Weather-based Fault Prediction in Electricity Networks with Artificial Neural Networks

Christina Brester
Department of Environmental and Biological Sciences
University of Eastern Finland
Kuopio, Finland
kristina.brester@uef.fi

Harri Niska
Department of Environmental and Biological Sciences
University of Eastern Finland
Kuopio, Finland
harri.niska@uef.fi

Robert Ciszek
A.I. Virtanen Institute for Molecular Sciences
University of Eastern Finland
Kuopio, Finland
robert.ciszek@uef.fi

Mikko Kolehmainen
Department of Environmental and Biological Sciences
University of Eastern Finland
Kuopio, Finland
mikko.kolehmainen@uef.fi

***Abstract*—Predicting weather-related outages in electricity networks is an important issue for distribution system operators. In this study, we apply a data-driven approach and train artificial neural networks to predict faults in the electricity network. In our experiments, we utilize the meteorological data and fault records collected for the period of 1.1.2011 – 31.12.2013 in central Finland. Assuming that there might be long-term dependencies between weather conditions and faults in the network, we investigate simple recurrent neural networks, long short-term memory networks, and traditional multilayer perceptrons. Taking into account the meteorological observations preceding faults and varying this period from several hours to several days, we found that 6 hours prior to faults included the sufficient information to make accurate predictions. Also, there was no need in more complicated recurrent neural networks as multilayer perceptron was able to predict events with the large number of faults more accurately. Besides, while forecasting all types of faults and wind-related faults only, oversampling allowed the model to predict rare high peaks.**

Keywords—electricity networks, fault prediction, artificial neural networks, weather-related outages, risk assessment

I. Introduction

Weather-related electrical outages are one of the major challenges for distribution system operators (DSOs). High-speed wind can fall trees on the overhead lines or make them swing, which results in a short circuit as lines contact each other. In winter, snow may weight on trees likewise causing them to fall on the overhead lines and low temperatures along with collected moisture may cause damage of equipment through freezing. In summer, lightning strikes occurring in the vicinity of power lines can cause high frequency transients and extreme temperatures may result in transformer failures. The extent of outages caused by weather-related faults ranges from minor day-to-day outages to outages affecting large portions of the grid leading to substantial financial losses.

In this context, data-driven forecasting models could play an important role for DSOs. Since restoration of power supply after an outage requires to dispatch the repair crew without delay, fault risk assessment would enable DSOs to reserve the needed workforce in advance. This would result in financial savings for the operator and increased customer satisfaction due to decreased outage lengths.

However, predicting weather-caused faults is challenging. Using common sense, it is possible to guess that a severe storm predicted for tomorrow also increases the odds of faults occurring tomorrow in power systems. Nevertheless, faults are known to have more stochastic nature. Therefore, data-driven models should be able to lessen uncertainty involved in forecasting by analyzing the weather conditions preceding faults and capturing subtle interactions that might be complicated for human experts.

In this study, we apply artificial neural networks (ANNs) to predict weather-based faults in the electricity network located in central Finland. We investigate the effectiveness of multilayer perceptrons (MLPs), simple recurrent neural networks (RNNs), and more complicated long short-term memory (LSTM) networks [1, 2]. We use the meteorological observations provided by Finnish Meteorological Institute (FMI) [3] and combine them with fault records received from the Finnish DSO Elenia Oy. First, we do not focus on a specific subtype of faults, e.g. faults caused by storms, and let ANNs learn all nonlinear effects of meteorological parameters on the fault occurrence. Then, we vary the size of time windows that include previous weather observations and check how the model performance changes in response. Larger time windows might allow the model to contextualize the current observations better. Also, for the best model, we perform the sensitivity analysis to expose the most important variables. Lastly, in the additional experiment, the best found ANN-architecture is used to predict wind-related faults.

The existing studies on the electricity grid fault risk assessment using the meteorological data have applied shallow ANN-models [4, 5], statistical approaches [6, 7, 8], and other methods such as fuzzy logic [9, 10]. On the one hand, RNNs and LSTMs have shown their effectiveness in time series forecasting, wherein sequences of inputs are effectively analyzed [11, 12]. We may assume that several previous weather observations might be relevant to predicting faults, and the memory of RNNs and LSTMs is an effective tool to model these dependencies. Besides, RNNs and LSTMs benefit from the large amount of the training samples [13, 14], which availability is increasing for the problem considered as digital records of faults and weather observations are being collected by DSOs and meteorological institutions correspondingly. On the other hand, some studies have demonstrated that MLPs can effectively utilize samples from consecutive time points and outperform

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

LSTMs when only a few recent inputs are necessary to make accurate predictions [15]. Therefore, a comparison of different ANNs in the electricity grid fault risk assessment can be seen as a topic of interest. Additionally, this study helps to reveal the number of previous weather observations sufficient to reach the highest model predictive ability.

The novelty of this study grounds on the application of ANNs for the weather-related fault prediction in the real grid environment in the Nordic conditions. The main emphasis is on the evaluation of forecasting performance as well as on the practical data processing stages required to enhance the capability of ANNs to deal with the characteristics of the fault data such as rare peak fault events. Moreover, we shed light on several important points for the weather-based fault prediction: in short-term forecasting, MLP is more suitable than LSTM or simple RNN, oversampling is helpful, and Index of Agreement enables to reveal models predicting rare peaks better. Developing a totally new algorithm was out of the scope in this study.

II. Data Description

The Elenia grid spans regions of Kanta-Häme and Päijät-Häme, Pirkanmaa, Central Finland and Ostrobothnia. The locations of the grid's substations are presented in Fig. 1. The dataset provided by Elenia contained in total 36874 unique faults that occurred between 1.1.2011 and 31.12.2013. The dataset entries consisted of the latitude and longitude of the substation which the fault was associated with, the start time of the fault in temporal resolution of one minute, the type of the fault, and the classification for the source of the fault. The exact location of the faults was not presented in the dataset, but the faults were associated with the Elenia substations within 50km distance.

The faults classified as undefined formed the majority of the faults, comprising 84.49% of all faults. The most prevalent defined type was the faults caused by wind and storm, which encompassed 7.26% of all faults. The remaining 8.25% was composed of the faults caused by snow and ice, thunder, other weather-related faults, structural faults, and faults of an unknown origin.

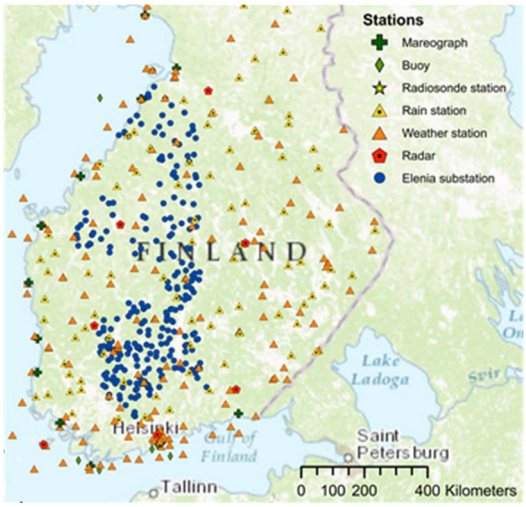

Fig. 1. Elenia substations and FMI observation stations.

TABLE I. Total Fault Counts During Years 2011-2013

Fault Source	Count	Percent
Undefined	311153	84.49%
Wind and storm	2676	7.26%
Snow and ice (tree)	1084	2.94%
Unknown	627	1.7%
Thunder	530	1.44%
Structural	456	1.24%
Snow and ice (load)	221	0.6%
Other weather related	127	0.34%

The faults occurred within the same hour were summed over the Elenia grid to create a single network level dataset. On the whole grid level, the majority of the faults occurred as single events, even when time resolution of one hour was used. However, there were several abrupt increases in the number of faults that needed to be predicted in advance. Table I provides an overview on the fault types and their number during the period of three years.

The Elenia fault datasets were combined with the meteorological observation data provided by FMI, which were queried using the Open Geospatial Consortium compliant API. The hourly weather observations from all the FMI observation stations made between 1.1.2011 and 31.12.2013 were queried. For each Elenia substation, the hourly weather measurements from FMI stations within the 100km-distance were averaged. The substation level observations were further averaged over the whole network to create a dataset describing the average weather over the network in time resolution of one hour.

The subnetwork level fault and weather time series were concatenated by combining the summed faults with the meteorological variables. Table II presents the whole vector of input variables used in modelling. In addition to the meteorological variables, discontinuous timing variables month and hour were included as sine and cosine transformed variables, resulting in four additional variables. Also, wind direction was included as two sine and cosine transformed variables. In total, the feature vector of the combined time series contained 15 variables.

TABLE II. Meteorological and Timing Variables Used For Predicting Faults

№	Variable	№	Variable
1	Air temperature, °C	9	Air pressure, hPa
2	Wind speed, m/s	10	Horizontal visibility, m
3	Gust speed, m/s	11	Cloud amount, 1/8
4	Relative humidity, %	12	Month (sine)
5	Wind direction (sine)	13	Month (cosine)
6	Wind direction (cosine)	14	Hour (sine)
7	Dew-point temperature, °C	15	Hour (cosine)
8	Snow depth, cm		

III. METHODS

A. Multilayer Perceptrons

Traditional MLPs are typically composed of stacked layers of interconnected cells [16]. Weighted connections between the cells define how much of the value of a cell from the previous layer of the network is propagated to each individual cell in the next layer. Cells sum the incoming values and transform the sum using an activation function, e.g. hyperbolic tangent or logistic sigmoid function. Each cell in the lowest layer of the network, i.e. the input layer, is given a numeric value based on the input vector, which is propagated upwards through the network using the aforementioned rules. The upmost layer, i.e. the output layer of the network, performs a classification or regression based on the summed inputs received from the lower layers and the error between the result predicted by the network and the known true result is calculated using an objective function, e.g. mean absolute error. The network is taught by propagating backwards the gradient of the loss function with respect to the weights and by adjusting the weights accordingly in a process known as backpropagation.

If we denote the input *d*-dimensional vector of MLP by $i = [i_1, \dots, i_d]^T \in \mathbb{R}^{\mathrm{d}}$, the input vector for layer $l \in [1, L]$ of MLP with L layers by $x = \left[x_{l,1}, \dots, x_{l,n_{l-1}}\right]^T \in \mathbb{R}^{n_{l-1}}$, the output of the *l*-th layer by $y_l = \left[y_{l,1}, \dots, y_{l,n_l}\right]^T \in \mathbb{R}^{n_l}$, the weight matrix of the *l*-th layer by $W_l \in R^{n_l \times n_{(l-1)}}$, the bias term vector $b_l \in \mathbb{R}^{n_l}$ and the number of cells in the layer l by n_l, we can present the MLP networks as:

$$x_0 = y_0 = i, \tag{1}$$

$$v_l = W_l y_{l-1} + b_l, \tag{2}$$

$$y_l = f_l(v_l). \tag{3}$$

where $f_l(v_l) = \left[f_{l,1}\left(v_{l,1}\right), \dots, f_{l,n}\left(v_{l,n_l}\right)\right]^T \in R^{n_l}$ denotes the vector of element-wise activation functions of the *l*-th layer.

In this study, we applied MPLs with three or four layers, including the input and output layers. As an activation function in hidden layers, we used sigmoid. For the output layer, an activation function was linear. Additionally, after each hidden layer, we applied batch normalization to standardize its output, i.e. the input for the next layer. Also, we employed l1- and l2-regularizers: $l1 = 0.00001$, $l2 = 0.00001$ to constrain weights of W_l and b_l as well as the output of each layer.

B. Simple Recurrent Neural Networks

Conventional MLPs accept inputs of a fixed size, for example a vector presenting the pixel intensity values of an image or a series of sequential measurements from a time series. Furthermore, as MLPs treat all inputs as unrelated events, they are unable to make decisions on temporal or sequential relationships between separate inputs. Conversely, RNNs possess a memory in a form of a feedback loop which enables the decision made based on the previous input to affect the decision made based on the current input [17, 18]. If a sequence of inputs $x = (x_1, \dots, x_N)$ corresponds to a sequence of outputs $y = (y_1, \dots, y_N)$, the following equations describe the RNN-module:

$$h_t = f(W_{xh} x_t + W_{hh} h_{t-1} + b_h), \tag{4}$$

$$y_t = W_{hy} h_t + b_y. \tag{5}$$

where W_{hy} is the RNN-output weight matrix, b_y is the output layer bias term, W_{xh} and W_{hh} are the weight matrices of the RNN-neuron for the input and feedback correspondingly, b_h is the bias term of the RNN-neuron. In this study, $f(x) = tanh$, i.e. the hyperbolic tangent activation function.

Due to multiplications involved in the feedback loop, the cost function derivatives in RNNs are prone to diminishing or considerable increasing during propagation – a phenomenon referred as "vanishing or exploding gradients". This limits the applicability of RNNs to problems involving sequences in which events of interest are separated across time.

C. Long Short-Term Memory Networks

LSTM networks circumvent the problem of vanishing or exploding gradients by increasing the architectural complexity of individual network modules [19]. In LSTM, in addition to the dedicated memory cell, each module contains an input gate *i* controlling the effect of the current input on the state of the cell, a forget gate *f* controlling the effect of the previous input on the state of the cell, and an output gate *o* controlling the passing of the value stored in the module memory cell on the following cells. Thus, for a sequence of inputs $x = (x_1, \dots, x_N)$ the corresponding sequence of outputs $y = (y_1, \dots, y_N)$ is calculated by applying $y_t = W_{my} m_t + b_y$ for each time point $t = 1, \dots, N$, where W_{my} is the LSTM-output weight matrix, b_y is the output layer bias term, and m_t is the output of the LSTM-module. The implementation of the LSTM-module as formulated by Gravez et al. [20] can be formally presented as a composite function, where σ presents the logistic sigmoid function and $tanh$ introduces the hyperbolic tangent activation function, W_{xi} presents the weight matrix :

$$c_t = f_t c_{t-1} + i_t tanh(W_{xc} x_t + W_{mc} c^{t-1} + b_c), \tag{6}$$

$$i_t = \sigma(W_{xi} x_t + W_{mi} m_{t-1} + W_{ci} c_{t-1} + b_i), \tag{7}$$

$$f_t = \sigma\left(W_{xf} x_t + W_{mf} m_{t-1} + W_{cf} c_{t-1} + b_f\right), \tag{8}$$

$$o_t = \sigma(W_{xo} x_t + W_{mo} m_{t-1} + W_{co} c_t + b_o), \tag{9}$$

$$m_t = o_t tanh(c_t). \tag{10}$$

This additional control over the internal memory allows cells to learn the appropriate conditions for recalling, forgetting, and memorizing, which overcomes the problem of diminishing or exploding gradients and, thus, enables the model to reach the higher temporal depth in practice. However, the increased complexity of the memory cells results in higher computational costs and the increase in the number of model parameters can result in difficulties with finding the optimal solution.

IV. Experiments and Results

In all the experiments, to train models, we used the data from 1.1.2011 to 31.12.2012 (first two years). As the validation and test sets, we employed the data from the first and second six months of 2013 and repeated training twice to have each of these periods as the test data. For every ANN, the experiment was repeated 10 times.

The input and output variables were normalized to the interval [0, 1], using minimum and maximum values from the training data [21]. Data preprocessing also included expanding the vector of input variables with values from previous time points. For RNN and LSTM, every input vector corresponding to the moment t was augmented with values from previous time points $(t-1)$, …, $(t-N_{hist}-1)$ so that we obtained sequences of N_{hist} length (i.e. a matrix N_{hist} x 15). Such sequence was used as an input to produce the number of faults for the moment t. For MLP, we did not include timing variables for all previous time points, only from the moment t. Thus, for MLP, the total number of inputs was defined as $4 + 11 \cdot N_{hist}$ (4 timing variables and 11 meteorological variables). In practice, the trained models are supposed to use weather forecast for the time t and several previous time points to predict faults for the moment t.

ANNs for this study were implemented using Keras [22]. As an optimizer, we selected Adam with a learning rate of 0.001. A batch size was equal to 64. After the input and hidden layers, we applied batch normalization. The cost function was Mean Absolute Error (MAE). The best model was found using the validation data: training stopped when there was no improvement in the validation loss for 50 epochs, and the model with the minimum MAE was applied to the test data.

While training ANNs, we realized that the decrease of MAE on the training and validation sets did not mean that the model was able to predict peaks. In Fig. 2, we present the predictions produced with MLP having 70 neurons in the input layer (N_{hist} = 6), 64 neurons in the hidden layer, and 1 neuron in the output layer. However, it turned out that for the same model, oversampling could make the model follow high peaks [23]. After data preprocessing, we added to the training dataset 10 copies of each sample which normalized output value was greater than 0.1. In Fig. 3, we show the predictions of the same MLP using oversampling. The model became more effective for periods with many faults but less accurate for periods with few faults.

In addition to MAE, we used Index of Agreement (IA) [24] to evaluate the model performance (11):

$$IA = 1 - \left(\frac{SSE}{\sum_{i=1}^{n} (|\hat{y}_i - \bar{y}| + |y_i - \bar{y}|)^2} \right), \tag{11}$$

where $SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$, y_i is the true number of faults, $\hat{y}_i$ is the predictied number of faults, $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$, $i = \overline{1, n}$, n is the sample size. IA $\in [0, 1]$, higher values of IA indicate better models.

Comparing Fig. 2 and 3, we may note that MAE does not reflect the model performance thoroughly. The second model (Fig. 3) is more preferable since it can predict the sudden increase of faults, whereas in terms of MAE, the first model without oversampling (Fig. 2) is more accurate. On the other hand, IA clearly indicates that the second model adequately predicts peaks. Therefore, we decided to use IA as the main metric in the model comparison.

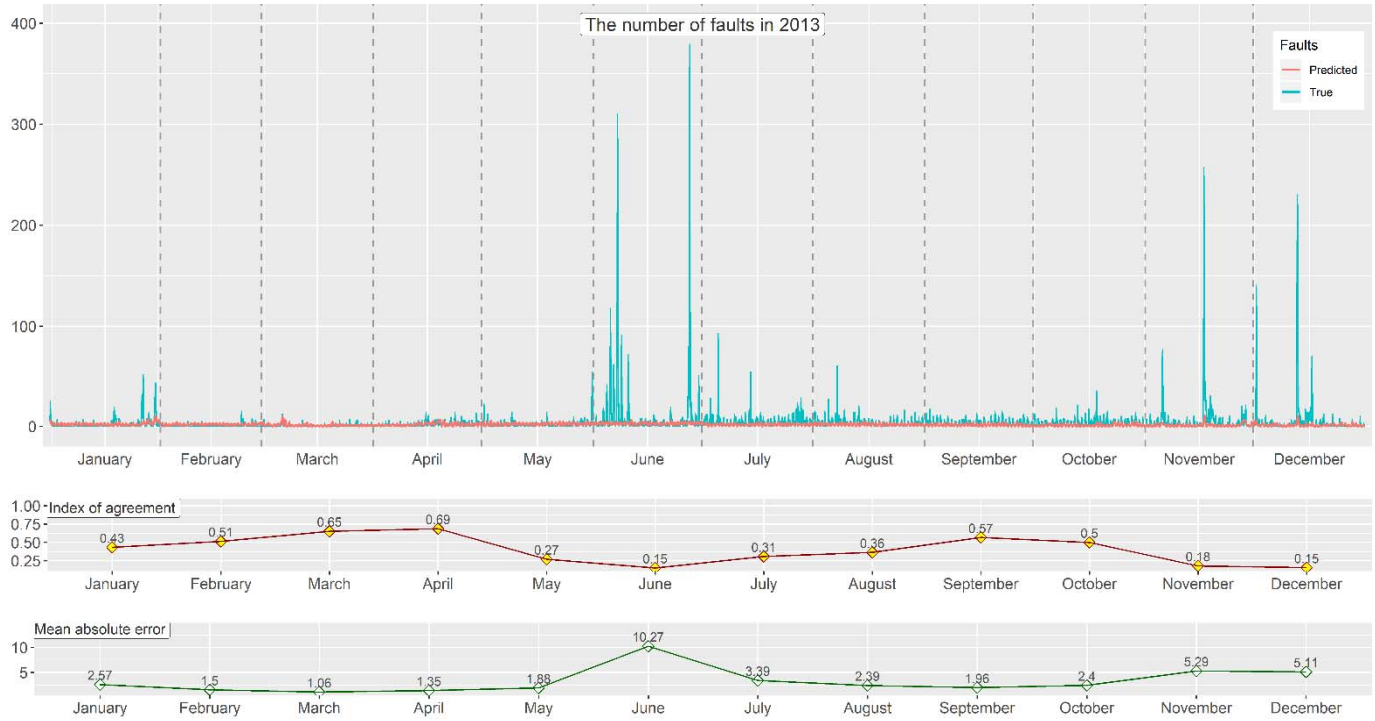

Fig. 2. MLP predictions without oversampling. The model trained cannot predict rare high peaks. It works better for periods with the low number of faults, which might be seen from the corresponding values of IA and MAE. In this example, the averaged MAE in 2013 is 3.26; the averaged IA is 0.12.

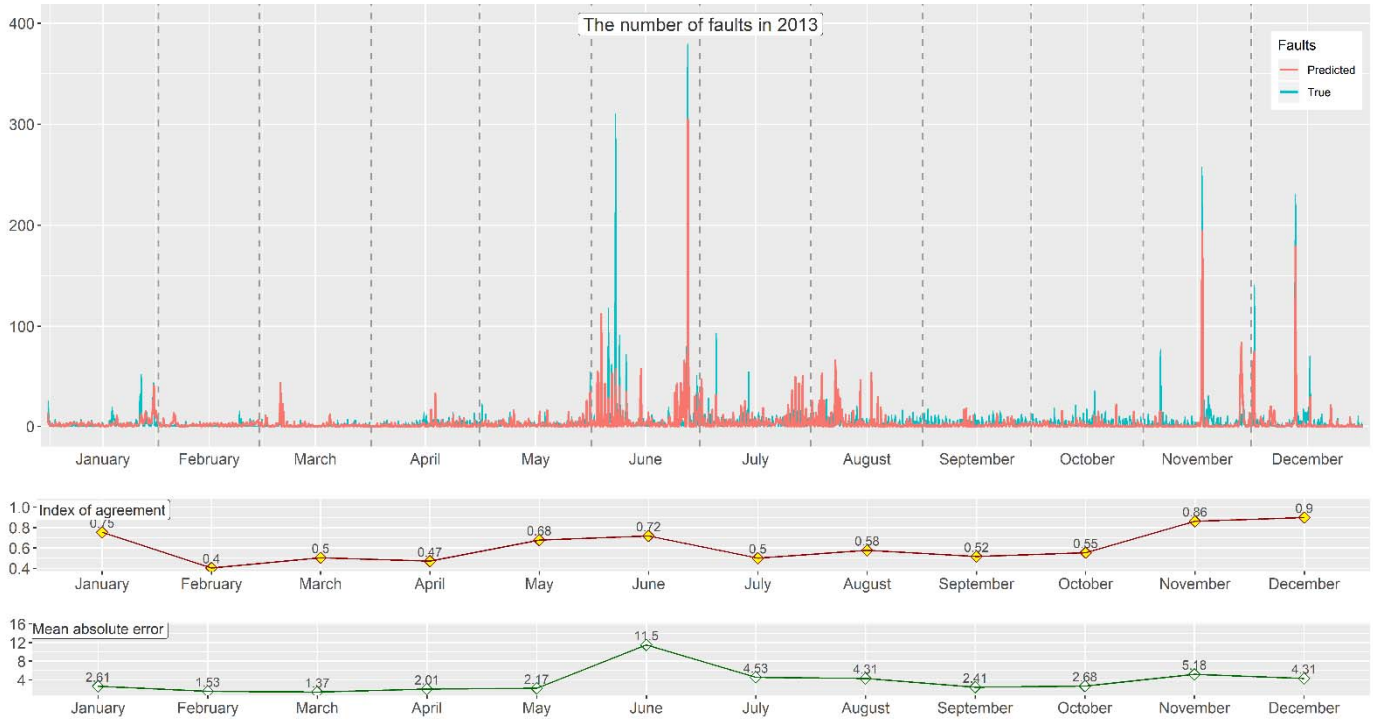

Fig. 3. MLP predictions with oversampling. The model trained predicts rare high peaks. It shows the better performance for periods with the large amount of faults, which might be seen from the corresponding values of IA. In this example, the averaged MAE in 2013 is 3.72; the averaged IA is 0.78.

The experiment with MLP revealed the importance of oversampling for the current problem, i.e. when there are rare peaks with many faults which should be predicted accurately. Therefore, while testing RNNs and LSTMs, we applied oversampling too.

In the comparative analysis of ANNs, we varied the number of previous time points, i.e. N_{hist}, which could be informative for the fault prediction. For MLP, we checked several N_{hist} from the interval [1, 12] so that the longest period taken into account was 12 hours prior to the predicted faults. In Fig. 4, first ten models represent MLPs with one or two hidden layers, with or without oversampling, with or without dropout (i.e. ignoring randomly chosen neurons while training a model). Every model's name contains the number of neurons in the input, hidden, and output layers. According to the results obtained, IA increases when N_{hist} changes from 1 to 6, and it starts decreasing after that. The best MLP model (number 5 in Fig. 4) had one hidden layer without dropout. The predictions obtained with this ANN-architecture are given in Fig. 3 above.

For RNNs and LSTMs, we tested a wider range of N_{hist}, starting with 6 hours and ending with 5 days. Since RNNs and LSTMs are capable of capturing long-term dependences, we could expose them in our data with the large values of N_{hist}. However, none of these models outperformed the MLP model operating with observations from previous 6 hours. We tried to increase the memory adding hidden layers (models 17 and 20) and neurons (model 18), enhance the architecture combining RNN- and LSTM-layers with Dense-layers (models 13 and 22). Dropout led to decreasing IA. Also, the best LSTM (model 13) had only 8 LSTM-neurons, whereas the more complex LSTM-architecture with two hidden layers and 64 neurons in each of them (model 17) could not reach the higher values of IA. Moreover, LSTM with 128 neurons in the hidden layer (model 18) demonstrated even much worse IA.

The experimental results tell us about the high importance of the weather conditions during 6 hours prior to the predicted faults. Indeed, strong wind causes outages immediately or within next hours rather than days, and snow depth in our data means the accumulated amount, not just the new amount of snow per hour. Apparently, with the weather measurements from a few hours (N_{hist} = 6) preceding the fault, we get all the information relevant to the prediction. Therefore, we assume that for predicting faults based on the presented meteorological data, LSTMs and RNNs might be redundant. The similar result has been discussed earlier in [15].

Next, we performed the analysis of variable importance for the best model found (number 5 in Fig. 4). In this experiment, we used a Python package called Eli5 [25] to execute a permutation importance technique [26]. Using this method, we estimated how IA decreased when a variable was "not available". In fact, values of each variable were shuffled to create a noise from the same distribution as real variable values had. After training the model, the permutation importance technique was applied to the test data. The variable importance obtained is presented in Fig. 5. At the top of this list, there are the timing variables and horizontal visibility, wind speed, gust speed from the time point (t – 5). This complies with the fact that taking into account up to 6 previous time points resulted in the highest model performance.

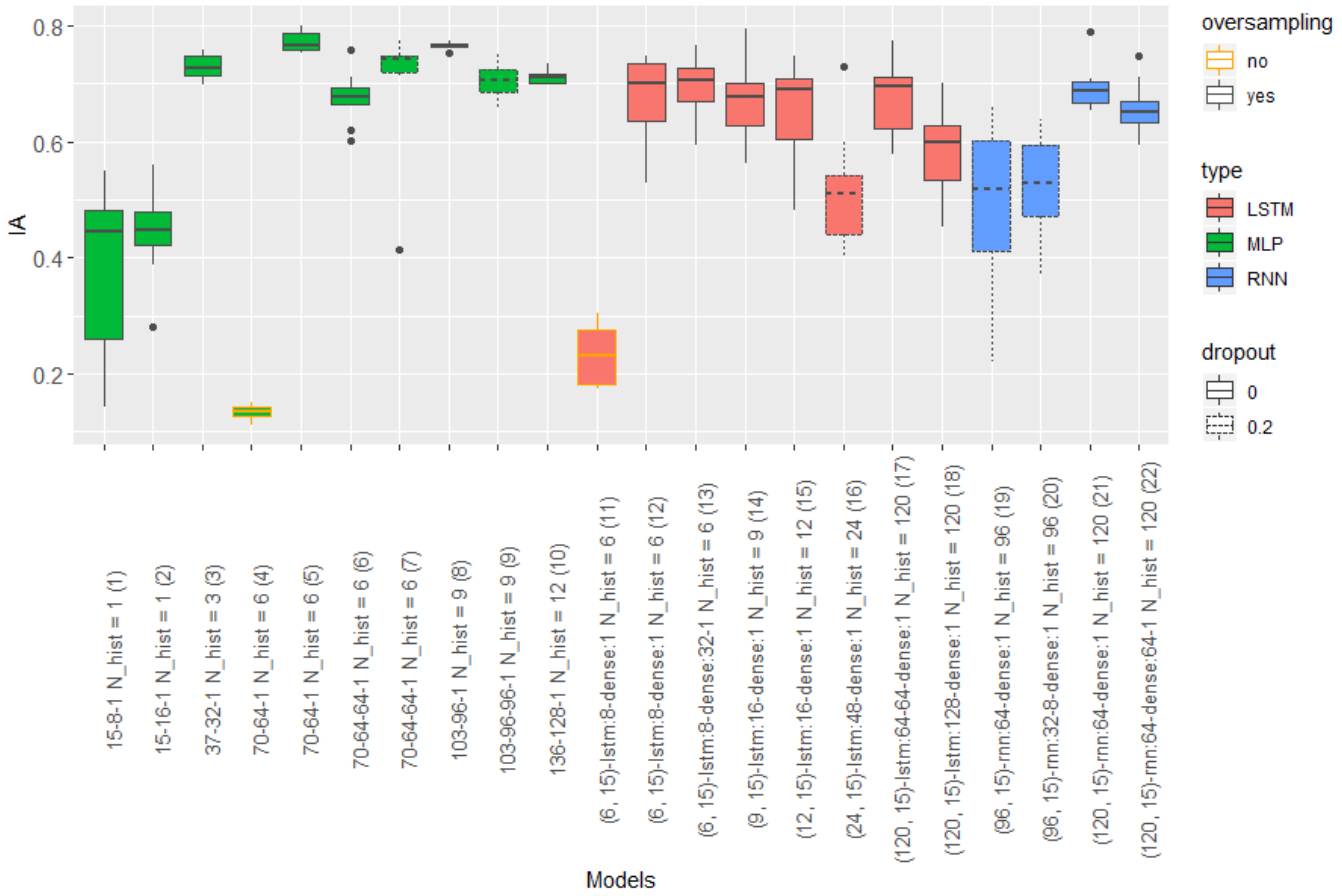

Fig. 4. Model evaluation. This figure shows that MLPs with the sufficient number of inputs from previous time points and neurons in the hidden layer compete with RNNs and LSTMs successfully. In terms of IA, model 5 is the best found ANN for the fault prediction in this study.

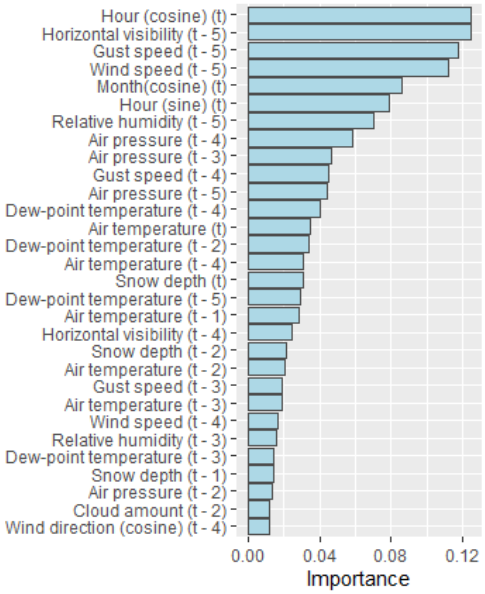

Fig. 5. Variable importance for the MLP model (70-64-1) estimated with the permutation importance technique. This figure shows the 30 most important predictors.

Additionally, the information about fault subtypes was utilized to build a model predicting wind-related faults only, which was the most prevalent defined type of faults in the sample. The best ANN-architecture found in the previous experiments, when predicting the total number of faults, was taken for modeling wind-related faults. In particular, the MLP model (70-64-1) with one hidden layer and N_{hist} = 6 was trained to predict the faults caused by wind and storm in 2013. As it was done previously, we repeated the model training 10 times and estimated the averaged IA that was equal to 0.68. The results of one of these runs are visualized in Fig. 6. As can be noted, the model could predict most of the high peaks at the end of 2013 (Fig. 6). However, since many faults in the data were not categorized at all ('undefined'), we may assume that some of wind-related faults are missing in the training, validation, and test data, which could affect the results.

All in all, with the found ANN-architecture we could predict both total faults and wind-related faults quite accurately. Since the biggest issue for DSOs is the events with the high number of outages, which should be predicted in advance, we may conclude that the MLP trained can handle this issue successfully.

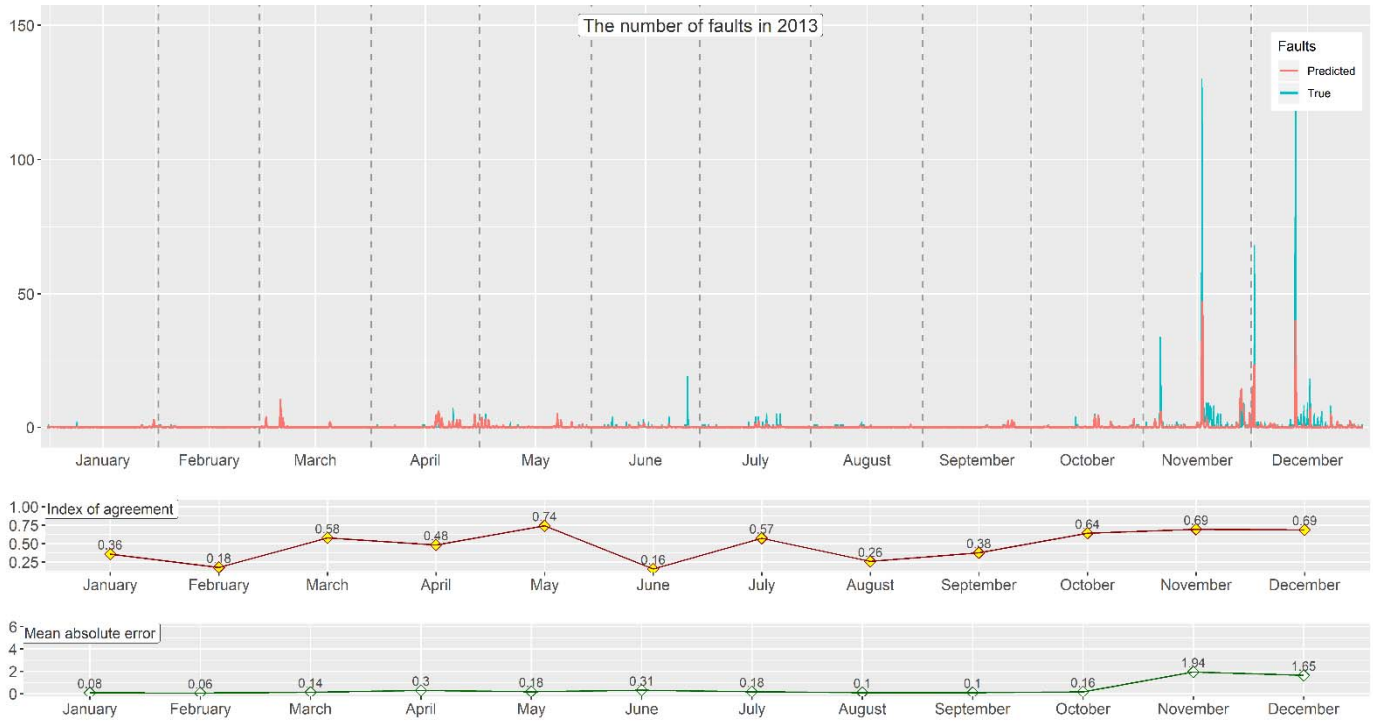

Fig. 6. MLP predicting wind-related fauls. The model predicts rare high peaks. In this example, the averaged MAE in 2013 is 0.43; the averaged IA is 0.69.

V. Conclusion

In this study, we compared MLPs, RNNs, and LSTMs in predicting faults in the electricity network. Despite the fact that RNNs and LSTMs are able to capture long-term dependencies in the data, we showed that for the problem considered all the relevant information was presented within a few hours prior to faults.

Then, we demonstrated that MAE averaged over a year could not express the model predictive ability as properly as IA did. Therefore, for the model comparison, when a time series contains rare high peaks, IA might be a good choice.

Besides, the experimental results proved that oversampling greatly contributed not only to the MLP performance but also to the performance of recurrent networks.

Acknowledgment

We wish to acknowledge the Academy of Finland for the financial support [324677 The Analytics project, 2019–2023] and CSC – IT Center for Science, Finland, for computational resources.

References

[1] J. Schmidhuber, “Deep learning in neural networks: An overview,” Neural Networks, 61, pp. 85–117, 2015.

[2] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, 521(7553), pp. 436–444, 2015.

[3] M. Honkola, N. Kukkurainen, L. Saukkonen, A. Petäjä, J. Karasjärvi, T. Riihisaari, and R. Ruuhela, “The finnish meteorological institute : Final report for the open data project”, Ilmatieteen laitos, 2013.

[4] A. Sarwat, M. Amini, A. Domijan Jr, A. Damnjanovic, and F. Kaleem, “Weather-based interruption prediction in the smart grid utilizing chronological data,” Journal of Modern Power Systems and Clean Energy, 4(2), pp. 308–315, 2016.

[5] H. Takata, M. Yanase, T. Waki, and T. Hachino, “A prediction method of electric power damage by typhoons in kagoshima via GMDH and NN,” Proceedings of the 41st SICE Annual Conference, vol. 4, 2002.

[6] G. Li, P. Zhang, P. B. Luh, W. Li, Z. Bie, C. Serna, and Z. Zhao, “Risk analysis for distribution systems in the northeast U.S. under wind storms,” IEEE Transactions on Power Systems, 29(2), 889–898, 2014.

[7] Y. Liu, “Short-term operational reliability evaluation for power systems under extreme weather conditions,” 2015 IEEE Eindhoven PowerTech, pp. 1–5, 2015.

[8] D. Zhu, D. Cheng, R. P. Broadwater, and C. Scirbona, “Storm modeling for prediction of power distribution system outages,” Electric Power Systems Research, 77(8), pp. 973–979, 2007.

[9] P. Chen, and M. Kezunovic, “Fuzzy logic approach to predictive risk analysis in distribution outage management,” IEEE Transactions on Smart Grid, 7(6), pp. 2827–2836, 2016.

[10] A. A. Nanadikar, V. N. Biradar, and D. V. S. S. Siva Sarma, “Improved outage prediction using asset management data and intelligent multiple interruption event handling with fuzzy control during extreme climatic conditions,” 2014 International Conference on Smart Electric Grid (ISEG), pp. 1–7, 2014.

[11] R. F. Berriel, A.T. Lopes, A. Rodrigues, F. M. Varejão, and T. Oliveira-Santos, “Monthly energy consumption forecast: A deep learning approach,” 2017 International Joint Conference on Neural Networks (IJCNN), pp. 4283–4290, 2017.

[12] S. Muzaffar, and A. Afshari, “Short-term load forecasts using LSTM networks,” Energy Procedia, vol. 158, pp. 2922–2927, 2019.

[13] A. Graves, N. Jaitly, and A. Mohamed, “Hybrid speech recognition with deep bidirectional LSTM,” 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 273–278, 2013.

[14] D. Zhang, X. Han, C. Deng, “Review on the research and practice of deep learning and reinforcement learning in smart grids,” CSEE Journal of Power and Energy Systems, vol. 4, no. 3, pp. 362–370, September 2018.

[15] F. A. Gers, D. Eck, and J. Schmidhuber, “Applying LSTM to time series predictable through time-window approaches,” In: Dorffner G., Bischof H., Hornik K. (eds) Artificial Neural Networks – ICANN 2001, ICANN 2001.

[16] S. Haykin, Neural Networks: A Comprehensive Foundation (2 ed.). Prentice Hall, 1998.

[17] J. L. Elman, “Finding structure in time,” Cognitive Science, 14, pp. 179–211, 1990.

[18] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016.

[19] S. Hochreiter, and J. Schmidhuber, “Long short-term memory,” Neural Computation, 9(8), pp. 1735–1780, 1997.

[20] A. Graves, A. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6645–6649.

[21] 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, and E. Duchesnay, “Scikit-learn: machine learning in Python,” JMLR, 12, pp. 2825–2830, 2011.

[22] C. François, “Keras documentation,” keras.io, 2015.

[23] R. Tervo, J. Karjalainen, and A. Jung, “Predicting electricity outages caused by convective storms,” 2018 IEEE Data Science Workshop (DSW), pp. 145–149, 2018.

[24] C. J. Willmott, S. G. Ackleson, R. E. Davis, J. J. Feddema, K. M. Klink, D. R. Legates, J. O’Donnell, and C. M. Rowe, “Statistics for the evaluation and comparison of models,” J. Geophys. Res, 900(C5), pp. 8995–9005, 1985.

[25] M. Korobov, and K. Lopuhin, “Permutation importance,” eli5.readthedocs.io, 2017.

[26] L. Breiman, “Random Forests,” Machine Learning, 45(1), pp. 5–32, 2001.