The exponential smoothing approach has been extensively studied in literature as a stochastic forecasting model. Initially introduced by Brown (Brown and Meyer 1961), the central concept behind the exponential smoothing method is to reduce the noise in past observations of the original time series and use this adjusted time series to make predictions about future values. This approach enables the generation of improved forecasts by smoothing out irregularities such as long-term trends and random fluctuations within the time series. Therefore, Souza et al. define exponential smoothing for time series with n observations $y_1,...,y_n$ without trend and seasonality, where α is the smoothing weight for each observation in time can be defined as:

$$L_t = \propto y_t + (1 - \propto) L_{t-1} \tag{6}$$

L_t refers to an exponential smoothed value at the time t, also known as the current level. The exponential smoothing method can be modified for time series with trend and seasonality components. The Holt-Winters Exponential Smoothing (HWES) technique expands the original exponential smoothing that adds three unique smoothing constants related to the makeup of the time series (Winters 1960). The approach

can be divided into a multiplicative and an additive algorithm, like the time series decomposition. We apply the statespace formulation by Hyndman et al. for HWES to produce a probabilistic forecast with appropriate prediction intervals (Hyndman et al. 2002). The prediction interval is 95% for this forecast and all ensuing ones. Using the statespace formulation, the model generates a thousand predictions for each forecast period. The prediction interval of 95% is obtained using the corresponding quantiles of 0.025 and 0.975.

Autoregressive methods The so-called Box Jenkins methodology, created in 1970 by Box and Jenkins, uses statistical models for time series research, such as forecasting future values of the time-dependent data (Cipra 2020). The autocorrelation analysis to ascertain the characteristics of the time series is one of the primary tools of this methodology. The linear forecasting method based on the Autoregressive Moving Average (ARMA) process is one of the most well-liked models of the Box Jenkins methodology (Brockwell and Davis 2016). The Moving Average (MA) and the AR components of ARMA models are used to model linear and stationary time series (Brockwell and Davis 2016). Cipra defines the MA as the following (Cipra 2020):

$$y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q} = \theta(B)\,\varepsilon_t \tag{7}$$

Here, the value of a time series at time t is described as a linear combination of the white noise ε of its present and past q observation (Montgomery et al. 2016). Additionally, θ is the q weights with values between 0 and 1, which are applied to each previous random error value ε. B denotes the backward shift operator and is defined as $B_{yt} = y_{t\text{-}1}$. Cipra (Cipra 2020) states that an MA model can only be used with linear and stationary time series. The same is true of the AR procedure. The observation is treated here as a linear mixture of its p-lagged values at time t:

$$y_t = \varphi_1 y_{t-1} + \ldots + \varphi_p y_{t-p} + \varepsilon_t \tag{8}$$

that is:

$$y_t - \varphi_1 y_{t-1} - \ldots - \varphi_p y_{t-p} = \varphi(B)\, y_t = \varepsilon_t \tag{9}$$

The weights φ given to the lagged values of the AR model of order p are comparable to those used in the MA process. The autoregressive technique is also rewritten more compactly using the backward shift operator. To generate a mixed *ARMA(p, q)* model of order *(p, q)*, the *AR(p)* and *MA(q)* processes are frequently coupled. The ARMA model can be defined as follows, according to Cipra (Cipra 2020):

$$y_t = \varphi_1 y_{t-1} + \ldots + \varphi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + .. + \theta_q \varepsilon_{t-q} \tag{10}$$

that is:

$$\varphi(B)\, y_t = \theta(B)\, \varepsilon_t \tag{11}$$

The weights θ and φ are assigned to both p lagged values y_t and q random errors values ε. The ARMA process is ideally suited to dealing with time series that are linear and stationary, but it cannot account for non-stationary phenomena (Brockwell and

Davis 2016). To cope with nonstationary time series induced by a trend in the dataset, time series differences are calculated until the differenced time series is proven stationary (Montgomery et al. 2016). The following $y_t - y_{t-1} = (1-B)y_t$ defines the first difference of a time series, while higher order differences are defined as $(1-B)^d y_t$, where d establishes the number of differences. Repeating differencing to achieve stationarity is called integration and, in combination with the AR and MA models, leads to an Autoregressive Integrated Moving Average (ARIMA) model (Montgomery et al. 2016). According to Montgomery et al. (Montgomery et al. 2016), an *ARIMA(p,d,q)* model with an order of *(p,d,q)* is defined as:

$$\phi(B)(1-B)^d y_t = \theta(B)\varepsilon_t \tag{12}$$

The backward shift operator B summarises the lags of the differenced time series and its random error components. ARIMA models are capable of mapping non-stationary time series with an existing trend. However, they cannot capture any seasonality (Brockwell and Davis 2016). To deal with seasonality, additional lagged and random error values of order P and Q are added to ARIMA models, depending on the season period of the time series s. Furthermore, adding a second differencing process $(1-B^s)^D$, where D denotes the order of the seasonal differencing, gives a Seasonal Autoregressive Integrated Moving Average (SARIMA) model. Montgomery et al. (Montgomery et al. 2016) define a SARIMA model as follows:

$$\phi(B)\Phi\left(B^s\right)(1-B)^d\left(1-B^s\right)^D y_t = \theta(B)\Theta\left(B^s\right)\varepsilon_t \tag{13}$$

where Θ and Φ represent additional weights assigned to the seasonal lagged values and random errors; further, when taking n exogenous variables into account, defined at each time step t, denoted by x^i_t for $I \leq n$, with coefficients β_i, the model is called SARIMAX. To find the optimal configuration of the autoregressive models, we use pmdarima's autoarima function (Smith 2023). For the information criteria, we use Akaike's information criterion (AIC) with a seasonal period of 24. The best model without taking into account exogenous variables according to the autoarima function that minimized the AIC is ARIMA(3,0,1)(2,0,0)(ENTSO-E. 2023). Taking into account exogenous variables, the autoarima function minimized the AIC for ARIMA(2,0,1)(2,0,2)(ENTSO-E. 2023). However, adding exogenous variables increased the fitting time by approximately forty times. We use the statsmodels API to construct a distributed forecast that includes the upper and lower quantiles of the 95% prediction interval and the mean of the prediction (Seabold and Statsmodels 2010).

Linear regression LR investigates the relationship between an outcome or response variable $y(t)$ and one or more given predictor variables $x(t) = (x_1(t), \ldots, x_p(t))^T$. Therefore, according to Olive (Olive 2017), the MLR is defined as:

$$y(t) = x(t)\,b + e \tag{14}$$

where e is the error and b are the LR coefficients calculated by the least squares algorithm (Olive 2017). LR is a simple algorithm that is easy to understand. It is, however, susceptible to outliers and performs poorly when the connection between independent

and dependent variables is nonlinear. It also presupposes that the input characteristics are independent of one another. To produce a valid model, multicollinearity in input characteristics must be managed appropriately (Rebala et al. 2019). Like the benchmark models, the LR is trained on 90% of the data since the validation phase is skipped. The MLR is implemented using the scikit-learn API (Pedregosa et al. 2012).

Non-parametric models

Machine learning techniques are regarded as non-parametric since they do not rely on parameters that depend on the statistical characteristics of the provided time series, in contrast to the parametric methods outlined in the preceding section (Parmezan et al. 2019). Time series forecasting using machine learning algorithms has recently yielded encouraging results (Athiyarath et al. 2020; Parmezan et al. 2019).

Ensemble models A growing trend in machine learning is the use of ensemble learning methods. Ensemble learning's fundamental premise is to mix several learning algorithms to improve the ensemble prediction's overall accuracy (Bühlmann xxxx). These techniques can lower the risk of overfitting by utilizing numerous base models. Overfitting is used to characterize a common issue with machine learning algorithms when a given model's estimations predict outcomes well when applied to known data but fall short when applied to unseen data (Liu 2000).

Bagging ensemble Bootstrap aggregation, or bagging as it is commonly known, was first proposed by Breiman in 1996 (Breiman 1996). Bagging creates numerous resampled datasets by randomly selecting data points from a single dataset. Additionally, data points are drawn using bootstrapping with replacement, guaranteeing that the resampled datasets are the same size as the original dataset (Marriott et al. 1995). This is to separate data to achieve the most significant information gain or reduce impurity in data. Following that, the bootstrapped data are trained using the same essential learning method, which produces a set of distinct predictors based on the number of bootstraps. The predictions made by the many base learners are then combined to get the final ensemble estimate. The decision tree is the most common option for the bagging method's underlying algorithm (Tsay and Chen 2019). A decision tree begins with a root node and progresses through a branched tree to a leaf node containing the algorithm's prediction (Tsay and Chen 2019). Table 2 lists the hyperparameter used by the grid search during the validation phase and the final hyperparameter configuration.

The bagging model's dispersed forecast is derived from the estimates of the several models that make up the ensemble. The bagging ensemble's estimators each offer

Table 2 Hyperparameter for bagging ensemble grid search

Hyperparameter	Space	Final
Base model	Decision tree	Decision tree
Number of estimators	250, 500, 750, 1000	750
Maximal samples	0.75, 1.0	0.75
Bootstrapping	True, False	True

a forecast for one forecast horizon. The average, upper, and lower quintiles are then determined.

Random forest A RF is an ensemble learning technique that, like the bagging method, bootstraps on the original dataset and employs decision trees as its foundational algorithm (Breiman 2001). This bagging strategy and random feature selection are combined by RFs (Breiman 2001). Each tree is generated from the resampled data using random feature selection after the algorithm generates each training set from the original training data with replacement. To ensure that each decision tree is constructed from newly produced datasets of the same size, the number of randomly picked trees is fixed before the trees are grown (Breiman 2001). The idea behind employing a collection of trees for prediction is to combine their predictions to reduce each tree's instability problem. Table 3 lists the hyperparameter used by the grid search during the validation phase and the final hyperparameter configuration.

As the RF performs best with estimators of 1000, the question arises if further increasing the hyperparameter would improve the model performance. Therefore, we tried to increase the number of estimators to 1250 and 1500. However, the performance increased only marginally, but the computation time increased considerably, which is why we kept 1000. Further, increasing the number of estimators bears the risk of overfitting. We did the same with the parameter *minimal sample leafs,* which did not improve performance.

Gradient boosting Boosting is another ensemble strategy that employs another way to achieve diversity from a single base learner. According to Freund and Schapire (Freund and Schapire xxxx), promoting is repeatedly running weak learning algorithms on distinct training data distributions and then merging the predictions made by each weak learner into a single prediction. A weak learner is a prediction method anticipated to generate inaccurate predictions. A decision tree with a modest depth, i.e., a small number of leaves, is an example of a poor learner, as explored by Freund and Schapire (Freund and Schapire xxxx). The boosting procedure is meant to minimize the mistakes generated by this type of model considerably. One well-known adaption of the boosting method is the Gradient Boosting (GradBoost) algorithm. Table 4 lists the hyperparameter used by the grid search during the validation phase and the final hyperparameter configuration. Additionally, we tried to vary the learning rate from 0.275 to 0.325 with increased estimators from 400 to 450. Furthermore. We increased the learning rate to 0.5 with 500 and 1,000 estimators. However, the first results showed that with these configurations, the

Table 3 Hyperparameter for RF grid search

Hyperparameter	Space	Final
Maximal depth	50, 100	50
Number of estimators	500, 750, 1000	1000
Maximal features	Sqrt, log2	Sqrt
Minimal sample leafs	1	1

Table 4 Hyperparameter for gradient boosting grid search

Hyperparameter	Space	Final
Learning rate	0.3	0.3
Number of estimators	50, 100, 150, 200, 250, 300, 350	350
Maximal depth	3	3
Minimal sample split	2	2
Subsample	1.0	1.0

validation score increased, therefore, we should have included these values in the final grid search.

As opposed to the prior bagging and RF models, GradBoost sequentially builds decision trees to improve the error of the previous tree by using gradient descent (Friedman 2002). As a result, the mean and quantiles of the base estimators cannot be used to determine the forecast distribution of the model since they interact with one another. GradBoost is instead applied using Hatalis et al.'s approach for probabilistic forecasting of machine learning models (Hatalis et al. 2017). The authors propose using quantile regression, where machine learning models are trained on a pinball loss function to generate probabilistic forecasts (see "Performance metrics" section). This strategy creates three models, each using the pinball loss as a loss function to estimate the forecast's quantiles.

Deep learning models Deep learning models have resolved various issues, including speech recognition and photo object detection (Dahl et al. 2012; Wang and Raj 2017). However, because some deep learning models directly predict the sequence, designs like feedforward neural networks are not well-suited for forecasting a series sequentially. Such networks ignore the temporal relationships prevalent in time series issues and produce predictions based purely on the current input, regardless of any earlier inputs (Sehovac et al. 2019). Deep learning models consist of neurons where data $x = x_0 \ldots, x_n$ is linearly combined with weights $w = w_0 \ldots, w_{n,}$ and a bias b to account for missing or false information. Afterward, a transformation function f is applied to the sum, and the result y is processed by the next neuron (Wang and Raj 2017). Wang and Raj define the process of a single neuron as follows (Wang and Raj 2017):

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{15}$$

Various activation functions like sigmoid rectified linear unit (RELU) and hyperbolic tangent can be applied (Sharkawy 2020). A one-layer neural network is produced by stacking numerous perceptrons on top of one another. The MLP architecture is created by stacking many one-layer neural networks. One or more layers are concealed in an MLP. According to Wang and Raj, an MLP has the universal approximation property, which allows it to approximate any function (Wang and Raj 2017). This has a cost for an MLP with few hidden layers since it takes more neurons exponentially to satisfy the universal approximation requirement. Instead of employing arbitrary numbers of neurons, the modern approach to that issue is to raise the depth, or the number of hidden

layers, of an MLP for increasingly tricky matters. However, as Hippert et al. point out, smaller MLPs are more robust against overfitting (Hippert et al. 2001). For the MLP, we use the mean squared error as it strongly punishes outliers (see "Performance metrics" section). Further L_2 regularization is added to the loss, as described by Lewkowycz and Gur-Ari (Lewkowycz and Gur-Ari 2020), which prevents overfitting and reduces model complexity. Another regularization technique we used is a standard dropout layer. The standard dropout layer comes after the first dense layer and has a dropout rate of 0.2 since this value performed best in reducing training and validation loss. Additionally, we use a second Monte Carlo (MC) dropout layer introduced by Gal and Ghahramani with a dropout rate of 0.6 (Gal and Ghahramani 2015). By using dropout during test time, the dropout layer calculates the uncertainty of the model. The dropout rate amount is directly related to the prediction interval's size. The higher the dropout rate, the more neurons will be dropped randomly during testing. Since each prediction only uses a small, random portion of the network's neurons, the model will become more uncertain if many neurons are removed. We use the ADAMX optimization introduced by Kingma and Ba (James et al. 2021) and a learning rate of 0.001 for the learning process. The MLP uses a batch size of 16 and a training duration of 100 epochs. The callback function cannot be used to follow the test data because test data is designed to remain unknown during training. To track the model's generalization capabilities using the keras callback function, the final 5% of each training set is not used to train the model. The callback further allows for the restoration of the weights. It applies early stopping, which follows the MSE loss of each validation set for each epoch. It stops training after a particular round of epochs if the validation MSE does not improve. The callback function is 25. Using the MC dropout, the MLP generates a distributed forecast for each prediction. It generates 75 forecasts for a single forecast. A prediction interval of 95% is obtained by averaging the 75 predictions and calculating the 0.025th and 0.975th quantiles.

The discipline of computer vision has typically employed convolutional neural networks (CNNs) to analyze picture datasets. They can also be applied to predict time series (Lim and Zohren 2021). The fundamental idea behind a CNN is to substitute convolutional layers for the fully connected ones mentioned before. The preceding layer is filtered using a kernel window by a convolutional layer, which lowers the model's parameter count. Due to the model's decreased computing complexity and storage demand, this technique is computationally beneficial. The CNN divides the training data into many batches with predetermined batch sizes, just like the MLP model. The convolutional filters then evaluate the features and smooth the data's information. With each convolutional layer, 128 filters are used. The first three convolutional layers' kernel window is set at 3. The final convolutional layer uses a kernel window of 2. Batch normalization layers are added following each convolutional layer. Max-pooling is done after each batch normalizing layer to minimize data dimensionality further. The data block for the max-pooling operations is set to 2, which means the maximum of a feature pair of 2 is calculated. This reduces the computational cost and avoids overfitting by halving the feature dimensionality after each convolutional layer (Lim and Zohren 2021). The last convolutional layer does not employ max-pooling because the output's feature dimensionality is already 1. The data is then changed back to having two dimensions using a flattened layer from Keras (2015). The dropout

layer is paired with the following four dense layers, each with a depth of 512, to allow for probabilistic predictions. The dimensionality of the CO_2 emission factor is determined by a final dense layer that employs a linear activation function. Convolutional and dense layers are subjected to L_2 regularization, and ADAM once more optimizes the MSE loss function with a learning rate of 0.001. Xavier initialization, or Glorot normal initialization as it is known in keras, is used to initialize the weights of the convolutional layers. The Xavier initialization did not enhance those layers' prediction performance on the validation set. Hence, their weights were initially distributed according to a standard normal distribution. RELU performed best on the validation set, the activation function of both the convolutional and dense layers. The dropout rate for the dropout layers is set at 0.5, and the callback function to 25 epochs. The model is trained for 200 epochs for each forecast, with a batch size of 12 for each training step during the walk-forward procedure. The model generates distributed forecasts by creating 75 predictions for a single forecast using the dropout layers.

Recurrent Neural Networks (RNN) are a subclass of neural networks where the connections between the processing units form a directed circle. RNNs, instead of feed-forward networks, may process inputs in any order using internal memory. By retaining the internal memory state, they account for past information. An RNN's computational units each have changeable weights and real-valued activations that change over time. Therefore, historically, RNNs have been used for sequential time series data (Lim and Zohren 2021). However, RNNs encounter specific challenges. They are recognized for having problems with exploding and vanishing gradients and are restricted in their ability to store long-term information in the data (Lim and Zohren 2021). Long Short-Term Memory (LSTMs) attempt to overcome these constraints. LSTMs are a special kind of RNN introduced by Hochreiter and Schmidhuber in 1997 (Hochreiter and Schmidhuber 1997). They address the fundamental issues with RNNs using LSTM cells instead of the typically hidden layers. Different gates that regulate the input flow comprise the cells: the input gate, cell state, forget gate, and output gate. Further, the sigmoid layer, the tanh layer, and the point-wise multiplication procedure are also included. The output of the LSTM is decided using the state components and enables long-term information storage. In our architecture, 1250 cells are applied to the input data by the LSTM layer. Following that, a dropout layer is paired with two dense layers that have linear activation functions. There are 32 layers in the first dense layer and 128 in the second. A non-linear activation of the dense layers leads to more computing expense and model complexity. However, it did not enhance the performance of the validation data. Once more, the dropout is utilized to both regularize the model and enable it to produce a distributed forecast. The output dimensionality is decreased to one using the final linear dense layer. Further, we used ADAM with a learning rate 0.001 for model optimization. The weights of both the LSTM and dense layers are initialized using Xavier initialization. Since the dropout rate is not applied to the LSTM layer but only to linear dense layers, the model architecture is robust to higher dropout rates than the MLP and CNN. Therefore, the dropout rate of the MC dropout layer is set to 0.8. Unlike the MLP and CNN, the LSTM had a less volatile learning behavior on the validation set. Therefore, the callback records the loss of the training data for early stopping. If the MSE does not improve during validation and testing, the callback terminates the training procedure after 15 epochs.

Performance metrics

It is crucial to assess the predictive effectiveness of a chosen model. Forecasting models are often assessed by looking at how well the prediction algorithm works on unobserved data, as Hyndman and Athanasopoulos note (Hyndman and Athanasopoulos 2021). Generally, the error of a time series forecast can be defined as (Hyndman and Athanasopoulos 2021):

$$e_{t+h} = y_{t+h} - \hat{y}_{t+h} \tag{16}$$

where h is the prediction horizon, y_{t+h} defines the test part, and $\hat{y}_{t+h}$ is the model's prediction. Based on the prediction error, the literature provides a variety of various evaluation techniques (Hyndman and Athanasopoulos 2021). Scale-dependent errors fall within the first group of accuracy measurements (Hyndman and Athanasopoulos 2021). The Mean Absolute Error (MAE), one of the most often used error measurements, also is referred to as (Hyndman and Athanasopoulos 2021):

$$\text{MAE} = \frac{1}{n}\sum_{i=0}^{n}|\hat{y}_i - y_i| \tag{17}$$

where n is the length of the forecast. it is simple to grasp and maintains the original unit of the prediction objective, the MAE is frequently used as an assessment metric. Another frequently used metric is the mean squared error (MSE). The MSE measures the average of the squares of the errors and is defined as (Ngoc and Phuc 2021):

$$\text{MSE} = \frac{1}{n}\sum_{i=0}^{n}(\hat{y}_i - y_i)^2 \tag{18}$$

The MSE is always positive, since it takes the square of the Euclidean distance and shifts towards zero with decreasing error. Taking the square root of the MSE yields the Root Mean Squared Error (RMSE) defined as (Hyndman and Athanasopoulos 2021):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}\left(\hat{y}_i - y_i\right)^2} \tag{19}$$

The RMSE penalizes significant prediction mistakes more severely than the MAE (Hyndman and Athanasopoulos 2021). The square root of the MSE is computed to represent the original unit of the prediction target. The scale-dependent error metrics do not allow for a comparison of forecast performance across various datasets since they maintain the unit of the prediction. The solution to this issue is to use unit-free percentage errors. The Mean Average Percentage Error (MAPE) is one of the most popular options for percentage errors (Hyndman and Athanasopoulos 2021):

$$\text{MAPE} = \frac{1}{n}\sum_{i=0}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| * 100\% \tag{20}$$

However, very large or even infinite values result for y_i towards zero, and the metric is asymmetric since negative errors are penalized more than positive ones. Assessing

a time series prediction model's goodness of fit is an additional method for evaluation. According to Witten et al. (James 2021), the ratio between the variance explained by the model and the total variance of the prediction target may be used to assess the quality of fit:

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=0}^{n} \left(\hat{y}_i - y_i\right)^2}{\sum_{i=0}^{n} \left(\bar{y} - y_i\right)^2} \tag{21}$$

where $\bar{y}$ is the mean of the target; this metric is called R^2 and will become 0 if the model forecasts the mean instead of its ideal value of 1. Until now, none of the metrics included how many additional features are used to make a prediction. Each new feature adds complexity to the model, which should be penalized if the predictions do not improve. The adjusted ${\mathrm{R}_{\mathrm{adj}}}^2$ addresses this issue (James 2021):

$$\mathrm{R}^2_{\mathrm{adj}} = 1 - \frac{n-1}{n-k-1}\left(1 - \mathrm{R}^2\right) \tag{22}$$

where k is the number of features.

Up to this point, only a point forecast's accuracy and goodness of fit have been examined. The quantile estimates and the prediction interval that come with a probabilistic forecast must be evaluated. A statistic used to assess the precision of a quantile forecast is the pinball loss function, often known as the quantile loss. An issue is determining how accurate a quantile forecast is. Regarding quantile predictions, the outcome is skewed on design, in contrast to conventional forecasts, where the aim is to make the forecast as near to the observed values as feasible. As a result, the naïve comparison of observations and projections is unsatisfactory. A metric that can be regarded as the precision of a quantile forecasting model is returned by the pinball loss (PL) function (Koenker and Machado 1999). Let τ be the target quantile and $\widehat{q}_{i,\tau}$ the quantile forecast, then the PL_τ which evaluates the upper and lower quantile separately, can be defined as:

$$PL_\tau\left(y_i, \hat{q}_{i,\tau}\right) = \begin{cases} \left(y_i - \hat{q}_{i,\tau}\right)\tau, & if y_i \geq \hat{q}_{i,\tau} \\ \left(\hat{q}_{i,\tau} - y_i\right)(1-\tau), & if \hat{q}_{i,\tau} \geq y_i \end{cases} \tag{23}$$

The lower the PL_τ score, the better the quantile prediction. Another metric to assess probability forecasts is the Interval Score (IS), which considers the width of the prediction interval, also known as sharpness. The IS is typically used with the PL_τ to assess the prediction model's total predictive uncertainty because it cannot adequately characterize the dependability of the prediction interval on its own (Hatalis et al. 2017).

Results

This section briefly analyzes the CO_2 emission factor time series, followed by a feature correlation analysis. Finally, the results of the various models are presented.

Data analysis

For this study, we use hourly data from 01.01.2019 to 31.12.2022. Compared to previous work, this dataset includes several years. The calculated CO_2 emission factor,

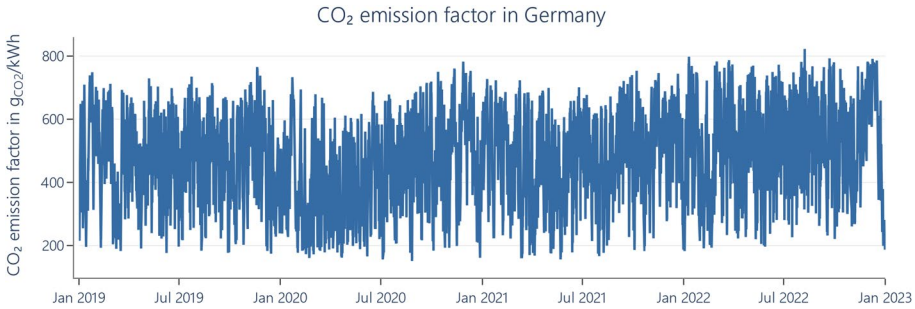

Fig. 3 German generation-based CO_2 emission factor between 01.01.2019 and 31.12.2022

according to "CO2 emission factor" section, over time is presented in Fig. 3. The CO_2 emission factor ranges from around 200 to 800 g_{CO2}/kWh.

First, we test if the time series consists of independent and identical distributed (iid) random variables using the BDS test (after the initials of W. A. Brock, W. Dechert, and J. Scheinkman), as suggested by Tsay and Chen (2019). The BDS statistic results in 35.78, and the p-value is 1.85×10^{-280}, indicating rejection of the null hypothesis since the p-value is close to zero. As a result, it implies that the generation-based CO_2 emission factor is most likely non-linear and does not consist of iid random variables. Further, we test if the time series is stationary using the Augmented Dickey-Fuller (ADF) test. Greene suggests a significance level of 0.05 to evaluate the test statistic (Greene 2003). The ADF statistic equals -15.47, while the p-value is 2.63×10^{-28}. Further, the critical values are -3.43, -2.86, and -2.57 for 1%, 5%, and 10% respectively. The p-value is considerably smaller than the suggested significance level, and the critical values are more significant than the test statistic. Therefore, the time series is most likely stationary. Next, the generation-based emission factor time series is examined to see if it includes a trend and seasonality component. It is possible to do both an additive and a multiplicative decomposition of a given time series. Figures 6, 7, 8 in appendix A show the decomposition graphs. The trend component shows no rise or decrease over time, indicating that the mean and variance are constant, further confirming the findings of the ADF test. The seasonal component shows a clear pattern that repeats over 24 h. The residue plot suggests that random or unforeseen events cause some of the time series. Additionally, we investigate the partial- and autocorrelation of the CO_2 emission factor time series. The perfect positive and negative correlations are denoted by autocorrelation values of 1 and -1, respectively. In contrast, the absence of any correlation between the lagged and current values is represented by a value of 0 (Brockwell and Davis 2016). In appendix A Figs. 9 and 10 show the graphs for the autocorrelation and the partial autocorrelation for the past 120 observations or five days. The autocorrelation between the present observation and the lagged value reduces the further in the past the lags are, and the most recent lags have the most substantial impact on the current observation. This conclusion is further supported by the autocorrelation of all lagged values, where the correlation between the observations slowly decreases to zero. As the partial autocorrelation rises noticeably during a repeating period of 24 h, the time series' daily seasonality is further fostered. The partial autocorrelation also gets less the further back the lagged

values are in time. The statistical models should have an autoregressive order of at least three, as the first three lagged data have the largest partial autocorrelation. The generation-based CO_2 emission factor time series demonstrates non-linear behavior, can be regarded as stationary, and displays seasonal activity, according to the statistical analysis carried out in this chapter.

Features and correlation analysis

As features, we use the data listed in Table 1. Additionally, we use several time features from the time stamps of the time series. Time features are weekday, the hour of the day, and the year. We use one-hot encoding for categorical weekday features to transform them into numerical features. Further, we apply cyclic feature encoding to account for periodic patterns in the time-based features hour of the day and hour of the year. This involves dividing a feature into a sine and cosine part. The problem with cyclical data for machine learning algorithms is the jump discontinuities. Mahajan et al. found that, e.g., LR benefits from using cyclic feature encoding and suffers when using ordinal encoding (Mahajan et al. 2021). Moreover, they discovered that regression trees suffer from the choice of one-hot encoding and might be more robust towards raw cyclical features. Since the offset of 24, 25, and 26 h has the highest autocorrelation of the CO_2 emission factor time series, we choose these three for the following correlation analysis. The best features for the forecasting models can be found with a correlation analysis between the various features and the target variable. A precise forecast will likely result from variables with a high positive or negative correlation with the prediction target (Edwards 1977). The Pearson and Spearman coefficients are two popular methods for assessing the degree to which two variables, x and y, are associated (Van Dongen and Enright 2012). Figure 4 shows the Pearson and Spearman coefficients for each feature.

Figure 2 shows that the Pearson and Spearman correlations between the emission factor and the features are almost identical. As a result, the characteristics and the

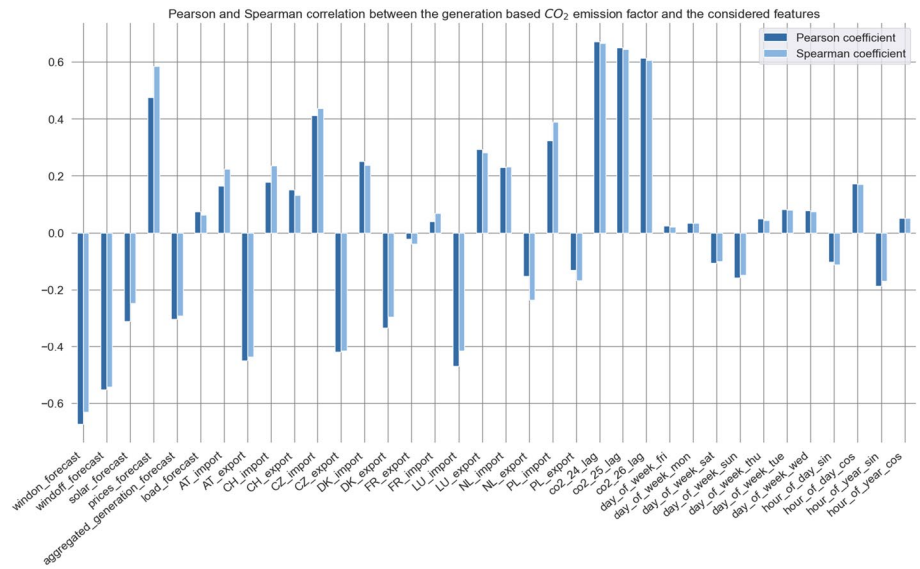

Fig. 4 Pearson and Spearman coefficients between the features and the CO_2 emission factor

prediction target do not generally have any hidden non-linear correlations. Additionally, Figure demonstrates a moderate to strong negative correlation between the forecast for renewable energy and the time series for the CO_2 emission factor. This is a highly reasonable outcome because renewable energy generation should decrease the CO_2 emissions factor as they produce nearly no CO_2. Further, there is only little positive correlation between the load forecast and the emission factor, which is also plausible: The load prediction refers to anticipated energy consumption, which needs to be considered when calculating the generation-based CO_2 emission factor. Next, Figure demonstrates a moderately negative correlation between the forecast of the aggregate generation and the CO_2 emission factor. Therefore, an increase in overall energy production is likely attributable to more renewable energy production, resulting in a drop in the CO_2 emission factor. The merit order in Germany explains the high correlation between the day-ahead price and the CO_2 emission factor. Fossil fuel-based energy generation, such as coal and gas, has higher electricity generation costs than renewable energy sources. Therefore, with higher electricity prices, the CO_2 emission factor rises. The scheduled imports and exports from various European Union nations are also among the features used for the correlation study. The highest correlation is found in the exports and imports between Germany and the Czech Republic and the German CO_2 emission factor. Increased renewable energy production in Germany leads to a low CO_2 emission factor, low prices, and more exports to other countries. On the other hand, low renewable energy production in Germany results in high prices and often more imports from other countries. Further, countries with a moderate to high correlation are Austria, Denmark, Luxembourg, Netherlands, Poland, and Switzerland. On the other hand, France shows almost no link with the CO_2 emission factor. We examine the correlation between the various features for the final feature selection. Leerbeck et al. state that a high correlation between features can lead to co-linearity and poor model performance, e.g., for LR or SARMAX (Leerbeck et al. 2020). The feature correlation analysis reveals a substantial correlation between the lagged features. The lag of 25 has the highest correlation with the lags of 24 and 26 h, with a Spearman and Pearson coefficient of 0.99. Due to their linear dependence, two highly correlated variables can have nearly the same ability to predict the outcome value for an observation. Therefore, we discard the lags 24 and 26. Finally, for the prediction models, all time-related characteristics are retained. The weekday features should be viewed as a single feature (Hyndman and Athanasopoulos 2021). The final feature set consists of 32 features listed in appendix A in Table 6.

Model comparison

This section compares the results of the various forecasting models described in the previous sections. The presented results correspond to the model's performance on the test set. The test set starts on 07.08.2022 at 22:00. It ends on 31.12.2022 at 23:00. The predictions are made at midnight with a 24 h horizon. Figure 5 shows the generation-based CO2 emission factor expost (yellow), the prediction (ex-ante forecast, blue), and the interval (blue channel) of the RF model between 05.11.2022 and 08.11.2022, exemplarily. Initially, the real value decreases compared to the prediction and remains lower. In some points, the real value lies outside the confidence interval. On 06.11.2022, the model predicts two small peaks before and after midday, with the actual value running

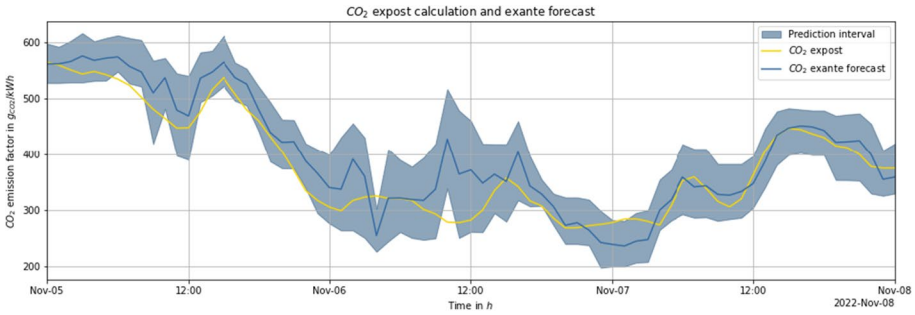

Fig. 5 Excerpts of the prediction on the test data for the RF model

Table 5 Overview of metrics per model

Model	MAE*	MAPE**	RMSE*	PL lower quantile*	PL upper quantile*	IS*	R2**	Adjusted R2**
Avg	137.08	26.73	161.66	–	–	–	−25.88	–
Mov. Avg	72.78	15.15	98.06	–	–	–	53.69	–
Naïve	108.45	23.18	141.81	–	–	–	3.14	–
LR	49.59	10.16	70.03	–	–	–	76.38	-
HWES	69.80	14.06	91.07	4.90	12.13	291.76	60.06	59.69
SARMA	65.03	13.12	93.69	5.68	6.20	360.68	57.72	57.72
SARMAX	48.02	9.16	63.64	**3.40**	6.02	176.43	80.49	80.31
Bagging	41.78	8.58	61.19	4.69	4.37	190.71	81.97	81.8
RF	42.70	8.62	**57.61**	4.29	**3.16**	259.66	**84.01**	**83.87**
GradBoost	**40.66**	**8.17**	62.10	5.05	5.32	165.13	81.43	81.26
MLP	51.15	10.06	72.10	6.19	19.95	**126.22**	74,96	74,73
CNN	42.40	8.70	64.08	7.48	9.86	129.41	80.22	80.04
LSTM	50.36	9.77	67.71	6.92	11.48	183.52	77.16	76.95

*in gCO_2/kWh, **in %

The value of the best-performing model per metric is highlighted in bold

in the opposite direction at the first peak and the trend only coinciding again after the second peak. From the morning of 07.11.2022 onwards, the courses of the actual value and the prediction match very well.

Table 5 shows the metrics of the different models. The value of the best-performing model per metric is highlighted in bold.

Looking at the MAE, MAPE, and RMSE, all applied models perform better than the three benchmark models: average, simple moving average, and the naïve forecast. According to Hyndman and Athanasopoulos, all models are worth further consideration, as their increased complexity led to a performance improvement compared to the benchmark models (Hyndman and Athanasopoulos 2021). The HWES and SARMA models perform the poorest compared to the benchmark models. Since these two use only data from the time series itself, it can be said that the chosen features have a more remarkable ability to predict future values of the CO_2 emission factor than do the time series' previous values alone. Otherwise, the R2 and adjusted R2 values are close for every model. This implies that the model's capacity to explain variation in the dependent variable beyond what is already captured by the basic model is not considerably improved by including new independent variables. Additionally, for models like HWES

and SARMA, the test data's prediction horizon of 24 steps or 24 h is relatively high. They focus mainly on the most recent time series observations. The model's performance declines if the prediction horizon is longer than the most recent data. Regarding the R2 or goodness of fit, all models using exogenous variables have a significantly higher score than the ones that don't. From the parametric models, the SARMAX performs best. The improvement of the SARMAX over the LR indicates that the residue adjustment was successful (Durbin and Koopman 2012). The MAPE and RMSE achieve better results than the best deep learning model, CNN. The CNN performs better from the three deep learning models than the MLP and the LSTM regarding the MAE, MAPE, and RMSE. CNN is the third-best model on the MAE, the fourth-best on the MAPE, and the fifth-best on the RMSE. The GradBoost model performs best on the MAE and the MAPE with 40.66 gCO_2/kWh and 8.17%, respectively. Huber et al. achieved a MAPE of 4.57% with their MLP model (Huber et al. 2021). However, when comparing our results to those of these older studies, it must be pointed out that they used marginal emission factors, a forecast horizon of 8 h, and data from 2017. Therefore, the results are only comparable to a minimal degree. Regarding the RMSE, the RF has the best result, with 57.61 gCO_2/kWh. A similar results was reached by Leerbeck et al. as their compound model resulted in a RMSE of 52.0 for the 24-h forecast on the average emission factor (Leerbeck et al. 2020). Again, the results are only comparable to a limited extent since they use emission factors of the DK2 zone for 2017. Further, the RF model has the best score for the PL upper quantile, the R2, and the adj. R2 with 3.16 gCO_2/kWh, 84.01% and 83.87% respectively. Since the RMSE punishes outliers more strictly than the MAPE, the applied RF is better suited to deal with outliers. At the same time, the GradBoost performs slightly better on average percentage deviation for the test data. The MLP, CNN, and GradBoost have the lowest IS or sharpest prediction interval. However, their PL is worse for the lower quantile than the other models. While the SARMAX, RF, and Bagging have the lowest PL for the lower quantile, and the RF, Bagging, and GradBoost have the lowest PL for the higher quantile, their IS is worse than the one from the MLP. This shows that the uncertainty of a time series prediction interval depends on a compromise between the sharpness of the prediction interval and the quantile accuracy. Therefore, increasing model uncertainty leads to decreasing accuracy that, in turn, reduces the PL.

Discussion

This section discusses the potential limitations and directions for future work. First, the generation-based emission factor is a relatively simple approach to determining the CO_2 intensity of electricity generation, as described in "CO2 emission factor" section. Therefore, when implementing, e.g., a smart charging scheme for electric vehicles, one should consider using the marginal emission factor. However, using the generation-based emission factor to forecast the time series is legit. Further, we used freely available data as input and our correlation analysis. When we started data collection, we did not find freely available recent weather data. However, the current approach and features allow a real-time prediction of the next 24 h. A representation of the recent forecast for the next 24 h can be found on the FfE website opendata.ffe.de (Ferstl 2023). The forecast is generated at 00:00 for the following day using an RF model. Furthermore, the forecast and actual results can be downloaded. Additional data, such as recent weather

data, could further improve the performance of the models. This might be an issue for future research to explore. We are currently working on a certified method for calculating the CO2 emission factor hourly, which will be part of an exchange platform. The specified hourly emission factors will then be used as a basis for forecasting, particularly for greenhouse gas verification, as is necessary, for example, for green hydrogen production. The feature correlation showed that the German day-ahead price forecast is a valuable feature for this prediction task. However, the course of the price and, thus, the forecast changed significantly towards the end of 2021 due to the effects of the Russian invasion of Ukraine. While the standard deviation of the German day-ahead price in €/MWh (averaged over the year) was 9.0 in 2019 and 9.4 in 2020, it was already 24.5 in 2021 and 57.3 in 2022 (ENTSO-E. 2023; Kern et al. 2022). Especially March, September, and December 2022 have recorded enormous price peaks of around 450, 500, and 700 €/MWh (ENTSO-E. 2023; Kern et al. 2022). It is, therefore, plausible that different validation and test periods (e.g., without 2022) could lead to significantly different results in terms of performance. This may constitute the object of future studies. With increasing amounts of renewable energy, the correlation of electricity price and CO_2 emission factor will increase due to the merit order effect. Therefore, depending on the application and goal, it might be sufficient in the future to solely predict the price or use the price forecast to minimize emissions. Another factor to consider is the aleatoric uncertainty caused by errors in the features utilized for the prediction. If the models rely on (market) data forecasts such as wind and price forecasts, which are prone to inaccuracies, such errors will propagate to the final emission forecast. On the one hand, our adjusted walk-forward helped us to decrease the computational costs. On the other hand, the models are tuned on different or longer prediction horizons than they are tested on. Adjusting the walk-forward length to the same as the test set could result in better model performance. Additionally, we set the MSE for the loss function for the deep learning and the RMSE for the machine learning models. Taking another loss function for the deep learning models might improve performance on the presented metrics. Further, when implementing the three deep learning models, we only had data available until mid-2022. During this phase of our work tests, the LSTM, CNN, and MLP performed slightly poorer than the GradBoost. However, we could not repeatably adjust the deep learning architectures and kept the initial architecture due to limited computational resources. Therefore, the deep learning architectures could be further improved when tuned against the latest data. In conclusion, we do not state that the ensemble models are, per se, better at this task than the deep learning methods. Further investigating different deep learning architectures or state-of-the-art deep learning architecture, such as temporal fusion transformers, might prove critical in future work. In a real-world implementation of CO_2 prediction for a smart charging app, for example, we believe the following things should be considered: In our opinion, the machine learning methods are faster and easier to implement than deep learning methods. Furthermore, the computational effort is lower, but it can be assumed that deep learning architectures achieve better results with careful implementation and tuning. The measure of implementing and maintaining more complex models should always be in proportion to the benefit of a better prediction.

Appendix

See Figs. 6, 7, 8, 9, and 10 and Table 6.

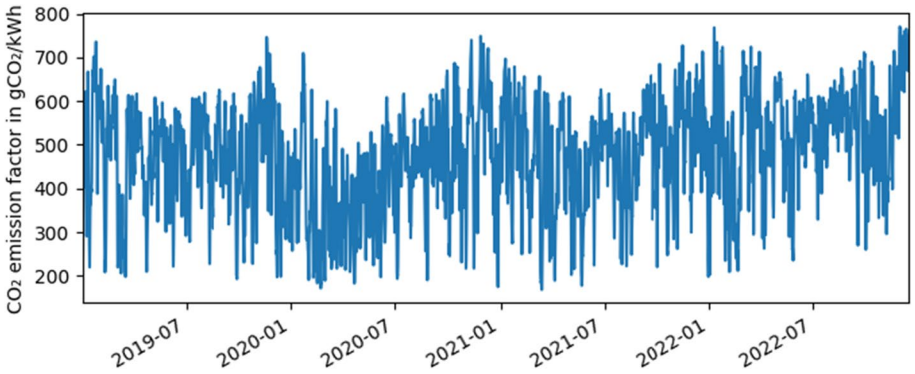

Fig. 6 The multiplicative trend component of the CO_2 emission factor time series between 01.01.2019 and 31.12.2022

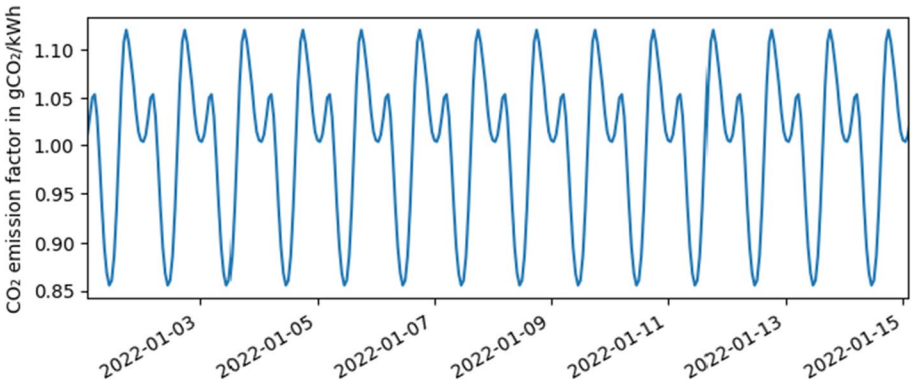

Fig. 7 The multiplicative seasonal component of the CO_2 emission factor time series between 01.01.2022 and 15.01.2022

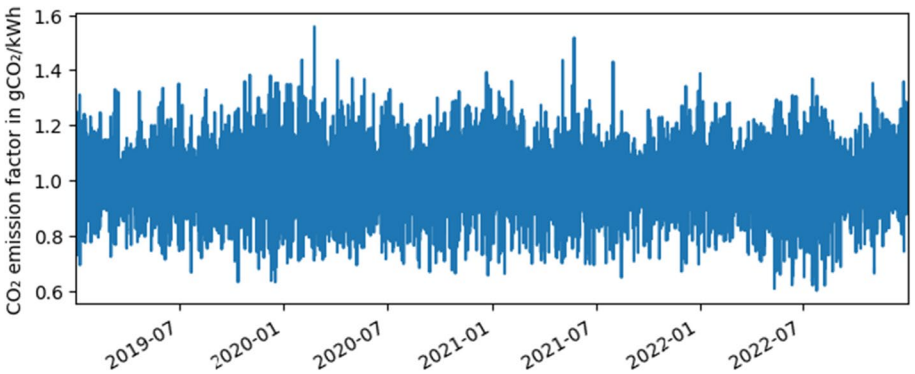

Fig. 8 The multiplicative residue component of the CO_2 emission factor time series between 01.01.2019 and 31.12.2022

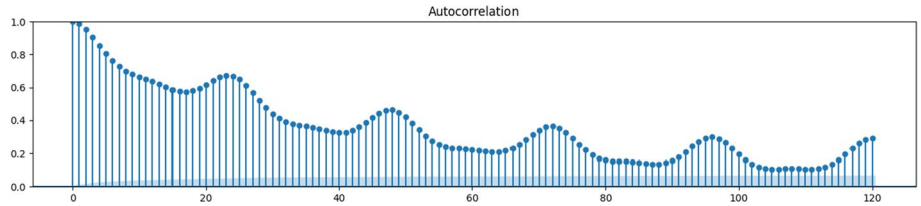

Fig. 9 Autocorrelation of the CO_2 emission factor with 120 lagged values

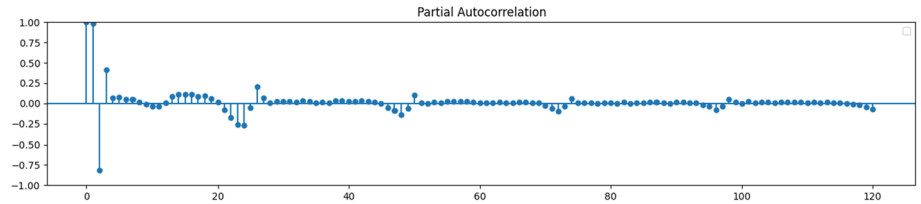

Fig. 10 Partial autocorrelation of the CO_2 emission factor with 120 lagged values

Table 6 Table of final feature selection

Feature name	Feature name
Forecast wind generation onshore	Forecast wind generation offshore
Forecast solar generation	Forecast aggregated generation
Scheduled import from Austria	Scheduled export to Austria
Scheduled import from Czechia	Scheduled export to Czechia
Scheduled import from Denmark	Scheduled export to Denmark
Scheduled import from Luxembourg	Scheduled export to Luxembourg
Scheduled import from Netherlands	Scheduled export to Netherlands
Scheduled import from Poland	Scheduled export to Poland
Scheduled import from Switzerland	Scheduled export to Switzerland
Forecast price	Lag of 24 h
Weekday: Monday	Lag of 26 h
Weekday: Tuesday	Hour of the day (sinus)
Weekday: Wednesday	Hour of the day (cosine)
Weekday: Thursday	Hour of the year (sinus)
Weekday: Friday	Hour of the year (cosine)
Weekday: Saturday	
Weekday: Sunday	

Acknowledgements
We thank our project partners in InDEED and unIT-e^2 for supporting the projects and our work.

Author contributions
Conceptualization, AO, ABo; methodology, AO, ABo and ABa; software, ABa; validation, AO, ABa; formal analysis, AO; investigation, AO, ABa; resources, AO, ABo; writing—original draft preparation, AO, ABa; writing—review and editing, AO, ABo; visualization, AO; supervision, ABo; project administration, AO; funding acquisition, AO, ABo All authors have read and agreed to the published version of the manuscript.

Funding
Open Access funding enabled and organized by Projekt DEAL. The described work was conducted within the projects "InDEED" and "unIT-e^2" by Forschungsstelle für Energiewirtschaft e.V. (FfE). Both projects are funded by the German

Federal Ministry for Economics and Climate Action (BMWK) under the funding codes (InDEED: 03EI6026A and unIT-e^2: 01MV21UN11).

Availability of data and materials
The German generation-based emission factors can be downloaded at this site: https://opendata.ffe.de/dataset/specific-greenhouse-gas-emissions-of-the-electricity-mix/.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests. The funders had no role in the study's design, in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Received: 1 September 2023 Accepted: 1 January 2024
Published online: 10 January 2024

References

Ahmed NK, Atiya AF, Gayar NE, El-Shishiny H (2010) An empirical comparison of machine learning models for time series forecasting. Econom Rev 29:594–621. https://doi.org/10.1080/07474938.2010.481556
Ameyaw B, Yao L (2018) Analyzing the impact of GDP on CO2 emissions and forecasting Africa's total CO2 emissions with non-assumption driven bidirectional long short-term memory. Sustainability 10:3110. https://doi.org/10.3390/su10093110
Athiyarath S, Paul M, Krishnaswamy S (2020) A comparative study and analysis of time series forecasting techniques. SN Comput Sci. https://doi.org/10.1007/s42979-020-00180-5
Barker J (2020) Machine learning in M4: what makes a good unstructured model? Int J Forecast 36:150–155. https://doi.org/10.1016/j.ijforecast.2019.06.001
Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35:1798–1828. https://doi.org/10.1109/TPAMI.2013.50
Georg Bieker. A global comparison of the life-cycle greenhouse gas emissions of combustion engine and electric passenger cars: White Paper. https://theicct.org/sites/default/files/publications/Global-LCA-passenger-cars-jul2021_0.pdf. Accessed 30 Mar 2023).
Bird L, Lew D, Milligan M, Carlini EM, Estanqueiro A, Flynn D, Gomez-Lazaro E, Holttinen H, Menemenlis N, Orths A et al (2016) Wind and solar energy curtailment: a review of international experience. Renew Sustain Energy Rev 65:577–586. https://doi.org/10.1016/j.rser.2016.06.082
Bistline JE (2017) Economic and technical challenges of flexible operations under large-scale variable renewable deployment. Energy Econ 64:363–372. https://doi.org/10.1016/j.eneco.2017.04.012
Bodke N, Tranberg B, Andresen GB (2021) Short-term CO2 emissions forecasting based on decomposition approaches and its impact on electricity market scheduling. Appl Energy 281:116061. https://doi.org/10.1016/j.apenergy.2020.116061
Breiman L (1996) Bagging predictors. Mach Learn 24:123–140. https://doi.org/10.1023/A:1018054314350
Breiman L (2001) Random forests. Mach Learn 45:5–32. https://doi.org/10.1023/A:1010933404324
Brockwell PJ, Davis RA (2016) Introduction to time series and forecasting. Springer International Publishing, Cham
Brown RG, Meyer RF (1961) The fundamental theorem of exponential smoothing. Oper Res 9:673–685. https://doi.org/10.1287/opre.9.5.673
Bühlmann P (2012) Bagging, boosting and ensemble methods. In: Gentle JE, Härdle WK, Mori Y (eds) Handbook of computational statistics. Springer, Berlin, pp 985–1022
Cipra T (2020) Time series in economics and finance. Springer International Publishing, Cham
Dahl GE, Yu D, Deng L, Acero A (2012) Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans Audio Speech Lang Process 20:30–42. https://doi.org/10.1109/TASL.2011.2134090
Denholm P, Hand M (2011) Grid flexibility and storage required to achieve very high penetration of variable renewable electricity. Energy Policy 39:1817–1830. https://doi.org/10.1016/j.enpol.2011.01.019
Durbin J, Koopman SJ (2012) Time series analysis by state space methods. Oxford University Press, Oxford
Duscha V, Wachsmuth J, Eckstein J Pfluger, B (2019) GHG-neutral EU2050 – a scenario of an EU with net-zero greenhouse gas emissions and its implications, on behalf of the German Environment Agency. 2019. Available online: https://www.umweltbundesamt.de/sites/default/files/medien/1410/publikationen/2019-11-26_cc_40-2019_ghg_neutral_eu2050_0.pdf. Accessed 4 Jan 2024
Edwards AL (1977) An introduction to linear regression and correlation. Freeman, San Francisco
ENTSO-E. Transparency Platform. https://transparency.entsoe.eu/. Accessed 30 Mar 2023.
European Environment Agency. EEA greenhouse gases - data viewer. https://www.eea.europa.eu/data-and-maps/data/data-viewers/greenhouse-gases-viewer. Accessed 30 Mar 2023.

European Environment Agency. National emissions reported to the UNFCCC and to the EU Greenhouse Gas Monitoring Mechanism. https://www.eea.europa.eu/data-and-maps/data/national-emissions-reported-to-the-unfccc-and-to-the-eu-greenhouse-gas-monitoring-mechanism-18. Accessed 30 Mar 2023.
Fattler S (2021) Economic and environmental assessment of electric vehicle charging strategies. Dissertation; Technische Universität München, München, Deutschland
Ferstl J (2023) Daily updated specific greenhouse gas emissions of the German electricity mix. http://opendata.ffe.de/daily-updated-specific-greenhouse-gas-emissions-of-the-german-electricity-mix/. Accessed 18 July 2023.
Freund Y, Schapire RE (1996) Experiments with a New Boosting Algorithm. In Proceedings of the Thirteenth International Conference on International Conference on Machine Learning; Morgan Kaufmann Publishers Inc: San Francisco, CA, USA; pp 148–156
Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38:367–378. https://doi.org/10.1016/S0167-9473(01)00065-2
Fushiki T (2011) Estimation of prediction error by using K-fold cross-validation. Stat Comput 21:137–146. https://doi.org/10.1007/s11222-009-9153-8
Gal Y, Ghahramani Z (2015) Dropout as a Bayesian approximation: representing model uncertainty in deep learning. https://arxiv.org/pdf/1506.02142. Accessed 4 Jan 2024
Geisser S (1975) The predictive sample reuse method with applications. J Am Stat Assoc 70:320. https://doi.org/10.2307/2285815
Greene WH (2003) Econometric analysis. Prentice Hall, Upper Saddle River
Guerra K, Haro P, Gutiérrez RE, Gómez-Barea A (2022) Facing the high share of variable renewable energy in the power system: flexibility and stability requirements. Appl Energy 310:118561. https://doi.org/10.1016/j.apenergy.2022.118561
Hatalis K, Lamadrid AJ, Scheinberg K, Kishore S (2017) Smooth pinball neural network for probabilistic forecasting of wind power. http://arxiv.org/pdf/1710.01720v1
Hippert HS, Pedreira CE, Souza RC (2001) Neural networks for short-term load forecasting: a review and evaluation. IEEE Trans Power Syst 16:44–55. https://doi.org/10.1109/59.910780
Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
Hong, T. Short Term Electric Load Forecasting. Dissertation; North Carolina State University, Raleigh, North Carolina, 10.09.2010.
Hosseini SM, Saifoddin A, Shirmohammadi R, Aslani A (2019) Forecasting of CO 2 emissions in Iran based on time series and regression analysis. Energy Rep 5:619–631. https://doi.org/10.1016/j.egyr.2019.05.004
Huber J, Lohmann K, Schmidt M, Weinhardt C (2021) Carbon efficient smart charging using forecasts of marginal emission factors. J Clean Prod 284:124766. https://doi.org/10.1016/j.jclepro.2020.124766
Hyndman R, Athanasopoulos G (2021) Forecasting: principles and practice, 3rd edn. OTexts, Melbourne
Hyndman RJ, Koehler AB, Snyder RD, Grose S (2002) A state space framework for automatic forecasting using exponential smoothing methods. Int J Forecast 18:439–454. https://doi.org/10.1016/S0169-2070(01)00110-8
James G, Witten D, Hastie T, Tibshirani R (2021) An introduction to statistical learning. Springer, New York
Keras FC, GitHub (2015). GitHub repository. https://github.com/fchollet/keras
Kern T, Ganz K, Wasmeier L (2023) Deutsche Strompreise im Jahr 2022 an der Börse EPEX Spot. https://www.ffe.de/veroeffentlichungen/deutsche-strompreise-an-der-boerse-epex-spot-im-jahr-2022/. Accessed 17 July 2023.
Koenker R, Machado JAF (1999) Goodness of fit and related inference processes for quantile regression. J Am Stat Assoc 94:1296–1310. https://doi.org/10.1080/01621459.1999.10473882
Leerbeck K, Bacher P, Junker R, Goranović G, Corradi O, Ebrahimy R, Tveit A, Madsen H (2020) Short-term forecasting of CO2 emission intensity in power grids by machine learning. http://arxiv.org/pdf/2003.05740v1. Accessed 4 Jan 2024
Lewkowycz A, Gur-Ari G (2020) On the training dynamics of deep networks with L_2 regularization
Lim B, Zohren S (2021) Time series forecasting with deep learning: a survey. Phil Trans R Soc A 379:20200209. https://doi.org/10.1098/rsta.2020.0209
Liu Y (2000) Overfitting and forecasting: linear versus non-linear time series models heating, ventilation and air-conditioning. Iowa State University. https://doi.org/10.31274/rtd-180813-15269
Lowry G (2018) Day-ahead forecasting of grid carbon intensity in support of HVAC plant demand response decision-making to reduce carbon emissions. Build Serv Eng Res Technol 39(6):749–760. https://doi.org/10.1177/0143624418774738
Mahajan T, Singh G, Bruns G (2021) An Experimental Assessment of Treatments for Cyclical Data. In: Proceedings of the 2021 Computer Science Conference for CSU Undergraduates, Virtual, 6 March 2021; Available online: https://cscsu-conference.github.io/index.html. Accessed 4 Jan 2024
Marriott P, Efron B, Tibshirani RJ (1995) An Introduction to the Bootstrap. J Royal Stat Soc Ser A 158:347. https://doi.org/10.2307/2983304
Montgomery DC, Jennings CL, Kulahci M (2016) Introduction to Time Series Analysis and Forecasting, Second edition. Wiley, Hoboken, New Jersey
FfE Munich. Daily updated Specific Greenhouse Gas Emissions of the German Electricity Mix Dataset. https://opendata.ffe.de/dataset/specific-greenhouse-gas-emissions-of-the-electricity-mix/. Accessed 30 Mar 2023.
Ngoc TT, van Dai L, Phuc DT (2021) Grid search of multilayer perceptron based on the walk-forward validation methodology. IJECE 11:1742. https://doi.org/10.11591/ijece.v11i2.pp1742-1751
Olive DJ (2017) Linear regression. Springer International Publishing, Cham
Parmezan ARS, Souza VM, Batista GE (2019) Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. Inf Sci 484:302–337. https://doi.org/10.1016/j.ins.2019.01.076
Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Müller A, Nothman J, Louppe G et al (2012) Scikit-learn: Mach Learn Python. https://doi.org/10.48550/arXiv.1201.0490
Rebala G, Ravi A, Churiwala S (2019) An introduction to machine learning. Springer International Publishing, Cham

Schnaubelt M (2019) A comparison of machine learning model validation schemes for non-stationary time series data. FAU Discussion Papers in Economics 11/2019, Nürnberg. http://hdl.handle.net/10419/209136. Accessed 4 Jan 2024
Seabold S, Perktold J (2010) Statsmodels: econometric and statistical modeling with python. SciPy, Austin
Sehovac L, Nesen C, Grolinger K (2019) Forecasting Building Energy Consumption with Deep Learning: A Sequence to Sequence Approach. In 2019 IEEE International Congress on Internet of Things (ICIOT). 2019 IEEE International Congress on Internet of Things (ICIOT), Milan, Italy, 08–13 Jul. 2019; IEEE; pp 108–116, ISBN 978-1-7281-2714-9
Sharkawy A-N (2020) Principle of neural network and its main types: review. J Adv App Comput Math 7:8–19. https://doi.org/10.15377/2409-5761.2020.07.2
Snijders TAB (1988) On cross-validation for predictor evaluation in time series. In: Dijkstra TK (ed) On model uncertainty and its statistical implications. Springer, Berlin, pp 56–69
Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958
Smith TG pmdarima: ARIMA estimators for Python. http://alkaline-ml.com/pmdarima/. Accessed 27 June 2023.
Tranberg B, Corradic O, Lajoiec B, Gibond T, Staffell I, Anderson GB (2018) Real-time carbon accounting method for the european electricity markets. Energy Strategy Rev. https://doi.org/10.1016/j.esr.2019.100367
Tsay RS, Chen R (2019) Nonlinear time series analysis. John Wiley & Sons, Hoboken NJ
2019 Twelfth International Conference on Contemporary Computing (IC3–2019): 8–10 August 2019, Jaypee Institute of Information Technology, Noida, India; Iyengar, S.S., Ed.; IEEE: Piscataway, NJ, 2019, ISBN 9781728135915.
Umweltbundesamt. Beobachtete und künftig zu erwartende globale Klimaänderungen. https://www.umweltbundesamt.de/daten/klima/beobachtete-kuenftig-zu-erwartende-globale#aktueller-stand-der-klimaforschung. Accessed 30 Mar 2023.
United Nations (2015) Paris Agreement. https://unfccc.int/sites/default/files/english_paris_agreement.pdf. Accessed 30 Mar 2023.
van Dongen S, Enright AJ (2012) Metric distances derived from cosine similarity and Pearson and Spearman correlations. Available online: https://arxiv.org/abs/1208.3145. Accessed 4 Jan 2024
Wang H, Raj B (2017) On the origin of deep learning. http://arxiv.org/pdf/1702.07800v4. Accessed 4 Jan 2024
Winters PR (1960) Forecasting sales by exponentially weighted moving averages. Manage Sci 6:324–342. https://doi.org/10.1287/mnsc.6.3.324
Zheng Z, Han F, Li F, Zhu J (2015) Assessment of marginal emissions factor in power systems under ramp-rate constraints. CSEE Power and Energy Syst 1:37–49. https://doi.org/10.17775/CSEEJPES.2015.00049

Publisher's Note