Electric Power Systems Research 187 (2020) 106496

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

A supervised learning approach for optimal selection of bidding strategies in reservoir hydro

Hans Ole Riddervold[*,a,b], Signe Riemer-Sørensen[c], Peter Szederjesi[b], Magnus Korpås[a]

[a] NTNU, Norway
[b] Norsk Hydro, Norway
[c] SINTEF, Norway

ARTICLE INFO

Keywords:
Reservoir hydro
Bidding strategies
Hydro power
Gradient boosting
Neural network

ABSTRACT

Power producers use a wide range of decision support systems to manage and plan for sales in the day-ahead electricity market. The available tools have advantages and disadvantages and the operators are often faced with the challenge of choosing the most advantageous bidding strategy for any given day. Since only one bid can be submitted each day, this choice can not be avoided. The optimal solution is not known until after spot clearing. Results from the models and strategy used, and their impact on profitability, can either be continuously registered, or simulated with use of historic data. Access to an increasing amount of data opens for the application of machine learning models to predict the best combination of models and strategy for any given day. In this article, historical performance of two given bidding strategies over several years have been analyzed with a combination of domain knowledge and machine learning techniques. A wide range of model variables accessible prior to bidding have been evaluated to predict the optimal strategy for a given day. Results indicate that a machine learning model can learn to slightly outperform a static strategy where one bidding method is chosen based on overall historic performance.

1. Introduction

One of the main tasks for an operator of hydro electric power in a deregulated market is to decide how much power should be produced the following day. Several strategies for bidding available production exist and have been described in the literature [4,35,38]. Each strategy will potentially lead to different commitments for production, which again will have an impact on profitability.

Market prices and inflow for the next day are uncertain, and the profit associated with any selected strategy is not revealed until after the decisions are made. Power producers might acknowledge the shortcoming of existing bidding strategies but are still confronted with the inevitable choice of choosing one strategy for the next day.

Historically, the intra-day price volatility in the Nordic market has been smaller than in the European continental market. However, with increased import and export capacity, the markets are becoming more connected and increased volatility can be expected [16]. In addition, climate changes lead to increased volatility in the reservoir inflows, and more extreme events. The increase in volatility might favour bidding based on a stochastic prediction of the day-ahead market, but input probability distributions are not always well described by the available data, and the computational requirements are higher than for deterministic predictions. Consequently, in some cases, a deterministic approach is preferred.

The question addressed in this article, is if the producer with access to sufficient amounts of information about historical performance of different strategies, can predict in advance which (existing) bidding strategy should be selected for a given day. The work is based on existing bidding strategy models and exemplified with historical data from a typical Norwegian cascading river system in order to demonstrate the relevance for operators.

Bidding day-ahead production to the power-exchange is typically done the day before the actual commitments are executed. “Issue date” is defined as the date when bidding is conducted, while “value date” is used for the date when commitments from the bidding are realized through costs and income. Only variables that can be identified on or before the issue date, can be used to classify the optimal strategy associated with performance of a corresponding value date. Fig. 1

* Corresponding author.
E-mail addresses: hans.o.riddervold@ntnu.no (H.O. Riddervold), signe.riemer-sorensen@sintef.no (S. Riemer-Sørensen), peter.szederjesi@hydro.com (P. Szederjesi), magnus.korpas@ntnu.no (M. Korpås).
URL: https://www.ntnu.no/ansatte/hans.o.riddervold (H.O. Riddervold).

https://doi.org/10.1016/j.epsr.2020.106496
Received 20 January 2020; Received in revised form 22 June 2020; Accepted 23 June 2020
Available online 30 June 2020

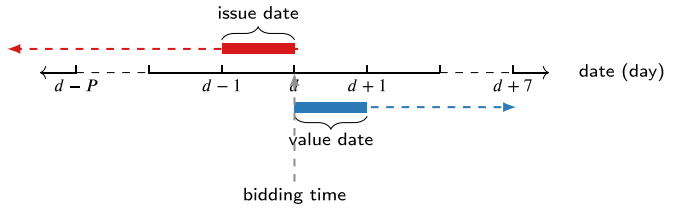


Fig. 1. Definition of issue- and value date where "d" is date, and "P" is any date before the bidding time.

describes the relationship between the two.

In the following sections, results from a case study investigating historical performance of two different bidding strategies will be presented [32]. Further, a brief overview of the steps associated with the machine learning process will be described. In Section 4 the characteristics of a river system is presented together with the key variables that will be used further in the analysis. Section 5 illustrates how the machine learning process can be applied to predict the optimal bidding strategy for the cascade river system. Finally, challenges related to deployment and a conclusion is presented in Sections 6 and 7. The aim of the article is to investigate if existing model framework used by many hydro power operators for bidding to the day-ahead market, can be combined with a machine learning process to improve the bidding performance and profitability.

2. Results from historical bidding strategies

In this analysis we demonstrate how to exploit machine learning to select between two different strategies, but the method can be used for any number of available strategies. In the first method, the expected volumes are found by deterministic optimization with price forecast and inflow and submitted as fixed hourly bids to the power exchange.

The optimization is performed with SHOP (Short-term Hydro Optimization Program), which is a software tool developed by SINTEF Energy Research, and is used by many hydro power producers in the Nordic market. The SHOP model determines the unit commitment (UC) and the dispatch plan for the coming hours and days for a price-taking, profit-maximising hydro power producer operating one or several river cascades [33]. It is a deterministic model where spot prices and inflow are assumed to be known during the entire optimization period [14]. The production schedule is found by considering all technical, hydraulic, environmental and strategic constraints in the river course, including a very detailed description of energy conversion, head-dependency and turbine characteristics. The objective function is to maximize the profit for the hydro power producer, where profit is the sum of all incomes and value of the end reservoir, minus the sum of all cost such as unbalance- and start-stop/ramping cost.

The second strategy is stochastic bidding. The stochastic model is based on the deterministic method, but allows for a stochastic representation of inflow to the reservoir and the day-ahead market prices. The model is formulated as a deterministic-equivalent program where a scenario tree is used to describe the uncertain parameters. In this case, bid-curves can be generated from the stochastic model as described in [3]. The stochastic input to the model is based on a simplified representation of uncertainty where nine scenarios have been synthetically generated around expected values for inflow and price. The deterministic model is based on the average values for inflow and price.

Example of results from evaluation of two strategies for some selected days for a specific river system are shown in Table 1. The performance quantification for the deterministic and stochastic models are the performance-gaps in EUR for these strategies.

In [32], a method for measuring performance of individual historical bidding days has been proposed, where $\Pi_{s,d}$ in Eq. (1) represents the profit calculated by the optimisation model for bidding strategy s on day d, and $\Pi_{opt,d}$ describe the profit for the relevant bidding date based on a deterministic strategy with perfect foresight of price and inflow.

The performance gap ($\beta_{s,d}$) reflects the loss of choosing strategy s for day d compared to an optimal deterministic strategy, and is calculated as the difference between $\Pi_{opt,d}$ and $\Pi_{s,d}$. A high value for $\beta_{s,d}$ indicate poor performance.

$$\beta_{s,d} = \Pi_{opt,d} - \Pi_{s,d} \tag{1}$$

$$\eta_d = \beta_{stoch,d} - \beta_{det,d} \tag{2}$$

Since the deterministic and stochastic predictions are based on pre-bidding values, even the best model will rarely correspond exactly to the perfect foresight strategy computed after the actual inflows and prices are known. Consequently, we define the best model as the one with the lowest gap relative to the perfect scenario.

If we define "strategy gap" (η_d) as "performance gap stochastic" ($\beta_{stoch,d}$) minus "performance gap deterministic" ($\beta_{det,d}$), each date (data point) is classified with "0" if (η_d) is negative, and "1" if (η_d) is positive as shown in Table 1. A high positive value is a strong signal to choose a deterministic model for that day. Negative values indicate that a stochastic model is preferred, and the more negative, the higher the importance of a stochastic model. Values around zero indicate a small difference in profit between the two strategies and therefore a negligible difference in the choice of model for the day. With this insight, we see that a pure classification model only measuring the correct number of classifications, will not necessarily give the best results if the overall target is to have a low deviation from optimum over time.

It seems obvious to apply supervised machine learning to the selection problem. We have tested two different approaches: 1) labelling each data point as stochastic or deterministic based on the performance gap, and then training a classification model to predict which category unseen data points belong to. 2) A regression model trained to predict the performances of each model directly and using a simple decision heuristic (minimum gap) to decide on the most advantageous strategy.

3. Machine learning

The use of machine learning to classify strategies or predict values have gained significant momentum during the last few years [9,21]. Machine learning is a set of techniques that allow a computer algorithm to improve performance as it gains experience, which in our case is through exposure to additional data. No explicit instructions are required, but instead, the algorithm needs some sort of training on representative input and output data. If the training is successful and the model can find some general patterns or behaviour in the data, it can subsequently be used to predict output for previously unseen input data. Machine learning basically covers everything from simple regression to deep neural networks.

Classification and regression problems, are categorised as supervised learning methods where the training takes place on pairs of input and output data, and subsequently applied to unseen input data to generate predictions.

Within the area of electricity power market analysis, neural networks have been used to investigate strategic and algorithmic bidding [6,12,30]. Neural networks and deep learning have successfully been applied for price- and load forecasting [5,7,23,36].

Gradient boosting methods have received less attention, but have been applied for load- and price forecasting [18,26].

Results from previously published articles, indicated that machine learning techniques successfully can be applied to improve forecasting of load and prices in the power market, but there exists only very few publications documenting operational use and added value from these techniques compared to what is state of the art in the industry. Several energy companies as well as software-, data- and consultancy companies supplying the energy sector advertise and promote the use of machine learning, and the European intra-day market has in particular been an area of interest in relation to application [27,28].

Table 1
Example of performance quantification in EUR for the deterministic and stochastic models for three days in 2017 for the use-case river system.

Issue date	Value date	Deterministic (β_{det})	Stochastic (β_{stoch})	Min	$\Delta(\eta)$	Best
2017-07-01	2017-07-02	69.2	137.9	69.2	68.7	1
2017-07-02	2017-07-03	16.5	65.1	16.5	48.6	1
2017-07-03	2017-07-04	31.1	29.9	29.9	-1.2	0

In this article, we assume that the data is time-independent in the sense that a strategy choice for the next day does not affect which strategy will be the best choice further in the future. Such time dependence could potentially be accounted for in a reinforcement learning framework which has received increased attention as a solution to the increasing amount of non-linear relationships and high-dimension problems associated with hydro power production planning [13,25,39]. However, in general, reinforcement methods are still fairly immature and require significant fine tuning on individual problems in order to work [19], so we consider it outside the scope of this work.

4. Description of cascade river system

An important prerequisite when building a learning-based model, is the access to data, and specifically in this case, historic bidding performance. To benefit from previous work, this analysis is based on the results obtained in [32] where bidding performance for a river system located in South-Western Norway has been analysed for 2018. The topology and relevant characteristics for the cascade are described in Fig. 2.

Given the large variations that are associated with operation of a hydro-based system, the period was extended with additional simulations for 2016 and 2017.

4.1. Input variables

Experienced production planners may have an intuitive perception of what the important input variables could be, and these can be used as initial values in the learning process.

The two strategies evaluated in this analysis are measured against a perfect foresight model where prices and inflow are known prior to bidding. All other factors going into the performance evaluation are equal. We expect that factors affecting the prices and inflows will have a significant effect on the performance gap. Another important factor affecting the production schedule is the water value. The water value can be defined as the future expected value of the stored marginal kWh of water, i.e. its alternative cost [10,37].

The basic concept is to produce when the water value is lower than the price. The lower the water value is compared to the price, the stronger is the signal to produce. Based on this insight and domain knowledge of what might effect production, we apply some general hypotheses to select an initial set of input variables for further analysis as given in Table 2.

All input variables in Table 2 are related to the issue date, and consequently available when the strategy for the next day is decided. In the “High” and “Low” columns, the preferred strategy based on operational experience is indicated for respectively high and low values of the variable. “U” indicates that domain experts are uncertain how that variable alone will affect the classification.

5. Application of machine learning process

There are several ways of approaching a machine learning problem, but they often involve some or all of the sub-tasks illustrated in fig. 3 and several iterative loops over the process. Our implementation of the process is discussed in Sections 5.1–5.6.

5.1. Data integration

First we visualize all the input- (X) and output variables (y) to detect missing or obviously incorrect data in the data set, or anything that should be corrected for during the analysis, in addition to providing ideas of interactions between input and output variables.

Fig. 4 shows performance gaps (β) together with inflow deviation from normal. It provides a clear indication of a connection between periods with high inflow and poor performance of deterministic bidding.

In Fig. 5 we plot the strategy gap (η_s) defined in Eq. (2), but only for absolute values larger than 200 €/day corresponding to days with a significant impact from the choice of strategy. These values typically concentrate around the second quarter every year, indicating that time of year e.g. month may be a relevant variable to include in the analysis.

5.1.1. Correlation

Fig. 6 shows the relative (Spearman) correlation between the variables listed in Table 2. Independent variables (X) that are un-correlated with the dependent variable(s) (y) can be removed. Independent variables that are correlated with each other can be reduced or combined.

We are mainly interested in the variables with the highest correlation to the prediction indicated in the “BEST” column; in this case the water value, inflow deviation and reservoir filling in reservoir 2. As seen in Fig. 2, reservoir 2 is a small reservoir in the middle of the cascade. With a high reservoir filling, the risk of flooding increases. This will favour a more risk-averse bidding strategy which can be provided be a stochastic model in accordance with the hypothesis listed in Table 2. There is also a strong correlation between prices associated with the issue date (`average_p`) and the prognosis for the value date (`average_prog`).

Additional information can be obtained from grid plots as shown in Fig. 7. Blue (triangular) markers indicated when the stochastic model performs best, while red (round) markers indicate when the deterministic model is preferred. When the level in the intake reservoir is high or inflow is well above normal, a stochastic model is preferred.

5.2. Feature engineering

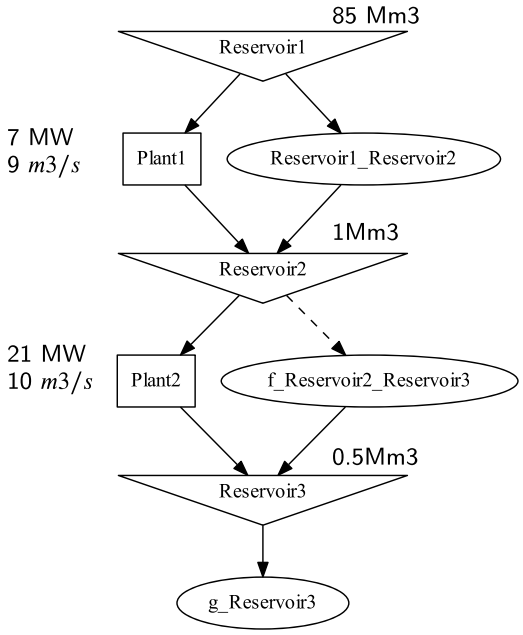
The terms variable and feature are often used without distinction. In this article, input variables are used when referring to variables collected in the data collection phase, while we use the term feature for constructed variables engineered to maximise the information available for the model.

In the ideal world with unlimited training data and computational power, the machine learning algorithm can be fed all the available variables and figure out the relevant features itself. However, with limited access to data, selection of features is often an iterative process with frequent reviews of the initial conditions and adjustments of the variables in order to maximise learning and minimise bias. Insufficient performance of a model should call for rethinking both related to input, model structure and system-design.

For the analysis with a limited number of variables, no additional feature engineering has been applied. The data-set is referred to as simple input, and the data is published in [31]. To evaluate the benefit of increasing the number of variables beyond the eight original variables, a second data set was investigated. The second data-set was

Table 2
Initial set of input variables, hypotheses for how they affect the choice of strategy, and the strategy that the domain experts presumably would select for high/low values of the variables.

Nr.	Variable	Hypothesis	High	Low
1	Inflow deviation from historic normal	Higher observed inflow increases the risk of flooding for the next day (value date)	S	D
2	Reservoir filling 1	High or low reservoir filling increases the risk of flooding or resource shortage during value date	S	S
3	Reservoir filling 2	High or low reservoir filling increases the risk of flooding or resource shortage during value date	S	S
4	Price volatility	High volatility in prices gives increased uncertainty for prices the next day	S	D
5	Price volatility in the prognosis	High volatility in the price prognosis give increased uncertainty for prices the next day	S	D
6	Water value	Water value is the primary deciding factor for production, and will clearly have an effect on the strategy choice when seen in relation to other variables	U	U
7	Average price	Price relative to water value is important	U	U
8	Avg. price prognosis	Price prognosis relative to water value is important	U	U



Characteristics of cascade
• Linked large and small reservoirs
• High head-loss in plant 2
• Time-delay between reservoir 1 and 2
• Gains from head-optimization
• Sensitive to flooding in periods with high inflow
• High utilization rate
• Buy-pass gate between Reservoir 1 and 2
• Flood gate between Reservoir 2 and 3
• Gate from Reservoir 3 out of cascade

Fig. 2. Topology and characteristics of investigated cascade riversystem.

extended to 113 variables without improving the results significantly. Additional feature engineering and variable reduction was also conducted. A summary of the variables associated with the extended dataset and main results are given in Appendix B.

5.3. Selecting the model and algorithm

The problem at hand can be solved as a classification problem

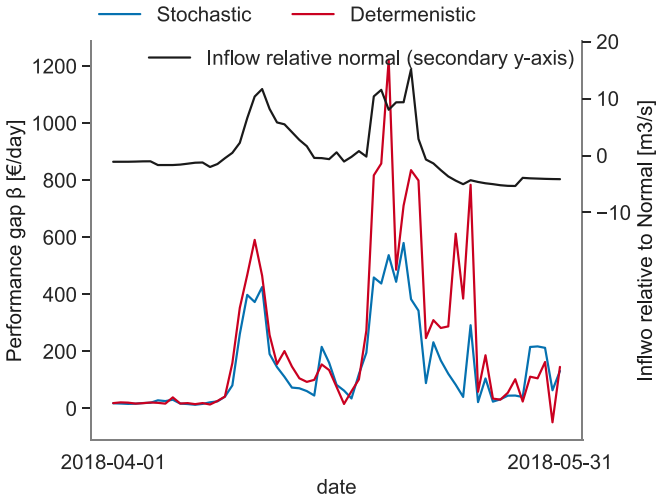


Fig. 4. Detailed results for performance gap from April to May 2018, compared to inflow relative to normal. Data from [32].

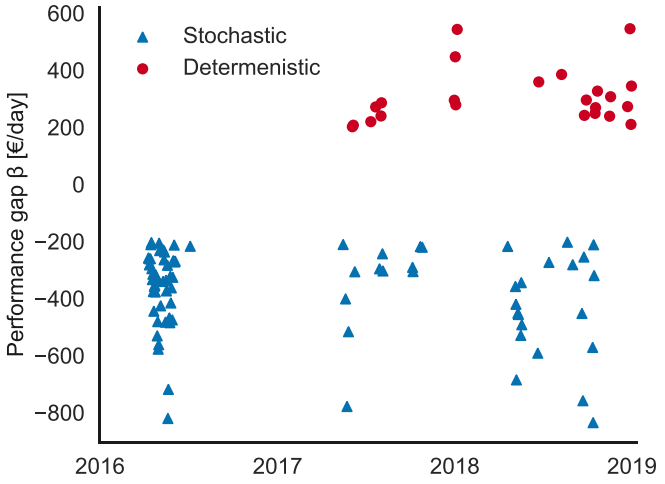


Fig. 5. Strategy gaps in the period 2016–2108.

where the target is to predict the best bidding strategy or by predicting the strategy gap directly, and based on this decide the optimal strategy.

An algorithm suitable for both classification and regression is decision trees as implemented in the XGBoost library [11]. XGBoost implements gradient boosting [15,17] decision tree algorithm using

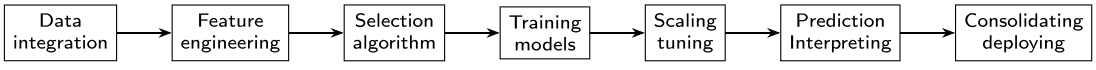


Fig. 3. Sub-tasks involved in the machine learning analysis process applied towards the cascade river system illustrated in Fig. 2 and with the variables in Table 2

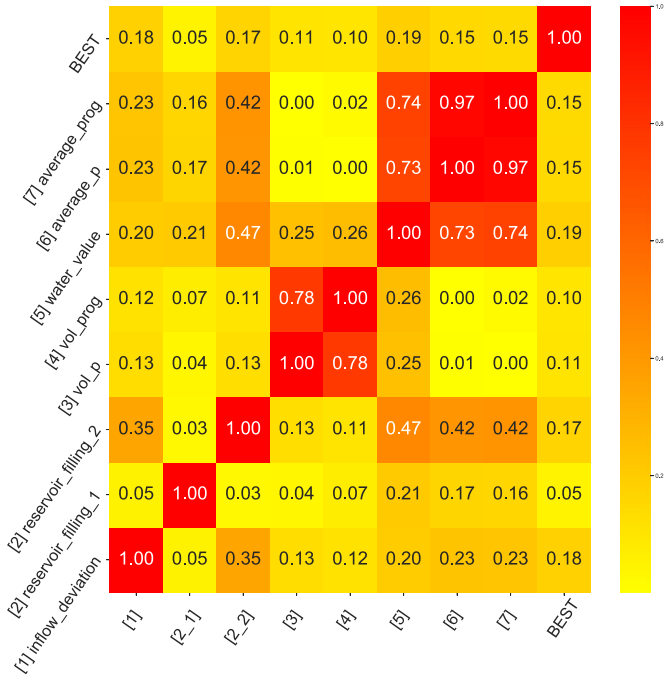


Fig. 6. Correlation matrix for the initial set of variables. Numbers to the left in labels denote "Nr." in Table 2. "BEST" denotes the dependent variable (prediction).

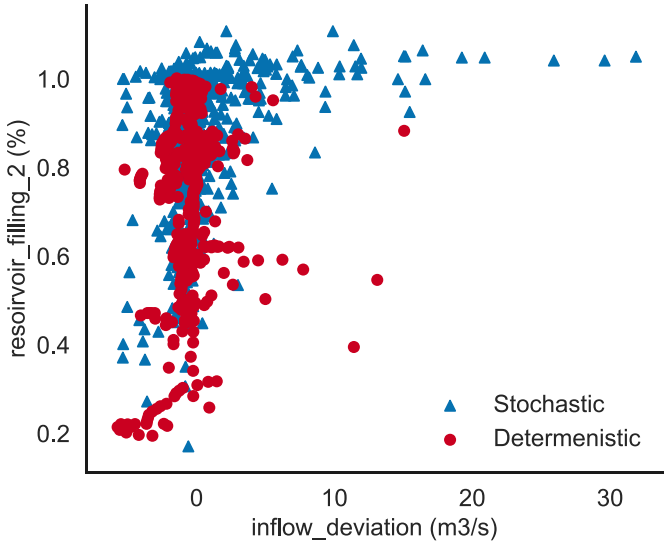


Fig. 7. Grid plot for variables inflow_deviation and reservoir_filling_2.

multiple trees to classify samples. This gives valuable insight on the feature importance and possible feature interactions. The features described in Section 4.1 are closely related to physical system and decision variables and consequently this insight translates directly to insight on the decision process. The decision trees and model structure are specified through the hyper parameters of which the most important ones are learning rate, maximum tree depth and number of estimators represented by trees [1]. The exact choice of hyper parameters affects the model performance and we discuss their tuning in Section 5.5.2.

Fig. 8 illustrates an example of a randomly selected tree from the classification model for a limited set of features. The first split feature is placed highest or to the left in the tree, followed by subsequent criteria. The total model is an average of multiple trees with different representations of the features. The importance of each parameter in the trees can be derived in several ways. In this article, we apply GAIN and SHAP values which will be described further in Section 5.5.

Fully connected neural networks (NN) and recurrent neural networks (RNN) have also been tested but with less success. Anyhow, we provide a comparison with these methods in Section 5.6.4.

5.4. Splitting the data

Before the model training, the data is split into training and test samples. The basic principle is to build and train the model on a fraction of the data (training data), and then test the performance of the model on data that are not used when building the model (test data).

Randomly splitting the data into training and test samples might conflict with underlying mechanisms behind the power market. When managing hydro power, weather plays an important role. The possible outcomes and combinations of reservoir levels, inflow, snow and prices are widely spread, and the power producers therefore tend to use a long history (typically 30–60 years) of observations to represent the possible scenarios in their decision support models. Consequently a random split of historic data may leak information from the historic future into the training sample, which is not desirable. Instead, we can use a sequential split of data, and test the trained model on years not previously seen by the model. Such a split will lead to a model that performs well on years with similar historic representation, but with very little predictive power in "freak" years. Three years of data are available in this analysis. If we assume that two years are used for training, and one year is used for testing, we get a split of 67/33 between training and test data. In Section 5.6.2 we present the results from splitting randomly and from training on 2016 and 2017 data and testing on 2018.

Another method, which has not been tested in this analysis, is to use all available historic information, or a rolling number of historic days to train the model. The model will then be used to classify the next day. To test performance of such a model, a simulation framework would be needed since we are depending on updating the training set every day, as well as re-calibrate the model.

5.5. Feature scaling and hyperparameter tuning

The main objective of the feature scaling and hyperparameter tuning processes is to find a set of model parameters that result in a model where the performance measures described by Eqs. (3) and (4) in Section 5.6.1 are taken into account, and the total profit for the operator cascade is optimized over time.

5.5.1. Feature scaling

Not all features have the same scale: Some have values of the order of 1000s, and some are 0.1. In order to let them equally influence the model, we need to "put everything on the same scale".

In general, decision tree algorithms such as XGBoost do not require scaling [1], but it might help with quicker convergence in relation to numerical processing. Scaling is required for neural networks, so to be able to apply the same pre-processing of data, similar scaling has been applied for both XGBoost and neural networks.

Depending on the sample size, the test data can either be scaled with their own scaling (for large samples), or with the training sample (small samples).

Time-dependent data may have trends that makes a simple scaling meaningless, in particular when training on historic data and applying to new data. Fig. 9a illustrate the distribution of water values for each of the three years. Fig. 9b describe the results after applying standard scaling for all years jointly. The distributions differ significantly between the three years. If two years with relative low values such as 2016 and 2017 are used for training, the values are not representative for the 2018 test data, which is dominated by higher values. If we instead assume that the fluctuations within the years are similar, but the overall scale may be characterized by non-controllable events, we can scale the water values per year, and obtain more similar distributions,

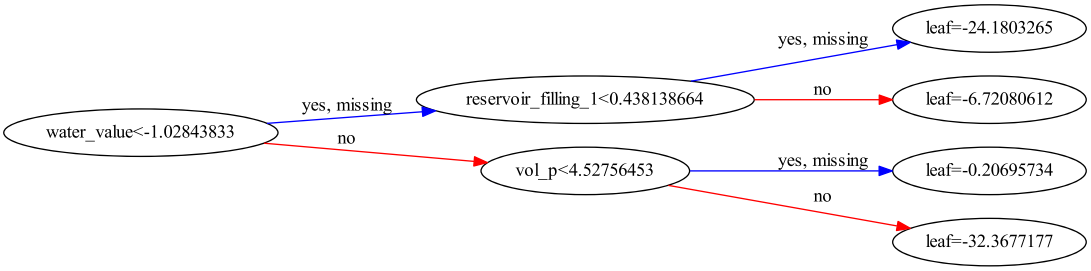


Fig. 8. An example of one classification tree from the model trained on a limited set of features. The total model is an average over many trees.

as shown in Fig. 9c. The same pattern is be observed for several of the input variables and consequently we scale all variables by individual years. In real life application, we are not able to scale the daily input variables with the yearly average since this information first is available after the year has passed. An implementable alternative would be to scale the input variable with the rolling average of the previous 365 days.

5.5.2. Hyperparameter tuning

There are several hyperparameters in the XGBoost model that should be tuned. We have applied the randomized search functionality in scikit-learn [29] to identify the values leading to superior model performance for a selection of parameters listed in Appendix A along with their final values. The resulting hyperparameter values after tuning vary both depending on the number of features that are included, and for limited data sets also as a result of the exact selection of training and test data. To avoid that the parameters are over-optimised towards the training data, the training data can be further split into subsets. A validation set is a subset of the training data, that is used during the training to describe the evaluation of model variation due to changes in hyperparameter values and data preparation.

In our case, we have split the randomly selected training set into five sub-sets (folds) of approximately equal size. The first fold is treated as a validation set, and the model is fit on the remaining $k-1$ folds [20]. These folds are then evaluated with 1000 sets of randomly generated hyperparameters. The resulting accuracy obtained for each fold for every combination of hyperparameters is plotted together with the average value in Fig. 10. Firstly, we observe a spread in accuracy within each fold illustrating the importance of tuning the hyperparameters. Further inspection reveals a complicated parameter landscape with multiple local minima. We also observe a spread in accuracy between each fold, indicating that the folds may not be representative for the full data set. This could be improved with additional data. The range in observed accuracy indicates that a fairly large spread can be expected when the model is applied on randomly generated training and test data. Finally, the hyperparameters associated with the best average score which can be observed around iteration number 400 in Fig. 10, is selected as the parameters used further in the analysis.

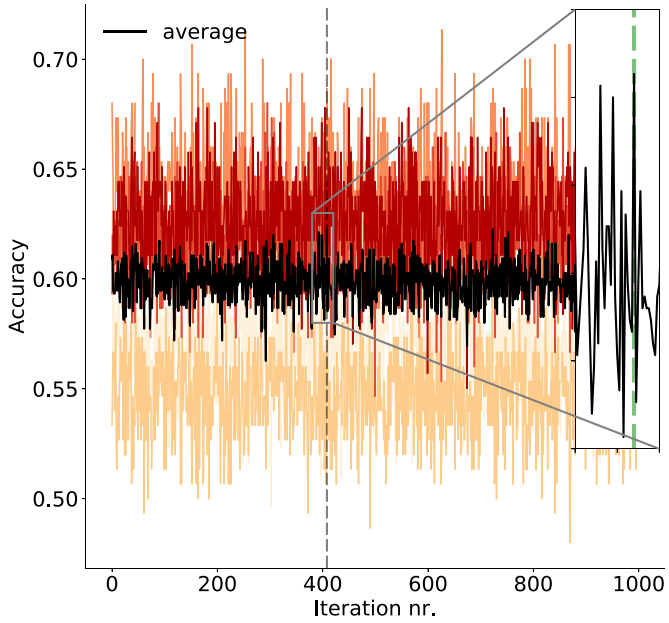


Fig. 10. Hyperparameter tuning over 1000 iterations of random combination of parameters.

5.5.3. Feature importance and explanations

A useful tool when evaluating the relevance and importance of different features is SHapley Additive exPlantions (SHAP). The framework interprets a target model by applying Shapley game theory for how a reward given to a team should be distributed between the individual players based on their contributions [24]. The features are interpreted as “contributors”, and the prediction task corresponds to the “game”. The “reward” is the actual prediction minus the result from the explanation model. The underlying idea is to take a complex model, which has learnt global non-linear patterns in the data, and break it down into lots of local linear models which describe individual data points.

The shap values can be interpreted either for individual predictions

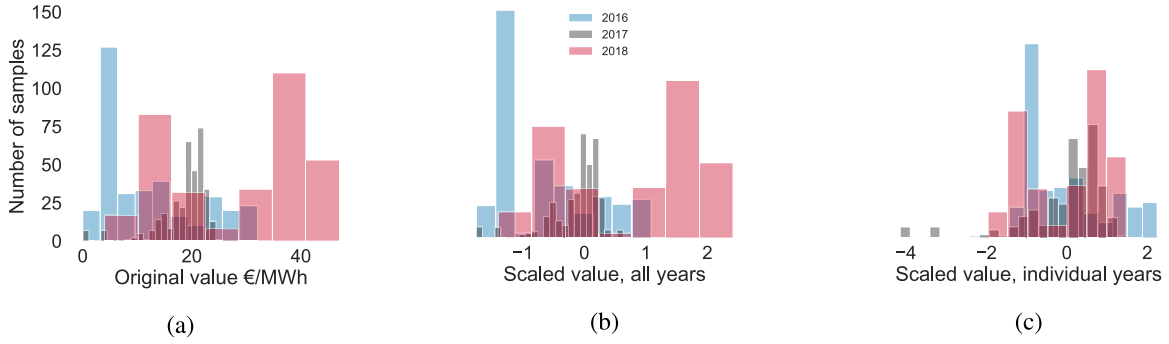
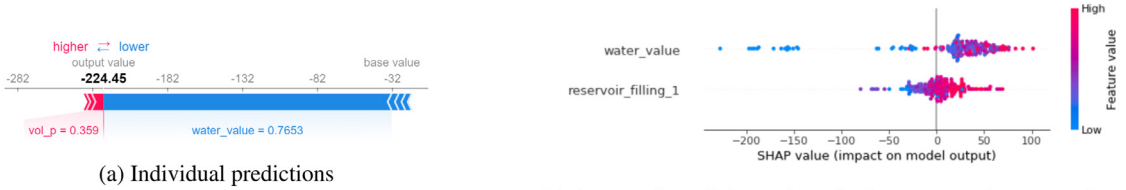


Fig. 9. Distribution profile for water values un-scaled (a), scaled together (b) and individually (c).



(a) Individual predictions

(b) Aggregation of shap values for the two most important features

Fig. 11. Shap values on individual and aggregated feature values.

or for the entire sample. Fig. 11a shows one sample classified using the regression model described in Section 5.6.3. The output value is the prediction for that observation which in this case is − 224.45 and consequently the sample is classified as stochastic. The base value of − 32 is the value that would be predicted if we did not know any other features for the current output. This is the mean value for "strategy gap" for all samples in the training set. This is logical, since in lack of other information, we would predict the average value for any new samples. Interestingly, if no other information is available, we would classify the sample as stochastic. The red and blue arrows illustrate how adding other feature values push the prediction towards higher or lower values. Red indicate a push towards higher values and deterministic bidding, and blue is a push towards lower values and stochastic bidding. In this sample there is a clear push towards stochastic bidding mainly driven by the water value. The only other feature with any significant impact on this sample is the volatility for today's price which is pushing in direction of deterministic bidding.

Fig. 11 b illustrates an aggregation of values for two of the most important features in the regression problem. Variables are ranked in descending order of feature importance. The shap values can be interpreted as "odds" i.e. what is the probability of "winning"/predict the higher/correct value which in our case what is the probability of predicting 1. Which again means predicting that the deterministic model is best. So for high (low) shap values that specific variable is contributing to increase (decrease) the probability of predicting deterministic (stochastic) bidding. This is illustrated by the horizontal location, which shows whether the effect of that value is associated with a higher or lower prediction. The color of the individual sample shows whether that variable is high (in red) or low (in blue) for that observation. If the colors are split similarly to the shap values, the relation is simple, otherwise it's probably complex and dependent on multiple variables. A low water value has a large and negative impact on the strategy gap pushing in favour of stochastic bidding, while a high water value favours deterministic bidding. The "low" comes from the blue color, and the "negative" impact is shown on the X-axis.

The computation of shap values assume independence between the features. Our features are not independent, which may affect the ranking somewhat [2]. However, since shap primarily is used to qualitatively understand and illustrate potential impact of different features in this article, the fact that some of the variables are dependent play a minor role in relation to prediction accuracy.

5.6. Prediction

The main target of the classification problem is to predict which bidding strategy to use for the next day. Fig. 12a illustrate predictions from the classification model together with actual historic "Best strategy" from the test data. Zeros indicate that a stochastic strategy should be chosen, while ones indicate a deterministic strategy. A nice attribute of the XGBoost classificator with binary logistic objective is that the outcome is given as probabilities and not pure classification. This means that a figure close to zero gives a high probability of the sample being stochastic, while a figure just beneath 0.5 still classifies as stochastic, but with lower probability.

When applying a single-output regression model, the output will be the strategy gap representing the predicted difference in value for choosing one strategy in favour of another. It has previously been explained that a negative strategy gap indicates stochastic bidding, while positive numbers indicate deterministic bidding. While the classification model gives probabilities between zero and one, the regression model will span out values in a much larger range capturing the values that are at stake (after inversion of any applied scaling). Fig. 12b illustrate the predicted strategy gap from the XGBoost regression model compared with the observations in the test data. The target is still to decide for a single bidding strategy for the next day. The results from the regression model must therefore be transformed to binary recommendations of either a stochastic or a deterministic strategy.

Both the classification and regression model open for the possibility of applying thresholds on the results/ probabilities. As an example, when applying a threshold of 0.5 in the classification model, all values above will be classified as deterministic, while all below will be classified as stochastic. If we choose a threshold of 0.4, and 0.6, only samples with values under or over this value will be classified into a category. Values in between will not be classified. The accuracy when applying a tighter threshold will typically increase, however with the price of an increased number of unclassified days. In both these cases the model is risk-neutral when deciding on strategy. It is also possible to have a skewed threshold where we are more risk averse in relation to choosing one strategy opposed to another. If a single strategy is to be selected for all days, previous analysis [32] have shown that a stochastic strategy is on average superior. If a threshold of for instance 0.6 is applied, a clear tendency in direction of deterministic bidding is required before this strategy is chosen in favour of a stochastic strategy. Applying thresholds might have an even higher impact on regression models. While the probabilities associated with prediction in the classification models only are proxies for the importance of choosing the correct strategy for a selected day, the regression model indicate directly the potential losses associated with choosing the wrong strategy. A strategy focusing on classification of samples with major cost impact could be a viable solution.

5.6.1. Evaluation of model performance

To evaluate the model performance, two measures are introduced:

$$A = \frac{Number\ \ of\ \ correct\ \ predictions}{Total\ \ number\ \ of\ \ predictions} \tag{3}$$

$$\delta_{realistic} = (\overline{\beta}_{gap} - \overline{\beta}_{gap,opt})/\overline{\beta}_{gap,opt} \tag{4}$$

Even though high classification accuracy is an important target, it is not the main objective related to strategy selection. The main objective is to reduce the average performance gap compared to a model where the best bidding strategy is selected every day. In this sense, identifying the correct strategy on dates where there is a large performance gap between the two strategies, previously defined as the strategy gap, will be more important. This is quantified as $\overline{\beta}_{gap,opt}$, which represents the average performance gap from the optimal plan for all samples in the test data if the optimal strategy had been selected, and $\overline{\beta}_{gap}$ represents the average performance gap from the optimal plan for all classified samples.

$\delta_{realistic}$ is then a measure of how far we are from the optimal bidding

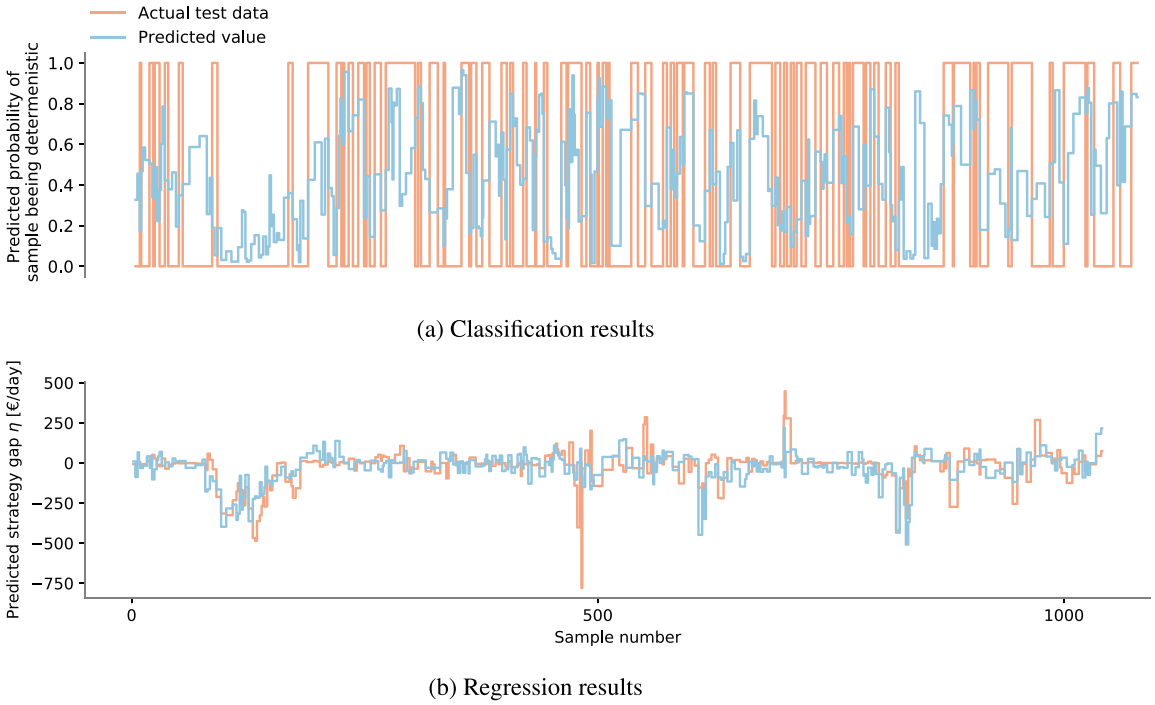


Fig. 12. Predictions with classification and regression for the simple model.

strategy in percent of the optimal bidding strategy. It will be referred to as "Realistic Performance Gap".

The model is designed to suggest an optimal bidding strategy, and performance should therefore be measured against a benchmark where an optimal bidding strategy is selected for every day, and not against an optimal plan with perfect foresight on prices and inflow. This is why the measure of $\delta_{realistic}$ is introduced, rather than using $\overline{\beta}_{gap}$ as performance measure.

Assuming that the consequences of wrongly classifying samples are normally distributed between stochastic and deterministic strategies, and that there is a fairly equal split between when the two strategies perform best, the accuracy represents a good measure for model performance. In other cases, for instance in medical diagnostics, the consequence of failing to identify disease of a sick person might be much higher than the cost of sending a healthy person to more tests. In this case we have to use more sophisticated methods for evaluating performance of the model [22].

In Fig. 13 the two measures of A and $\delta_{realistic,i}$ are plotted as a function of number of features that are removed in the classification problem. Here we have used a rule-based algorithm where model

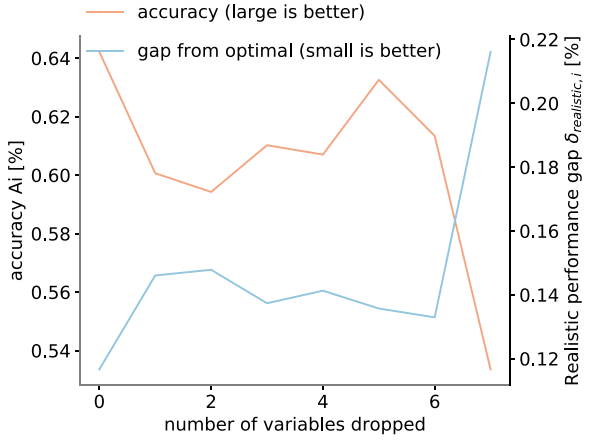


Fig. 13. Performance as a function of feature reduction.

performance is evaluated as we gradually remove features with the lowest impact on the classification results for a fixed set of hyperparameters. The feature importance has been evaluated as the "gain" computed from the decision trees in XGBoost. The gain is defined as the improvement in accuracy from adding a split on a given feature to a branch in the classification tree [6]. In this example, the best accuracy is obtained with the eight original features. It can also be observed that this coincides with when the gap to a optimal bidding model is the lowest.

Table 3 summarizes the results from different modelling approaches described in the Sections 5.6.2–5.6.5 on various splits of the available data.

5.6.2. Classification

Two main approaches for selecting training and test samples have been investigated when applying the classification model, sequential and random split.

For the random sampling, we observe a variation in performance for different random seeds. This indicates noise in the data, and that finding a universal set of features and hyperparameters is difficult based on the available data. In order to determine the variance of possible outputs on the random sample, we bootstrap [34] the test sample and determine the performance on each sub-sample individually.

As expected, applying the model on sequential data give poorer accuracy performance than for random samples. When 2016 and 2017 data are used to predict 2018 strategies, the results are barely better than random guessing. Investigating the true values, stochastic bidding is best for 58% of the days in 2016–2017 and there are 52% stochastic samples in 2018. Consequently, applying a purely binomial selection with a probability factor for stochastic results equal to 0.58 would increase the accuracy to above 50% without taking into account any variables.

One would however, expect that the results from sequential splitting would approach the accuracy obtained by random split as more data will become available.

The random sampling give an accuracy around 62%, but also here the training and test set consists of an average of 54% stochastic samples, so we are only increasing accuracy by 8% compared to applying a

Table 3
Results for predicted accuracy (A) and realistic performance gap ($\delta_{realistic}$) with different modelling approaches. A^*_{mean} is the mean value and A^*_{std} the standard deviation after bootstrapping on 100 samples. $Stoch_{red}$ is the percent reduction in days using the stochastic model.

Model	description	A	$\delta_{realistic}$	A^*_{mean}	A^*_{std}	$Stoch_{red}$
(1) XSCR	Xgboost Simple Classification Random	0.62	0.18	0.61	0.03	40
(2) XSCS	Xgboost Simple Classification Sequential	0.56	0.21	0.55	0.03	47
(3) XSRR	Xgboost Simple Regression Random	0.60	0.15	0.60	0.03	43
(4) NNRR	Neural Network Regression Random	0.59	0.19	NA	NA	43
(5) RNRS	Recurrent Neural Network Regression Sequential	0.57	0.22	NA	NA	48
(6) CCRR	Combined Custom Regression Random	0.59	0.17	NA	NA	44
(7) PDR	Pure Deterministic Random	0.46	0.43	NA	NA	100
(8) PSR	Pure Stochastic Random	0.54	0.21	NA	NA	0

binomial selection without any variable input.

An interesting observation is that even though there is an average of 18% realistic performance gap (δ) for the prediction model when applying random sampling, the gap is still 3% better than when applying a pure stochastic strategy for all days in the evaluation period. Relying solely on a deterministic bidding strategy is not a good idea, since this would give a realistic performance gap (δ) of 43%.

5.6.3. Regression

The classification process described in Section 5.6.2 is designed and optimized towards the best prediction accuracy. The best accuracy correlate weakly with situations where the lowest realistic performance gap (δ) is observed. If the primary target is to obtain as low a gap as possible, several adjustments could be made.

First, the classification model has been trained with a binary logistic objective, which specifically optimizes the accuracy regardless of the resulting gap. Fitting for the actual gap values with e.g. a "Mean Squared Error" (MSE) objective would penalise large gaps more than small gaps.

It is also possible to introduce more categories and perform multi-class classification. That would allow for weighting different strategy gaps differently in the loss function. E.g. the range $\delta \in [-400, -200]$ would in most cases benefit from being predicted as stochastic. The disadvantage is that the weighting must be tuned by hand.

To take into account the importance of the numeric values of the strategy gap, an alternative method is to perform regression on the strategy gap directly, and rather classify the best strategy with simple heuristics after predicting the strategy gap. The reason for not pursing this method as primary approach, is that the relative limited amount of data, and amount of noise in the input data, which lead to a model with relative poor performance. However, if we are only interested in the sign of the performance gap the regression model allows for direct optimisation on the gap. With a mean squared error (MSE) objective, the GAIN loop takes features that contribute to reducing the gap into account. MSE is given by Eq. (5), where η_i is the strategy gap for sample i and $\hat{\eta}_i$ is the predicted value. Consequently, the features can be selected based on the strategy gap rather than classification accuracy.

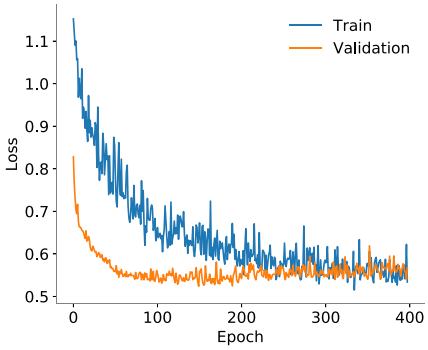
$$MSE = \frac{1}{n}\sum_{t=1}^{n}(\eta_i - \hat{\eta}_i)^2 . \tag{5}$$

As expected, the classification accuracy is 2% worse than for the pure classification model, but the performance gap for the model is 3% better.

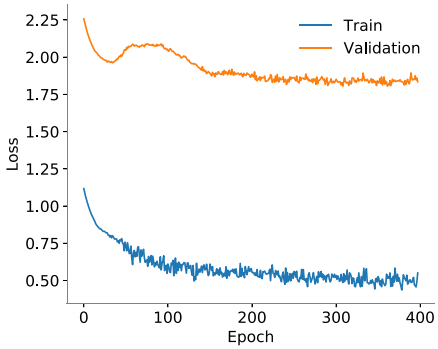
5.6.4. Neural networks

The XGBoost algorithm was chosen for two main reasons. Firstly, it allows for transparent tracking of feature importance for the predictions, and thereby it avoids the "black-box" perception associated with neural networks. Secondly, it is suitable for both classification and regression. However, it is useful to benchmark the performance of XGBoost against a neural network approach.

Here we compare the XGBoost regression model for the original feature sample (XSRR) with a fully connected multilayer perceptron (MLPs) model, often referred to as a classical type of neural network. The benchmark neural network model consists of two hidden layers with 50 and 20 neurons. The hyperparameters have been manually tuned to minimise the mean squared error with the resulting values given in Appendix A. Typically, with neural networks, we seek to minimize the error of the prediction model, also referred to as the loss. The loss provides information of the goodness of fit as well as potential over-fitting if compared between the training and validation data as illustrated in Fig. 14a, where the loss on the validation data increases beyond 150 epochs while the training loss still decreases. The resulting model provides a good representation of the training data but lacks the ability to generalize. We apply early stopping and use the neural



(a) The training and validation losses as a function of training epocs for a well-fitted neural network with eight features. Beyond 150 epochs the validation loss increase while the training loss decrease, indicating over-fitting.



(b) The training and validation losses as a function of training epoch for a poorly fitted recurrent neural network.

Fig. 14. Plot of model loss for Neural Networks.

network model after 150 training iterations for benchmarking. The prediction accuracy and realistic performance gap (δ) for the neural network (NNRR) are somewhat worse than for the regression using XGBoost (XSRR) as shown in Table 3.

In the fully connected neural network we assume that all data points are independent of each other timewise. To exploit any time dependence in the data, we have applied a recurrent neural network (RNN) on the sequential split of training and test data. The network consists of a single layer of 3 recurrent units. As for the fully connected neural network, the hyperparameters are manually tuned to the values given in Appendix A, Table 4. The network is trained for 400 epochs on the 2016–2017 data with a lookback time of 30 days, and tested on the 2018 data.

The high values of the training loss shown in Fig. 14b indicate that the model fails to converge on a good fit. This is likely due to limited data samples. The poor fit is reflected in poor prediction performance as seen in Table 3.

5.6.5. Alternative and combined approaches

Both XGBoost and neural networks have strength and weakness that can influence the results of the analysis. A nice attribute of the XGBoost algorithm is the ability to track the importance of different features, and thereby remove features that have little or no impact on the results. A useful attribute of neural networks is the ability to simultaneously predict multiple features, and combine this with a custom made loss function. A possible approach for combining the strengths of the methods could be a hybrid model where XGBoost is used to select the most important features, and a neural network is applied with the recommended set of features and a customized loss function (CL).

This approach has been investigated, and indicate that the results might improve marginally for the neural network model, but the results are still poorer than for using the XGBoost model alone. The used custom loss function was designed to focus on the days where the highest performance gaps were observed.

$$CL = \frac{1}{n}\sum_{t=1}^{n} |min(\beta_{det}, \beta_{stoch}) - min(\hat{\beta}_{det}, \hat{\beta}_{stoch})|^n \quad (6)$$

In comparison with Eqs. (5), (6) evaluates directly on the performance gap (Π_s) for each bidding strategy directly, and not on the strategy gap (η_s). This makes it possible to "punish" days where one strategy perform poorly compared to another. The difference can be raised to a power of n to emphasize the difference even more.

Anther concept that could be an alternative to the investigated approaches is Bayesian model averaging, where multiple models can contribute to the decision support, with weights based on how sure each model is.

6. Challenges for operational deployment

Computational time associated with running the stochastic model depend on the topology and number of scenarios but is in the range 10–20 times higher than for a deterministic model [8] for a system comparable to the one analysed in this article. The time associated with establishing hyper parameters for the model on an annual or monthly basis will only be a small fraction of the potential time saved by reducing the number of days where the stochastic model is used. To be able to choose between different bidding strategies, a company will need enough computational power to manage the most demanding bidding process within a relative short time period. For a stochastic model with a significant number of scenarios applied on a complex river system, this might potentially pose a challenge as time might be a critical factor for companies wishing to include the most recent and updated prognoses in the decision process.

If a power producer decides to implement a strategy selection model, including new observations and re-calibrating the model on a daily basis could be the most robust solution. Time associated with parameter tuning will increase, and the time saved compared to running a stochastic model will be less important. One difference between neural networks and boosting methods is that neural network can be updated on the fly, whereas boosting methods must be fully retrained when the training data changes.

Tracking performance of a strategy selection model will be important, and the power producers should consider running the model in parallel with existing systems for a period of at least one year to account for seasonal effects. If performance of the model turns out to be poor, it is an indication that the historic information available might be insufficient.

The investigated approach of combing machine learning with existing model framework used by many hydro power producers has demonstrated to provide additional value for the investigated river system. Establishing a simulation framework and simulating data for historic performance for new typologies is a comprehensive task. Direct application of the method described in this article on new cascades might be challenging unless power producers already are in possession of data describing historic bidding performance. Further analysis of benefits for other river systems with different topology and/or market conditions is a topic for further research. The stochastic model presented in this article is based on a very limited set of scenarios. Improvements to the stochastic model will potentially change the performance of a stochastic strategy considerably. For further analysis, it will be of interest to investigate the performance of a more well-adjusted stochastic model against either a deterministic or a simple stochastic model.

7. Conclusion

We have tested various techniques for classification and regression on historical data representing bidding performance for a reservoir-based river-system in the Nord Pool market. The primary objective has been to investigate the possibility of predicting prior to day-ahead bidding whether stochastic or deterministic bidding would be the preferred strategy under the prevailing market and hydrological conditions.

A simple plot of inflow deviation together with performance gap for the year 2018 as shown in Fig 4 give an indication that periods with high inflow to a certain degree are correlated with periods with negative stategy-gaps and hence favouring stochastic bidding.

Inflow deviation together with water value has also proved to play a large role in the prediction model. This has been demonstrated using two different techniques for investigating feature importance that are associated with the gradient boosting in XGboost.

If historical data are assumed to represent future instances with sufficient accuracy (some trends can be removed through scaling), applying a prediction model trained on available data will be able to predict the optimal strategy with an accuracy of 62–63%. If the benchmark is a strategy where only stochastic bidding is performed, the prediction model will outperform this strategy, and the number of days that must be analysed with the exemplified stochastic model is reduced by almost 40%. With an improved representation of sample space and probabilities for the stochastic input more in line with best practice for these models, the number of days where the deterministic model outperforms the stochastic model will be reduced. A prediction model with only 62% accuracy and a standard deviation of 3% indicate that there is considerable noise in the data. Given that only one combination of hyperparameters and variables can be selected in an operational environment, the risk of over-fitting a model increases when the model is fit to a large set of variables. This might favour choosing a simple model with a limited set of features.

Neural networks have been tested to benchmark result from XGBoost however, they were less successful. Using regression rather that classification has proved to reduce the performance losses, but at

Table 4
Model architecture and hyper-parameters.

Param	XSCR	XSCS	XSRR	NNRR	RNRS	CCRR
Algorithm	Xgb	Xgb	Xgb	Keras Sequential Dense	Keras Sequential LSTM	Keras Sequential Dense
objective	binary: logistic	binary: logistic	reg: squarederror	mse	mse	custom Eq. (6)
learning_rate	0.075	0.075	0.092	0.001	0.001	0.001
max_depth	4	4	3			
n_estimators	178	178	259			
number_boost_round	9282	9282	2509			
gamma	4.26	4.26	4.34			
subsample	0.77	0.77	0.64			
dropout_frac				0.2	0.3	0.2
number_neurons1				50		50
number_neurons2				20		20
LSTM layers/neurons					1 and 3	
lookback					30	
L2				0.005	0.005	0.005
kernel_regularizers				l1(L2)	l1(L2)	l1(L2)
activation				linear	linear	linear
optimizer				adam	adam	adam

the expense of somewhat lower accuracy score.

In order for the prediction models to be valid, the training data must be representative of the actual data the model will be applied to. In some cases this can be achieved through proper scaling of the data. In general more historical data will alleviate the problem, but it may not always account for the volatile nature of power markets and weather.

CRediT authorship contribution statement

Hans Ole Riddervold: Conceptualization, Methodology, Writing - original draft, Project administration, Investigation, Software. **Signe Riemer-Sørensen:** Methodology, Software, Writing - review & editing, Validation, Formal analysis. **Peter Szederjesi:** Data curation, Investigation. **Magnus Korpås:** Supervision.

Declaration of Competing Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Appendix A. Model architectures and hyper-parameter values

Appendix B. Extended data-set

To evaluate the benefit of increasing the amount of variables, a second data-set has been investigated. The second set is referred to as *extended data-set* and consists of the following variables:

- the eight simple variables (8)
- all hourly prices for both issue date and prognosis for value date (48)
- bid-ask curves (48)
- rolling volatility (2)
- month, year, day and performance of similar week-days (4)
- strategy gap for issue date (1)
- rate of change for reservoir filling (1)
- difference between price and water value (1)

In total, the complex-input data-set consist of 113 variables. Results : A = 0.63, $\delta_{realistic} = 0.15$

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.epsr.2020.106496

References

[1] Xgboost documentation.
[2] K. Aas, M. Jullum, A. Løland, Explaining individual predictions when features are dependent: more accurate approximations to shapley values, ArXiv abs/1903.10464(2019).
[3] E. Aasgård, C. Naversen, M. Fodstad, H. Skjelbred, Optimizing day-ahead bid curves in hydropower production, Energy Syst. 9 (2) (2018) 257–275.
[4] E.K. Aasgård, H.I. Skjelbred, F. Solbakk, Comparing bidding methods for hydropower, Energy Procedia 87 (2016) 181–188, https://doi.org/10.1016/j.egypro.2015.12.349. 5th International Workshop on Hydro Scheduling in Competitive Electricity Markets
[5] S. Anbazhagan, N. Kumarappan, A neural network approach to day-ahead deregulated electricity market prices classification, Electr. Power Syst. Res. 86 (2012) 140–150, https://doi.org/10.1016/j.epsr.2011.12.011.

[6] S. Baltaoglu, L. Tong, Q. Zhao, Algorithmic bidding for virtual trading in electricity markets, IEEE Trans. Power Syst. 34 (1) (2019) 535–543, https://doi.org/10.1109/TPWRS.2018.2862246.
[7] J. Bedi, D. Toshniwal, Deep learning framework to forecast electricity demand, Appl. Energy 238 (2019) 1312–1326, https://doi.org/10.1016/j.apenergy.2019.01.113.
[8] M. Belsnes, O. Wolfgang, T. Follestad, E. Aasgård, Applying successive linear programming for stochastic short-term hydropower optimization, Electr. Power Syst. Res. 130 (2016) 167–180, https://doi.org/10.1016/j.epsr.2015.08.020.
[9] C. Bordin, H.I. Skelbred, J. Kong, Z. Yang, Machine learning for hydropower scheduling: state of the art and future research directions, Submitted to 24th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, (2020).
[10] S.H. Brovold, C. Skar, O.B. Fosso, Implementing hydropower scheduling in a european expansion planning model, Energy Procedia 58 (2014) 117–122, https://doi.org/10.1016/j.egypro.2014.10.417. Renewable Energy Research Conference, RERC 2014
[11] T. Chen, C. Guestrin, Xgboost: a scalable tree boosting system, ArXiv abs/1603.02754(2016).
[12] G. Cocchi, L. Galli, G. Galvan, M. Sciandrone, M. Cantú, G. Tomaselli, Machine learning methods for short-term bid forecasting in the renewable energy market: a case study in italy, Wind Energy 21 (5) (2018) 357–371, https://doi.org/10.1002/we.2166.
[13] J.T. Fidje, C.K. Haraldseid, O.-C. Granmo, M. Goodwin, B.V. Matheussen, A learning automata local contribution sampling applied to hydropower production optimisation, in: M. Bramer, M. Petridis (Eds.), Artificial Intelligence XXXIV, Springer International Publishing, Cham, 2017, pp. 163–168.
[14] O. Fosso, M. Belsnes, Short-term hydro scheduling in a liberalized power system, 2004 International Conference on Power System Technology, 2004. PowerCon 2004, 2 IEEE, 2004, pp. 1321–1326Vol.2.
[15] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. 29 (5) (2001) 1189–1232.
[16] I. Graabak, M. Korpås, S. Jaehnert, M. Belsnes, Balancing future variable wind and solar power production in central-west europe with norwegian hydropower, Energy 168 (2019) 870–882, https://doi.org/10.1016/j.energy.2018.11.068.
[17] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference and Prediction, second ed., Springer, 2009.
[18] J. Iria, F. Soares, A cluster-based optimization approach to support the participation of an aggregator of a larger number of prosumers in the day-ahead energy market, Electr. Power Syst. Res. 168 (2019) 324–335, https://doi.org/10.1016/j.epsr.2018.11.022.
[19] A. Irpan, Deep reinforcement learning doesn't work yet, 2018, (https://www.alexirpan.com/2018/02/14/rl-hard.html).
[20] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning: with Applications in R, Springer Texts in Statistics, Springer New York, 2014.
[21] M.I. Jordan, T.M. Mitchell, Machine learning: trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260, https://doi.org/10.1126/science.aaa8415.
[22] L.A. Krukrubo, The confusion matrix for classification.
[23] S. Li, P. Wang, L. Goel, Short-term load forecasting by wavelet transform and evolutionary extreme learning machine, Electr. Power Syst. Res. 122 (2015) 96–103, https://doi.org/10.1016/j.epsr.2015.01.002.
[24] S. Lundberg, G. Erion, H. Chen, A. DeGrave, J. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S.-I. Lee, Explainable ai for trees: from local explanations to global understanding, ArXiv abs/1905.04610(2019).
[25] B.V. Matheussen, O.-C. Granmo, J. Sharma, Hydropower optimization using deep learning, in: F. Wotawa, G. Friedrich, I. Pill, R. Koitz-Hristov, M. Ali (Eds.), Advances and Trends in Artificial Intelligence. From Theory to Practice, Springer International Publishing, Cham, 2019, pp. 110–122.
[26] V. Mayrink, H.S. Hippert, A hybrid method using exponential smoothing and gradient boosting for electrical short-term load forecasting, 2016 IEEE Latin American Conference on Computational Intelligence (LA-CCI), (2016), pp. 1–6, https://doi.org/10.1109/LA-CCI.2016.7885697.
[27] NN1, Machine learning on the intraday power market.
[28] NN2, Wattsight launches price forecast for continuous intraday trading.
[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[30] T. Pinto, T.M. Sousa, H. Morais, I. Praça, Z. Vale, Metalearning to support competitive electricity market players strategic bidding, Electr. Power Syst. Res. 135 (2016) 27–34, https://doi.org/10.1016/j.epsr.2016.03.012.
[31] H.O. Riddervold, Replication Data for: a gradient boosting approach for optimal selection of bidding strategies: Simple model - Original variables, 2020, 10.18710/WNKSVX.
[32] H.O. Riddervold, E.K. Aasgård, H.I. Skjelbred, C. Naversen, M. Korpås, Rolling horizon simulator for evaluation of bidding strategies for reservoir hydro, 2019 16th International Conference on the European Energy Market (EEM), (2019), pp. 1–7, https://doi.org/10.1109/EEM.2019.8916227.
[33] H.I. Skjelbred, Unit-based short-term hydro scheduling in competitive electricity markers, 2019.
[34] L.B. Statistics, L. Breiman, Out-of-bag estimation.
[35] G. Steeger, L.A. Barroso, S. Rebennack, Optimal bidding strategies for hydro-electric producers: a literature survey, IEEE Trans. Power Syst. 29 (4) (2014) 1758–1766.
[36] J. Torres, A. Galicia, A. Troncoso, F. Martínez-Álvarez, A scalable approach based on deep learning for big data time series forecasting, Integr. Comput. Aided Eng. 25 (4) (2018) 335–348.
[37] O. Wolfgang, A. Haugstad, B. Mo, A. Gjelsvik, I. Wangensteen, G. Doorman, Hydro reservoir handling in norway before and after deregulation, Energy 34 (10) (2009) 1642–1651, https://doi.org/10.1016/j.energy.2009.07.025. 11th Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction
[38] H. Yamin, Review on methods of generation scheduling in electric power systems, Electr. Power Syst. Res. 69 (2) (2004) 227–248, https://doi.org/10.1016/j.epsr.2003.10.002.
[39] M. Zarghami, Short term management of hydro-power system using reinforcement learning, 2018.