Massive-Scale Simulation of Electrical Load in Smart Grids Using Generalized Additive Models

Pascal Pompey, Alexis Bondu, Yannig Goude, and Mathieu Sinn

Abstract The emergence of Smart Grids is posing a wide range of challenges for electric utility companies and network operators: Integration of non-dispatchable power from renewable energy sources (e.g., photovoltaics, hydro and wind), fundamental changes in the way energy is consumed (e.g., due to dynamic pricing, demand response and novel electric appliances), and more active operations of the networks to increase efficiency and reliability. A key in managing these challenges is the ability to forecast network loads at low levels of locality, e.g., counties, cities, or neighbourhoods. Accurate load forecasts improve the efficiency of supply as they help utilities to reduce operating reserves, act more efficiently in the electricity markets, and provide more effective demand-response measures. In order to prepare for the Smart Grid era, there is a need for a scalable simulation environment which allows utilities to develop and validate their forecasting methodology under various what-if-scenarios. This paper presents a massive-scale simulation platform which emulates electrical load in an entire electrical network, from Smart Meters at individual households, over low- to medium-voltage network assets, up to the national level. The platform supports the simulation of changes in the customer portfolio and the consumers' behavior, installment of new distributed generation capacity at any network level, and dynamic reconfigurations of the network. The paper explains the underlying statistical modeling approach based on Generalized Additive Models, outlines the system architecture, and presents a number of realistic use cases that were generated using this platform.

P. Pompey (✉) • M. Sinn
IBM Research, Damastown Industrial Estate, Dublin 15, Ireland
e-mail: papompey@ie.ibm.com; mathsinn@ie.ibm.com

A. Bondu • Y. Goude
EDF R&D, 1 Avenue du Général De Gaulle, 92140 Clamart, France
e-mail: alexis.bondu@edf.fr; yannig.goude@edf.fr

A. Antoniadis et al. (eds.), *Modeling and Stochastic Learning for Forecasting in High Dimensions*, Lecture Notes in Statistics 217,
DOI 10.1007/978-3-319-18732-7_11

1 Introduction

The French electrical grid is currently being fundamentally modernized by deploying Information and Communication Technology at a massive scale. The emerging "*Smart Grid*" is designed to meet multiple objectives: (i) optimizing the control of the grid and the quality of the electricity supply, despite the fact that power generation is becoming more decentralized; (ii) scheduling the production of energy while taking into account the uncertainty related to renewable energy sources (e.g., photovoltaics, hydro and wind); (iii) coordinating and shaping the energy demand to flatten consumption peaks and limit their impact on the networks and on the electricity markets.

"*Smart Meters*" constitute the fundamental building block of the Smart Grid architecture. Within the next few years, these digital meters are expected to be installed at all French households.[1] Smart Meters record the individual power consumptions in real time, and send this information to a data center through a communication network. The expected volume of Smart Meter data (in France: 35 millions signals sampled every 30 min) poses a significant challenge for utility companies. In France, one year of Smart Meter data amounts to more than 600 billion data points, which is equivalent to 4.4 Terabytes.[2] Electricité de France (EDF), the main French provider of electricity, needs to anticipate managing such amounts of data in terms of **storage**, **querying** and **data analysis** capabilities. Currently, only a small subset of the 35 million Smart Meters has already been deployed, mostly through pilot studies in specific geographic areas. In order to prepare for the full deployment and test different types of distributed data management systems, EDF needs to simulate consumption data for individual households at a massive scale.

Previous studies on massive-scale processing of electrical load time series have been carried out using the Hadoop framework [8]. Also the data storage and querying aspects have been investigated in this context. The present paper describes a platform for **more realistic** simulations of electricity consumption in order to validate forecasting approaches at different levels of the electrical grid. The platform also supports the generation of **what-if-scenarios** to foresee the impact of changes in electricity usage on the quality of the forecasts. Note that electricity consumption data at the level of individual households have several distinctive features: (i) the overall number of time series is very large; (ii) the diversity of individual behavior induces a wide variety of shapes; (iii) the volatility of these time series is very high; (iv) the sum of these time series is a smooth time series with cyclical patterns. The upper time series in Fig. 1 shows the total consumption in France during 1 week, and the lower time series gives an example of an individual consumption time series during the same period of time. As can be seen, the characteristics of the load profiles at these different aggregation levels are very different.

[1] More details are available at http://www.erdfdistribution.fr/linky/

[2] Assuming that each data point requires 8 bytes memory.

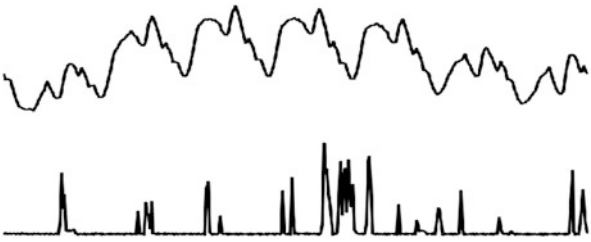

Fig. 1 Example of an individual consumption signal during 1 week *(lower time series)*, in comparison with the sum of individual consumption signals during the same period of time *(upper time series)*

This paper is organized as follows. After a review of related work on the simulation of electrical networks in Sect. 2, Sect. 3 introduces the statistical modeling approach for simulating and forecasting electrical load based on Generalized Additive Models. Section 4 describes the architecture of the simulation platform. Use cases demonstrating applications of the simulation platform are presented in Sect. 5. Finally, Sect. 6 proposes a benchmark method to evaluate how realistic are the simulations generated by the platform at different aggregation levels. Section 7 concludes with an outlook on directions for future work.

2 Related Work

There exists a wide body of literature and software tools for simulating electrical networks. Most of these tools focus on physical properties of the grid (e.g., power flows, voltage drops), typically under steady-state conditions and for a limited part of the network (e.g., transmission or distribution), and with a great level of detail in modeling the physical assets of the grid (lines, transformers, etc.). The purpose of the simulation platform presented in this paper is to emulate **statistical** properties of electrical load. In this context, bottom-up and top-down approaches have been proposed in the literature (see [19] for a detailed review). Bottom-up methods start by modeling the usage of individual electrical appliances (e.g., by a Multi-Agent System) and then compute the aggregated load, e.g., at the household or neighborhood level. While those approaches yield detailed and realistic simulations at a high temporal resolution, they are computationally expensive, require considerable modeling effort, and typically rely on assumptions about the usage of appliances that are difficult to justify empirically. Typically, bottom-up methods are used for loads only at low-level aggregations, e.g., to simulate Microgrids.

Top-down methods start by modeling aggregated load curves which are then iteratively disaggregated using statistical methods to obtain the consumption at lower levels. The main advantage of this approach is that a variety of models can be used to accurately represent features of aggregated load, and usually high-quality data for fitting those models is available at the top aggregation levels. However, top-down approaches often fail to reproduce distinctive features of disaggregated loads,

e.g., the volatility of loads at lower aggregation levels, and the localized effects of meteorological and socio-economic variables.

The simulation platform presented in this paper is designed to emulate loads throughout the entire electrical network (from individual households over low- to medium-voltage network assets up to the transmission and national level) for a country the size of France, over multiple years and under various what-if-scenarios. To the best of the authors' knowledge, there exists no previous solution for simulation studies of this scale. Another special feature of the platform presented in this paper is the modeling approach based on Generalized Additive Models, which will be discussed in the following section. As will be shown in Sect. 6, while this approach does not capture all the distinctive features of loads at individual households, it reflects well the characteristics of aggregates of 70 households or more. Hence, it can be argued that the modeling approach proposed in this paper offers a good compromise between top-down and bottom-up methods.

3 Generalized Additive Models

3.1 Background

Generalized Additive Models (GAMs) are a class of semi-parametric regression models introduced in [12] and [13]. Originally, the learning of GAMs was done using the backfitting algorithm, but recently more efficient methodologies have been introduced, among them boosting procedures (see [3]) and penalized regression methods (see [22]). GAMs have been successfully applied to electrical load forecasting at different geographical scales and network aggregation levels. For example, [18] uses GAMs to forecast the French load at the national level, achieving a Mean Absolute Percentage Error (MAPE) of less than 2 %. Ba et al. [1] studies the same data set and proposes an online learning algorithm for GAMs which is shown to further improve the forecasting accuracy. Fan and Hyndman [9] applies GAMs to regional data in the National Electricity Market of Australia, [16] shows results on data from a US utility company, and [11] demonstrates forecasting at the substation level in France. Experiments in Sect. 6 of the present paper suggest that GAMs are applicable to small aggregates of down to 70 households.

GAMs have properties which make them useful both for simulation and forecasting: They are able to capture complex non-linear relationships (e.g., between electrical load and temperature), and their estimation and prediction are straightforward. Another interesting feature of GAMs is their simplicity due to their additive structure, which makes them easy to use and understand by practitioners. This property is of particular importance in the simulation context, because it allows domain experts to design specific what-if-scenarios.

Mathematically, GAMs have the following form:

$$y_i = f_1(x_{1,i}) + f_2(x_{2,i}) + \ldots + f_p(x_{p,i}) + \varepsilon_i$$

where y_i is a univariate response variable (here the electrical load), $x_{q,i}$ are the covariates that shape y_i (e.g., meteorological conditions, the time of day, the day of week, etc.). ε_i denotes the model error at time i, also called "*noise*" in this paper. The simulation platform presented in this paper supports different types of noise: White noise, Autoregressive noise, and Heteroscedastic noise where the variance of ε_i at time i could depend on the covariates $x_{q,i}$. The functions f_q, called "*transfer functions*" in this paper, are centered around 0 to achieve model identifiability and represented using splines (in particular, they can be non-linear). A penalization term in the model estimation enforces smoothness of the transfer functions. More specifically, using the spline representation each transfer function can be written as follows:

$$f_q(x) = \sum_{j=1}^{k_q} \beta_{q,j} b_j^q(x)$$

where k_q is the dimension of the spline basis, and $b_j^q(x)$ are the corresponding basis functions (e.g., cubic B-splines) with the spline coefficients $\beta_{q,j}$. In order to estimate the spline coefficients of all the transfer functions while enforcing smoothness, the following objective is minimized:

$$\sum_{i=1}^{n} (y_i - \sum_{q=1}^{p} f_q(x_i))^2 + \sum_{q=1}^{p} \lambda_q \int \|f_q''(x)\|^2 dx.$$

Here $\Lambda = (\lambda_1, \ldots, \lambda_p)$ is a vector of penalty parameters controlling the degree of smoothness of each transfer function (the higher λ_q, the smoother f_q). This parameter is optimized through a model selection criterion, e.g., see the methodology in [21] and [23] which minimizes the Generalized Cross Validation criterion proposed in [7]. For practical computations in this paper, the implementation in the R package `mgcv` (see [20] and [22]) is used.

3.2 Load and Wind Farm Modeling

This subsection provides examples of GAMs which will be used in Sect. 5 to configure different use cases running on the simulation platform. The data set used for learning the load models was compiled by the Irish Commission for Energy Regulation (CER) in a Smart Metering trial (see the reports [5] and [6]). The data were collected half-hourly for every meter participating in the trial from

July 14th, 2009, to December 31st, 2010. In this paper, meters with missing values or replications were discarded; the resulting cleaned data set consisted of 4,623 m (residential customers and small-to-medium enterprises), each with 48 half-hourly meter readings per day over 536 days. For simplicity, days corresponding to daylight savings were dropped: October 25th, 2009, March 28th and October 31st, 2010. As the location of the individual meters is anonymized for confidentiality reasons, the weather data from the Dublin airport (downloaded from wunderground.com) were used as the meteorological covariates in the load models.

As part of the CER Smart Metering trial, one out of five different tariff classes was offered to each residential household. For the experiments in this paper, the load of households using the same tariff was aggregated, and one GAM per class was estimated. Figure 2 shows 2 weeks of data for each of the five classes. The GAM learned for each class is given by

$$y_i = \sum_{k=1}^{7} s_k(\text{TimeOfDay}_i) I_{\text{WeekDay}_i=k} + s(\text{Temperature}_i) + s(\text{TimeOfYear}_i) + \varepsilon_i \quad (1)$$

where y_i is the electrical load, TimeOfDay_i is the time of day (ranging from 0 to 47, corresponding to the half-hourly measurements at 0:30, 1:00, ..., 24:00), WeekDay_i is the day of week (1 = Sunday, 2 = Monday, ..., 7 = Saturday), Temperature_i is the temperature at the Dublin airport, and TimeOfYear_i is the time in the year (ranging between 0 on January 1st and 1 on December 31st). Note that $I_{\text{WeekDay}_i=k}$ denotes the indicator function which evaluates to 1 if $\text{WeekDay}_i = k$, and to 0, otherwise. Hence, the model includes a transfer function depending on the time of day which is specific for each week day. The transfer functions are represented using cubic B-splines, and cyclic splines for the TimeOfYear effect which enforces continuity between December 31st and January 1st. In the simulations, the noise term ε_i is sampled from a normal distribution with zero mean and a standard deviation proportional to 1 % of the signal, i.e., as explained in the previous subsection, the variance also depends on the model covariates (here: TimeOfDay_i, WeekDay_i, Temperature_i and TimeOfYear_i).

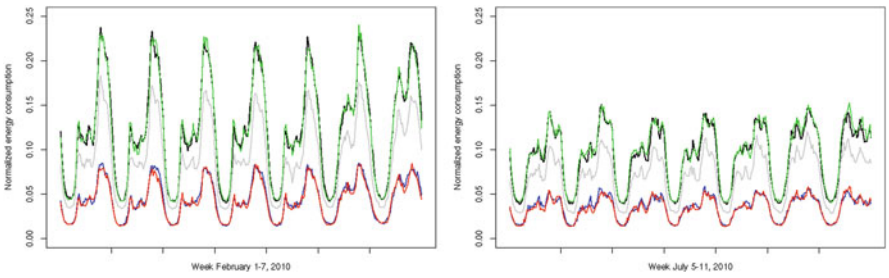

Fig. 2 Irish CER data set: Electricity consumption of residential customers signed up to five different tariff classes (represented by the curves in different colors)

For the learning of a wind farm model, a public data set from the wind power forecasting track of the GEFcom competition (see [16]) was used. In the experiments of this paper, this model was standardized and, in order to simulate wind farms of different sizes, scaled to the desired level. The GAM model is given by

$$y_i = \sum_{k=1}^{12} s_k(\text{WindSpeed}_i) I_{\text{WindDirection}_i = k} + \varepsilon_i \tag{2}$$

where y_i is the wind power, WindSpeed_i is the wind speed, and WindDirection_i the wind direction (1 = N, 2 = NNE, 3 = NE, ..., 12 = NNW). Note that the wind direction was discretized into 12 sectors (instead of using a bivariate transfer function) for parsimony reasons. In the simulations, the noise term ε_i is sampled from a normal distribution with zero mean and a standard deviation proportional to 5 % of the signal (to simulate higher uncertainty of production data), i.e., the variance again also depends on the model covariates (here: WindSpeed_i and WindDirection_i).

4 Simulator Platform Architecture

This section describes the architecture and design of the Smart Grid simulation platform, with particular emphasis on the modeling of the electrical network, the representation of load at the Meter level, and design considerations related to the scalability of the platform.

4.1 Network Modeling

Simulating the load at each level of an electrical grid requires a model of the network. The simulation platform presented in this paper models the initial network structure, and dynamic changes (e.g., reconfiguration events) applied to it. The initial **network structure** is a tree of depth six, with the nodes – from the lowest level to the root – representing Meters, Low-Voltage Stations (LVS), High-Voltage Stations (HVS), Source Substations, Regional Agencies, and the National Level. An example of a subtree, up to the Regional Agency level, is shown in Fig. 3. The numbers on the right hand side correspond to the number of nodes per network level for a country the size of France. Note that the tree structure only allows for the representation of radial networks; modeling meshed networks is a direction for future work.

To ensure the resilience and security, numerous backup lines exist in real electrical networks that enable to redirect the electrical flow from one element to

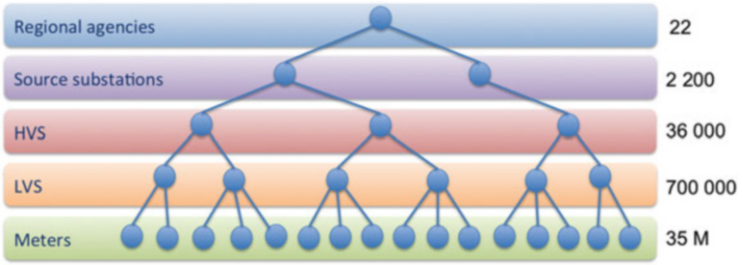

Fig. 3 Tree-based representation of the network structure grid with the approximate number of nodes per network level for a country the size of France

another. The simulation platform can take into account **dynamic reconfigurations** where, at a given point in time, a leaf node or an internal node (with its subtree) connect to a different parent node at the upper level. This can also be used to emulate mobile network elements like electric vehicles which might change their connection point to the grid depending on their location. In most networks, backup lines only exist between few, but not all the nodes. The simulation platform is able to enforce "can connect/cannot connect" constraints to ensure that dynamic reconfigurations only connect network elements that are physically linked with each other.

The **aggregated load** at any internal node in the network structure is obtained by simply taking the sum of the loads from all children elements in the tree. Electricity production (e.g., from distributed renewable energy sources) can also be taken into account and modeled as negative load. The simulation platform supports separate aggregation of load, production and net load (i.e., the difference between load and production); moreover, load can be aggregated separately for different customer classes. Note that the simulation platform does not model physical properties and only aggregates active powers. In particular, line losses are neglected, and there is no calculation of currents, voltages and other physical quantities in the network.

4.2 Representation of Load at the Meter Level

The simulation platform uses two attributes for characterizing load at the Meter level in the network: The statistical model which is used for simulating the load at a particular Meter, and the geographical location of the Meter. Typically, the simulation model is chosen from a set of "*customer classes*", e.g., representing the behavior of customers signed up to different tariffs as shown in Sect. 3.2.[3] Similarly, also simulation models for energy production (e.g., from wind farms)

[3]It is important to note that the GAMs learned on aggregated load data do not really represent load at the individual Meter level, but more an "average consumer". As will be shown in Sect. 6, GAMs fit well for aggregates of 70 households or more. The purpose for using GAMs, nevertheless, at

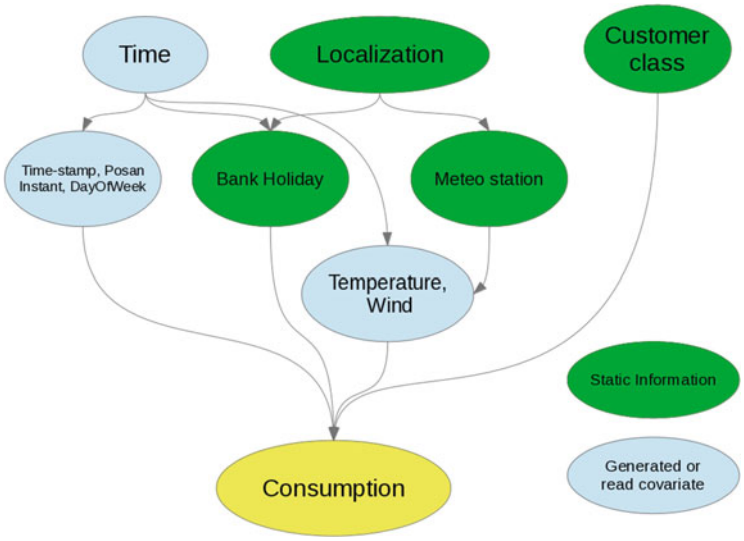

Fig. 4 Bayesian Belief Network representing the dependencies among the simulation models and covariates based on localization, time and customer type information

can be deployed at the Meter level. The geographic location of the Meter allows the simulation platform to retrieve the relevant covariates for the simulation model, e.g., the temperature data from the nearest weather station.

By taking into account the location of Meters, the simulation platform can represent complex **spatial correlations** among the simulated time series. In particular, by using the meteorological information from the nearest weather station, nearby Meters will use similar covariates in their simulation models. Another way to induce correlations between Meters is via the customer class, i.e., the type of model that is used for simulation. Figure 4 shows a Bayesian Belief Network representing the dependencies among the simulation models and covariates based on localization, time and customer type information.

Finally, the simulation platform allows for changes in the simulation model assigned to a particular Meter at given points in time. This capability can be used to represent consumers changing their behavior (e.g., due to dynamic pricing or the usage of novel electrical appliances such as electric vehicles or heat pumps), to model changes in the customer portfolio of an energy supplier, and to simulate installment of new wind farms and solar systems. Use cases illustrating this capability are described in Sects. 5.1 and 5.2.

---

the Meter level, is to represent shifts in the customer portfolio and changes in the consumers' behaviors, as will be explained at the end of this subsection.

4.3 Scalability Aspects

An important aspect in the design of the architecture of the simulation platform was scalability to enable massive-scale simulations of extended time periods much faster than realtime (e.g., simulate one year of half-hourly data from 35 millions Meters in less than 30 h). A key paradigm to achieve scalability was to use **parallel processing for streaming data**. Streams processing is a computational model designed for handling large amounts of data flows in a parallel and distributed manner. The rationale is similar to assembly-lines for manufacturing: each data element goes through different processing units, is processed and then forwarded to the next unit. Storage of the processed elements is avoided throughout the processing pipeline and performed only for the finished end product of the computations. IBM InfoSphere Streams [14] is a computing platform designed to enable high-performance, parallel and distributed processing of data streams. The challenge in designing a streaming application is to carefully design the processing line to take maximal advantage of distributed computing resources while keeping the volume of communication among these resources at a reasonable level.

A full description of the design is beyond the scope of this paper. The most important consideration was that, in the simulation platform, most of the data volume is generated at the lowest levels of the network (the Mete' and LVS levels in Fig. 3). In the case of very large networks, this requires to heavily distribute the computation at those levels. Also the volume of communication between network elements at those low levels is significant (in particular, when aggregating loads from the Meter to the LVS level), which requires to fuse Streams operators into single processing elements in order to avoid impractical communication overhead.

Scalability results from experiments with the simulation platform are shown in Fig. 5. The horizontal axis shows the number of parallel processing elements used

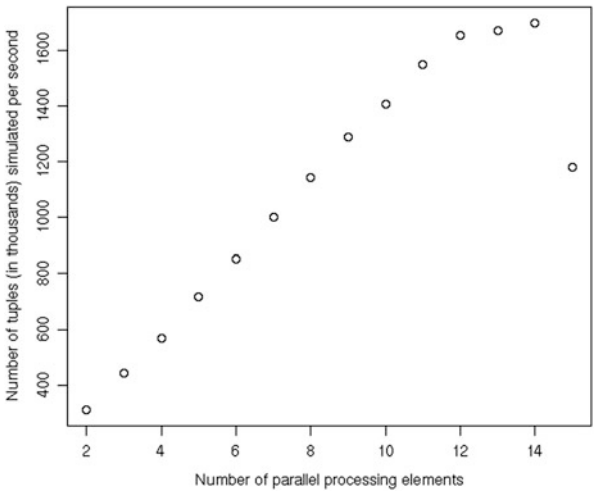

Fig. 5 Scalability of the simulation platform: The horizontal axis shows the number of parallel processing elements used in the simulation, the vertical axis the number of simulated data points per second. As can be seen, the platform scales almost perfectly linearly until the number of parallel processing elements reaches the number of physical CPUs (which was 12 in this experiment)

in the simulation, the vertical axis the number of simulated data points per second. As can be seen, the platform scales almost perfectly linearly until the number of parallel processing elements reaches the number of physical Central Processing Units (CPUs) which was 12 in this experiment. Approximately 140,000 data points can be simulated per CPU in one second. Based on this experiment, it can be estimated that 40 cores are sufficient to simulate one year of half-hourly data from 35 millions Meters (corresponding to 613.2 billion data points) in approximately 30 h.

5 Use Cases

This section presents three different use cases generated with the simulation platform presented in this paper, each of them addressing a specific challenge for utility companies from the emerging Smart Grids.

5.1 Forecasting a Time-Varying Portfolio

The first use case studies the impact of losses and gains of customers in a utility company's portfolio on the aggregated consumption. It is motivated by the deregulation and competition in retail electricity markets which will allow customers to change their electricity provider. Another goal of this use case is to illustrate the effectiveness of the online learning algorithm for GAMs introduced in [1] to forecast the aggregated consumption.

To simulate the changes in the portfolio, the five different customer classes learned from the Irish CER data set (see Sect. 3.2) are used. Two different kinds of changes are simulated in this use case: abrupt and gradual changes. Let $P_t = (p_{t,k})_{k=1,\ldots,5}$ denote the proportion of customers in the portfolio belonging to each class at a given time t. An abrupt change occurs at time t_0 if there is a significant difference between P_{t_0} and P_{t_0+1}. A gradual change is a linear transition of P_t between two points in time t_0 and t_1. Losses and gains of customers can be simulated by introducing a sixth "void" class which represents zero consumption, and simulating customers switching from/to this class to/from any of the five tariff options in the portfolio.

Figure 6 shows an example: Here, a portfolio of residential customers was simulated, uniformly distributed over the five tariff classes, with a loss of 20 % of the customers over the course of two years. The black line in the left plot shows a simulated abrupt change, while the blue line depicts a gradual, linear loss over the two years. The right plot illustrates the performance of forecasting algorithms in the gradual loss scenario. Here the black line shows the actual loads, the blue line shows

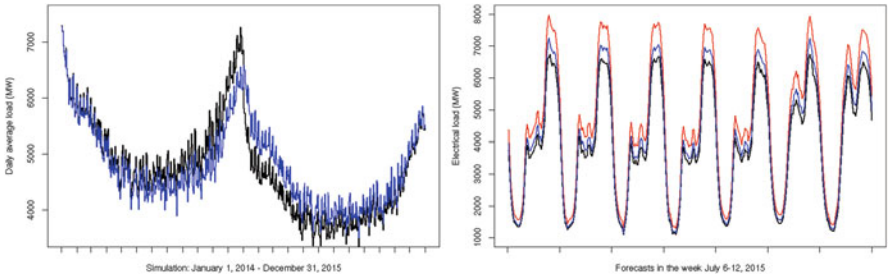

Fig. 6 *Left*: Simulation of a customer portfolio with a loss of 20 % of the customers over two years. *The black line* shows an abrupt loss after the first year, the *blue line* a gradual, linear loss over the two years. *Right*: Performance of forecasting algorithms in the gradual loss scenario. Here the *black line* shows the actual loads, the *blue line* shows the forecasts obtained by a GAM with online learning, and the *red line* the forecasts obtained by a GAM without online learning. As can be seen, the online learning is able to track some of the losses, resulting in a higher forecasting accuracy

the forecasts obtained by a GAM with online learning (using the algorithm proposed in [1]), and the red line the forecasts obtained by a GAM without online learning. As can be seen, the online learning is able to track some of the losses, resulting in a higher forecasting accuracy than the non-adaptive method. More generally, this example shows the usefulness of the simulation platform for comparing the performance of forecasting algorithms under different what-if-scenarios.

5.2 Impact of Wind Power Generation on the Distribution Grid

Managing the injection of power from renewable energy sources into the electrical grid, particularly wind power, raises high levels of concern for utility companies. Electricity providers and network operators need to optimize their production and grid management, respectively, to cope with those intermittent energy sources. Due to the high variability of wind power and its localized properties, simulations are an important tool for making decisions in this context.

Figure 7 shows examples of the simulations generated by the platform. The blue curves represent actual loads, generated using the same models as in the previous use case. The green curves show the simulated amount of wind power injected into the distribution network. For the simulation of wind power, the GAM introduced at the end of Sect. 3.2 was used. The difference between the two curves (i.e., the net load) is shown by the red curves. The plot on the left-hand-side displays a detail of 1 week, while the right plot shows the evolution over one year with an increase of 20 % in wind power capacity, corresponding to the installment and connection

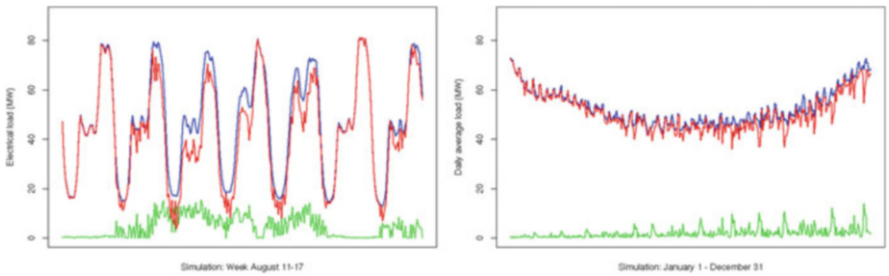

Fig. 7 Simulation of actual loads (*blue*), power from distributed wind farms (*green*), and the resulting difference, i.e., the net loads (*red*). The *left graph* shows a detail of 1 week, the *right graph* the daily averages over one year, with a simulated 20 % increase in wind power capacity

of new wind farms to the grid.[4] Note that, similarly, the simulation platform also supports the simulation of distributed power generation from photovoltaic systems.

5.3 Network Reconfigurations

In the last use case, the effect of network reconfiguration events is simulated. Such events, where loads are transferred over alternative lines or to different substations, become increasingly important in the operation of distribution networks where the trend is towards a more active management of the grid in order to increase the efficiency while coping with the challenges, e.g., due to power injections from distributed renewable energy sources. In this paper, only reconfigurations between the LVS and HVS network levels (see Sect. 4.1) are considered, where an LVS node connects to a different HVS parent node. In general, however, the simulation platform can represent reconfiguration events at any level in the network. Interestingly, the same logic can be applied to simulate electric vehicles (nodes at the Meters level) connecting to different charging stations (nodes at the LVS level), e.g., related to changes in location. Note that the platform presented in this paper can read reconfiguration events either from static files (e.g., generated by the user based on statistical assumptions and/or historical data), or dynamically receive them via a web server interface.

Figure 8 shows an example. The graph on the left shows how network entities and their current status (load, outside temperature etc.) are displayed on a map. The same interface can be used to dynamically introduce reconfiguration events by selecting a new HVS parent node for a particular LVS node. Typically, the new

[4]The installment of new wind power capacity can be represented by network nodes which, at specified time points, change their simulation model from a "void" GAM (producing zero values) to a GAM model simulating wind farm output. Compare with the remark at the end of Sect. 4.2.

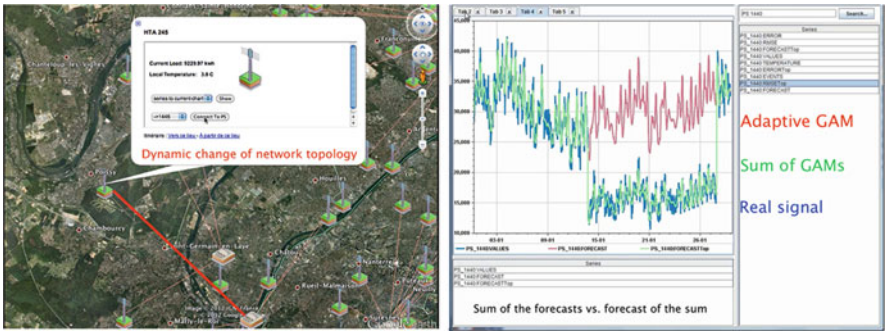

Fig. 8 Dynamic reconfiguration events simulated by the platform. The *left picture* displays LVS and HVS network elements on a map. By using the menu options in the *white balloon*, the user can manually connect the LVS element to a new parent at the HVS level. Changes in the connectivity will be reflected by the *red lines* displayed on the map. The *right hand side* shows how reconfiguration events impact the load signal and forecasts at the parent node. Here the *blue line* represents the actual load, the *red line* the forecasts at the parent node, and the *green line* the sum of forecasts from all children nodes. After approximately half of the displayed time period, one of the children is connected to a different parent node at the HVS level, resulting in a significant decrease in load (*blue line*). While the forecasts at the parent node (*red line*) are unable to quickly adapt to this change, taking the sum of forecasts from all children nodes (*green line*) reflects the actual configuration. Shortly before the end of the displayed time period, the children node is reconnected to its original parent, hence the load goes back to the original level

parent node is chosen from a list of candidates to which physical connections exist. The blue curve in the right graph shows the load at an HVS node. As can be seen, there is a significant load decrease after 2 weeks, which is due to a child of this node connecting to a different parent at the HVS level. After 2 weeks, the child reconnects to its original parent, and the load reaches the previous level. The red curve shows the load forecasts for the HVS node using an adaptive GAM model. While these adaptive models are very effective in tracking long-term trends and changes (see Sect. 5.1), they are not capable to adapt to such sudden shifts. The green curve represents the load forecasts obtained by taking the sum of the load forecasts for the children of this HVS node. Clearly, this approach is favorable in the presence of reconfiguration events.

6 Statistical Evaluation

The goal of this section is to evaluate how realistic are the load simulations generated by the platform, both at an aggregated and at the individual Meter level. Most approaches in the literature for this purpose use statistical hypothesis tests to assess whether the simulated and the real data have the same distribution. For

instance, in [15] a Mann-Whitney U test is used to test the similarity of the real and simulated data distributions. In [17], different statistics of the simulated and real data set are compared to assess how realistic the simulations are. The evaluation protocol in this paper is based on a classifier which aims at discriminating real and simulated data. The more difficult it is for the classifier to distinguish these data, the more realistic the simulations are. This approach is motivated by previous work which combines supervised and unsupervised approaches in order to evaluate the quality of the unsupervised task. For instance, the cascade evaluation [4] enriches a supervised dataset with the *cluster id* of each example. Then the *cluster id* is used by a classifier as an additional explicative variable. The cascade evaluation estimates the quality of the unsupervised task by measuring the improvement of the classifier when the *cluster id* is used. Another example is the use of a classifier to detect changes in the distribution of a data stream [2]. In this approach, two time windows are used to capture the "*current*" and the "*normal*" behavior of the observed system, respectively. Changes are quantified by the ability of the classifier to discriminate the both classes.

6.1 Experimental Protocol

The goal of the first experiment in this section is to assess the accuracy of GAMs depending on the size of the groups over which the load is aggregated. Same as in Sect. 3.2, the data set for this experiment is the Irish CER Smart Metering trial, and the GAM is given by Eq. (1). For aggregation sizes between $k = 10$ and $k = 800$, a random sample of k meters is drawn from the CER data set and then aggregated into a single time series. A GAM is learned on the first 70 % of this time series, then the model's Mean Squared Error (MSE) and standard deviation of the error (Sd) is calculated on the remaining 30 % of the time series. Overall, this procedure is repeated $n = 1{,}000$ times for each aggregation size k, and the average MSE and Std are computed for each k.

The results of this experiment are shown in Fig. 9. As to be expected, the models become more accurate (i.e., the MSE decreases) with increasing sample sizes, essentially illustrating the Law of Large Numbers which states that aggregating independent random variables following the same distribution yields stabilized variables around the mean value. Noteworthy is the inflection point in the Sd curve around the sample size $k = 70$: Beyond this point, the standard deviation of the model errors is slightly increasing. Similarly, the decrease in the MSE beyond this point is much less pronounced. A possible explanation is that the distributions of the individual meter signals are not identical, therefore, if too many signals are aggregated, information specific to some meters is lost while the benefit of aggregation to reduce noise does not compensate that loss of information. Therefore, the variance of a model learned on that sample will increase.

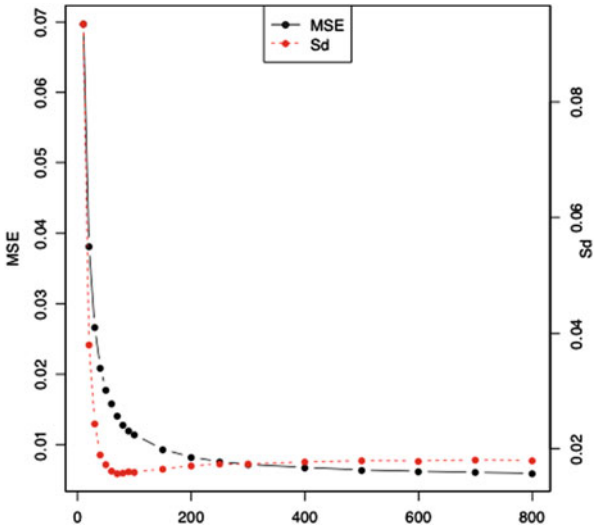

Fig. 9 Accuracy of GAM depending on the size of randomly aggregated groups of meters, measured in terms of Mean Squared Error (MSE) and Standard deviation of the error (Sd)

This experiment suggests two directions how to improve the quality of the simulations. First, the GAM approach is effective for simulating aggregations of 70 (or more) households, but not suitable for smaller sizes. Hence, for those low-level aggregations, other modeling approaches will be required. Second, blindly aggregating meters can lead to an information loss and an increase of variance of the error. Therefore, clustering meters into similar classes could improve the modeling accuracy.

Next, the effectiveness of this clustering approach using the k-means algorithm with the Euclidean distance is investigated. The clustering of the meters is used to build a **generative model** which is obtained by learning different GAMs for the aggregation of meters from each cluster. The k-means algorithm is parameterized in two different ways:

1. **Naive setting:** The number of clusters is arbitrarily fixed at $k = 10$. The corresponding generative model is used as a base line.
2. **Taking into account GAM performance:** Using the results from Fig. 9, an aggregation size of 70 m per cluster is found to be optimal, because it yields a good performance in terms of the MSE and the minimal standard deviation of errors. Correspondingly, the number of clusters is fixed at $k = 60$, leading to an average group size of 70 m (n.b.: the total number of residential meters in the data set is approximately 4,000).

For both settings, the k-means algorithm is applied to one year of half-hourly meter data.

6.2 Evaluation Protocol

The generative model obtained from the k-means clustering is first evaluated on simulations of the aggregated consumption. In particular, the sum of the 4,000 simulated individual meter signals is compared with the sum of the 4,000 real signals from the CER data set over the same time period. The Mean Absolute Percentage Error (MAPE) and the Root Mean Square Error (RMSE) are calculated as evaluation criteria.

In order to assess how realistic the simulated individual meter signals are, a classifier for discriminating real and simulated signals is used. The classification task is defined as follows: the data set consists of 8,000 time series (4,000 simulated and 4,000 real ones), each described by 336 numerical explicative variables (denoted by v_i), corresponding to 48 data points per day over 7 days. The target class variable c is equal to "0" for the simulated signals, and equal to "1" for the real ones. The data set is split into two disjoint parts: 70 % of the data are used for training, and 30 % for testing. For the classification, a simple Naive Bayes classifier is used. In particular, the range of each explicative variable v_i is discretized into 10 intervals, such that the numbers of training observations lying in each interval are equal. The conditional probabilities $P(v_i|c)$ for $i = 1, 2, \ldots, 336$ are estimated by the corresponding sample frequencies, and then $P(c|v_1 \ldots v_{336})$ is computed by applying Bayes' rule. The classifier is evaluated by using the Area Under Curve (AUC) metric [10]. Recall that a perfect classifier reaches an AUC equal to 1, and a random classifier an AUC equal to 0.5.

6.3 Results

Table 1 reports the RMSE and MAPE of the GAMs for the two different numbers of clusters $k = 10$ and $k = 60$. These two metrics assess the ability of the simulator to fit aggregated individual load signals. In both cases, the value of k has an insignificant impact on the RMSE and the MAPE. Note that a MAPE of 10 % is relatively high, however, it needs to be taken into account that the GAMs were learned on small aggregates and not at the national level.

Table 1 also reports the AUC score of the Naive Bayes classifier for the generative models with $k = 10$ and $k = 60$. In both cases the classifier is able to separate almost perfectly the simulated signals from the real signals, which underlines the difficulty of building a realistic simulator for individual load signals. This result can be intuitively explained by the fact that the GAMs in this experiment were learned on aggregated loads, which are much smoother than the individual signals. The Gaussian noise added to the simulated signals fails to exactly reproduce the characteristics of individual consumption signals. Alternative approaches will be discussed in the conclusions of this paper. Nevertheless, a significant drop in the classifier accuracy from AUC 0.927 for $k = 10$ to 0.806 for $k = 60$ can be observed.

Table 1 Comparative evaluation of generative models based on $k = 10$ and $k = 60$ clusters. The MAPE and RMSE measure how accurately the models are fitting the real data, while the AUC indicates how difficult it is for a classifier to distinguish between real and simulated data (hence, how realistic the simulations are). Note the drop in the classifier accuracy from AUC 0.927 for $k = 10$ to 0.806 for $k = 60$, which indicates that optimizing the granularity of the generative model can significantly improve the authenticity of simulations

	Criterion	$k = 10$	$k = 60$
How accurate?	Mean Absolute Percentage Error (MAPE)	10.81 %	10.73 %
	Root Mean Squared Error (RMSE)	283.21	283.05
How realistic?	Area Under Curve (AUC)	0.927	0.806

This means that using the clustering of consumer signals in the generative model can significantly improve the authenticity of the simulated signals.

7 Conclusion

In this paper, a platform for massive-scale simulation of electrical load in Smart Grids has been presented. The paper has provided details on the underlying statistical methodology, based on Generalized Additive Models (GAMs), and explained the architecture of the platform, with particular emphasis on scalability aspects. Experiments have shown the scalability and computational power of the platform, which is able to simulate one year of half-hourly load data for the entire electrical network in a country the size of France. The paper has presented three different use cases generated by the simulation platform, illustrating the value of the platform for power system engineers, statisticians and econometricians to study various what-if-scenarios, e.g., related to dynamic reconfigurations of the electrical network, changes in the customer portfolio and consumers' behavior, and increasing capacity of distributed renewable energy sources such as solar and wind.

In an evaluation study, the paper has shown that GAMs provide realistic simulations for aggregated load signals of at least 70 individual households. However, it has been demonstrated that novel modeling approaches are needed for simulating lower-level aggregates. Possible ideas for future research in this direction are: (i) using point processes (e.g., non-homogeneous Poisson); (ii) taking into account ancillary information (e.g., higher-resolution meteorological data and socio-economic indicators); (iii) considering GAMs with random effects and spatio-temporal correlations.

References

1. Ba, A., Sinn, M., Goude, Y., & Pompey, P. (2012). Adaptive learning of smoothing functions: Application to electricity load forecasting. In P. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 25, pp. 2519–2527). Curran Associates, Inc.
2. Bondu, A., & Boullé, M. (2011). A supervised approach for change detection in data streams. In *IJCNN (International joint conference on neural networks)*, San Jose (pp. 519–526). IEEE.
3. Bühlmann, P., & Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting (with discussion). *Statistical Science, 22*, 477–522.
4. Candillier, L., Tellier, I., Torre, F., & Bousquet, O. (2006). Cascade evaluation of clustering algorithms. In J. Fürnkranz, T. Scheffer, & M. Spiliopoulou (Eds.), *17th European conference on machine learning (ECML'2006)*, Berlin (Volume LNAI 4212 of LNCS, pp. 574–581). Springer.
5. Commission for Energy Regulation. (2011). *Electricity smart metering customer behavior trials findings report* (Technical report). Commission for Energy Regulation, Dublin.
6. Commission for Energy Regulation. (2011). *Results of electricity cost-benefit analysis, customer behavior trials and technology trials* (Technical report). Commission for Energy Regulation, Dublin.
7. Craven, P., & Wahba, G. (1979). Smoothing noisy data with spline functions: Estimated the correct degree of smoothing by the method of general cross validation. *Numerische Mathematik, 31*, 377–403.
8. dos Santos, L. D. P., Picard, M. L., da Silva, A. G., Worms, D., Jacquin, B., & Bernard, C. (2012). Massive smart meter data storage and processing on top of Hadoop. In *International workshop on end-to-end management of big data, VLDB (International conference on very large data bases)*, Istanbul.
9. Fan, S., & Hyndman, R. J. (2012). Short-term load forecasting based on a semi-parametric additive model. *IEEE Transactions on Power Systems, 27*(1), 134–141.
10. Fawcett, T. (2003). *ROC graphs: Notes and practical considerations for data mining researchers* (Technical report HPL-2003-4). HP Labs.
11. Goude, Y., Nedellec, R., & Kong, N. (2013, to appear). Local short and middle term electricity load forecasting with semi-parametric additive models. *IEEE Transactions on Smart Grid, 5*(1), 440–446.
12. Hastie, T., & Tibshirani, R. (1986). Generalized additive models (with discussion). *Statistical Science, 1*, 297–318.
13. Hastie, T., & Tibshirani, R. (1990). *Generalized additive models*. Boca Raton: Chapman & Hall/CRC.
14. International Technical Support Organization. (2013). Addressing data volume, velocity, and variety with IBM InfoSphere streams V3.0. http://www.redbooks.ibm.com/redbooks/pdfs/sg248108.pdf. March 2013.
15. Muratori, M., Roberts, M., Sioshansi, R., Marano, V., & Rizzoni, G. (2013). A highly resolved modeling technique to simulate residential power demand. *Applied Energy, 107(C)*, 465–473.
16. Nedellec, R., Cugliari, J., & Goude, Y. (2014, to appear). Electric load forecasting and backcasting with semi-parametric models. *International Journal of Forecasting, 30*(2), 375–381.
17. Paatero, J. V., & Lund, P. D. (2006). A model for generating household electricity load profiles. *International Journal of Energy Research, 30*(5), 273–290.
18. Pierrot, A., & Goude, Y. (2011). Short-term electricity load forecasting with generalized additive models. In *Proceedings of ISAP power*, Hersonissos (pp. 593–600).
19. Swan, L., & Ugursal, V. (2009). Modeling of end-use energy consumption in the residential sector: A review of modeling techniques. *Renewable and Sustainable Energy Reviews, 13*(8), 1819–1835.
20. Wood, S. (2001). mgcv: GAMs and generalized ridge regression for R. *R News, 1*(2), 20–25.

21. Wood, S. (2004). Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association, 99*, 673–686.
22. Wood, S. (2006). *Generalized additive models, an introduction with R*. Boca Raton: Chapman and Hall/CRC.
23. Wood, S. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semi-parametric generalized linear models. *Journal of the Royal Statistical Society Series (B), 73*(1), 3–36.