Day-ahead Forecasting of Losses in the Distribution Network

Nisha Dalal,[1*], Martin Mølnå[1*], Mette Herrem[1], Magne Røen[1], Odd Erik Gundersen[1,2]

[1]TrønderEnergi Kraft AS, Trondheim, Norway
[2]Norwegian Open AI Lab, Norwegian University of Science and Technology, Trondheim, Norway
{nisha.dalal, martin.molna, mette.herrem, magne.roen}@tronderenergi.no, odderik@ntnu.no
* contributed equally

Abstract

Utility companies in the Nordics have to nominate how much electricity is expected to be lost in their power grid the next day. We present a commercially deployed machine learning system that automates this day-ahead nomination of the expected grid loss. It meets several practical constraints and issues related to, among other things, delayed, missing and incorrect data and a small data set. The system incorporates a total of 24 different models that performs forecasts for three sub-grids. Each day one model is selected for making the hourly day-ahead forecasts for each sub-grid. The deployed system reduced the mean average percentage error (MAPE) with 40% from 12.17 to 7.26 per hour from mid July to mid October, 2019. It is robust, flexible and reduces manual work. Recently, the system was deployed to forecast and nominate grid losses for two new grids belonging to a new customer. As the presented system is modular and adaptive, the integration was quick and needed minimal work. We have shared the grid loss data-set on Kaggle.

Introduction

Every day at noon utility companies in the Nordics have to nominate to Nord Pool[1] how much electricity is expected to be lost in the power grid for each hour the next day. This is called day-ahead nomination of grid losses. The grid loss is correlated with the length of the path that the electricity is routed and the amount of electricity that is transported through the power grid. The path through the power grid changes daily and seemingly stochastically. The electricity is routed based on operational decisions made by the power grid operator. The consumption also changes every hour according to the weather, the season, time of the day, day of the week and whether or not it is a holiday.

The power company TrønderEnergi Kraft AS nominates losses day-ahead for the utility company Tensio as a service. In the past, operators used a numerical method to forecast the grid losses based on relatively simple heuristics and an off-the-shelf energy consumption forecasting model. It required manual work to recalculate constants part of the numerical method, and the output of the forecasts were adjusted manually from time to time if they looked off to the operators. The quality of these adjustments depended on the
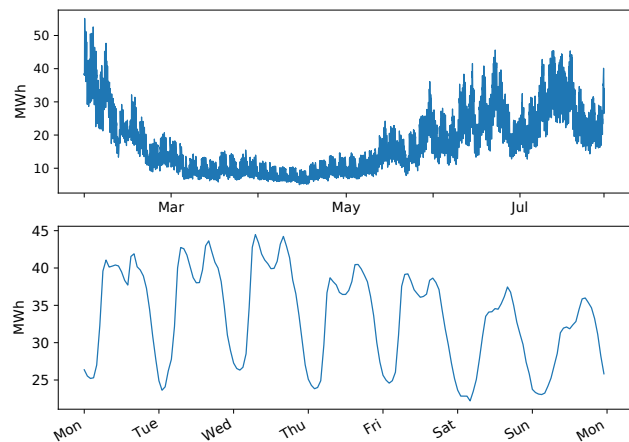
Figure 1: Grid loss over an year (top) and a week (bottom).

experience of the operators, and hence they were only as good as the experts making these. As a small set of operators used the above-mentioned numerical method to manually forecast and adjust these loss forecasts, it was not very robust to changes. Although not very time-consuming, using it required manual work.

In this paper, we present a system that has been deployed to automate the day-ahead prediction of grid losses for the utility company Tensio. The deployed system reduces the MAPE with 40% from 12.17 MW to 7.26 MW per hour for the period July 17 to Oct 21, 2019. We used MAPE as the error metric since it has same characteristics as mean absolute error (MAE) but it is normalized hence easy to compare (de Myttenaere et al., 2016). The results translate to a reduced imbalance cost of about 15 000$ per year for a relatively small part of the power grid in Norway. Other advantages include reducing financial risk that the utility is exposed to because of the imbalance between the nominated loss and the actual loss in the grid, reduced manual work (∼100 hours/year) and systematic nomination not relying on individuals. Reducing manual work reduces the potential for human errors. Automation also frees time for the operators that can be spent on other, more valuable tasks. It also standardizes the process, so that it relies less on the subjective

judgment of individual operators. Since October 2019, the previous model is not in use and hence not maintained.

Since June 2020, the machine learning system was deployed to forecast and nominate grid losses in the distribution network for a another utility company. The customer provided one year of historic grid loss data for two new sub-grids. The system was developed with such an expansion in mind. It is flexible and robust and supports easy adaptation to changes in the grids as well as changes in the available feature set.

Power Grids, Losses and Electricity Markets

The *power grid* transports electricity from the producers to the consumers and is divided into the transmission network and the distribution network. The *transmission network* transports electricity from the generation site, such as an electrical power plant, to electrical substations, while the *distribution network* distributes the electricity from the electrical substations to the consumers. The transmission system operator (TSO) operates the transmission network while the distribution system operator (DSO) operates the distribution network. In the Nordics, the state owned public utilities Statnett, Fingrid, Svenska kraftnät and Energinet are the TSOs responsible for the national transmission networks. The distribution networks are typically owned by local utility companies, such as Tensio.

Grid loss is defined to be the difference in electricity between what has been produced by the power plants and what has been sold to the consumers. Grid losses can be divided into technical and non-technical losses. Technical losses are due to both transport and transformation and show themselves as reduced voltage. Some of these are variable while others are fixed. The fixed losses do not depend on the amount of electricity that is transported, but the applied voltage. The variable losses vary with the current carried by the conductor and depend on the resistance, as the resistance causes energy to be absorbed by the conductor. Non-technical losses include theft and failing electricity meters. The physics of grid losses are well understood and can be calculated quite accurately given the grid configuration. Still, as the configuration of the grids are not known or changes all the time, calculating grid losses is not straight forward. Parts of the losses are in the transmission network and parts in the distribution network. The utilities are responsible for the losses in their networks, and they have to nominate the expected loss day-ahead to the market so that the electricity price can be decided. We assume that all electricity is accounted for and that there is no theft, as the theft or commercial losses are very small in Norway (NVE, 2016).

Electricity is sold in several different physical markets. In these markets, the sellers have to produce the agreed upon amount and the buyers have to buy the amount they bid for. The electricity price in the Nordics is decided in the Nord Pool spot price market. The spot price is decided based on an auction where producers and consumers make bids on how much electricity they can produce or consume and at what price. The auction closes at noon the day before pro-duction starts, and the spot price for each hour the next day is presented 42 minutes later.

As wind and solar power are variable power sources that cannot be dispatched on demand and vary from hour to hour, the amount of electricity that they will produce is highly uncertain and hard to forecast day-ahead. Electricity consumption is also hard to predict. Its dependencies are among other the weather and season, which is not surprising as electricity is the main heating source for most houses in Norway. Figure 1 shows the variation of grid loss over a year and a week. As the top graph in Figure 1 shows, grid loss is higher during winter and lower during summer. The reason is of course the increased electricity usage caused by the need for heating during winter. The lower graph in Figure 1 shows the how the grid loss changes over a typical week with higher values during the weekdays and lower during the weekends. Even on daily basis, you can see a pattern with two peaks during the day (morning and afternoon) and then a big dip during the night.

As mentioned above, grid loss is correlated with the amount of energy in the power grid. Hence, the total energy demand, which includes consumption and loss, is also hard to forecast. The uncertainty in both the production and the demand results in imbalances in the energy market that the system operator has to settle on behalf of the non-compliant parts. The uncertainty results in forecasting errors which again result in deviation from the nominated positions. These deviations result in imbalances that are settled in the imbalance marked by the TSO on behalf of the non-compliant parties. The TSO buys position changes from portfolio owners with flexible assets, and the imbalance price is set uniformly for each hour determined by the bid/ask price of the last activated asset. The imbalance price is highly unpredictable, and the imbalance price has a two-price logic which ensures that anyone causing imbalance always will be worse off compared to the day-ahead market price. Often the imbalance price is fairly similar to the spot price, which on average is between 20 and 30 EUR/MWh, but the upper limit is 5000 EUR/MWh. Thus, producers and consumers benefit from improving their forecasts, as nominating with a low error leads to a reduced risk of paying high imbalance prices and thus lowers the *imbalance cost*.

Problem Description and Quadratic Model

Grid loss is represented as a time-series with 1 hour granularity, where each value is the average loss in the network over the past hour. The objective is to nominate the grid losses for the next day at noon, so we need to forecast grid loss for 24 hours of the next day before noon. Formally, at time $t = 0$, noon, for network n, the objective is to nominate the 24 future losses $L_{n,t+\Delta}$ where $\Delta \in <13, ..., 36>$, alternatively annotated as $L_{n,t} = (L_{n,t+13}, ..., L_{n,t+36})$. We annotate the forecasts with **bold**, so the grid loss forecasts are denoted $\mathbf{L_{n,t}}$. The grid losses that we forecast are the losses in the distribution network that Tensio is responsible for. This distribution network is composed of three non-overlapping sub-networks, and hence:

$$L_{t+\Delta} = L_{1,t+\Delta} + L_{2,t+\Delta} + L_{3,t+\Delta}, \qquad (1)$$

where $L_{t+\Delta}$ is the grid loss for the whole distribution network, $L_{1,t+\Delta}$, $L_{2,t+\Delta}$ and $L_{3,t+\Delta}$ are the grid losses for each of the sub-networks for hour $t + \Delta$. $L_{t+\Delta}$ is the actual loss in the distribution network, and $E_{t+\Delta}$ is the error for hour $t + \Delta$:

$$E_{t+\Delta} = L_{t+\Delta} - \mathbf{L}_{t+\Delta}. \qquad (2)$$

The error is the amount of electricity that has to be traded in the balancing market for the imbalance price that is decided for that specific hour. Total average energy in a target network for hour $t + \Delta$ can be described as follows:

$$I_{t+\Delta} = C_{t+\Delta} + L_{t+\Delta}, \qquad (3)$$

where $I_{t+\Delta}$ is the average total energy in the target network (load) for hour $t + \Delta$, $C_{t+\Delta}$ is the average consumption by consumers in the target network for hour $t + \Delta$ and $L_{t+\Delta}$ is the grid loss for the target network for hour $t + \Delta$. The previous method for estimating the average grid loss for hour $t + \Delta$ is based on a quadratic polynomial regression model:

$$\mathbf{L}_{t+\Delta} = L_0 + k\mathbf{C}_{t+\Delta}^2, \qquad (4)$$

where $\mathbf{L}_{t+\Delta}$ is the estimated loss for the target network, L_0 is idle loss, k is a constant and $\mathbf{C}_{t+\Delta}$ is the expected power consumption in the target network for hour $t + \Delta$. Both L_0 and k are computed numerically by fitting equation 4 so that equation 3 is correct for historical data. $\mathbf{I}_{n,t+\Delta}$ is estimated for each hour the next day using an off-the-shelf demand model that takes as input historical consumption in the target region and the temperature prognosis for the next day. Then, based on the expected power consumption, the grid loss $\mathbf{L}_{t+\Delta}$ for each hour was calculated using equation 4. $\mathbf{L_t} = (L_{t+13}, ..., L_{t+36})$ was nominated to the spot market before noon.

A complicating factor is that the DSO is responsible for the electricity consumption of consumers that do not have a contract with an electricity retailer. Retailers buy electricity on behalf of many small consumers, and they have to nominate the expected consumption. The reasons why some consumers do not have contracts with retailers are: 1) Consumers are in-between retailers; 2) consumers that have not paid their bill to the retailer and the retailer has stopped selling them electricity still get electricity until the DSO cut the supply physically; 3) some consumers did not get a retailer since the market opened. The consumption by consumers without a retailer are nominated as part of the day-ahead grid loss nomination. Predicting this consumption is hard as the amount of consumers that do not have a retailer is seemingly stochastic.

Related Work

Grid losses has a high cost for society and work has been done on identifying and reducing it. Navani et al. (2008) gives an overview of technical and non-technical losses and provides an analysis of the consequence of losses to the Indian economy. Bernheim et al. (2018) present a system for detecting and localizing non-technical losses by comparing current and voltage flowing through different meters at the same time and uses these to see whether there are un-metered flows between the meters and the transformer,

while Carquex and Rosenberg (2018) use state estimation and smart meters to detect and locate theft in distribution systems. Glauner et al. (2017) provides an overview over AI techniques for detecting non-technical loss. Kang et al. (2006) and Leal et al. (2006) use artificial neural networks to perform analysis and evaluation of losses in distribution systems. Agüero (2012) reviews technologies, methodologies and operational approaches aimed at improving the efficiency of power distribution systems. Both Han and Li (2019) and Hu et al. (2017) present methods for reducing losses by distributing resources, as "decentralized generation can avoid grid losses and save primary energy". Oliveira et al. (2001), presents a method for computing losses offline after the fact.

Although losses have to be nominated daily, the literature on methods for predicting grid losses is sparse. To the best of our knowledge, we have only found two other publications dealing with predicting grid losses. This should indicate the novelty of our research presented here. Sulakov (2017) presents a system that is used to nominate hourly grid losses day-ahead in the Bulgarian electricity market. It is a statistical approach that takes meteorological forecasts, hourly load forecasts, the net export and forecasts of wind and solar power production as inputs to forecast the hourly transmission losses. Corona losses are part of the transmission losses and Sulakov (2016a) presents a hourly method very similar to the method presented in Sulakov (2017) for making hourly forecasts of corona losses in order to trade imbalances in the intraday market. Sulakov (2016b) discusses how the forced renewables wind and solar impact variable technical losses.

Practical Constraints and Issues

A new system had to handle the following issues and constraints:

Delayed grid loss measurements: While the preliminary estimates of the actual grid loss for each grid are available the day after, these values are unreliable and are overwritten for the next five-six days. These changes are significant (up to 40% change from day to day), and hence the measured data cannot be used before these changes are accommodated.

Missing measurements: Due to technical issues, sometimes we do not receive the measured grid loss for days. For example, for grid 3, measured historical data was unavailable at the forecasting hour for a total of 533 hours over the last eleven months. Sometimes these missing measurements are updated later but sometimes, they stay missing. A robust prediction system needs to make reliable predictions even when measured data is missing.

Incorrect measurements: In the past, detecting incorrect data was an irregular and manual process. Figure 2 shows that for two months in 2018, we received incorrect data with an approximate error of 25-40 MW per hour. The measurements were never corrected, and it took months to detect this issue. The incorrect measurements affect the quality of
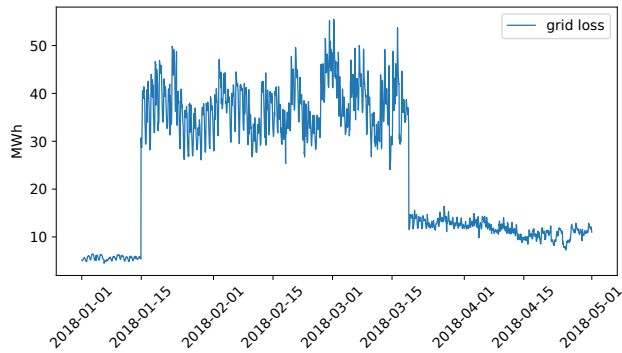
Figure 2: Incorrect measurements for grid 2.

the training and test data. Depending upon the scale of error, predictions for those periods can be way off leading to incorrect bidding of grid losses.

Incorrect measurement detection not grid specific: The manual process for detecting errors in the measured grid loss was based on the sum of the grid losses, so incorrect data from individual sub-grids were not detected. There might be scenarios when even the major errors in the sub-grids might not affect the total grid loss significantly. Additionally, since this method was based on knowledge about these three grids, it will not work for any new grids added to the system.

Changing measurements: While the historical measurements for the grid loss are stored and maintained, changes to them were not tracked. In current implementation, historical measurements were sometimes overwritten. This poses a problem for any model that uses the historical data.

Small data-set: In order to capture yearly seasonal effects properly, data-sets should span several years. Unfortunately, we only had access to less than two years of data.

Grid specific predictions: The previous solution forecasted the total grid loss for the three separate grids, which leads to less transparency. For example, errors in the data that cause erroneous results are harder to spot, as large errors in the smaller grids might not change the overall results enough to be noticed. The DSO wanted to improve the transparency.

Robustness to missing features: The previous solution was not robust to missing features. If either weather or consumption were missing, the system would break and forecasts had to be done manually by an operator. Missing features is a common issue in real-time systems, so strong dependence on having all features available in order to being able to perform a prediction will increase downtime.

Manual retraining: Operators estimated values of L_0 and k in equation 4 manually once in six months, typically. Due to high seasonal effects and abrupt changes in grid configurations, this was not ideal, and should be done more frequently.

Manual alterations: Domain experts regularly changed the predictions from the previous numerical model. They made manual changes when the predictions looked off. Such updates were based on subjective expert intuitions. Hence, the system in use was unsystematic, not reproducible and dependent on expert intervention.

Lack of a monitoring infrastructure: There was no monitoring system to support the experts in detecting anomalies. This made it hard to detect the problems with the forecasts like missing or incorrect measurements. Performance of the resulting forecasts were not monitored either.

Poor scalability: The previous system was not developed for scalability and did not easily support adding new grids and customers. Adding new grids and customers would lead to more manual work in both training the model and monitoring the deployed solution. Any errors would lead to an increased workload.

While some of the above issues can easily be solved by implementing a machine learning (ML) system that automates the process, some of the issues, such as delayed, incorrect and missing grid loss measurements, changing measurements and a small data-set needed to be solved explicitly.

Grid Loss Forecasting

The main results that we report are the experimental findings and deployed results for the three sub-grids from Tensio. Towards the end, we will provide some preliminary results for the grid loss forecasts for the two additional grids belonging to the new customer. We chose CatBoost from Prokhorenkova et al. (2018), an open-source implementation of gradient boosting on decision trees library, to forecast grid loss for each hour the next day. CatBoost with minimal hyper-parameter tuning performed well on our small data-set. We conducted different experiments to design and evaluate the effect of possible features. We report results from three of these in more detail in the subsection *Experiments*.

We identified the historical measurements of grid loss as an important feature. Since the correct measurement of the grid loss was only available six days after the fact, we used the measurements from the same hour the week before as one of the features. Since temperature directly affects the electricity consumption (heating requirements are extensive during winter) and thus affects load and grid loss, we included meteorological forecasts as features. Figure 1 shows that the grid loss is seasonal in nature and calendar features such as month, week, day of the week, hour of the day affect the grid loss.

Experiments

In total, we had 19 months of data from Dec 2017 to June 2019. 13 months of these were used for training and cross-validation and six months were used for testing. We evaluated four different algorithms on the training data using cross-validation. These were: 1) A Multi-layer perceptron with 5 hidden layers, 2) a decision tree regressor, 3) a gradient boosting regressor ensemble from sci-kit

learn and 4) CatBoost. Their respective MAE were 3.07, 1.52, 1.02 and 0.95 MW. Since we did not have enough data for hyper-parameter optimization, we chose CatBoost which performed the best with minimal hyper-parameter tuning. Also, we have important categorical features (season, month, weekday) that CatBoost handles well. We selected features and took design decisions based on the experiments we conducted on the training and cross-validation data. Since grid 3 is relatively new, it does not have enough data and has low impact on total grid loss due to its size, we excluded it from these initial experiments. We designed the model for grid 3 based on the results from the experiments we conducted using data from grid 1 and 2.

Load predictions as a feature: We had an hypothesis that an estimate of the load in a grid could be an important feature for the grid loss. To test this hypothesis, we used the load for the same hour one week before as a feature. We refer it as the one-week persistence. The effects were clear, and we then decided to make a load prediction model. We trained a separate CatBoost model for predicting the load for each of the grids. For this model, we used the historical measurements for the load, calendar features and weather predictions as features. We used MAPE (de Myttenaere et al., 2016) as the error metric. The load forecast performed 60% better than a one week persistence baseline (MAPE 3.39 vs 8.49). Using the load prediction as a feature for the grid loss forecast reduced the MAPE for Grid 1 with 25% (from 15.6 to 11.8) and 22% for grid 2 (from 13.0 to 10.2).

Grid-wise losses versus total losses: One of the requirements was to provide separate grid loss forecasts for the individual grids. However, this requirement was not more important than reducing the error, so we had to identify whether it was possible to provide forecasts for each grid with the same or lower error. For this comparison, we trained three models, one for each grid, and compared the sum of their output to a model trained with the same set of features predicting the total loss for all three grids. We found that predicting the grid loss separately for each grid improved the predictions with 9% reduction of MAPE (from 6.4 to 5.8).

Size of training data: We knew that both seasonal effects and concept drift would affect the predictions. To capture seasonal effects, more data is expected to improve the model. However, the energy consumption changes with changes in the grid and consumer behavior. To test what worked best for the amount of data we had access to, models were trained with different amount of training data in a sliding window fashion. For some models, we used all the historical data, and for others we used the different number of days like 180 or 90. It means for making a grid loss prediction for day d, we trained the models on the data from $d - 186$ to $d - 6$ days (due to delay in target for 6 days) for the training size of 180 days. The test showed that 180 day of training data was an optimal choice for both Grid 1 and Grid 2, as it reduced the error with around 20% for both grids.

Testing

We selected the best features and training period for each time series from cross-validation and then evaluated the corresponding models on the testing period (Jan-June 2019). The CatBoost models for the two grids performed better than the one-week persistence. For Grid 1, MAPE was reduced from 9.85 to 4.77, and for Grid 2 the reduction was from 10.01 to 8.45.

Handling missing data

In the real-world, we need to predict with the data available at hand at a given time. This complexity is often hidden when working with historical data sets. Training and testing models on historical data sets provides a good understanding of how well a model might perform, but it does not prepare the inference engine for handling missing, incorrect, and overwritten data. An individual model that performs the best given all the data might not perform well when some of the data is missing or incorrect. Over a period of almost 11 months, the features load prediction, measured grid loss and weather forecasts were missing between 1% and 8% of the time for each of the grids. We found that Grid 3 was especially prone to missing data.

Robustness: A robust system must be able to predict even if some of features are missing or unreliable. When forecasting grid loss for day-ahead bidding, the cost of not making a prediction is typically much higher than the cost of making a slightly worse prediction. For example, if the temperature forecast service is down, the model should still be able to predict the grid loss reliably even though weather forecast is an important feature. To facilitate this robustness in our system, we trained a set of models using unique subsets of features we found useful in our experiments. For example, in the above scenario when temperature forecasts were not available, the system could still provide grid loss predictions from a model trained without these. This model generally had a worse performance than the model that used temperature as a feature, but better in the cases where these were missing.

Detecting incorrect data: Since the system was deployed (July 17, 2019), it detected outliers and/or incorrect historical data and features for 570 hours for different sub-grids. When the system detects the values as extreme, they are tagged as incorrect and are not used for predictions. So, the models that use those features/values, will be discarded and won't be used for the final prediction.

Model selection: Due to multiple models, multiple predictions are available for the same grid at the same hour. Hence, we needed a process to select which predictions should be chosen to be nominated as the grid loss for a given hour. This model selection is based on availability of features and past performance of the models. First, the system discards the predictions from models using missing features. From the set of remaining models, it selects the prediction from the model that performed the best in the past (same day, last week). For example, five out of eight models use measured grid loss from last week as a feature. If grid loss

measurements from last week are missing or tagged incorrect, the system will discard these five models relying on the historical grid loss and select the prediction to be nominated from the remaining three models. Finally, the prediction that will be nominated will be chosen from the model that performed better the same day last week. Model selection is performed independently for each grid.

We compared this way of selecting models to a simple ensemble method that calculated the average of all the available models for a particular hour. While the performance was similar to our model selection method, the ensemble method performed worse with a MAPE of 8.92 versus 8.11 for the non-ensemble method. The period we tested this is the same as for the deployment period, that is July 17 to October 21, 2019. Nevertheless, we added average (mean of all the model predictions for one grid at each time point) as one of the models in our deployed system.

Deployed Application

The result of the project is a service that forecasts hourly grid losses for the next day. These daily forecasts of the grid loss are integrated with the current workflows of the operators through writing to the time-series service where the previous model wrote its forecasts. In this way, the operators who manually submit the forecasts as bids to the day ahead market follow the same workflow as they have always used. The predictions should be ready in due time before noon, so that operators have enough time to submit the nominations.

Architecture

The system has three main parts: 1) Data storage, 2) Machine Learning (ML) pipeline and 3) visualization. The deployed application mainly uses two types of data storage: Object storage and a relational database. The object storage is used to store the trained models while the relational database stores features, the predictions and values calculated for monitoring purposes. The features include measured data and weather forecasts, which we retrieved from external sources, as well as calendar features and load predictions. We keep track of all predictions made by all the models, as well as which predictions that are selected for nomination. Additionally, we store the detected outliers, economic results, spot and imbalance prices as well as information required for monitoring. We keep track of missing features and changes to the deployment setup and models as well. The training and prediction workflows read from and write to the relational database and the object storage. Grafana[2] is an open source tool analytics and monitoring solution that supports querying and visualising metrics from different data sources. We used it for making dashboards for both the operators and the ML DevOps team to monitor performance and the status of the system.

Workflow

We organized the production code as workflows, one for training the models and one for making predictions.

A workflow consists of several steps where outputs from one or more steps serve as inputs to the consequent steps. We
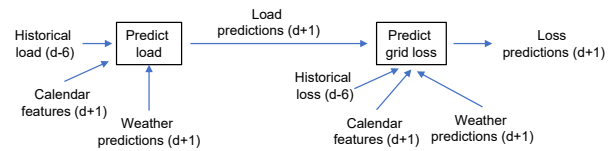
Figure 3: Grid loss prediction where *d* is the current day.

first implemented the system using Jupyter Notebooks. Each of the notebooks implemented one or more of the workflow sub-steps. We scheduled the notebooks to run in a sequence, timed manually so that the next notebook was executed after the previous one was completed. There are several reasons why we decided not to use this solution in production: 1) Steps in the workflow are not started until previous ones are completed, but this temporal dependency was not straightforward with notebooks where each step had to be scheduled manually causing a lot of run-time trouble, 2) each workflow step implements a single task, which can make the system easier to test and maintain, but notebooks are not easy to maintain. 3) as each sub-step in the workflow depends only on its inputs and not on the implementation of other sub-steps, developers can work in parallel, which simplifies collaboration. Combining multiple sub-steps into one notebook made parallelization difficult.

Due to these problems, we did not use notebooks in production. Instead we implemented each step as a Python script and deployed the workflows in the cloud with Azure Machine Learning Pipelines[3]. *The training workflow* has three steps: 1) Retrieve data, 2) detect and remove outliers, and 3) train models using cleaned data. *The prediction workflow* consists of five steps: 1) Getting and cleaning data, 2) detect and remove outliers, 3) make predictions, 4) choose predictions using model selection, and 5) report predictions.

Figure 3 shows a breakdown of the prediction step for each grid. On day *d*, we first predict the load for the next day *d+1* using historical load from *d-6*, calendar features for *d+1* and weather predictions for *d+1*. Using these load predictions, historical loss from *d-6* and the same calendar features and weather predictions, we predict the grid loss for *d+1*, for each grid.

After a few months of deployment, AzureML proved unreliable. Some technical faults led to processes waiting for hours and resulted in problems with meeting the nomination deadline. The overhead of executing each process step was large as well. In order to resolve these issues, we deployed the system on Kubernets mirroring the workflow that was implemented on AzureML. The service running on Kubernets is significantly faster and has so far proved reliable.

Dashboards

We created the dashboard for domain experts to provide a quick glance of the performance without overloading them with the technical details of the underlying system. We gave special attention to metrics and plots the domain experts are already familiar with, such as an overall status of the incoming measured data, grid loss predictions, comparisons

with the previous approach and financial savings made by both the new and the previous model. We designed two dashboards for monitoring and visualizing the performance of the deployed application. The first dashboard was designed in collaboration with the domain experts to ensure it fulfilled their requirements. Among many other relevant things, it shows two important plots: one comparing the actual grid loss, polynomial predictions and new improved machine learning predictions and another one comparing the imbalance volumes from both the previous polynomial and the new ML model. We designed a second dashboard for the ML DevOps team for monitoring and evaluating model performance. It visualizes the performance of different models, error metrics and model selection statistics etc. Both dashboards show statistics for each grid individually as well as for the total grid loss.

We developed a separate notification system. The first iteration was implemented on AzureML. The notification system, sent emails to the operators and the ML DevOps whenever something unexpected happened and human intervention was needed, for example, when jobs failed, prediction error were huge, and outliers were detected. After moving the rest of system to Kubernets, we implemented the notification system as alarms in Grafana.

Deployed Results

The service trained eight different models for each grid, and they are referenced as M1, ..., M8 (8 models * 3 grids). Each of these models were trained on subsets of the features from the experimental setup. Every day, hourly predictions from these models are stored in the database. During model selection, system selects one model per grid and its predictions for all 24 hours are nominated day-ahead.

Model selection

Figure 4 (top) shows the grid-wise performance (MAPE) of our eight models (M1, M2, .., M8) since deployment (from July 17, 2019 to May 31, 2020). M8 is the one-week persistence (last week's measurements), and M1 is the model that uses all features in the feature set. Blue indicates models for grid 1, orange for grid 2 and green for grid 3.

The other six models use a unique subset of the features. Both for grid 1 and grid 3, model M1 performs the best. M8 performs the worst for grid 1 and 2.

Figure 4 (bottom) shows how many times each of the eight models that are deployed for each grid are selected during deployment. As you will notice, model selection count does not necessarily align with the model performance. Due to unavailability of all the features needed for that model (if they are missing or incorrect), predictions from the best model cannot be used.

For grid 2 and 3, the persistence, M8 is selected more than the other models, even though it does not perform the best.

Historical measurements for grid 3 are missing often and hence the models that use them as a feature (M1, M2, M3, M4 and M5) are often discarded and hence not selected for the final predictions.

A big structural change occurred in Grid 2 on August 26, 2019 when a high energy consuming device was connected to the grid for long term. The models learned these changes in about 10-12 days. In the meantime, model selection picked Model 8 (last week measured values) since it was closest to the measurements and was the best performing model.

Up-time

The Grid loss prediction system utilizing machine learning has been deployed since July 17, 2019. Since deployment, only once (24 hours) the predictions could not be delivered to the operators. The problem was due to computation cluster not starting at AzureML. The operators were notified beforehand to use the fallback method that is last week's measured values, equal to what is used by the one week persistence model. Other than that, the model has always provided reasonable predictions to the operators, even when features have been missing.

System performance

We evaluated the performance of the deployed ML system, the one-week persistence and the polynomial regression method over a three month period. In this period, the ML system reduced MAPE from 9.00 to 7.26 compared to the one-week persistence, which corresponds to a 19% reduction of error. In the same period, the polynomial regression method had a MAPE of 12.17, which corresponds to an increase in error of 35%. The polynomial regression method being worse than the one-week persistence is not always the case historically. However, after finishing this evaluation, the polynomial regression method is no longer in use. There are several reasons for this: 1) the results were inferior compared to the whole test and deployment period of the new system (Jan to October, 2019); 2) The constants had to be recalculated, which requires manual work; and 3) the previous system required manual adjustments every day, which means that we did not get value from automating the process until we decommissioned the previous system.

After about 11 months of deployment, the performance of the ML system is 29% better than the one-week persistence with a MAPE of 8.7 compare to 12.28 for the persistence measure. This includes the performance of the system in winter when the consumption and losses are higher. Since the polynomial regression model is not maintained anymore, we only compared to one-week persistence. The MAPE for the best performing models for grid 1, 2, and 3 are 9.3, 12.5 and 13.3 respectively (Figure 4). However the overall performance of the deployed system is 8.7, which is better than any of the individual grids. This is because we nominate the sum of the three grid losses together and errors in the individual grid loss predictions often cancel each other to some degree, which leads to an overall better performance.

Improvements

Requirements of a deployed service typically change over time: Features may become unavailable, grid structures can
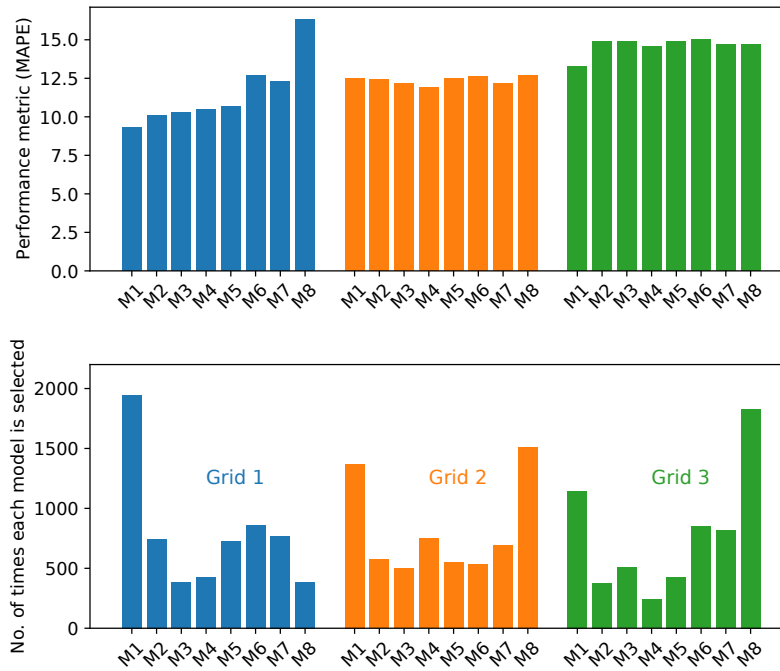
Figure 4: Top: model performance (MAPE). Bottom: model selection (no. of times models were selected) from July 17, 2019 to May 31, 2020.

change, additional grids can be added or removed. Maintenance of this service is not restricted to successful runs everyday, but also to adapt the system to changing requirements, as efficiently as possible.

Adding new grids

The introduction of the presented system has allowed TrønderEnergi to become a service provider related to grid loss nomination and won a bid for providing the nomination responsibility for a retail company that is responsible for two grids in a different part of the country. The presented system was adaptive enough to add new and completely independent grids easily into the deployed system. We were provided historical data for the past year. From these historical data, we used eight months for training and four months for testing the grid loss predictions. Compared to thee one-week persistence that has a MAPE of 11.96, the error of the ML system is 25% lower for the test period with a MAPE of 9.00. Apart from this quick initial testing, the deployment of the model for new grids did not require any other major setups as retraining is one of the steps of the current solution. The two new grids were integrated in the deployed system to predict grid loss from June 1, 2020.

Dealing with restricted feature sets

For the new grids, we only had access to a subset of the original feature set. We did not have access to grid load to train the model as Figure 3. However, as the model selection selects the best performing model from a group of models trained on the subsets of features, the system still worked.

Moreover, even for the original three sub-grids, these load measurements became unavailable, starting from June 1, 2020. Since load prediction was an important feature, most of the models included it. This unavailability lead to a large number of the models being discarded during the model selection, leaving just a handful of usable models.

To deal with this, more subsets of feature sets were created, excluding the load prediction. As the system is flexible enough to entertain different subsets of features, integrating the new models in the system was smooth. Since the model selection is a dynamic process in our system, adding new models is relatively safe. If the new models being added do not perform well in production, they will not be chosen for the final prediction.

Alerts

When erroneous predictions are detected, the domain experts are consulted to understand the root cause. Multiple reasons could lead to substantial errors in the predictions, but the two major ones are: 1) *Incorrect measured data:* when we receive incorrect data, the error calculation cannot be trusted. This data is detected as incorrect and marked in the database (automatically) so that is it not used for training. 2) *Changes in the grid:* when sudden big changes happen in grid configuration or demand, our first few predictions will be off since they are naive to these changes. Since we train our models everyday, they will start learning these changes. Model selection will choose the persistence model M8 (measured values from last week) if it is a better prediction than our other models.

Grid loss data

We have shared all data used for training and testing these algorithms on Kaggle (Dalal et al., 2020). The web page also describes how to implement the persistence baseline we used. We have shared the data used for training the models presented here and an implementation of a Jupyter notebook[4] that reads and calculates the one week persistence baseline on the test set. The following paragraphs gives a short introduction to the data-set. Detailed description is also provided on the above Kaggle link.

The data file contains time-series data for the three Tensio grids from December 2017 to May 2020 (two and a half years). The exception is grid 3 which has only one year and nine months of historical data available. The data-set is further divided into train and test sets with two years and six months of data respectively.

The data-set contains grid-specific features and common features. Grid-specific features are different for each grid and depend on the size and location of the grid. These features include the load, loss, weather forecast of the underlying region etc. The common features, on the other hand, are shared between all three Tensio grids. They include features like calendar features, demand in Trondheim and weather the data from the grids was considered incorrect.

Conclusion

We have presented a system for day-ahead forecasting of hourly grid losses in the distribution grid reduces the MAPE with 40% from 12.17 to 7.26 compared to the previous solution. By reducing the error, the system also reduces the financial risk. The presented system performs *grid specific predictions* for each of the five sub grids separately, which provides the transparency to the DSO. It also does this with improved results over predicting the total grid loss. *Delayed, missing or incorrect measurements* are handled explicitly by having multiple models that are trained on the subsets of features, so that the system will provide results even with delayed and missing data. Incoming data deemed incorrect will not be used for future training. The system requires *less human intervention* as the predictions do not need manual alterations, avoiding subjective biases in the predictions and the corresponding bids. *Automatic retraining* of the ML models are done every night, and the performance is monitored by providing a *monitoring infrastructure* visualizing results in dashboards and firing alarms if something is unusual. The *small data-set* is accommodated by the retraining. No data is lost by being *over-written*, as the historical data is stored in a separate database. The system is *robust to missing features* as it will provide forecasts even if one or more features are missing. The performance will deteriorate with missing features, but will not be worse over time than the one week persistence as it is one of the models the system selects from. *Scaling* the system to include new grids was a fairly quick process as it did not require a completely different set up. Hence, the deployed system meets all the presented constraints and issues.

While some of our solutions for solving the practical constraints are domain specific, others are generalizable to similar forecasting problems. Tackling small data-sets is important and retraining the model regularly will improve the performance over time. Time sensitive systems must handle delayed, missing and incorrect data, and training models on subsets of the features and choosing the models that use the available features is a reasonable solution. Methods for detecting incorrect data should be implemented as well although the method we applied is domain specific. Finally, close collaboration with the domain experts that are responsible for the task at hand when developing the solution will help ensure a successful deployment. The presented system can easily be deployed at all Norwegian DSOs and thus could have a high societal impact by substantially reducing imbalances nationally.

Notes

[1]https://www.nordpoolgroup.com.
[2]https://grafana.com/.
[3]https://docs.microsoft.com/en-us/azure/machine-learning/service/concept-ml-pipelines.
[4]https://www.kaggle.com/ndalal01/grid-loss-demo-and-persistence-model.

References

Julio Romero Agüero. Improving the efficiency of power distribution systems through technical and non-technical losses reduction. In *PES T&D 2012*, pages 1–8. IEEE, 2012. doi: 10.1109/TDC.2012.6281652.

Henrik F Bernheim, Jerritt H Hansell, and Marcia Reid Martin. System and method for detecting and localizing non-technical losses in an electrical power distribution grid, June 19 2018. US Patent App. 10/001,514.

Côme Carquex and Catherine Rosenberg. Multi-timescale electricity theft detection and localization in distribution systems based on state estimation and pmu measurements. In *Proceedings of the Ninth International Conference on Future Energy Systems*, e-Energy '18, pages 282–290, New York, NY, USA, 2018. ACM. doi: 10.1145/3208903.3208908.

Nisha Dalal, Martin Mølnå, Mette Herrem, Magne Røen, and Odd Erik Gundersen. Grid loss prediction dataset, 2020.

Arnaud de Myttenaere, Boris Golden, Bénédicte Le Grand, and Fabrice Rossi. Mean absolute percentage error for regression models. *Neurocomputing*, 192:38 – 48, 2016. ISSN . doi: https://doi.org/10.1016/j.neucom.2015.12.114.

Patrick Glauner, Jorge Augusto Meira, Petko Valtchev, Franck Bettinger, et al. The challenge of non-technical loss detection using artificial intelligence: A survey. *International Journal of Computational Intelligence Systems*, 10(1):760–775, 2017. doi: 10.2991/ijcis.2017.10.1.51.

Bing Han and Yonggang Li. Optimization method for reducing network loss of dc distribution system with distributed resource. *Photonic Network Communications*, 37(2):233–242, Apr 2019. ISSN 1572-8188. doi: 10.1007/s11107-018-0805-5.

Jing Hu, Robert Harmsen, and Wina Crijns-Graus. Developing a method to account for avoided grid losses from decentralized generation: the eu case. *Energy Procedia*, 141:604 – 610, 2017. ISSN 1876-6102. doi: https://doi.org/10.1016/j.egypro.2017.11.080. Power and Energy Systems Engineering.

Meei-Song Kang, Chao-Shun Chen, Chia-Hung Lin, Chia-Wen Huang, and Meng-Fu Kao. A systematic loss analysis of taipower distribution system. *IEEE Transactions on Power Systems*, 21(3):1062–1068, 2006. doi: 10.1109/TPWRS.2006.879307.

A. G. Leal, J. A. Jardini, L. C. Magrini, S. u. Ahn, H. P. Schmidt, and R. P. Casolari. Distribution system losses evaluation by ann approach. In *2006 IEEE PES Power Systems Conference and Exposition*, pages 1658–1662, Oct 2006. doi: 10.1109/PSCE.2006.296161.

JP Navani, NK Sharma, Sonal Sapra, Shiv K Gupta, PK Sharma, M Bansal, B Kumar, Mamta Goyal, Durga Nath Dhar, SK Singh, et al. Technical and non-technical losses in power system and its economic consequence in indian economy. *South Asian Survey*, 15(1): 5–32, 2008.

NVE. Overview of Norway's Electricity History. Technical report, Norwegian water resources and energy directorate (NVE), 2016.

CCB de Oliveira, N Kagan, A Meffe, S Jonathan, S Caparroz, and JL Cavaretti. A new method for the computation of technical losses in electrical power distribution systems. In *16th International Conference and Exhibition on Electricity Distribution, 2001. Part 1: Contributions. CIRED.(IEE Conf. Publ No. 482)*, volume 5, pages 5–pp. IET, 2001. doi: 10.1049/cp:20010889.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, pages 6638–6648, 2018.

S. I. Sulakov. Forecasting hourly corona losses applying statistical approach. In *2016 19th International Symposium on Electrical Apparatus and Technologies (SIELA)*, pages 1–4, May 2016a. doi: 10.1109/SIELA.2016.7543053.

S. I. Sulakov. Forced renewables penetration impact on ohmic transmission losses. In *2016 19th International Symposium on Electrical Apparatus and Technologies (SIELA)*, pages 1–4, May 2016b. doi: 10.1109/SIELA. 2016.7543052.

S. I. Sulakov. Forecasting software for hourly transmission losses. In *2017 15th International Conference on Electrical Machines, Drives and Power Systems (ELMA)*, pages 110–114, June 2017. doi: 10.1109/ELMA.2017.7955412.

Author biography

Nisha Dalal is a machine learning engineer at Trønderenergi Kraft. Currently, she is working towards modeling and automation of the prediction systems in the green energy sector in Norway. She has a PhD from Ludwig Maximilian University Munich, Germany and a Masters of Engineering in Information Technology from Indian Institute of Information Technology Allahabad, India.

Martin Mølnå is a machine learning engineer at Trønderenergi Kraft. His current research focus is deployed AI applications. He Received a M.Sc. in Computer Science from the Norwegian University of Science and Technology.

Mette Herrem is an electrical power engineer at TrønderEnergi. She works with different forecasting models such as wind power, inflow and grid loss. Herrem received a BS from Trondheim Ingeniørhøgskole (now NTNU).

Magne Røen leads the hydrology department at Trønderenergi Kraft. He has a bachelor in electronics and a master in electrical engineering. He has worked with hydro power production since 1996 and wind power production since 2007. In addition, he works with forecasting of grid losses and end users demand in the day ahead market.

Odd Erik Gundersen (PhD, Norwegian University of Science and Technology) is the Chief AI Officer at TrønderEnergi AS and an Adjunct Associate Professor at the Department of Computer Science at the Norwegian University of Science and Technology. Gundersen has applied AI in the industry since 2006. Current research includes how AI can be applied in the renewable energy sector, how to better teach drivers using AI and the reproducibility of AI algorithms.