2018 IEEE Fourth International Conference on Big Data Computing Service and Applications

From data points to ampacity forecasting

Gated Recurrent Unit networks

Nicholas Popovic
Institute for Information Processing Technologies
Karlsruhe Institute of Technology
Karlsruhe, Germany

Gabriela Molinar
Institute for Information Processing Technologies
Karlsruhe Institute of Technology
Karlsruhe, Germany

Wilhelm Stork
Institute for Information Processing Technologies
Karlsruhe Institute of Technology
Karlsruhe, Germany

***Abstract*—Predicting the maximum current capacity of high voltage overhead lines (the so-called, ampacity forecasting) is an interesting and still explored subject in the energy industry. Due to its dependency on weather conditions (and the infamously complex mechanisms which affect these), calculating the maximum amount of power a conductor can transmit in the future remains challenging. The project this paper originates from, proposes the installation of several weather stations along an overhead line in order to acquire meteorological data in the locality of the conductor. This data is then used to model the local atmospheric dynamics using machine-learning algorithms. In this paper, experiments based on gated recurrent unit networks to forecast the ampacity of simulated overhead lines up to 24 hours ahead are presented and analyzed. The use of geographically distributed weather stations brings with it an improvement in the accuracy of the forecasts, while a higher precision remains as a goal to be accomplished. Moreover, the impact of feature selection and extraction, bearing in mind the underlying meteorological concepts, is also discussed.**

Keywords—ampacity forecasting, weather prediction, machine learning, gated recurrent unit, recurrent neural network, overhead line monitoring.

I. INTRODUCTION

The conductor temperature of an overhead line depends on the amount of transmitted current and the weather conditions in the locality of the line [1]. This temperature must neither reach nor exceed a fixed maximum temperature (normally 80°C). If it does, the conductor ages faster or can even be damaged. Based on this, a conservative margin for the maximum allowed amount of current that can be transmitted through an overhead line is fixed by the Transmission System Operators (TSOs). This margin (also called static rating) is used independently of the weather conditions and is calculated assuming the worst-case scenario (40°C ambient temperature, 0.6m/s transversal wind speed and 1000W/m^2 solar radiation [2])[1]. Since many lines are already working at their static rating limits however, this safety margin does not allow for an increment on operational renewable energy sources (RES), which are known to exhibit sudden surges in power output.

An expansion of the electrical network infrastructure could be the most straightforward solution. There are however several lines with available capacity, which in the end are not used due to those conservative margins. In winter, for example, the amount of energy, which can be transmitted on a conductor, before its temperature reaches the maximum permissible value could be more than twice the amount allowed by the static rating margin [3]. Using appropriate monitoring systems instead, a more efficient use of the capacity of the network could be achieved.

If, moreover, a forecast of the maximum current capacity of an overhead line is available (the so-called ampacity forecasting), TSOs can better schedule the power generation and transfer, which ensures a more reliable use of the electrical network. The ampacity of an overhead line depends on the conductor temperature, which in turn depends on the local weather conditions. If a local weather forecast along the line is available, an ampacity prediction for that conductor can be calculated. To accomplish this, local weather data shall be collected along the line and used to train a model, which can then be used to predict changes in the ampacity rating of the line.

Among the main objectives of this project are:

- To understand what meteorological processes can be observed and modeled using the data generated by the sensor network and how important these are for predicting the ampacity one day ahead, based on historical weather data.
- To identify a machine learning algorithm capable of forecasting line ratings without exceeding 20% mean average percentage error (MAPE) [4];
- To specify the optimal structure of the sensor network (density of weather stations, in particular) so as to provide sufficient spatial coverage and resolution, while remaining practical and financially feasible.

The experiments described in this paper were designed primarily to provide insight into the first and second objectives. Based on the test results, the next experimental steps and an insight relevant to the third objective are expected and described in the outlook section (part X).

The next sections are organized as follows: first (sec. II), an overview of existing algorithms or methods for ampacity and weather forecasting is presented; then the selected algorithm and datasets are explained considering advantages and

[1] There is also a seasonal static rating, where the fixed margins are changed twice or four times a year.

0-7695-6396-1/18/$31.00 ©2018 IEEE
DOI 10.1109/BigDataService.2018.00037

disadvantages with respect to other possibilities; in section V, the features to be tested and the reasons for their selection are presented. Afterwards, in sections VI and VII, the hyperparameter optimization process and the cost function are described. Section VIII presents and discusses the set of experiments and their results. Finally, conclusions and outlook are found in the last two sections of this paper.

II. State-of-the-art: Ampacity and Weather Forecasting

As mentioned above, the ampacity of an overhead line is related to weather data, i.e. if a weather forecast is known, an ampacity forecast can be calculated. In literature, just a few methods and experiments for ampacity forecasting can be found.

In [5] numerical weather prediction (NWP) methods are used. These methods solve a set of differential equations based on physical phenomena, using numerical methods. Running those models at slightly different initial conditions, the so-called ensemble forecasting, a range of possible future values for each weather parameter is generated.

NWP describes the weather dynamics in a grid, which can be as small as 2.8Km wide [6]. This resolution is excellent for many applications. It is however not enough for an ampacity calculation. The conductor is too sensitive to weather changes in its vicinity (especially under small changes on wind speed and direction). Hence, having fixed values across a grid unit of 2.8Km wide generates errors in the calculation of the ampacity.

To obtain a better ampacity forecasting of an overhead line, the measurement of weather parameters along and in the locality of the conductor is necessary [4]. The gathered data is then applied to machine learning algorithms, which are tuned to *learn* the most probable behavior of the parameters in the near future.

Applying machine learning algorithms to weather or ampacity forecasting is still being explored by researchers. The chaotic behavior of weather makes a prediction on several days ahead a challenging task. However, some steps forward can be seen in [7], where probabilistic methods are used to predict weather parameters 24 hours ahead. The proposed architecture combines a bottom-up predictor for each individual variable with a top-down deep belief network that models the joint statistical relationships. Besides, in [8] ARIMA models are compared to Feedforward Neural Networks for a prediction of wind speed in the following 4 hours. These models, however, cannot predict farther into the future. Moreover, wind predictions are of great importance for ampacity forecasting, which means that especially for wind data, an improvement of the accuracy of state-of-the-art solutions is necessary.

III. Gated Recurrent Unit Networks for Weather Forecasting

The experiments shown in this paper are developed using Recursive Neural Networks (RNN). The selection of this algorithm was based on the possibility of the network to learn long-range dependencies in time [9]. This means it can learn the behavior of sequences of values, which is a key-point when doing weather forecasting. A simple Feedforward Neural Network has the disadvantage that it performs regression considering the measurements as independent samples. The weather conditions in a time t_0, however, are dependent on the conditions at t_{-1}, t_{-2}, t_{-3}, etc.

Specifically, from the family of RNN, Gated Recurrent Unit (GRU) networks were implemented. As seen in figure 1, a GRU can be seen as a black box. Given the current input (x_t) and its previous hidden state (s_{t-1}), it computes the next hidden state (s_{t+1}). Based on this, the network ends up having memory at each cell, what allows the model to learn sequences of data.

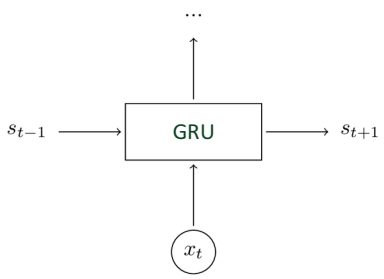

Fig. 1. A Gated Recurrent Unit can be seen as a black box. Given the current input (x_t) and its previous hidden state (s_{t-1}), it computes the next hidden state (s_{t+1}). [9]

IV. Dataset

To run and test the experiments, weather data distributed along overhead lines are necessary. As a practical, easy and cost effective solution, a simulation of the line routes was developed based on nine already installed weather stations from the National Oceanic and Atmospheric Administrations (NOAA) Idaho National Laboratory Mesonet (see figure 2). The relatively high density of weather stations provided by this database allowed a simulation of five different sensor networks on five hypothetical overhead lines, as figure 3 shows (each point in the figure is a weather station and the dashed lines are the simulated line routes). The length of each line shall not exceed 80 to 90 km, otherwise the ampacity of the line is no longer determined by the conductor temperature, but other constraints that are out of the scope of this paper (for more information refer to [10]).

Fig. 2. Locations of the selected weather stations from the National Oceanic and Atmospheric Administrations (NOAA) Idaho National Laboratory Mesonet.

The weather measurements used to train the GRU network were sampled hourly from 2007 to 2013 (35244 samples). Data from the whole year 2014 was selected for validation (4768 samples), and from 2015 to 2016 for testing (7922 samples). The

weather parameters, used as input variables for the model, were carefully selected and different combinations were considered. Each set of variables formed a different experiment. Those parameters and the reasons for their selection are explained in the next section.

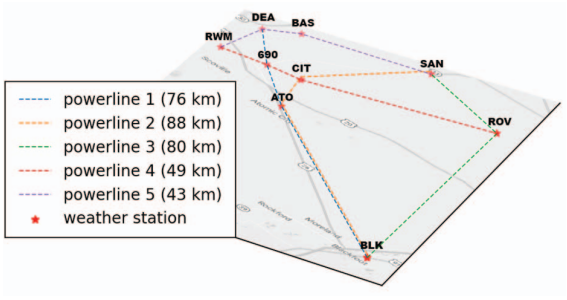

Fig. 3. Set of weather stations selected to train the GRU Network and their arrengement as simulated overhead lines.

V. Feature Identification

Machine learning algorithms are theoretically capable of separating relevant from irrelevant information during the training process. Given a limited amount of training data, however, reducing the set of input features to its most important components can drastically improve performance. It is therefore advantageous to process the raw data in such a way that the information density of the feature set is maximized. Besides normalization and transformations such as Principal Component Analysis (PCA), deriving other relevant values from the raw data can lead to better results (often referred to as feature extraction).

From the following features, different input sets were created and used at each experiment described in section VIII:

- Wind: The wind speed is first normalized using the maximum measured wind speed value. This results on elements in the range 0 to 1.

  Then, the normalized wind speed and direction are transformed to their Cartesian components (tuples of the form (X,Y)) and rotated using PCA. This was done in order to make regular wind patterns easier to identify. In [11], the analysis of historical data has shown that drainage and upslope flow from the surrounding mountains result in two main wind directions (Northeast and Southwest). Applying the PCA makes this more evident for the training process.

- Ambient temperature: normalized using the mean temperature for the respective day of the year (in order to remove seasonality) and 3 times the standard deviation. Hence, the resulting values are between -1 and 1.

- Adiabatic lapse rate: this parameter describes the change in ambient temperature w.r.t. height from the ground (see eq. 1). The units of the lapse rate Γ are K/m. It includes information about the vertical inversion of the temperature gradient, one of the main factors in the change in wind direction at night [11].

$$\Gamma = -\frac{dT}{dz} \qquad (1)$$

- Relative humidity: this feature comes from the measurements as a percentage. These are simply transformed to the range from 0 to 1.

- Air pressure: normalized dividing the value between the mean pressure and 3 times the standard deviation.

- Geostrophic wind approximation [12] [13]: is a theoretical wind vector, whose magnitude is in m/s and is directed parallel to the isobars (lines of constant pressure). This vector is generated by calculating the Coriolis force and the horizontal pressure gradient force for a given location.

VI. Hyperparameter Optimization

Hyperparameters define the structure of a network and the characteristics of its training process. The number of layers, number of neurons, batch size, momentum, learning rate, the learning rate decay and the L2 regularization constant were the parameters to optimize in this project. For any different selection of each of these elements, a different learning capacity can be achieved, i.e. different accuracies. Therefore, to attain the best possible results from a learning algorithm, the optimization of the hyperparameters is crucial.

For this paper, seven hyperparameters were tuned to optimize the performance of the algorithm. Each hyperparameter has a different range of values:

- Number of Neurons: a discrete variable was used for this hyperparameter, since there are no appreciable differences between networks with n or $n+1$ neurons, when n is relatively big. The values that this variable can take are 8, 16, 32, 64, 128, 256 or 512.

- Number of layers: the optimization was run trying networks with up to 3 layers.

- Batch size: the amount of samples per batch can be 32, 64, 128, 256 or 512.

- Momentum: For a specific weight in the network, the rate at which the learning rate can increase was optimized continuously in the range from 0 to 1.

- Learning rate: the working range was set up from 10^{-8} to 0.1.

- Decay: the rate at which the learning rate decays can take continuous values from 0 to 1.

- L2-Regularization (λ): To reduce model overfitting, this variable can take values from 10^{-5} to 0.1.

There are several methods for finding the optimum hyperparameters for an application. The most used are explained in [14] [15] and are summarized here:

- Manual Search: each hyperparameter is changed manually depending on information previously known about the system.

- Grid Search: Based on knowledge about the problem, ranges for the hyperparameters are identified. Then a manual search is executed inside those ranges, selecting uniformly distributed elements.
- Random Search: This is similar to grid search but the elements to be tried are not uniformly distributed but randomly distributed inside the established ranges.
- Bayesian Optimization: This method selects first randomly hyperparameters and samples their performance. Then using the information obtained from the experiments as a heuristic, it decides how to select the hyperparameters for the next experiment until it finds the optimal combination.

Because of their performance, Bayesian optimization and random search were selected to optimize the hyperparameters of the GRU network. Here three different acquisition functions for the Bayesian optimizer were compared to random search (see figure 4): Maximum Probability of Improvement (MPI), Lower Confidence Bound (LCB) and Expected Improvement (EI). The comparison was made in terms of the mean distance to the minimum error obtained from all the experiments. For this, the libraries GPyOpt and NumPy from Python were employed.

As the figure 4 shows, actually random search had a better performance for this dataset compared to the more complex three Bayesian optimizers. The first one is hence used in the rest of this paper.

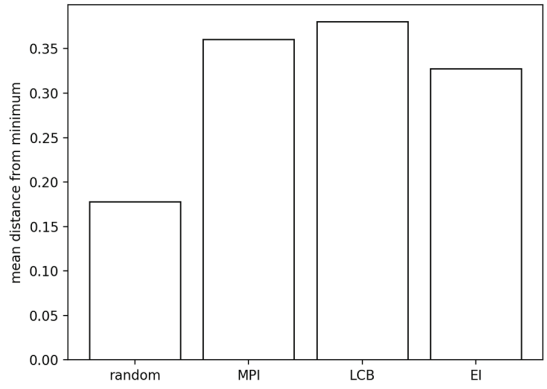

Fig. 4. Performance comparison over the same dataset of random search and three Bayesian optimizers: Maximum Probability of Improvement (MPI), Lower Confidence Bound (LCB) and Expected Improvement (EI). Random search has less errors than the more complex Bayesian optimizers.

VII. Cost function

The Root Mean Squared Error (RMSE) is often used as cost function to train machine-learning algorithms. During the training process, this error is minimized by changing the weight and bias values between neurons in the network through backpropagation.

Since the end-parameter to be predicted is the ampacity of an overhead line, it seems reasonable to consider the importance of each weather parameter in the ampacity model. The following graphics show how is the relationship between the absolute ampacity error of a line and the ambient temperature (Fig 5.a), solar radiation (Fig 5.b) or wind speed (Fig 5.c) errors.

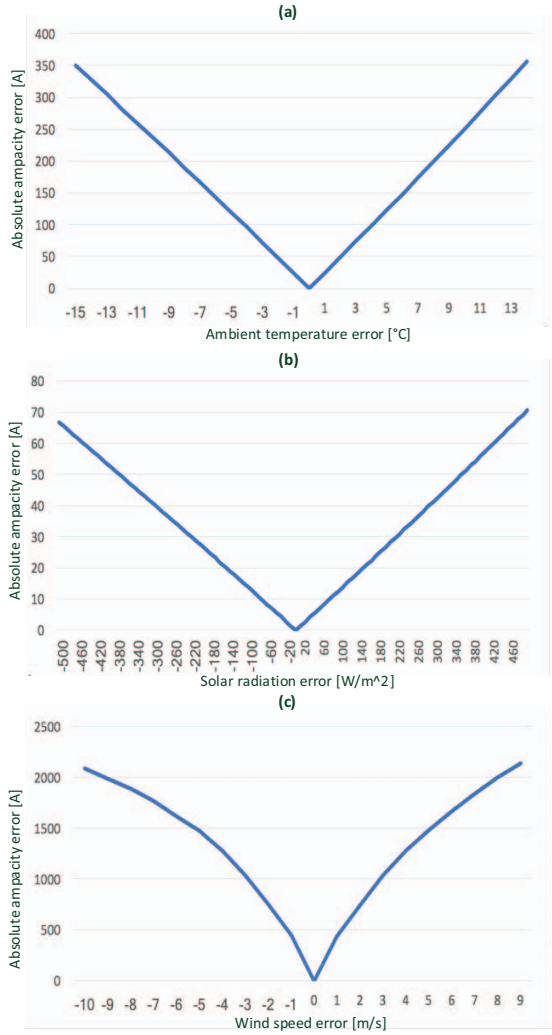

Fig. 5. Relationship between the absolute ampacity error [A] and (a) the ambient temperature error [°C], (b) solar radiation error [W/m^2] and (c) wind speed error [m/s].

Considering the relationships in figure 5, a weighted RMSE cost function was tested for training the network. The selection of the weighting factors was done bearing in mind that a strong difference between weights can result in an oversight of the temperature and solar radiation predictions quality. Therefore, the selected weights were 26.2 A/°C for the ambient temperature, 450 A/ms^{-1} for wind speed and 0.15 A/Wm^{-2} for solar radiation[2]. This means that during training, the error of the

[2] Tests comparing different values for the weighting of each weather parameter have shown that the wind direction weight can be left as 1 for simplicity and the results are not considerably altered.

wind forecast is prioritized over the error of the temperature and solar radiation. The effect of this weighted cost function is illustrated in figure 6, where it is seen clearly how the wind speed parameter have been given greater importance, hence first optimized, than temperature and solar radiation.

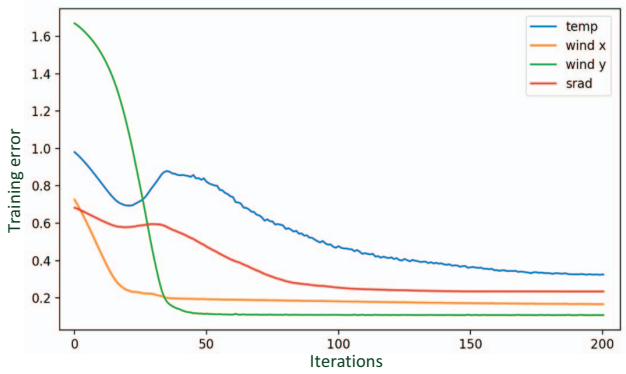

Fig. 6. Example of RMSE per variable over training using the adjusted cost function. As expected, the optimizer begins by focusing on reducing the errors regarding wind (which was weighted significantly higher) and adjusts temperature and solar radiation accuracy later.

Both cost functions (weighted and unweighted RMSE) were compared (see figure 7). They show a similar Mean Absolute Percentage Error (MAPE) after training the model for 1000 iterations. Therefore, for simplicity, the unweighted RMSE is employed in the rest of this paper, i.e.

$$J = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad (2)$$

From equation (2), n is the number of forecasted values $\hat{y}_i$ and y_i are the measured data values.

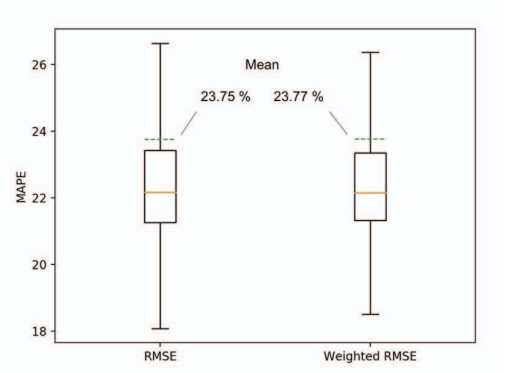

Fig. 7. Comparison of weighted and unweighted RMSE functions, with respect to the MAPE on the validation dataset for 540 randomly selected configurations of hyperparameters. Each configuration was run once per cost function. A significant performance improvement was not observed.

VIII. Experiments, Results and Discussion

One of the main goals of this research project is to identify how the geographical distribution of a sensor network could improve the forecast over that of a single weather station. Therefore, several sets of inputs were arranged and their results were compared to a baseline experiment. All tests were based on a GRU network, which was implemented using TensorFlow.

The output labels, which the algorithm attempts to predict, were the same for all experiments and consisted of the three weather parameters needed to calculate the ampacity of an overhead line:

- Ambient temperature
- Wind (speed and direction)
- Solar radiation

The input features were organized, through all the experiments, as blocks of 48 hours. The samples of the first 24 hours are "past data" (features) for the algorithm and the next 24 hours work as labels to check the accuracy of the model.

Applying the CIGRE approximation for the steady state thermal rating of a conductor [1] on these predicted values and comparing this result to the actual line rating (CIGRE approximation for true, measured weather conditions) leads to a Mean Absolute Percentage Error (MAPE). This value is the metric used hereafter to calculate the performance of an experiment, which will also help to compare it to the baseline test.

A. Preparation: baseline model

An initial run (also referred as the baseline experiment) had as input features the normalized measurements recorded at the same station for which the output was to be predicted. This meant six input variables, namely ambient temperature, wind vector (X and Y components), solar radiation, barometric pressure and relative humidity. In all of the subsequent experiments, these input variables were used and new features were added. This approach resulted in a forecast accuracy of 19.22% (MAPE) / 318.01A (MAE).

B. First stage: single station inputs

For the first two experiments, the lapse rate was added to the input feature set. The idea behind this selection was to include information about the vertical inversion of the temperature gradient, one of the main factors in the change of wind direction at night [11].

Two experiments were conducted at this stage: one in which the lapse rate value was added to the input set without change, and a second experiment in which the added feature consisted of only its sign (two possible values: -1 or 1). The addition of these features brought a slight improvement in the mean errors of the individual outputs (especially for the solar radiation). However, an improvement in the ampacity forecast was not observed (see comparison to baseline in figure 11).

C. Second stage: auxiliary station measurements

For these experiments, data from the weather stations forming the shown overhead lines in figure 3 was added to each corresponding feature set. When adding all weather parameters from all auxiliary sensor nodes, however, the experiment results in very poor performance (+0.29% (MAPE) / +1.02% (MAE)). Therefore, a different approach was developed. Wind and pressure data from the auxiliary sensor nodes was added separately to the input sets. Wind data was included in the

feature set in order to mimic the approach for wind pattern recognition used in [11]. Pressure data was added due to the direct correlation between pressure changes and wind conditions.

While the addition of pressure measurements from auxiliary weather stations on the simulated overhead lines resulted in an improved MAE (-1.36%) the MAPE was worse than the baseline (+0.27%). Feature sets including only wind measurements (X and Y components) from the auxiliary stations, on the other hand, brought the first improvements in both the MAE (-3.60%) and the MAPE (-0.12%) of the ampacity prediction.

D. Third stage:synoptic pressure gradient

The third group of tests was focused on using additional, remote weather stations for feature extraction. This was done in order to include information about the synoptic scale (approx. 1000 km range) of the weather system. Observing processes at this scale is a key part to understand weather conditions [13].

A total of 13 weather stations, at distances ranging from 80 to 950 km from the primary sensor nodes, were used to reconstruct the distribution of air pressure surrounding the local weather stations using interpolation. This information was then used to derive the geostrophic wind approximation, which can be calculated using the pressure gradient (see figure 8). The addition of this value to the input set resulted in a mean improvement of 0.29% over the baseline MAPE and an improvement of 2.79% in MAE.

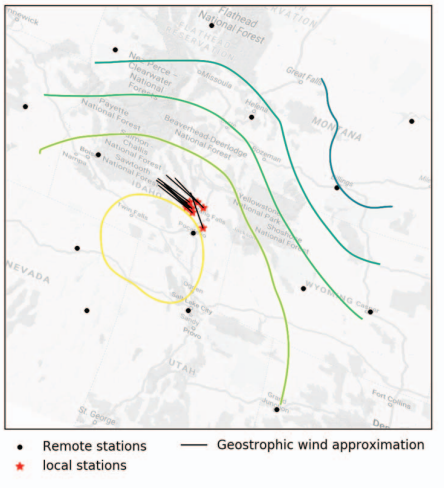

Fig. 8. The geostrophic wind approximation for the considered region of weather stations is drawn in this image as straight lines pointing in the same direction as the calculated vectors. This calculation is based on the air pressure gradient of the remote weather stations measurements.

E. Combination of experiments

From all experiments described, the two best feature-sets, namely the wind measurements from the auxiliary weather stations simulating overhead lines and the geostrophic wind approximation, were combined. Together they formed the last experiment described in this paper. This resulted in an improvement of 0.63% over the baseline MAPE and a 6.12% improvement in MAE, making it the best performing configuration tested.

Figure 9 shows the mean absolute error (MAE) of both the baseline and the combined experiment, with respect to the forecast scope in hours. It can be seen that the improvement in the forecast of the combined experiment is evident throughout the scope.

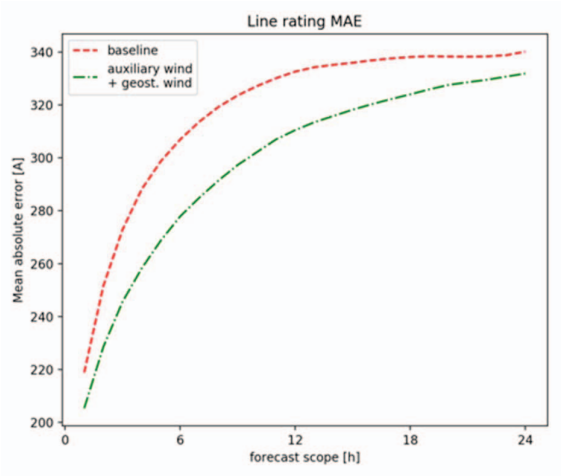

Fig. 9. Mean absolute error of the line rating forecast over the forecast scope for both the baseline experiment, and the final combined data set.

Figure 10 shows the error distribution of predicted line ratings for both the baseline and the combined experiment. Both approaches tend to predict lower values than the true ones for the ampacity of the line. The cause of this is the use of the MAPE for hyperparameter optimization. When using the MAPE, a lower prediction than the real value cannot exceed an error rate of 100%, while a higher prediction can do it. Since an underestimation of the ampacity is preferable for TSOs than an overestimation (because of conductor safety), it makes sense to use this value for scoring forecasts. However, during hyperparameter optimization this kind of scoring might be less advantageous, since a small improvement in MAE might lead to a worse MAPE.

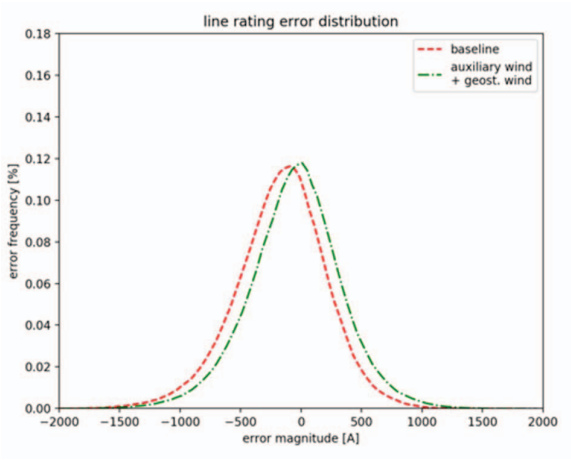

Fig. 10. Error distribution of predicted line ratings for both the baseline experiment, and the final combined data set.

F. Comparison of all experiments

A comparison of the baseline performance and the other experiments explained in the last sections is shown in figure 11. Here can be seen that the best result was obtained from the combination of the wind measurements from the auxiliary weather stations simulating overhead lines and the geostrophic wind approximation.

Experiment	Change in MAE	Change in MAPE
Lapse rate	+ 0.96 %	+ 0.16 %
Lapse rate (sign)	+ 0.40 %	+ 0.05 %
Auxiliary pressure	- 1.36 %	+ 0.27 %
Auxiliary wind	- 3.60 %	- 0.12 %
Auxiliary all	+ 1.36 %	+ 0.29 %
Geostrophic wind	- 2.79 %	- 0.29 %
Geos. + Aux. wind	- 6.12 %	- 0.63 %

Fig. 11. Comparison between 7 different tests with respect to the baseline model in terms of the MAE and MAPE.

IX. Conclusion

There exists a vast amount of knowledge, product of the intensive research done by meteorologists, which should be noticed when applying learning algorithms in this area. Even though a large amount of data is available, the careless use of it is clearly not the way to obtain successful results.

Combining the geostrophic wind approximation with distributed wind measurements on the mesoscale proved to be the most effective method. In fact, a smaller MAPE than 20% was obtained, namely 18.59%, as required by the second objective of this project. The set of features present in this experiment is sufficient to separate the effects of local wind patterns and synoptic scale weather systems, making current wind conditions easier to understand. A noticeable overall performance increase in wind and solar radiation forecasts was led by additional information supplied to the machine-learning algorithm. This resulted in higher accuracy and precision of the ampacity prediction (6.12% improvement in MAE and 0.63% improvement in MAPE). An increase in temperature accuracy, while observed, was less pronounced.

One goal of this project was to understand how useful the distribution of weather stations along an overhead line would be for forecast accuracy. While a weather station's proximity to the conductor plays a big role in the accuracy of a line rating calculation [4] and a high density of weather stations should therefore be used, a wide spatial coverage is more important for forecast accuracy. As seen in the final experiment, combining both approaches appears to be the best solution.

While noticeable improvements were made using the right input features, it was found that the largest improvements were made with respect to the forecasts accuracy, as opposed to its precision. This revealed a systematic problem concerning the use of the mean absolute percentage error (MAPE) during hyperparameter optimization.

X. Outlook

From all the experiments done and described in this paper, several questions have arisen, which could be start-points for future research. First, a forecast of weather parameters in the region between weather stations and along the overhead line has not been attempted in this work. However, it would be of great interest for the ampacity forecasting application. The interpolation of the data should be done considering the terrain and geographical factors from the surrounding of the line routes.

Second, the data recorded at the remote weather stations could be used more extensively: (1) the improvements achieved in the experiments were most noticeable within the first 8 hours of the forecast. The horizontal pressure distribution, used here for calculation of the geostrophic wind approximation, could be used to allow the tracking of synoptic scale weather systems. This might increase the temporal scope of the improvements. (2) The temperature gradient, not used so far, is another important variable in atmospheric processes and it could be calculated using the available data.

Third, more experiments focusing on increasing the forecasts precision need to be conducted. An improved accuracy is a good first step. However, an increase in precision would be desirable regarding the robustness of the system.

With respect to the methodology used in this project, there are a few more changes worth exploring. The evaluation of the experiments revealed that when generating forecasts on real data sets, errors in individual climate variables are not as closely linked to errors in ampacity forecasts as an examination of the equations would imply. Although large improvements in climate variable accuracy led to improvements in the line rating accuracy, small improvements shifting the mean error towards zero might lead to a worse score due to the characteristics of the MAPE. This difference in the loss function of the ANN and the objective function used in hyperparameter optimization might explain why the random search outperformed Bayesian optimization. This discovery calls into question the usefulness of the line rating accuracy as the objective function for optimization and model selection. Perhaps it would be better to optimize a weather forecast entirely without ampacity calculations and use these only for the final evaluation.

In the results shown in this paper, only GRUs were tested. A change of the type of the RNN algorithm (instead of GRU, the use of LSTM for example) could also lead to improvements in performance.

References

[1] Cigré, *Guide for Thermal Rating Calculations of Overhead Lines (601),* 2014.

[2] D. Douglas, W. Chisholm und others, „Real-Time Overhead Transmission-Line Monitoring for Dynamic Rating," Bd. 31, pp. 921-927, December 2016.

[3] M. Muhr, S. Pack und S. Jaufer, „Usage and Benefit of an Overhead Line Monitoring System," in *2008 International Conference on High Voltage Engineering and Application*, Chongqing, China, 2008.

[4] N. Doban, „Building predictive models for dynamic line rating using data science techniques," Stockholm, 2016.

[5] Energy&Meteo, „ORKA 2: Ensemble-Methoden und -Modelle," [Online]. Available: https://www.energymeteo.de/projekte/orka_2.php.

[6] D. W. (DWD), „Regionalmodell COSMO-DE," [Online]. Available: https://www.dwd.de/DE/forschung/wettervorhersage/num_modellierung/01_num_vorhersagemodelle/regional modell_cosmo_de.html.

[7] A. Grover, A. Kapoor und E. Horvitz, *A Deep Hybrid Model for Weather Forecasting,* Sydney, Australia, 2015.

[8] J. C. Palomares-Salas, J. J. G. de la Rosa, J. G. Ramiro, J.Melgar, A. Agüera und A. Moreno, *ARIMA vs. Neural Networks for Wind Speed Forecasting,* 2009.

[9] D. Britz, „Recurrent Neural Network Tutorial, Part 4 – Implementing a GRU/LSTM RNN with Python and Theano," October 2015. [Online]. Available: http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/.

[10] M. Rashid, Power Electronics Handbook: Devices, Circuits and Applications, Burlington, MA : Academic Press, 2009.

[11] K. Clawson, R. Eckman, N. Hukari, J. Rich und N. Ricks, „Climatography of the Idaho National Laboratory 3rd Edition," NOAA, USA, 2007.

[12] J. Holton, „An Introduction to Dynamic Meteorology," Elsevier, USA, 2004.

[13] F. Lutgens und E. Tarbuck, The Atmosphere: An Introduction to Meteorology, New Yersey, USA: Prentice-Hall, 1986.

[14] J. Snoek, H. Larochelle und R. P. Adams, „Practical Bayesian optimization of machine learning algorithms," *Advances in Neural Information Processing Systems,* Bd. 25, pp. 2960-2968, 2012.

[15] Y. Bengio, *Practical Recommendations for Gradient-Based Training of Deep Architectures,* 2012.