Evaluating Combined Load Forecasting in Large Power Systems and Smart Grids

***Abstract*—We present here a combined aggregative short-term load forecasting method for smart grids, a novel methodology that allows to obtain a global prognosis by summing up the forecasts on the compounding individual loads. More accurately, we detail here three new approaches, namely bottom-up aggregation (with and without bias correction), top-down aggregation (with and without bias correction), and regressive aggregation. Further, we have devised an experiment to compare their results, evaluating them with two datasets of real data and showing the feasibility of aggregative forecast combinations for smart grids.**

I. Introduction

The term smart grid has been coined to describe the evolution of the current power networks into the future computer-aided grids. Among other novelties, all low-voltage meters will be remotely managed and this fact arises new chances for load forecasting. Short-term load forecasting (STLF) is a major column in everyday's life of power networks nowadays. An accurate prediction is necessary to issue day-ahead network operation plans and, hence, any inaccuracy or deviation may result in a loss of hundreds of thousands or even millions of dollars [1] for the operator.

Historically, there has been an estimable effort around short-term load forecasting. The solutions proposed can be divided into two main groups, depending on the strategy followed. On the one hand, statistical methods aim at estimating a regression function that matches the points registered in the historical load data (i.e. consumption records). There exist very effective ways to approach regular curves but, since load forecasting usually lacks of regularity, statistical methods alone normally present poorer results than their counterparts [1], [2]. On the other hand, Artificial Intelligence has designed several techniques, methods, and models that deal with risk and uncertainty (the main aspects behind prediction). The most popular due to their efficiency are Support Vector Machines (SVM) and Neural Networks (NN) (see Section III for a more accurate description).

Still, despite their accuracy, they also present a number of drawbacks such as difficult parametrisation, non-obvious selection of variables and over-fitting. Moreover, in the praxis they normally require much historical data to *learn* the patterns inherent on it [2]. The most widely-used of these methods, NN, further adds several extra inconveniences such as a very time-consuming learning process, the risk of local minima, the lack of an exact rule for setting the number of hidden neurons to avoid over-fitting or under-fitting, the inability to generate explanations for their results, and their poor scalability [3].

Besides, some models may perform adequately under certain conditions whereas fail in others. Similarly, each one has been designed with a certain purpose in order to offer distinct information and precision. If we simply choose the one whose error is minimal as the optimum, we may lose some important feedback. Model combination deals exactly with this issue: it is a well-established methodology for improving forecasting accuracy [4] and has already been successfully applied in other disciplines (see [5] for a survey).

According to [5], [6], research in this area has issued two primary conclusions, one expected and one surprising. The expected one says that combining forecasts reduces the error compared to the average error of the singular forecasts (conclusion also highlighted in [4], [7]). The unexpected one shows that a simple average of the component forecasts performs as well the most sophisticated statistical approaches.

This idea has already been applied to STLF with classifiers such as average [8], multiple linear combination [9], or diverse machine learning techniques to determine the weights with which individual forecast methods are mixed [8], [10], [11].

Nowadays, in transmission and distribution networks, the consumption prognosis is issued on the overall consumption since there is a huge amount of single loads and nowadays many of them are not yet remotely metered (which makes data collecting very time-consuming). Moreover, this approach is reinforced by the fact that, according to the *Central Limit Theorem*, the same reason renders measurement very smooth. Therefore, the overall load can be more easily predicted. In this way, model combination can be understood as finding the proper mixing of different predictors to achieve a more accurate global forecast.

Yet with the advent of the smart grids, this situation will change since low-voltage meter data will be available to issue single predictions and model combination will be able to be applied to improve these. Hence, there will exist two possibilities of issuing a forecast on a certain part of a smart grid: adding up the single consumptions and perform a global forecast (*top-down* method) and adding up the sum of the forecasts on the single consumptions (*bottom-up* method). Additionally, the latter allows a slight modification: forecasting a regression of the individual loads recorded by the meters (*regression* method).

Please note that the goal of this study is not to improve the performance of other forecasting methods and models, but to evaluate whether combined forecasting will work on smart grids. The actual performance of combined forecasting depends on the individual methods used for prediction and we have already coped with that objective in other works [12], [13].

Against this background, we will advance the state of the art in three main ways. We present for the first time three different approaches for combined *aggregative* forecasting in STLF in smart grids. They are namely a bias correction method, a bottom-up and top-down approaches and a regression method for forecast combinations. We have tested all of them thoroughly on a number of real consumption datasets and we will discuss the comparison of their results.

The remainder of the paper is structured as follows. Section III presents the different combinations algorithms. Section IV details the tests, describes the used datasets and discusses the obtained results. Section II presents related work. And, finally, Section V concludes and draws the avenues of future work.

II. Background

STLF presents a large research tradition applied to country loads (see [14], [2], [15] for a comprehensive survey on STLF) where as it has not been so prolific restricted to more accurate goals (e.g. buildings). Research on STLF mainly focuses on two branches. The first one deals with *statistical* methods and *causal models* like Dynamic Linear or Non-Linear models, ARMAX models [16], or Non-parametric Regression [17], with ARIMA as the method that achieves most promising results [18]. The second group is related to Artificial Intelligence methods that address and try to cope with the non-linear characteristics of the historical data (e.g SVM [19], [20], or NN [21], [22], [23]).

As aforementioned, meta models are not a new approach. The branch of work that has gathered the most attention is focused on *meta-heuristics* (term already coined in 1986 [24]), an upper-level strategy that controls and modifies other heuristics in order to produce solutions of higher quality [25], [26]. In this way, the research on meta-heuristics has concentrated on two areas. The first one uses a meta-heuristic to calculate the best set of parameters of a SVM or a NN [27], [28], [29], [30], [31], [32], [33] (see [3] for a survey on NN-based hybrid methods) but these works suffer from the same flaw single models do (i.e. without heuristics). The second area has explored the optimal way of combining the output of the single models, usually by assigning weights (see [7] for different approaches to this end). For instance, a very simple but effective approach consists of defining equal weights, usually referred to as the *Simple Average* combination method (which, despite being simple, has shown to be surprisingly effective [6], [8]). More sophisticated approaches include Linear Combination [9] (e.g. diverse Machine Learning techniques to determine the weights [8], [10]), Dynamic Optimal Weight Combination [11], Genetic Algorithms as best model selectors [34] or Rule-Based best model selection [35], [36], [37] (which is similar to the first classifier we have designed). Please see [38] for a survey on meta-heuristics and forecast combination applied to power systems in general.

Finally, [39] proposes a multiple classifier system combined with NN. The dataset used is divided into several parts: 24 hours, 3 days, 1 week, and 1 month before the predicting hours. This dataset is used as a dynamic weight to be added to the base MAPE classifiers in order to obtain the integrated total result of the forecasting. The error obtained was the 15.12%, which is very high. We use only the last q days for the forecast, showing an error of approximately the 6%.

III. Aggregative Combined STLF

In order to maintain the precarious equilibrium between generation and load in every grid we need a very reliable prediction of both values. In this paper we focus on the latter case.

As aforementioned, the major problem in predicting energy consumption is the lack of a meter that registers the *actual* load flowing. The only possibility is to aggregate the loads of all consumers and to take into consideration line losses. We will compare here different forms of *aggregation* in order to find the optimal one.

The methodology we put forward comprises two steps (see Fig. 1 for further details):

1) *Local Prediction:* For every consumer node i we issue a load forecast l_i as accurate as possible. This problem has been already dealt with in the following papers: [12], [13], so we will just provide a brief introduction to the methods used.
2) *Aggregation post-process:* We then combine these forecasting results to calculate the actual load L of the overall grid. In this article we will group the different post-process aggregation methods in three different families described in the following subsection.

A. Local Prediction

1) Time Series model: We have chosen an *Autoregressive Model* (AR) which is commonly used for modelling univariate time series for every hour and day type:

$$s_t^{h,d} = \sum_{i=1}^{q} \varphi_i^{h,d} s_{t-i}^{h,d},$$

where $\varphi_i^{h,d}$ are the model parameters for the hour h and day type d. Note that we have computed the q last values of the *same* day type (e.g. with $q = 3$, from a Tuesday, the previous Monday, Friday, Thursday) and not the q last chronological values (e.g. from a Tuesday, the previous Monday, Sunday, and Saturday).

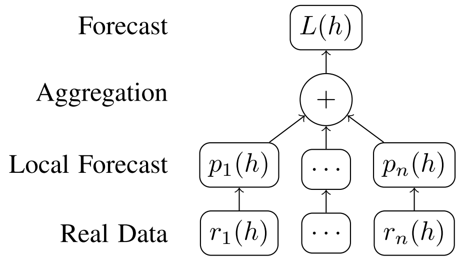
Moreover, we assign weights (model coefficients l) for those days of the prediction window, in order to give a higher priority to the latest data against the oldest values, by $polynomial$ or $exponential$ methods. Polynomial methods produce the following parameters:

$$\varphi_i = \frac{(q-i)^l}{\sum_{i=0}^{q}(q-i)^l},$$

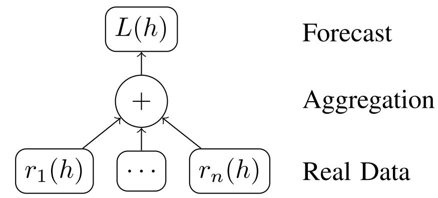
whereas the exponential method produces:

$$\varphi_i = \frac{2^{(q-i)}}{\sum_{i=0}^{q} 2^{(q-i)}},$$

where q is the value of learning window and l can take values $l \in \mathbb{Z}$ for polynomial case. We have used different values for the parameter l. Namely we have carried out our tests with $l \in \{0, 1, 3, exp\}$. Note that $l = 0$ corresponds to the mean of the previous values and exp denotes the exponential method.



(a) Bottom-up method's flowchart



(b) Top-down method's flowchart

Fig. 1. Flowchart of the different forecasting methods. Note that the regression method has the same flowchart as the bottom-up method changing the + node for an arbitrary function.

2) Polynomial model: The second model consists of univariate polynomial that tries to (clumsily) capture the load curve. It is defined as follows:

$$l_\delta(h) = \sum_{i=0}^{\delta} \alpha_i h^i,$$

where $h \in [0, 23]$ denotes the hour of a day, δ is the degree of the polynomial and α_i for $i \in \{0, \ldots, \delta\}$, the polynomial coefficients. There is one polynomial for every day type that is re-adjusted every day using the *least squares* technique over the data of the previous q days of the same day type. We have tested several degrees, namely $\delta \in \{2, 4, 6, 8\}$.

3) Support Vector Machines: SVMs construct a hyperplane or set of hyperplanes in a high (or infinite) dimensional space, which can be used for classification, regression, or other tasks. SVM have been already applied to load forecasting in buildings [40]. In this case we have used a ν-SVR using a Radial Basis Function as kernel and parameters: threshold $\nu = 0.9$, soft margin parameter $C = 10$ and kernel parameter $\gamma \in \{1, 10^{-1}, 10^{-3}, 10^{-5}\}$. The explanation of these parameters are out of the scope of this paper (see [41]).

B. Aggregation post-process

1) Bias Correction: This post-process method adds a *Gaussian Random Value* with the same mean and standard deviation as the error measured in the learning windows. The aim of this strategy is to mimic the error behaviour of the forecast (hoping) to *correct* the typical historical error of the output. In this case we have tested two approaches:

- *Bottom-Up:* We add the error measured locally to the forecast of every node and then we obtain the simple average of the post-process forecast. Suppose that $\hat{\mu_i}$ and $\hat{\sigma}_i^2$ represent the mean and the variance (respectively) of the local error measured in the forecasting of the load of the node i. Then, if we denote by e_i a random value following a Gaussian random variable of parameters $\hat{\mu_i}$ and $\hat{\sigma}_i^2$, we may compute the total forecast L with the following expression:

$$L := \sum_{i=0}^{n} (l_i + e_i).$$

- *Top-Down:* We add the error measured globally to the simple average of the forecast of every node. *I.e.* suppose that $\hat{\mu}$ and $\hat{\sigma}^2$ represent the mean and the variance (respectively) of the total error measured when forecasting the total load, then, if we denote by E a random value following a Gaussian random variable of parameters $\hat{\mu}$ and $\hat{\sigma}^2$, we may calculate the total forecast L with the following expression:

$$L := E + \sum_{i=0}^{n} l_i.$$

Please note that these methods are fairy different because the aggregation process can overlap or compensate the errors and any correction measure (as the exposed above) should take care of that. Also, keep in mind that we can apply both techniques to any of the local prediction methods not just the presented here.

2) Regression Methods: In this case we suppose that there exists a function f that relates the forecasts of every node to the actual load L of the grid. In other words, we have modelled the actual load L in the following terms:

$$L(h) := f(p_1(h), \ldots, p_n(h)) + \xi_h,$$

where $p_i(h)$ is the forecast of the node i in the hour h, and ξ_h is a Gaussian random variable with mean 0 and variance σ_h^2. To this extent, we can use several methods to approach the function f, for example: Polynomial Models, Neural Networks, Support Vector Machines or Genetic Programming. Note that this method does not only aim at finding the relation between all the forecasts p_i and L (since this enterprise is trivial) but also at defining the relation between the *error* in the forecasts and L.

For that purpose, we have used a ν-SVR using a Radial Basis Function as kernel performing a grid search in a deeper degree than the one used in the learning method. More accurately, we have tested all the possible combinations of the parameters in Tab. I to train the SVM (see [41]).

TABLE I
PARAMETERS USED TO TRAIN THE SVM IN THE REGRESSION METHOD.

Parameter	Values
n	3,5,10,30
C	0.1,1,10,100
γ	10,1,0.1
ν	0.1,0.5,0.9
ε	0.1,0.001,0.0001

IV. Experimental Results

A huge drawback in smart-grids research is the sparse or non-existing consumption data. It is difficult to obtain real individual load data acquiring the overall one is also non trivial. The methodology presented herewith basically aims at testing whether aggregating the forecasts on the compounding loads is more accurate than directly predicting the whole load. Therefore, we decided to use similar data. Specifically, we have tested this approach with two different sets. The first one contains the primary substation records of New York City "aggregated in a manner that best represent each zone" [42]. The records of the dataset have been collected from June 2001 to October 2011, detailing the hourly consumption of ten primary areas (including the famous August 2003 blackout). Also note that in 2005 one of the substations was split up in two, with the subsequent difficulties it arises. The second dataset belongs to the PJM Interconnection that comprises the states of Delaware, Illinois, Indiana, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia, West Virginia and the District of Columbia, being one of the biggest electrical market system of the world. In this case, this data presents substation records aggregated by regional markets since January 2008 until December 2010.

In both cases we have aggregated all substation data to obtain the overall consumption, so we could issue prognosis on that value as well in order to use that new data to validate the predictions. Indeed, we have compared the predicted result with the real consumption value and then computed the Mean Absolute Percentage Error (MAPE) to measure it. We have selected this error to evaluate performance of the models since it is unit free; this is, it allows comparisons between forecasting errors from different measurement units. Moreover, it is the error measure most widely used in forecasting [21]. It is calculated as follows:

$$MAPE := \frac{1}{days}\sum_{j=1}^{days}\left(\frac{1}{24}\sum_{i=1}^{24}\frac{|r_i^j - p_i^j|}{r_i^j}\right)\times 100,$$

where p_i^j is the predicted value of the load for the hour i of the day j, r_i^j the actual one and $days$ represent the numbers of days in that particular datasets (1 091 days in the PJM dataset and 3 760 days in NYC).

Tables II to VII summarise the results of applying the aggregative post-processes as described in Section III to the polynomial, SVM and AR local predictors and applied to both datasets. Red figures highlight worse performance in contrast to green ones (black means that both bottom-up and top-down accomplished the same result), the best result in each table is underlined. Note that the Regression column contains the results with the best parameters found for the SVM used (in all cases were $n = 3$, $C = 0.1$, $\gamma = 10$ and $\nu = 0.9$).

Finally, we have estimated the best MAPE that can be expected given that the load curve follows any of the local models (see the Appendix A for a detailed explanation on the procedure). Please note that the expected MAPE is only related to the forecast accuracy in an asymptotic way (see [43] for a detailed discussion on the issue). Tab. VIII present a comparison of this expected MAPE with the results obtained by the Transmission System Operator (TSO) in the PJM Interconnection and NYC and finally with the MAPEs obtained with our methods. This table show evidences that the method developed *ex-profeso* for the NYC TSO [42] is improved by using of weather forecast, local work calendar and characterisation of special days (such as important sport events or local holidays and celebrations) and hence is far better than our general-purpose method. On the other hand, the methods used in the PJM interconnection are probably not tuned the same way. It is true that the load in PJM is more difficult to forecast, but our method show that it is possible to improve their results, even with a general-purpose method (as we are able to slightly beat the PJM operator forecast by a 0.42%). Again, the goal of this paper is not to top methods tailored to certain scenarios but to illustrate how the explained methodology shows promising results even without any special adaptation.

The first conclusion one can draw is obvious: bias correction

TABLE II
MAPE RESULTS FROM MODEL POLY IN DATASET PJM

Parameters		With Bias Correction		Without Bias Correction		
n	d	Bottom-Up	Top-Down	Bottom-Up	Top-Down	Regression
3	2	9.945%	9.944%	9.844%	9.844%	11.024%
	4	9.629%	9.632%	8.786%	8.786%	10.422%
	6	9.534%	9.533%	8.256%	8.256%	10.274%
	8	9.484%	9.484%	7.902%	7.902%	10.002%
5	2	8.415%	8.416%	9.551%	9.55%	9.962%
	4	8.086%	8.084%	8.505%	8.505%	9.47%
	6	7.962%	7.965%	7.926%	7.926%	9.322%
	8	7.852%	7.854%	**7.555**%	**7.555**%	9.023%
10	2	8.353%	8.349%	10.129%	10.129%	10.08%
	4	8.043%	8.043%	9.13%	9.13%	9.564%
	6	7.899%	7.899%	8.588%	8.588%	9.408%
	8	7.775%	7.773%	8.239%	8.239%	9.092%
30	2	8.993%	8.993%	11.224%	11.224%	9.878%
	4	8.689%	8.69%	10.33%	10.33%	9.361%
	6	8.571%	8.569%	9.953%	9.953%	9.253%
	8	8.443%	8.442%	9.644%	9.644%	8.951%

TABLE III
MAPE RESULTS FROM MODEL SVM IN DATASET PJM

Parameters		With Bias Correction		Without Bias Correction		
n	γ	Bottom-Up	Top-Down	Bottom-Up	Top-Down	Regression
3	0.00001	10.588%	10.483%	11.703%	11.622%	13.224%
	0.001	9.537%	9.521%	9.167%	9.151%	11.041%
	0.1	9.16%	9.128%	7.681%	7.621%	9.678%
	1	9.156%	9.114%	7.67%	7.608%	9.659%
5	0.00001	8.711%	8.707%	11.017%	10.98%	11.812%
	0.001	8.068%	8.062%	8.801%	8.772%	9.968%
	0.1	7.881%	7.836%	7.534%	7.457%	8.974%
	1	7.893%	7.836%	7.534%	**7.45**%	8.975%
10	0.00001	8.719%	8.709%	11.354%	11.34%	11.688%
	0.001	7.986%	7.979%	9.112%	9.119%	9.913%
	0.1	7.679%	7.66%	7.989%	7.985%	8.898%
	1	7.687%	7.666%	7.989%	7.983%	8.949%
30	0.00001	9.329%	9.319%	12.138%	12.12%	11.404%
	0.001	8.644%	8.641%	10.202%	10.199%	9.547%
	0.1	8.472%	8.452%	9.526%	9.512%	8.813%
	1	8.47%	8.452%	9.516%	9.511%	8.834%

TABLE IV
MAPE RESULTS FROM MODEL AR IN DATASET PJM

Parameters		With Bias Correction		Without Bias Correction		
n	l	Bottom-Up	Top-Down	Bottom-Up	Top-Down	Regression
3	0	9.262%	9.261%	7.643%	7.643%	9.832%
	1	9.189%	9.189%	7.558%	7.558%	10.042%
	exp	9.164%	9.166%	7.559%	7.559%	10.144%
	3	9.317%	9.315%	7.758%	7.758%	10.552%
5	0	7.672%	7.67%	7.289%	7.289%	8.912%
	1	7.883%	7.883%	7.275%	7.275%	9.347%
	exp	8.012%	8.012%	7.338%	7.338%	9.804%
	3	8.218%	8.218%	7.484%	7.484%	10.042%
10	0	7.639%	7.639%	7.967%	7.967%	8.959%
	1	7.441%	7.442%	7.432%	7.432%	9.004%
	exp	7.706%	7.707%	7.309%	7.309%	9.763%
	3	7.502%	7.498%	**7.216**%	**7.216**%	9.292%
30	0	8.328%	8.328%	9.415%	9.415%	8.791%
	1	8.062%	8.062%	8.544%	8.544%	8.889%
	exp	7.404%	7.407%	7.308%	7.308%	9.762%
	3	7.703%	7.7%	7.824%	7.824%	8.944%

TABLE V
MAPE RESULTS FROM MODEL POLY IN DATASET NYC

Parameters		With Bias Correction		Without Bias Correction		
n	d	Bottom-Up	Top-Down	Bottom-Up	Top-Down	Regression
3	2	9.459%	9.484%	9.407%	9.426%	9.985%
	4	9.086%	9.108%	8.128%	8.149%	9.189%
	6	9.033%	9.058%	7.688%	7.709%	9.16%
	8	8.975%	9.001%	7.376%	7.396%	8.79%
5	2	7.495%	7.523%	8.931%	8.949%	8.629%
	4	7.087%	7.117%	7.641%	7.66%	8.158%
	6	6.981%	7.011%	7.136%	7.155%	8.154%
	8	6.881%	6.912%	**6.789**%	6.808%	7.757%
10	2	7.521%	7.545%	9.201%	9.216%	8.662%
	4	7.103%	7.126%	7.942%	7.96%	8.223%
	6	6.986%	7.005%	7.46%	7.476%	8.191%
	8	6.869%	6.893%	7.109%	7.124%	7.774%
30	2	7.693%	7.704%	9.748%	9.757%	8.458%
	4	7.307%	7.32%	8.549%	8.561%	8.037%
	6	7.207%	7.218%	8.159%	8.169%	7.993%
	8	7.052%	7.061%	7.787%	7.796%	7.546%

TABLE VI
MAPE RESULTS FROM MODEL SVM IN DATASET NYC

Parameters		With Bias Correction		Without Bias Correction		
n	γ	Bottom-Up	Top-Down	Bottom-Up	Top-Down	Regression
3	0.00001	10.673%	10.56%	11.822%	11.755%	13.112%
	0.001	9.076%	9.014%	8.668%	8.632%	9.941%
	0.1	8.24%	8.239%	6.605%	6.593%	8%
	1	8.236%	8.24%	6.596%	6.587%	7.999%
5	0.00001	8.132%	8.137%	10.733%	10.729%	10.826%
	0.001	7.208%	7.204%	8.124%	8.113%	8.6%
	0.1	6.787%	6.766%	6.422%	6.383%	7.44%
	1	6.79%	6.768%	6.417%	**6.377**%	7.431%
10	0.00001	8.14%	8.167%	10.814%	10.818%	10.405%
	0.001	7.141%	7.138%	8.15%	8.141%	8.533%
	0.1	6.629%	6.636%	6.616%	6.614%	7.351%
	1	6.638%	6.638%	6.612%	6.609%	7.361%
30	0.00001	8.243%	8.261%	11.069%	11.075%	10.295%
	0.001	7.221%	7.252%	8.475%	8.498%	8.302%
	0.1	6.904%	6.928%	7.442%	7.472%	7.249%
	1	6.899%	6.929%	7.433%	7.465%	7.25%

TABLE VII
MAPE RESULTS FROM MODEL AR IN DATASET NYC

Parameters		With Bias Correction		Without Bias Correction		
n	l	Bottom-Up	Top-Down	Bottom-Up	Top-Down	Regression
3	0	8.74%	8.76%	6.866%	6.886%	8.441%
	1	8.69%	8.709%	6.907%	6.928%	8.706%
	exp	8.687%	8.706%	6.953%	6.974%	8.806%
	3	8.871%	8.891%	7.236%	7.257%	9.241%
5	0	6.69%	6.709%	**6.282**%	6.301%	7.47%
	1	7.067%	7.087%	6.475%	6.495%	8.008%
	exp	7.381%	7.403%	6.698%	6.718%	8.507%
	3	7.611%	7.632%	6.867%	6.888%	8.715%
10	0	6.68%	6.7%	6.622%	6.638%	7.497%
	1	6.65%	6.67%	6.379%	6.397%	7.617%
	exp	7.149%	7.168%	6.661%	6.681%	8.476%
	3	6.846%	6.865%	6.418%	6.438%	7.984%
30	0	6.895%	6.904%	7.406%	7.416%	7.317%
	1	6.733%	6.741%	6.86%	6.871%	7.374%
	exp	6.783%	6.802%	6.659%	6.679%	8.473%
	3	6.554%	6.566%	6.49%	6.505%	7.487%

generally does not improve the results of the algorithms since none of the tables' best results belongs to the bias-corrected experiments. Still, the difference between bias-corrected and non-corrected best results is usually tiny (around 0.3%). Therefore, bias-correcting does not seem to pay off.

Second, the bottom-up approach is slightly better than top-down but in most cases the performance shown was the same or very close (about less than 0.1%). Moreover, there is no rule of thumb to determine when top-down is better that bottom-up or *viceversa*. It depends on the dataset and the local predictor but varies irregularly. These results are aligned with these in the literature (see [44] and references therein).

Third, the best record was obtained by the Auto Regressive predictor using a top-down aggregation approach on the the NYC dataset. Worth to mention, in the majority of the cases performance improves with more training days (in future experiments we will explore the exact amount of training days that gives us the absolute maximum result for each local predictor). This tendency, however, does not apply for the AR, confirming also other researches [45], [46].

Fourth, all local forecast methods seem to converge and behave in a quite similar manner. This phenomenon happens due to the fact that we have used a **very long** period of time to analyse these techniques (10 years in NYC dataset and 3 years in the PJM). In future works we will explore data-aging techniques against this drawback. Anyway, it is clear that AR produces better forecasts than SVM and the latter, in turn, is an improvement over the polynomial method.

TABLE VIII
SUMMARY OF THE MAPE RESULTS.

Dataset	Operator	Expected	Our Best
NYC	2.87%	5.78%	6.28%
PJM	7.64%	7.45%	7.22%

Last but not least, regression does not outperform the rest of the aggregators: it works adequately when the other aggregators do, but usually a couple of points worse. This behaviour appears because the training error of the regressor does not compensate the reduction given by the non-linear pattern found between the local forecasts.

V. Conclusions

One of the minor revolutions arising from the smart grid vision will cause a major impact in short-term load forecasting: being able to remotely retrieve low-voltage data will allow applying forecasting techniques infeasible so far. Following the outstanding results of forecast model combination (a common banner identifying several post-process techniques that combine different models' predictions on a common variable), we have developed a new sort of model combination that will be possible to use in smart grids: combined aggregation.

With this novel approach, we have devised three methods to obtain the overall consumption prognosis by adding up forecasts on the compounding loads. These new models are namely bottom-up aggregation (with and without bias correction), top-down aggregation (with and without correction), and regression aggregation. Due to the impossibility of obtaining real data, we have designed an experiment to test a similar problem and have fed it with real TSO data.

The results obtained show that bias correction implies an unnecessary effort since it does not improve the performance. Moreover, the best local predictor, parameter set, and aggregation vary from one dataset to another but the results of the botton-up methods equal or even top the top-down method and, therefore, we can conclude that aggregative model combination is a useful technique in STLF in smart grids.

We have observed that in almost all cases the performance improved as we increased the amount of training days. Outperforming other forecasting results was not the goal of this work but further experiments will explore to this end the exact number of training days that gives us the absolute maximum result for each local predictor. Moreover, we will implement different methods for the regression aggregation, such as NN and Genetic Programming. We did test a fourth local predictor based on a NN but its results were very poor and it lasted too long in comparison with the rest of the predictors. Future works will also include tuning-up the NN to improve its results and executing it simultaneously on several computers to speed up this process.

Appendix

In this section we present how we have computed the estimation of the minimum MAPE. Suppose that the load curve $l(h)$ of an specific day type has the following expression:

$$l(h) := f(h) + \xi_h, \tag{1}$$

where f is an unknown function and ξ_h is a Gaussian random variable with mean 0 and variance σ_h^2. In our experiments we have measured (via a Gaussian Test) that this is a rational hypothesis, at least in the case of the Time Series model.

Any method that successfully forecasts the load curve l will have learned $f(h)$. We may estimate the min expected MAPE for that case as follows:

$$min := \mathrm{E}\left[\frac{100}{24}\sum_{i=1}^{24}\left|\frac{f(h)+\xi_h-f(h)}{f(h)+\xi_h}\right|\right] = \\ = \frac{100}{24}\sum_{h=1}^{24}\mathrm{E}\left[\left|\frac{\xi_h}{f(h)+\xi_h}\right|\right], \tag{2}$$

Up to this point we do not have any evidence on how to compute the exact value of this expected value. Note that this would be the best theoretical error we may achieve. Our next steps aim at giving a rude estimation on Equation (2). Suppose the following bound applies:

$$f(h) + \xi_h < \max(l). \tag{3}$$

Using the bound in Equation (3) in Equation (2) leads to:

$$min \geq \frac{100}{24}\frac{1}{\max(l)}\sum_{h=1}^{24}\mathrm{E}[|\xi_h|].$$

As $\mathrm{E}[|\xi_h|] = \sqrt{\frac{2}{\pi}}\sigma_h$ (see [47] for example) we have that:

$$min \geq \frac{100}{24}\sqrt{\frac{2}{\pi}}\frac{1}{\max(l)}\sum_{h=1}^{24}\sigma_h.$$

We may then estimate σ_h for instance by $\mathrm{Var}(l(h))$.

References

[1] H. Alfares and M. Nazeeruddin, "Electric load forecasting: literature survey and classification of methods," *International Journal of Systems Science*, vol. 33, no. 1, pp. 23–34, 2002.

[2] V. Hinojosa and A. Hoese, "Short-term load forecasting using fuzzy inductive reasoning and evolutionary algorithms," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 565–574, 2010.

[3] A. ul ASAR, S. R. ul HASSNAIN, and A. U. KHATTACK, "A multi-agent approach to short term load forecasting problem," *International Journal of Intelligent Control and Systems*, vol. 10, pp. 52–59, 2005.

[4] M. Hibon and T. Evgeniou, "To combine or not to combine: selecting among forecasts and their combinations," *International Journal of Forecasting*, vol. 21, no. 1, pp. 15–24, 2004.

[5] R. Clemen, "Combining forecasts: A review and annotated bibliography," *International Journal of Forecasting*, vol. 5, pp. 559–583, 1989.

[6] J. Scott-Armstrong, "Combining forecasts: The end of the beginning or the beginning of the end," *International Journal of Forecasting*, vol. 5, pp. 585–588, 1989.

[7] L. DeMenezes, D. Bunn, and J. Taylor, "Review of guidelines for the use of combined forecasts," *European Journal of Operational Research*, no. 120, pp. 10–204, 2000.

[8] R. Prudencio and T. Ludermir, "Using machine learning techniques to combine forecasting methods," in *In Lecture Notes in Artificial Intelligence*, 2004, pp. 1122–1127.

[9] K.-B. Song, Y.-S. Baek, and D.-H. Hong, "Short-term load forecasting for the holidays using fuzzy linear regression method," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 96–101, 2005.

[10] R. Prudencio and T. Ludermir, "A machine learning approach to define weights for linear combination of forecasts," in *In 16th International Conference on Artificial Neural Networks*, 2006, pp. 274–283.

[11] Y.-X. Jin and J. Su, "Similarity clustering and combination load forecasting techniques considering the meteorological factors," in *Proceedings of the 6th WSEAS International Conference on Instrumentation, Measurement, Circuits and Systems*. World Scientific and Engineering Academy and Society (WSEAS), 2007, pp. 115–119.

[12] I. Fernández, C. Borges, and Y. Penya, "Efficient building load forecasting," in *Proceedings of the* 16^{th} *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2011, pp. 1–8.

[13] C. Borges, Y. Penya, and I. Fernández, “Optimal combination of short-term load forecasting models in non-residential buildings,” in *2011 IEEE PES Innovative Smart Grid Technologies, ISGT Asia 2011, Nov. 13-16, Perth, Australia*, 2011, pp. 1–7.

[14] E. Feinberg and D. Genethliou, “Load forecasting,” in *Applied Mathematics for Power Systems, Chapter 12*, 2005, pp. 269–285.

[15] E. Kyriakides and M. Polycarpou, “Short term electric load forecasting a tutorial,” *Studies in Computational Intelligence (SCI)*, vol. 35, pp. 391–418, 2009.

[16] H. Yang and C. Huang, “A new short-term load forecasting approach using self-organizing fuzzy armax models,” *IEEE Transactions on Power Systems*, vol. 13, no. 1, pp. 217–225, 1998.

[17] W. Charytoniuk, M. Chen, and P. Van-Olinda, “Non parametric regression based short-term load forecasting,” *IEEE Transactions on Power Systems*, vol. 13, no. 3, pp. 725–730, 1998.

[18] M. Hagan and S. Behr, “The time series approach to short term load forecasting,” *IEEE Transactions on Power Systems*, vol. 2, no. 3, pp. 785–791, 1987.

[19] A. Jain and B. Satish, “Clustering based short term load forecasting using support vector machines,” in *Proceedings of the IEEE Bucharest PowerTech*, 2009, pp. 1–8.

[20] S. Lin, Z. Lee, S. Chen, and T. Tseng, “Parameter determination of support vector machine and feature selection using simulated annealing approach,” *Applied Soft Computing*, vol. 8, no. 4, pp. 1505–1512, 2008.

[21] P. González and J. Zamarreño, “Prediction of hourly energy consumption in buildings based on a feedback artificial neural network,” *Energy and Buildings*, vol. 37, no. 6, pp. 595–601, 2005.

[22] S. E. Papadakis, J. B. Theocharis, S. J. Kiartzis, and A. G. Bakirtzis, “A novel approach to short-term load forecasting using fuzzy neural networks,” *IEEE Transactions on Power Systems*, vol. 13, no. 2, pp. 480–492, 1998.

[23] R. Sadownik and E. P. Barbosa, “Short-term forecasting of industrial electricity consumption in brazil,” *International Journal of Forecasting*, vol. 18, no. 3, pp. 215–224, 1999.

[24] F. Glover, “Future paths for integer programming and links to artificial intelligence,” *Computers and Operations Research*, no. 5, pp. 533–549, 1986.

[25] C. Johnson, “A design framework for metaheuristics,” *Artificial Intelligence Review*, vol. 29, pp. 163–178, 2008.

[26] C. Lemke and B. Gabrys, “Meta-learning for time series forecasting and forecast combination,” *Neurocomputing*, vol. 73, no. 10-12, pp. 2006–2016, 2010.

[27] Z. Liuzhang, “Short-term electric load forecasting with combined data mining algorithm,” *Automation of Electric Power Systems*, 2006.

[28] C.-H. Wu, G.-H. Tzeng, and R.-H. Lin, “A novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression,” *Expert Syst. Appl.*, vol. 36, pp. 4725–4735, April 2009.

[29] G.-C. Liao, “Hybrid chaos search genetic algorithm and meta-heuristics method for short-term load forecasting,” *Electrical Engineering (Archiv fur Elektrotechnik)*, vol. 88, pp. 165–176, 2006.

[30] Z. Yang, X. Nie, W. Xu, and J. Guo, “An approach to spam detection by naive Bayes ensemble based on decision induction,” in *Proc. of the* 6^{th} *International Conference on Intelligent Systems Design and Applications (ISDA’06)*, 2006, pp. 861–866.

[31] G.-s. Hu, F.-f. Zhu, and Y.-z. Zhang, “Short-term load forecasting based on fuzzy c-mean clustering and weighted support vector machines,” in *Proceedings of the Third International Conference on Natural Computation - Volume 05*, ser. ICNC ’07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 654–659.

[32] Z. Ismail, F. Jamaluddin, and F. Jamaludin, “Time series regression model for forecasting malaysian electricity load demand,” *Asian Journal of Mathematical Statist*, no. 1, pp. 139–149, 2008.

[33] V. Ferreira and A. Pinto-Alves-da Silva, “Automatic kernel based models for short term load forecasting,” in 15_{th} *International Conference on Intelligent System Applications to Power Systems (ISAP)*, 2009, pp. 1–6.

[34] P. Cortez, M. Rocha, and J. Neves, “A meta-genetic algorithm for time series forecasting,” 2001.

[35] F. Collopy and J. S. Armstrong, “Rule-based forecasting: development and validation of an expert systems approach to combining time series extrapolations,” *Management Science*, vol. 10, pp. 1394–1414, 1992.

[36] K. Hwang, “A stlf expert system,” in *Proceedings of* 5^{th} *Russian-Korean IEEE International Symposium on Science and Technology*, 2001, pp. 112–116.

[37] X. Wang, K. Smith-Miles, and R. Hyndman, “Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series,” *Neurocomputing*, vol. 72, no. 10-12, pp. 2581 – 2594, 2009, lattice Computing and Natural Computing (JCIS 2007) / Neural Networks in Intelligent Systems Designn (ISDA 2007).

[38] M. M. Teresa, M. Teresa, P. Leo, J. T. Saraiva, J. Nuno, F. V. Mir, J. Lus, P. J. Peas, L. J. Rui, F. Jorge, and M. C. Pereira, “Meta-heuristics applied to power systems,” 2001.

[39] P. Chan, W.-C. Chen, W. Ng, and D. Yeung, “Multiple classifier system for short term load forecast of microgrid,” in *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on*, vol. 3, july 2011, pp. 1268 –1273.

[40] B. Dong, C. Cao, and S. Lee, “Applying support vector machines to predict building energy consumption in tropical region,” *Energy and Buildings*, vol. 37, no. 5, pp. 545–553, 2005.

[41] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines: and other kernel-based learning methods*, 1st ed. Cambridge University Press, 2000.

[42] *Day-Ahead Scheduling Manual*, New York Independent System Operator, 2001.

[43] G. Shmueli, “To explain or to predict?” *Statistical Science*, vol. 25, no. 3, pp. 289–310, 2011.

[44] R. J. Hyndman, R. A. Ahmed, G. Athanasopoulos, and H. L. Shang, “Optimal combination forecasts for hierarchical time series,” *Comput. Stat. Data Anal.*, vol. 55, no. 9, pp. 2579–2589, Sep. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.csda.2011.03.006

[45] Y. Penya, C. Borges, D. Agote, and I. Fernandez, “Short-term load forecasting in air-conditioned non-residential Buildings,” in *Proceedings of the* 20^{th} *IEEE International Symposium on Industrial Electronics (ISIE)*. IEEE, 2011, pp. 1359–1364.

[46] Y. Penya, C. Borges, and I. Fernández, “Short-term load forecasting in non-residential buildings,” in *Proceedings of the* 10^{th} *IEEE Region 8 Conference (AFRICON)*, 2011, pp. 1–6.

[47] J. Patel and C. Read, *Handbook of the normal distribution*, ser. Statistics, textbooks and monographs. Marcel Dekker, 1996.