Recalibration of LSTM-based Models for Day-Ahead Wind Power Forecasting

Pierre-David DAPOZ, Jérémie BOTTIEAU, Aurélien WAUTIER, Zacharie DE GREVE, François VALLEE, Jean-François TOUBEAU
Electrical Power Engineering Unit (EPEU), University of Mons, 7000, Mons, Belgium

***Abstract*—This paper is focused on the day-ahead prediction of the onshore wind generation. This information is indeed published each day, ahead of the market clearing, by European Transmission System Operators (TSOs) to help market actors in their scheduling strategy. In that regard, our first objective is to efficiently capture the complex temporal dynamics of the wind power using recurrent neural networks. To that end, advanced architectures of Long Short Term Memory (LSTM) networks are implemented and compared. Secondly, in order to continuously improve the prediction accuracy, different techniques for recalibrating the model during its practical utilization are analyzed. This procedure consists in adjusting the parameters of the neural networks by taking advantage of the new information revealed at each time step, without the (time-consuming) need to retrain the model from scratch over the whole available dataset. Finally, the financial impact for the system operator due to the wind prediction error is estimated. Outcomes from the Belgian case study show that an optimal model recalibration can significantly improve the quality of the forecasts, thereby decreasing the balancing costs of the system.**

***Index Terms*—Deep Learning, Electricity Markets, Bidirectional Decoder, LSTM Neural Networks, Recalibration Forecast**

I. Introduction

The liberalization of the electricity sector has introduced new prerogatives for the Transmission System Operators (TSOs), among which the task of facilitating the access to the market for all actors. In that regard, TSOs must provide various information to all market participants such as the real-time evolution of the system balance or the anticipated onshore and off-shore wind generation. With the current decarbonization of the electricity sector, the increased contribution of weather-dependent (and thus, uncertain and intermittent) renewable generation has made this forecasting task very important for ensuring a cost-effective system operation.

Many researches have thereby introduced new techniques for wind prediction. Firstly, statistical approaches based on the inference (from observed data) of basic statistics such as the mean, variance and autocorrelation have emerged [1], [2]. However, the underlying assumptions often involve that such forecasters rely on simple linear models which are not able to capture the nonlinear characteristics (such as the different ramp rates) of the wind. In parallel, physical models were also developed, but they necessitate a complex mathematical description of the environment, which is computationally intensive, and often based on arbitrary simplifying assumptions [3]. To address these issues, machine learning (data-driven) approaches have recently been tested by the prediction community, and have progressively exhibited better performances than classical methods [4]-[6]. This trend is mainly driven by the ability of data-driven approaches to accurately capture and represent hidden characteristics of complex variables, without the need to arbitrarily define the model complexity. Specifically, neural networks are powerful tools whose flexible nature can be tailored to the characteristics of the forecasting problem, thereby improving their accuracy. Particularly, this property has led to the advent of recurrent neural networks (RNNs), advanced deep learning structures that are specifically designed to propagate through time relevant information from past observed data (by relying on a dynamic memory). In this way, such recurrent networks have shown a high potential in processing times series data such as wind generation.

However, one of the main challenges that still needs to be properly studied relates to the recalibration of the model. Indeed, once the forecaster is trained (using historical observations), it is then used for actual field operation (on new data), but the model is usually not updated with the new information that is revealed at each time step. In [7], the models are re-estimated from scratch (using all the historical database) on a daily basis but at the expense of a continuous utilization of large computational resources. In this work, we aim at improving this naïve approach by retraining the existing forecaster at optimal time intervals (e.g. every day, week, etc.) with a sliding window that includes the relevant set of past observations. The idea is to identify whether it may be beneficial to generate a slight bias in the model (by dynamically over-fitting to recent conditions) rather than to rely on a single static model that performs well in average along the year but that is suboptimal for each of its constituting sub-periods. Overall, the contributions of the work can be summarized as follows.

Firstly, we exploit the flexible nature of neural networks by implementing three different recurrent architectures, based on Long Short Term Memory (LSTM) cells [8]. The objective is to predict (at 11:00 a.m. in day-ahead) the expected wind generation for the 24 hours of the next day. The three models, i.e. (i) the encoder, (ii) the decoder, and (iii) the bidirectional decoder differ in the way they capture space-time dependencies, which affects their predictive capabilities. In that regard, their accuracy is not only compared to state-of-the-art techniques (such as ensemble methods), but also with the predictions performed and published by the TSO.

Secondly, the development of a recalibration procedure is developed. This process allows to adjust the parameters of the neural networks by taking advantage of the new information continuously revealed over time (during the actual daily utilization of the forecaster), without the time-consuming need to retrain the model over all the historical data set.

Thirdly, the financial impact of prediction errors (on both the TSO and wind producers) is estimated. To that end, the balancing costs required to compensate the wind imbalances are computed based on (publicly available) market data.

The paper is organized as follows. In Section II, we develop different LSTM architectures to capture the dynamical behavior of wind generation, and we discuss several strategies for recalibrating the model over time. Section III focuses on the prediction accuracy of the models, which are compared with outcomes from TSO and state-of-the-art methods. The best model is then selected, and incorporated into the recalibration policy, which allows to significantly improve the quality of the predictions. Section IV finally evaluates the financial gain of reducing the forecasting error incurred by the forecasting error (by saving the balancing costs). Finally, in Section V, conclusions and perspectives are exposed.

II. Methodology

This section is divided into two parts. Firstly, different LSTM-based architectures of recurrent neural networks are presented (Section II.A). Secondly, the methodology to identify the best recalibration policy is discussed (section II.B).

A. Development of LSTM-based forecasting tools

This work focuses on neural networks, which are flexible tools (theoretically able to learn any complex nonlinear functions) that combine multiple advantages. In that regard, the complexity of the model can be tailored to the complexity of the task (thereby avoiding both under- and over-fitting issues), and the architecture can be adapted to the specificities of the problem. Given that wind generation is an inherently dynamic process, we consider recurrent networks (Fig. 1), which are purposely tailored to process temporal dependencies.

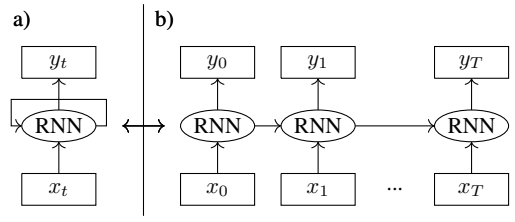

Fig. 1. General representation of recurrent neural networks (RNN) with cyclical connections that act as a dynamical memory (a), i.e. the network is unrolled though time to seamlessly represent time dependencies (b).

The general principle of recurrent neural networks (RNN) is to generate the prediction y_t based on the input information x_t, for each time step $t \in T$ of the prediction horizon of interest. Based on N historical data, the RNN is trained to minimize the error between its output y_t and actual observation d_t.

The RNN is made up of different stacked layers, each one composed of multiple neurons, which overall define the model complexity. The recurrent architecture is also characterized by cyclical links, connecting the state of the neurons among consecutive time steps, thereby propagating information through time. This property is illustrated in Fig. 1 where it is observed that the prediction a time t depends on both input variables at time t and the information from previous time steps.

In recent years, RNN applications have been very successful for a variety of problems such as speech recognition, language modelling and translation, etc [9]. However, RNNs are known to struggle in capturing long-term dependencies, such that relevant information arising from longer term periodicities (such as seasonal effects) can be lost. To address this issue, LSTM neurons were developed, and rely on gating units that regulate the flow of information that is propagated through time. The principle of LSTM cells is depicted in Fig. 2.

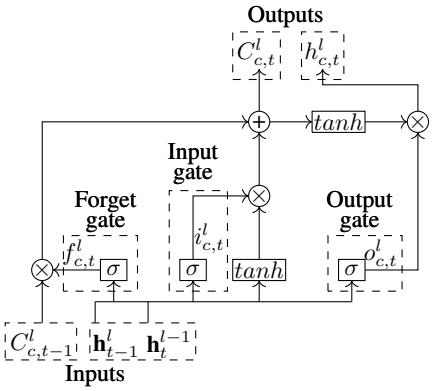

Fig. 2. Single-cell LSTM memory block c (pertaining to layer l at time t).

In Fig. 2, we observe that the LSTM cell c at layer l at time step t is fed by three different contributions, i.e. $\mathbf{h}^{l-1}_t$ the output vector (of all LSTM cells) of the layer below at the same time, $\mathbf{h}^l_{t-1}$ the output vector (of all LSTM cells) of the same layer at the previous time step, and $C^l_{c,t-1}$ the state of the cell c at the previous time step (which acts as a dynamical memory). Overall, the LSTM neuron is composed of 3 gated units (input, output and forget gates) and the LSTM layer l is thus characterized by the following composite function:

$$\mathbf{f}^l_t = \sigma\left(\mathbf{W}_f\ \mathbf{h}^{l-1}_t + \mathbf{W}_f\ \mathbf{h}^l_{t-1} + \mathbf{b}_f\right) \quad (1)$$

$$\mathbf{i}^l_t = \sigma\left(\mathbf{W}_i\ \mathbf{h}^{l-1}_t + \mathbf{W}_i\ \mathbf{h}^l_{t-1} + \mathbf{b}_i\right) \quad (2)$$

$$\mathbf{C}^l_t = \mathbf{f}^l_t\ \mathbf{C}^l_{t-1} + \mathbf{i}_t\ \tanh\left(\mathbf{W}_c\ \mathbf{h}^{l-1}_t + \mathbf{W}_c\ \mathbf{h}^l_{t-1} + \mathbf{b}_c\right) \quad (3)$$

$$\mathbf{o}^l_t = \sigma\left(\mathbf{W}_o\ \mathbf{h}^{l-1}_t + \mathbf{W}_o\ \mathbf{h}^l_{t-1} + \mathbf{b}_o\right) \quad (4)$$

$$\mathbf{h}^l_t = \mathbf{o}_t\ \tanh(\mathbf{C}^l_t) \quad (5)$$

where σ is the logistic sigmoid function, and $\mathbf{i}_t$, $\mathbf{f}_t$ and $\mathbf{o}_t$ are the activation vectors of the input, forget and output gates respectively, whereas $\mathbf{C}_t$ stands for the cell activation vector. The weight matrices $\mathbf{W}_\bullet$ (i.e. links between LSTM neurons) and the bias vectors $\mathbf{b}_\bullet$ are the parameters of the network that need to be optimized during the learning procedure.

In this work, three different LSTM-based architectures, which differ by the way they process temporal information, will be developed and compared, i.e. (i) the encoder, (ii) the decoder, and (iii) the bidirectional decoder.

The encoder is a topology tailored to sequentially process the past information, and to generate the predictions at the end of the input sequence. The issue with this architecture consists thus in feeding the network with the available (known or estimated) information about the future. Such information typically comes from numerical weather forecasts that can provide estimation of the future temperature, cloud cover or wind characteristics, and it is thereby essential to properly include these features as input data for the prediction model. In the encoder, it is done by providing those data at the last time step of the input sequence, which may not be optimal. This architecture is shown in Fig. 3

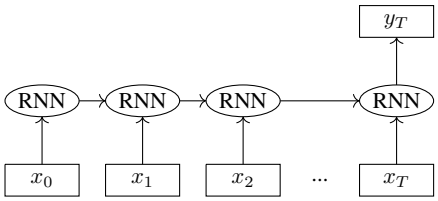

Fig. 3. General representation of the encoder architecture.

Another option for incorporating the temporal information is to rely on a decoder, which generates a prediction at each time step of the horizon. This design, which is represented in Fig. 4, is traditionally used for on-line tasks (such as sequence generation), and is thus is not well suited to take advantage of past information. Indeed, these data need to be incorporated at the first time step of the decoder, which may thus struggle to properly extract the relevant information from both short- and long-range past features.

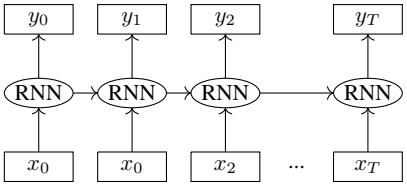

Fig. 4. General representation of the decoder architecture.

To improve on both encoder and decoder architectures, a third topology, the bidirectional decoder, is investigated. This design aims at optimally exploiting (at each time step of the prediction horizon) the complete information about the whole horizon. For instance, for the prediction at time t, the network is not only fed by the past information (by exploiting the traditional recurrent connections) but also by the available future data (such as the estimation of weather variables at next time steps). The underlying idea is that the available information at time $t + k$ (e.g. through weather forecasts) can help explaining what will happen at time t. As we can see in Fig. 5, the bidirectional decoder is composed of two different recurrent neural networks, which are connected to the same output layer (providing the predictions of interest). The resulting topology processes (simultaneously) the input sequence forwards and backwards, thereby leveraging the whole contextual information.

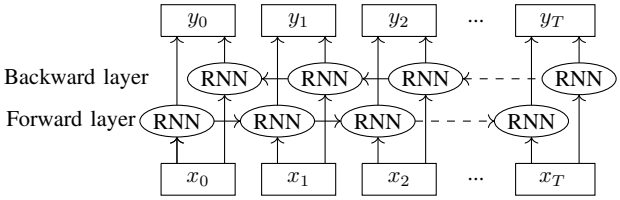

Fig. 5. General representation of the bidirectional decoder

B. Recalibration strategy

When the same prediction model is used each day (in a static manner), two problems inevitably arise. Firstly, the model does not take advantage of the new information that continuously becomes available over time (and that can be used to improve the accuracy of the data-driven model). Secondly, the model may be good in average, but not optimal for each sub-period of the year. To address both these issues, a recalibration of the model is investigated, where the model can be slightly biased to new data (e.g. the inner dynamics of the model will differ between winter and summer months).

In practice, when identifying the best recalibration strategy, two questions need to be answered :

- what is the frequency at which the model needs to be calibrated, i.e. the optimal number of days p between two calibrations ?
- what is the size of the sliding window, i.e. the number of days r whose information is exploited to adjust the parameters of the forecaster ?

To identify the best values of r and p, a design of experiments is carried out in Section III.C, by assuming that $r \geq p$.

In parallel, for determining the extent to which the model needs to be modified, three different strategies are investigated. Firstly, an ideal (non-realistic) benchmark is considered, which yields the best outcome that can be expected from the calibration. To that end, the model is trained on the r past days, but the p days to predict are used as validation set. In reality, these days cannot be used as validation (since they are not yet realized). In that way, we ensure that the model is calibrated in such way that it will provide the best outcomes for the days to predict. A second method selects the validation set in a classical way (using 10% of the historical information), so that the model is trained on the remaining 90% data, until convergence is achieved on the validation set. The third model is trained with a fixed number of epoch (i.e. we impose the number of iteration of the gradient descent algorithm through the training sequence of r days), so that no data are discarded for the validation set.

III. Case study

In this work, we focus on the deterministic onshore wind generation in Belgium. Our results can thus be compared with those of the system operator (i.e. Elia), which publishes each

day (at 11.00 a.m., i.e. 1 hour before the closure of the Day-Ahead Market) its hourly forecasts in order to promote a more competitive and transparent market. Indeed, a better prediction will result in better information for the market players, hence reducing the risks when participating in competitive electricity markets. To compare our models on a fair basis, our predictions are also carried out at 11.00 a.m. for the 24 hours of the following day. Thus, the prediction horizon of interest ranges from m = 13 to 37 hours into the future. The prediction tool used by the TSO is not disclosed for confidentiality reasons.

A. Data pre-processing

The available dataset includes the onshore wind power (aggregated at the Belgian level) for four years, starting from 2014 to the end of 2017. These four years are separated into a training, a validation and a test set. The training set starts on January 1, 2014, and ends on September 31, 2016, the validation set is composed of the next three months, and the year 2017 is used as test set.

The prediction tool is fed by input (explanatory) variables of different types. Firstly, we use weather data (such as temperature, cloud cover, etc.) that are expected for each hours of the next day. This information typically comes from advanced meteorological models. For this work, we add only access to the data from a single station (located at the center of the country). Secondly, the last measured values (typically the previous 6 to 48 hours) of wind generation are highly important to capture the dynamics of the variable, and are thus provided to the models. Thirdly, temporal information (hours of the day, day of the week and month of the year) is also used to better capture multi-scale time characteristics [10]. Finally, the installed capacity of wind generation is also exploited.

Before training the model, it is necessary to standardize the data for two main reasons. First, different variables are typically associated with different ranges, e.g. the scale of temperature values (in °C) is naturally lower than the historical wind generation (in MW) by several orders of magnitude. However, it does not mean that the latter variables is that much more important that the first one. Such differences will lead to more difficulty in correctly adjusting the weights of the neural network, resulting in poor outcomes and longer simulation times. Secondly, the range of variables must be adapted to the activation function of the LSTM. For instance, the hyperbolic tangent in (3) and (5) reaches saturation when the input is higher than 2. Feeding the network with higher values thereby wipes off the processing power of the network. The scaled variables $X_{scaled} \in [0, 1]$ are computed as:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{6}$$

where X_{min} and X_{max} are the minimum and maximum values of the database for each variable X.

B. Comparison with state-of-art approaches

In this part, we calculate the prediction accuracy (over the test year 2017) for the three developed LSTM-based architectures, the encoder (Enc.), decoder (Dec.) and bidirectional decoder (B.Dec.). The models are trained using the "Adam" optimization algorithm [11]. These models are compared to the predictions published by the Belgian TSO, as well as to other classical methods, i.e. the feedforward neural network (also referred to as multilayer perceptron, or MLP), and XGBoost [12]. In practice, Python 3.6.0 and the Keras library (with the TensorFlow backend) has been used for implementing neural networks, whereas the scikit-learn library has been employed for XGBoost. The results are represented in Table I. The root mean square error (RMSE) is used as error metric :

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - d_t)^2} \tag{7}$$

with $n = 8760$ the number of predicted values (i.e. hourly data over the 2017 test set), y_t the output of the prediction model and d_t the actual measured value.

TABLE I
COMPARISON OF LSTM-BASED MODELS WITH OTHER METHODOLOGIES

Methodology	MLP	XGBoost	Enc.	Dec.	B.Dec.	TSO
RMSE (MW)	128	140	127	125	115	111

Interestingly, the bidirectional decoder (B.Dec) outperforms other LSTM-based tools, which can be explained by its tailored architecture that empowers traditional RNN by better capturing temporal dependencies. Overall, all recurrent models are more accurate than classical methods (MLP and XGBoost). The optimal complexity of the bidirectional network is given by a single hidden layer with 32 LSTM neurons in its two constitutive forward and backward layers (Fig. 5). Moreover, the best results were obtained by feeding the models with 2 days of historical wing generation.

Overall, those results are very promising since they are closely challenging the performances of the TSO, which has potentially access to more input features (such as several meteorological stations in Belgium). Indeed our best model (i.e. bidirectional decoder) has an error metric of 115 MW while the TSO has an error of 111 MW (over the year 2017). In the next sub-Section III.C, we will investigate (for the B.Dec.) whether adjusting the model at regular intervals throughout the test year can improve the prediction accuracy.

C. Performance of the recalibration

Firstly, we define the ideal benchmark for the (B.Dec) model calibration. The results are shown in Table II, where the calibration is performed in different conditions, i.e. for a calibration performed every p days, using the information from a number r of past days.

From Table II, we see that recalibrating the initial model (RMSE of 115 MW) can significantly improve its accuracy (to reach a RMSE of 100 MW), thereby surpassing the performance of the TSO model. Outcomes show that the ideal frequency for recalibrating the bidirectional decoder model is 7 days, with an historical database composed of the past 30 days. These parameters will therefore be used in the rest of the paper

TABLE II
ACCURACY OF DIFFERENT RECALIBRATION STRATEGIES

RMSE (MW)		r			
		1 day	7 days	30 days	90 days
p	1 day	102.76	102.06	101.64	104.22
	7 days	X	101.96	100.3	102.91
	30 days	X	X	101.95	102.94
	90 days	X	X	X	105.5

(for other calibration methods). The value of these parameters can be explained by the nature of the learning procedure. Indeed, training the model on a lower number of days (on a very regular basis) results in over-fitting the calibrated model to these new observations (thereby loosing the generalization capabilities of the prediction tool). On the other hand, when the model is too rarely updated, we do not take advantage of the beneficial effect of slightly adapting the model parameters to the current conditions.

As a reminder, the stopping criterion of the ideal benchmark is triggered by the performance on the days to predict, allowing the model to perfectly overfit on these days.. It thereby yields an upper bound of the gain that can be expected by the recalibration. In actual field operation, these outcomes cannot be achieved. Different practical methods are thus studied to try achieving comparable performances.

In that regard, the most straightforward strategy consists in relying on a conventional validation set (in a similar fashion than the the one used to train the original model). This approach allows to regularize the model, by avoiding that the parameters are overly adapted to the training data (thus reducing the generalization capacities of the prediction tool). Unfortunately, this validation set decreases the amount of data that can be used during the training phase, and we thus choose a validation test containing 10% of the training set. Then, we assess the impact of the position of the validation set within the historical database. Practically, four cases are studied : (i) the validation set is chosen at the beginning of the dataset (older data), (ii) in the middle, (iii) at the end (more recent data), and (iv) randomly within the whole training sequence. However, this sensitivity analysis shows that modifying the position of the validation set does not influence the accuracy of the prediction (with a difference of at most 0.1 MW).

Another approach for calibrating the model is to bypass the use of a validation set (that decreases the number of data available to adapt the weights of the model) by considering a fixed number of epochs (i.e. number of iterations of the gradient descent algorithm). Finding the optimal number of epochs is a challenging task since smaller values do not allow to fully exploit the new revealed information, while large values inevitably result in over-fitting issues. In the two cases, we do not learn optimally. Fig. 6 shows the evolution of the prediction error if we modify the number of epochs. A number between 100 and 500 epochs is relatively stable and introduce a RMSE close to 102 MW, which is very close to the optimal value of the ideal benchmark. Evidently, it should be kept in mind that retraining the model on a higher number of epochs will inevitably increase the simulation time.

Fig. 6. Evolution of the error regarding the number of epochs

Finally, these methods are compared with a more simple (and time-consuming) methodology where the model is re-trained from scratch every $p = 7$ days.

Monthly errors are summarized in Fig. 7, where we observe that the ideal (non realistic) way for recalibrating the model systematically improve the results (for all months of the year). Then, we see that using a fixed number of epochs (i.e. 250 in accordance with Fig.6) seems to be the best strategy (outperforming all other approaches), and leads to results close to the ideal benchmark. In this way, retraining from scratch is not only computationally demanding, but also less efficient than the proposed calibration method. Besides, after recalibration, this optimal model (Epoch fixed) shows higher accuracy than the model of the TSO.

In general, we can also note than the prediction error (quantified through the RMSE) is slightly lower during summer months. However, the winter period is the more critical in terms of generation adequacy, and it is thus important to have reliable information during that time. In that regard, it is interesting to notice that our models are significantly better than the TSO for these important months.

IV. FINANCIAL COSTS ARISING FROM FORECAST ERRORS

In this part, we evaluate the costs that can be saved from recalibrating the wind generation forecaster. Indeed, in case of real-time imbalance, the TSO restores the system frequency by relying on (costly) operating reserves. Both downward and upward reserves are needed to respectively compensate excesses and shortages of wind power [13].

The costs associated with this balancing mechanism result from two contributions, (i) the capacity allowance (€/MW/h) that remunerates the procurement of power margins (that can be activated by the TSO in case of need), and (ii) the actual deployment of the requested energy (€/MWh). However, these costs are supported by different actors. The TSO is responsible to size and build the reserve capacity, and the resulting costs (i) are transferred to the electricity bill of end-users [14]. The reserve activation costs (ii), on the other hand, are supported by market actors who are responsible for creating the imbalance. In this way, by enhancing the forecast reliability, we decrease

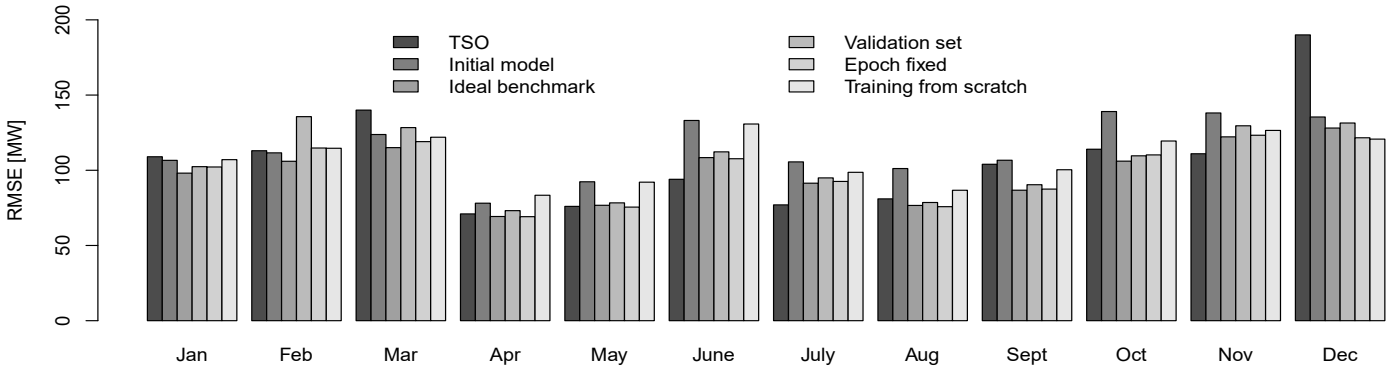

Fig. 7. Comparison of monthly evolution of each recalibration model

the (costly) reserve capacity to be contracted by the TSO. Likewise, reducing the forecast error decreases the penalties incurred to wind producers, thus boosting their profitability.

In this work, we assume that the real-time system imbalance originates only from the wind forecast error (i.e. the dispatch of other resources strictly follows their committed day-ahead schedule, and the failures of network components are neglected). In accordance with the current European legislation, i.e. the System Operation Guidelines, we consider that the TSO defines the minimum reserve capacity (required to maintain the balance in the control zone) with the goal of covering the imbalances for at least 99% of the time, taking into account historic imbalance observations [15]. Hence, based on the distribution of wind forecast errors computed at each of the 8760 hourly time step of the year 2017, we infer the resulting need of both upward R^+ and downward R^- reserve capacity (as depicted in Fig. 8). Once the sizing is determined, considering an average price of 10 €/MW/h, the annual costs (in €) can be computed according to $(R^+ + R^-)$*10*8760.

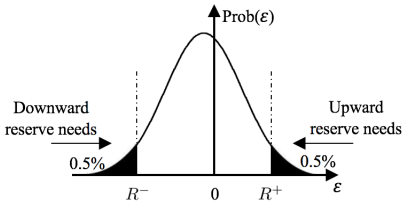

Fig. 8. Representation of the method for sizing the reserve capacity.

In this Fig. 8, the prediction error ε_t is defined as the difference between the prediction y_t and the actual value d_t:

$$\varepsilon_t = y_t - d_t \tag{8}$$

A positive error corresponds thus to overestimating the wind production (such that upward reserves R^+ are needed), while a negative error underestimates the generation (resulting in the activation of downward reserves R^-). Table III provides the results of the different forecasting models, i.e. the TSO model (TSO), the static bidirectional decoder (Static), and its recalibrated version with a validation set (Val.), from scratch (Scratch), and with a fixed number of epochs (Epoch). Specifically, we represent the need of upward R^+ and downward R^- reserve capacity, and their associated costs C_r^+ and C_r^-. The total system costs are thus $C_r = C_r^+ + C_r^-$.

TABLE III
ANNUAL BALANCING COSTS ASSOCIATED WITH EACH METHODOLOGY.

	R^+[MW]	R^-[MW]	C_r^+[M€]	C_r^-[M€]	C_r[M€]
TSO	-153.35	374.57	13.43	32.81	46.24
Static	-265.49	318.72	23.26	27.92	51.18
Val.	-326.16	262.84	28.57	23.02	51.83
Scratch	-283.81	299.18	24.86	26.21	51.07
Epoch	-291.88	255.11	25.57	22.35	47.92

We see that prediction errors can significantly differ between tools. For instance, the TSO tends to strongly underestimate the wind generation, leading to high costs C_r^- for downward capacity. For most of our models, the prediction errors tend to be symmetrical (around zero), which is logical since positive and negative errors are equally penalized into the learning procedure. However, we also observe that our LSTM-based model (and their subsequent recalibrations) lead to heavy-tailed distributions of prediction errors in which large inaccuracies are more frequently encountered. In that regard, even though our models are more effective in general, they necessitate to rely on higher balancing needs to cover 99% of the imbalance situations. However, we observe that recalibrating the static model decrease these costs by 3.26 M€, i.e. a reduction of 6.3%, which stresses again the added value of this re-training phase. From these observations, an interesting perspective is to modify the model training to further penalize large errors.

Then, the financial penalties incurred to wind producers (due to the reserve activation) are computed. In general, these activation costs increase with the severity of the imbalance position, and vary with respect to the direction of the error (Fig. 9). Two cases can occur. On the one hand, if the wind producer generates less than expected (i.e. positive error ε_t), upward reserve will be activated, and he will pay the resulting activation price (which is higher than the price he has received in the energy market). This penalty cost Λ^+ is calculated by (9). On the other hand, if the generation exceeds the forecasted value (i.e. negative error ε_t), the producer will sell the surplus

energy at the downward activation price (which is lower than the price that he would have received in the energy market). The resulting opportunity loss Λ^- is calculated by (10):

$$\Lambda^+ = \sum_{t=1}^{n} (\lambda_t^{res+} - \lambda_t^{\mathrm{DA}}) \cdot \varepsilon_t \tag{9}$$

$$\Lambda^- = \sum_{t=1}^{n} (\lambda_t^{\mathrm{DA}} - \lambda_t^{res-}) \cdot \varepsilon_t \tag{10}$$

with λ_t^{res+} and λ_t^{res-} the upward and downward reserve prices, and λ_t^{DA} the electricity price on the day-ahead market.

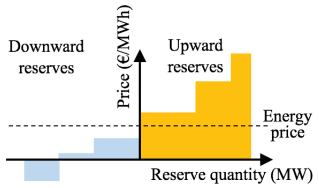

Fig. 9. Merit-order activation of reserves.

The total reserve activation costs over the year 2017 (for each prediction tool) are computed using the actual offers in the Belgian reserve market [16], and given in Table IV. We see that all recalibration techniques allow to decrease the costs of the static forecaster, up to a factor 2 for the model calibrated with an optimal number of epochs. This impressive gain is explained by the merit order effect (Fig. 9), in which large deviations are more heavily penalized. Hence, even slight improvements can significantly reduce the balancing fees. In addition, we also observe that opportunity losses Λ^- are much higher than penalty costs Λ^+, which arises from the fact that the price spread between the energy price λ_t^{DA} and the price for the generation surplus λ_t^{res-} is usually much higher than the difference between λ_t^{DA} and λ_t^{res+}. Wind producers are thus incentivized to overestimate their future generation (and thus to pay the moderate penalty λ_t^{res+}) rather than to receive the very low λ_t^{res-} when they generate more than expected.

TABLE IV
ANNUAL ENERGY COSTS FOR WINDS PRODUCERS.

	Λ^+ [M€]	Λ^- [M€]	Total costs [M€]
TSO	5.78	66.54	72.32
Static	13.13	78.28	91.41
Val.	19.09	30.05	49.14
Scratch	14.6	57.42	72.02
Epoch	17.82	27.29	45.11

In general, we conclude that relying on an (optimally-calibrated) model allows to save 3.3 M€ (for the reserve capacity) and 45 M€ (for the reserve activation) compared with a static model.

V. CONCLUSION

This paper was devoted to the day-ahead prediction of the onshore wind power generation. Firstly, we exploited the flexible nature of recurrent neural networks to implement different LSTM-based topologies, which all provided accurate results in regards to other state-of-the-art approaches. Secondly, we observed that recalibrating the model during its actual utilization can strongly improve the accuracy of predictions. In that regard, it appears that a recalibration with a fixed (optimally-chosen) number of epochs is a very effective solution compared to the traditional use of a validation set. Finally, we quantified the financial impact of prediction errors on both the TSO and wind producers. It was observed that, due to the structure of the balancing costs, even small prediction improvements can lead to substantial costs savings, thereby paving the way to further research in wind forecasting.

VI. ACKNOWLEDGEMENTS

The work is supported via the energy transition funds project "BEOWIND" organized by the FPS economy, S.M.E.s, Self-employed and Energy.

REFERENCES

[1] S. Rajagopalan and S. Santoso, "Wind power forecasting and error analysis using the autoregressive moving average modeling," IEEE Power & Energy Society General Meeting, Calgary, pp. 1-6., 2009.
[2] J. W. Taylor, P. E. McSharry and R. Buizza, "Wind Power Density Forecasting Using Ensemble Predictions and Time Series Models," *IEEE Trans. Energy Conver.*, vol. 24, no. 3, pp. 775-782, Sept. 2009.
[3] J. Figa-Saldaña, J. J. W. Wilson, E. Attema, R. Gelsthorpe, M. R. Drinkwater and A. Stoffelen, "The advanced scatterometer (ASCAT) on the meteorological operational (MetOp) platform: A follow on for European wind scatterometers," Canadian Journal of Remote Sensing, vol. 28, no. 3, pp. 404-412, 2002.
[4] G. N. Kariniotakis, G. S. Stavrakakis and E. F. Nogaret, "Wind power forecasting using advanced neural networks models," *IEEE Trans. Energy Conver.*, vol. 11, no. 4, pp. 762-767, Dec. 1996.
[5] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong and K. P. Wong, "Probabilistic Forecasting of Wind Power Generation Using Extreme Learning Machine," *IEEE Trans. Power Syst.*, vol. 29, no. 3, pp. 1033-1044, May 2014.
[6] J.-F Toubeau, et al., "Deep Learning-Based Multivariate Probabilistic Forecasting for Short-Term Scheduling in Power Markets," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1203-1215, March 2019.
[7] J. Lago, F. De Ridder, B. De Schutter,"Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms", Applied Energy, Volume 221, 2018, Pages 386-405.
[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.
[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, 2015.
[10] J.-F. Toubeau, J. Bottieau, F. Vallée, and Z. De Grève, "Improved day-ahead predictions of load and renewable generation by optimally exploiting multi-scale dependencies," Proc. 7th IEEE Conf. Innovative SmartGrid Technol., Dec. 2017.
[11] D. P. Kingma and J. L. Ba, "Adam : A method for stochastic optimization," 2014. arXiv:1412.6980v9
[12] X. Liao, N. Cao, M. Li and X. Kang, "Research on Short-Term Load Forecasting Using XGBoost Based on Similar Days," 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Changsha, China, 2019, pp. 675-678.
[13] J. Bottieau, L. Hubert, Z. De Grève, F. Vallée, J.-F. Toubeau, "Very Short-Term Probabilistic Forecasting for a Risk-Aware Participation in the Single Price Imbalance Settlement" IEEE Trans. Power Syst., in press.
[14] M. Bucksteeg, L. Niesen and C. Weber, "Impacts of Dynamic Probabilistic Reserve Sizing Techniques on Reserve Requirements and System Costs," IEEE Trans. Sustain. Energy, vol. 7, no. 4, pp. 1408-1420, 2016.
[15] K. De Vos, N. Stevens, O. Devolder, A. Papavasiliou, B. Hebb and M.-D. Johan, "Dynamic Dimensioning Approach for Operating Reserves: Proof of Concept in Belgium," Energy Policy, vol. 124, pp. 272-285, 2019.
[16] Elia website, "Available regulation capacity," https://www.elia.be/en/grid-data/balancing/available-regulation-capacity.