# A Scalable Load Forecasting System for Low Voltage Grids

Marisa Reis, André Garcia, Ricardo J. Bessa
INESC TEC
Campus da FEUP, Rua Dr Roberto Frias, Porto 4200–465
marisa.m.reis@inesctec.pt, andre.f.garcia@inesctec.pt, ricardo.j.bessa@inesctec.pt

***Abstract*—A recent research trend is driven to increase the monitoring and control capabilities of low voltage networks. This paper describes a probabilistic forecasting methodology based on kernel density estimation and that makes use of distributed computing techniques to create a highly scalable forecasting system for LV networks. The results show that the proposed algorithm outperforms three benchmark models (one for point forecast and two for probabilistic forecasts) and demonstrate the applicability of the distributed in-memory computing solution for a practical operational scenario. The ultimate goal is to integrate information about net-load forecasts in power flow optimization frameworks for low voltage networks in order to solve technical constraints with the available home energy management system flexibility.**

***Index Terms*—Forecasting, smart grid, low voltage, probabilistic, scalability**

## I. Introduction

The installation of smart grid equipment, such as smart meters and data concentrators, contributes to increase the monitoring and control capabilities of low voltage (LV) grids. However, advanced computational solutions are needed to fully explore the information collected by the different devices.

A new paradigm is the preventive control of distribution grids, where a key input is information about net-load forecasts in each node of the distribution grid, which can be used to run (optimal) power flow algorithms in order to detect and solve potential technical problems in a predefined time horizon (e.g., 24 hours-ahead). This approach can be complementary to the real-time control and avoid expensive control actions (e.g., load or renewable energy curtailment). Information from forecasting algorithms is a key requirement for these management functions.

Forecasting the consumption of individual LV clients is challenging due to a high variability of the time series and a vast type of consumption profiles. For instance, Mirowski et al. [1] showed that the Mean Absolute Percentage Error (MAPE) can range between 40% and 90% for one LV consumer, in contrast to 8% for a LV feeder. In this problem, Hayes et al. [2] showed that a naive method (i.e., forecast equal to the load at the same time period in the previous equivalent day) can outperform linear and non-linear autoregressive models. Taieb et al [3] applied component-wise gradient boosting with P-splines to generate quantile forecasts for LV clients.

Compared to the state of the art, this paper makes the following contributions: (i) apply a weighted kernel density estimation (KDE) method that outperforms the naive model from [2] and generates conditional density forecasts; (ii) describes a distributed in-memory computing framework that ensures high scalability of the forecasting methodology and makes it suitable for operational deployment.

The rest of the paper is organized as follows: section II describes the forecasting methods used in this article, both point and probabilistic forecast. The approach for online parameter estimation and the computational solution is also described in this section. Section III introduces the evaluation metrics along with the results for the proposed solution. Section IV presents the conclusions.

## II. Forecasting methodology

### A. Conditional Kernel Density Estimation

Let $L_t$ be the the load during the hour $t$, and let $\hat{L}_{t+h|t}$ be the load forecast for the time horizon $t+h$ given the information available at hour $t$. This approach considers that the prediction can be made by a linear weighted combination of past observed values of the load. A weighted average can be written as

$$\hat{L}_{t+h} = \frac{\sum_{i=1}^{n} L_i \omega_{i,t+h}}{\sum_{i=1}^{n} \omega_{i,t+h}} \tag{1}$$

where the smoothing coefficients $\omega_{i,t+h}$, measure of similarity between the new measures and each correspondent historical measure, so the forecast is given by a weighted average of the load observations.

The only information required for this approach is a historical set of data pairs $(v_i, L_i), i = 1, , n$, with $L_i$ the corresponding normalized load power, and the vector $v_i$ the covariates that benefit the prediction task such as calendar variables (e.g. weekday, hour of the day) and observed power consumption (e.g. previous day $t-24$, previous week $t-168$).

This forecast methodology relies on the premise that if two situations are similar (analogous), the observations that occur shortly after them will be also similar. Therefore, the outcome of a similar past situation may be used as a prediction of what will happen after the current situation. Following this idea, there are three major steps: (1) how to measure similarity; (2) how to select the number of neighbors (or analogs) according to the computed similarities; (3) how to weight past historical values to generate the load forecast.

The similarity of each past situation to the current situation $v_{t+h}$ is evaluated by calculating a distance between the two

$d_{i,t+h} = d(v_i, v_{t+h})$, were $v_i$ is the historical vector at time $i$. In our experiments, absolute distance is used, but other distance functions may also be considered.

$$d_{i,t+h} = \frac{1}{K} \sum_{k=1}^{K} |v_{k,i} - v_{k,t+h}| \tag{2}$$

When $v_{t+h}$ is available, the distance between this vector and each historical vector $v_i, i = 1, , n$ is computed, resulting in a n-dimensional vector of distances $\mathbf{d}_{t+h} = (d_{1,t+h}, , d_{n,t+h})$. The next step is to apply a criterion to select a number $M$ of the total $n$ observations that are close to $v_{t+h}$.The parameter $M$ is determined based on a bandwidth $\delta_s$ that acts as a threshold. Distances above $\delta_s$ lead to null coefficients of $\omega_{i,t+h}$. The bandwidth $\delta_s$ can be computed in a several ways. Here, is computed from a percentage of the range of distances, as described in [4].

$$\delta_s = d_{min} + p_r(d_{max} - d_{min}) \tag{3}$$

where $d_{min}$ and $d_{max}$ are the minimum and the maximum of the distance. Other alternatives to the maximum distance could be chosen for computing $\delta_s$, like the median or the mean distance. Once selected historical data, a weighting strategy is applied. For this purpose, we consider the negative exponential function, defined by

$$\omega_m^d = exp(-\alpha d_m) \tag{4}$$

where $m = 1, ..., M$ is the *m*th historical data out of the $M$ selected based on the distance threshold given by Eq.3, and $d_m$ is the distance between $v_m$ and the new vector $v_{t+h}$. The weighting function is only applied to the $M$ selected historical observations. The value of the parameter $\alpha$ regulates how local the weighting is, so that the larger the exponent $\alpha$, more localized the model is. The weighting function must have a maximum value at zero distance, and its value must decay as distance grows. Additionally, these weights are multiplied by an age weighting function aimed to give more importance to recent observations that to old ones. The age-weighting function is given by

$$\omega_m^\tau = \lambda^{\tau_m} \tag{5}$$

where $\omega_m^\tau$ is the age-weighting coefficient, $\tau_m$ is the age in hours of the selected historical element m, and $\lambda$ is the forgetting factor with $0 < \lambda < 1$.

Lastly, the weights to be used in the local model are given by

$$\omega_m = \omega_m^\tau \omega_m^d \tag{6}$$

The kernel density estimation based on a non-parametric approach was employed for the probabilistic forecast, following the approach described in [5]. The kernel density estimation (KDE) is a well-known memory-based learning method that estimates an unknown density function by smoothing the observations. The density at a given point is computed by assigning to each sample point a weight representing the contribution to the density. Thus, given a random sample $Y_1, ..., Y_N$ of the random variable $Y$, the kernel density estimator can be written as

$$\hat{f}(y) = \frac{1}{N} \sum_{i=1}^{N} K_{h_u}(y - Y_i) \tag{7}$$

with $K_{h_u}(\cdot) = K(\cdot/h_u)/h_u$, where $K$ is a kernel function and $h_u$ is the smoothing parameter or bandwidth.

The first step is to choose the kernel, and in the literature, there are several possible kernels, namely, normal, biweight, beta, Epanechnikov, and logistic, among others. For this case, the beta kernel estimator is chosen, so the unknown density function for values within the range $[0, 1]$ can be estimated by,

$$\hat{f}(y) = n^{-1} \sum_{i=1}^{n} K_{y/h+1,(1-y)/h+1}(Y_i) \tag{8}$$

which is similar to the standard kernel estimator, but with beta kernels [6]. The choice of the beta kernel is given by the fact that, the estimators proposed in [6] produces estimates that are free of boundary bias and are always non-negative. In [7] a boundary correction approach is used to deal with this.

The estimation of the predictive density, using also the weighting coefficients with time decay, is

$$\hat{f}(y) = \frac{\sum_{t=1}^{n} w_i K_{y/h+1,(1-y)/h+1}(Y_i)}{\sum_{t=1}^{n} w_i} \tag{9}$$

The density function at the value $y$ is constructed by applying kernel density estimation to the set of observations $\{Y_1, Y_2, \ldots, Y_n\}$ with each $Y_t$ value weighted in accordance with the closeness of the corresponded $v_t$. Then, the resulting density estimation is used for producing probabilistic forecasts, represented as a set of quantiles. The quantiles are obtained by means of the cumulative distribution function. So, let $Y$ be a real valued random variable with cumulative distribution function $F_Y(y) = P(Y \leq y) = \int_{\infty}^{y} f_Y(t)dt$, the $\tau-$quantile, issued at time $t$ for lead time $t+h$ can be computed as follows:

$$\hat{q}_{t,h}^{\tau} = F_Y^{-1}(\tau) \tag{10}$$

where $\tau \in [0, 1]$. To obtain the cumulative distribution function from the estimated density function in Eq.9 two more steps are added. Firstly, the probability distribution function (*pdf*) is obtained by properly normalizing the estimated density function, so that, the sum of its events (integral) sums up to one. Secondly, the cumulative distribution is obtained using numerical integration through the estimated *pdf*. This way, once the cumulative distribution function is estimated, performing the quantiles computation is straightforward and computationally cheap. Altogether, this procedure produces a valid cumulative distribution function at horizon $t$, so the following monotonicity property is satisfied

$$\hat{q}_{t,h}^{\tau_1} \leq \hat{q}_{t,h}^{\tau_2} \quad \forall \tau_1, \tau_2 \quad \text{such that} \quad \tau_1 \leq \tau_2 \tag{11}$$

This is of special importance, since *quantile crossing* is avoided and no additional corrections are needed to be implemented to dealt with this.

### B. Parameter Estimation

The parameter estimation of the KDE method is time consuming and a batch (or offline) estimation is unfeasible if thousands of time series need to be forecasted. In contrast to cross validation approaches, where the overall possible parameter combinations are tested, a dynamic continuou optimization ensures a continuous tracking of the optimum (which might be changing with time), in a so-called dynamic simplex (or Nelder-Mead) algorithm [8]. This is ideal for large volumes of streaming data, such as this case, where highly dimensional power observations are constantly added as new observations arrive.

Recalling the parameters to be optimized, we have $p_r$ (percentage of points to be used in the weighting step) and $\alpha$ (smoothing coefficient of the weighting function). The dynamic simplex solution is employed (see [8] for deepe details). So the optimization is done using few evaluation as possible. And by evaluations is meant testing the algorithm with different parameters. The criterion for evaluating which pairs of parameters are better is given by an objective function, in this case the mean absolute error is considered.

A dynamic optimum search is performed using a simplex structure. A simplex $S$ is defined as a convex hull with N+1 vertices $\{x_j\}_{j=1}^{N+1}$ in an N-dimensional space $\mathbb{R}^N$, where $N$ is given by the number of parameters to be estimated. In this case $N = 2$, so a triangle is used a simplex. Given a initial pair of parameters, the iterative process starts by defining an initial simplex (three pairs of parameters) and the objective function is computed for each of these pairs. The worst point of the simplex is chosen for reflection, this is, the method expands the simplex in the opposite direction. Figure 1 illustrates the dynamic simplex procedure for one LV forecast, the starting point (blue point) is defined, from that, a number of successive reflections are considered, searching the best portion of the space in which to move.

### C. Computational Solution

In order to sustain a high demand for processing multiple LV forecasts, a solution based in parallel computing is adopted to improve performance and scalability. For this purpose we choose Gearman[1], for being simple and a reliable proven software in distributing tasks to multiple processes and computers. This last feature allows horizontal scaling (adding more computers to the layer) in case of a growing demand for processing power. To illustrate the relation between time complexity and the number of LV forecasts processed, we simulate 3 different scenarios aimed to reproduce a production environment, where the algorithm updates its coefficients and a forecast is generated for the next 24 hours. The scenarios are as follows:

1) Synchronous execution (single thread) to process a batch of LV forecasts;
2) Asynchronous execution (multiple thread) using 10 processes and one computer;

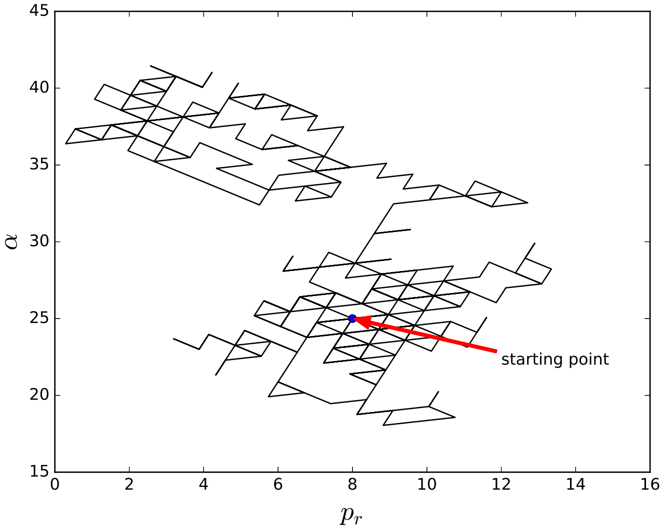

Fig. 1. Tracking parameter path using dynamic simplex algorithm, with initial pair of parameters $(p_r = 8.0, \alpha = 25.0)$

3) Asynchronous execution (multiple thread) using 30 processes distributed by two computers.

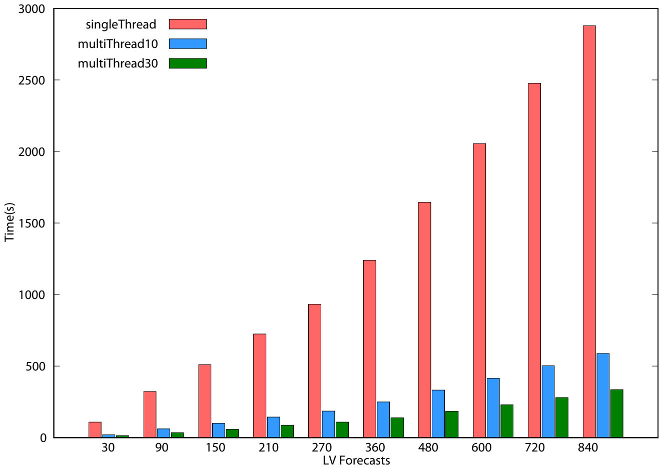

Fig. 2. Gearman run time simulation using single and multiple thread processes.

The simulation results illustrated in Figure 2 show a clear difference between single thread and multiple thread execution times, where the multiple threads executions significantly exceed in performance the single thread ones. It also reveals a linear progression between the time to finish the execution and the number of LV clients to forecast. By calculating the results, we realize that the usage of the multiple thread technology improves the time required to complete the same tasks by at least 80% compared to the traditional single thread method. The simulation also demonstrates that better results can be achieved by increasing the number of available processes to execute the LV forecast, which offers scalability for the future in case of higher number of LV to forecast.

[1] gearman.org

Finally, one limitation of the conditional KDE algorithm is the need to calculate the distances and search for analogs in a batch of historical data, which access must be very fast. In order to overcome this limitation, the memcached[2] framework is used since it is a simple, highly scalable key-based cache that stores data and objects wherever dedicated or spare RAM is available for quick access by applications, without going through layers of parsing or disk I/O. The main goal is to have fast access speed to information, in contrast to loading the information each time from disk.

## III. Evaluation

### A. Setup

The focus is on day-ahead point and probabilistic forecasts using smart meter data (active power) from 103 individual LV clients in Spain, with hourly temporal resolution. One year of historical smart meter data from 2015 is used. The first 9 months contains 6528 observations as historical data and it is used for training, while the renaming 3 months are used to simulate and test the online forecast scenario. In the experiments, $p_r = 0.08 * 100$ and $\alpha = 25$ were used as initial starting vertex, and the step size $s_i = 1, i = 1, \dots, k$. By the choice of the same step size for all the span parameters, the algorithm produces an equilateral simplex. Note that the parameter $p_r$ has been re-scaled so it has the same scale than $\alpha$, since different scales with the same step size result in different changes in response. For the generation of the probabilistic forecast at each horizon, we compute the $\tau$-quantiles from the cumulative distribution function given by Eq.10 for a set of probabilities $\tau_i = \{0.01, ..., 0.99\}$ with $i = 1, ..., Q = 99$. Our method will be denoted by DynKDE.

### B. Benchmark methods

A naive approach (denoted by Naive) is used as benchmark, where the load consumption forecast for instance $t$ is given by the observed power consumption from the previous day $t - 24$. This approach is the same one reported in [2] to outperform linear and non-linear autoregressive models.

For the probabilistic approach two other benchmark methods were considered:

1) Unconditional quantiles (denoted by UnCond): We compute the $\tau$-quantile of the distribution of the active all historical observations.
2) Quantiles conditioned to the time-of-day (denoted by PeriodOfDay). This method segments the data into 24 sub-datasets, and different quantiles are computed for each sub-dataset.

Both methods have been considered before in the context of load forecasting[3]. Due to the similarity of these two methods with the proposed method we also include them in the analysis. The UnCond benchmark method does not take into account any kind of seasonality, while the PeriodOfDay allows the demand distribution to change for each period of day. In the DynKDE the period of the day and recent demand effects are included with a kernel density estimation strategy.

### C. Evaluation

Despite the popularity of the *mean absolute percentage error* MAPE usage in the load forecasting context, this error metric can be misleading when applied to smart meter data. In particular, when load are close to zero, MAPE values become very large, regardless of the actual absolute errors. The use of scaled errors as a robust alternative to percentage errors when comparing forecast accuracies across series on different scales is endorsed. Therefore, in order to evaluate the deterministic forecast, the Mean Absolute Scaled Error (MASE) [9] is used:

$$\text{MASE} = \frac{\text{MAE}}{\text{MAE}_{naive}} \tag{12}$$

where $MAE$ is the mean absolute error produced by the actual forecast; while $MAE_{naive}$ is the mean absolute error produced by the naive forecast. $MASE > 1$ implies that the actual forecast does worse than the benchmark.

We also consider *Better* as a relative measure for reliability. This metric reflects a frequency of predictions by a proposed model that is better than a baseline model.

$$\textit{Better} = \frac{1}{n} \sum_{i=1}^{n} I(|p_i - o_i| \leq |b_i - o_i|) \tag{13}$$

where $o_i$, $p_i$ and $b_i$ are the observed, model predicted and baseline prediction for interval $i$, respectively.

For probabilistic (density) forecast evaluation, the CRPS (Continuous Ranking Probability Score), as it both measures the calibration and the sharpness (see [10]), is used:

$$\text{CRPS}(\hat{F}_{t,h}, y_{t+h}) = \int_{-\infty}^{\infty} (\hat{F}_{t,h} - \mathbb{1}(z \geq y_{t+h}))^2 dz \tag{14}$$

where $\mathbb{1}$ is the indicator function, $\hat{F}_{t,h}$ the predicted cumulative distribution function and $y_{t+h}$ the actual observation at horizon $h$.

Given a test set of size $N$, it is possible to compute the average CRPS for an $h$-step-ahead forecast as:

$$\text{CRPS}(h) = \frac{1}{N} \sum_{t=1}^{N} \text{CRPS}(\hat{F}_{t,h}, y_{t+h}) \tag{15}$$

## IV. Results

Figure 3 shows an example of probabilistic forecast generated for the active power consumption by a LV client. The evaluation results are outlined in Fig. 4 and Fig. 5. To promote the interpretation of the results, we made the forecast horizon equivalent to the period of the day, in order to be possible to relate the error measurements with the intra-daily consumption pattern.

The UnCond method forecasts a flat density shape across all time horizons (top panel of Fig. 3), this is due to the fact that the method does not condition to recent demand or takes into account hourly seasonality of the data. The PeriodOfDay improves the shape of the density forecast (middle panel of Fig. 3) in relation to UnCond. The DynKDE exhibits better calibration with sharper forecasted densities surrounding the observations.

[2] memcached.org

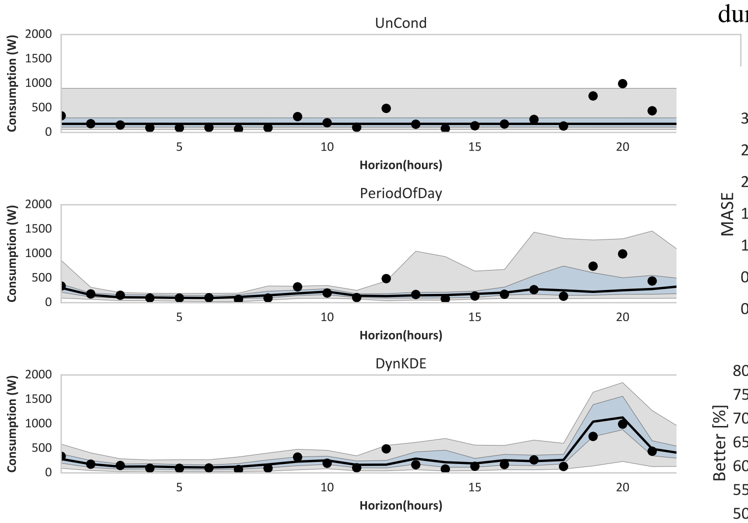

Fig. 3. One-day ahead load forecast example. The blue and gray region show 50% and 90% forecast intervals respectively. The black points serve as the real values and the median quantile forecast is the black line.

The MASE results shows a direct comparison between the deterministic solution versus the naive approach (top panel Fig.3). In general, the adopted solution outperforms the naive approach. This is more noticeable in hours of higher variability on consumption, where the the most part of the error distribution is bellow 1 (red dashed line). This result shows that an analog based method can outperform naive models and improve the forecast skill, which contrasts with conclusions from the state of the art.

The bottom panel in Fig 4 shows the reliability of the proposed approach over the benchmark methods presented in Section III-B. The DYNKDE is compared with the NAIVE, and the median (50% quantile) of the UNCOND and PERIODOFDAY methods. The lower improvement of the proposed approach can be found when compared with the baseline NAIVE method, coherent with the findings reported in [2]. Forecasting procedures that are not carefully calibrated fail to pass this “naive test” surprisingly often.

The probabilistic evaluation is based on the CRPS. In order to compare the CRPS values across all LV series, the consumption values were standardized to lie between 0 and 1. As expected, the CRPS is higher during the periods of the day where large consumption values occur. Being a generalization of the MAE error it is highly dependent on the range of observed values.

Fig. 5 depicts the CRPS average values obtained with the probabilistic benchmark methods (dashed lines) and the DYNKDE. The UNCOND has worst performance. By conditioning on period of the day the error decreases [3]. The DYNKDE has the best performance as it exhibits lower CRPS values. The results also reveals that, for a shorter forecast horizon (or equivalently with the beginning period of the day), PERIODOFDAY and DYNKDE have similar errors. But, for higher consumption and consequently higher variability, as in during the day, the proposed method is superior.

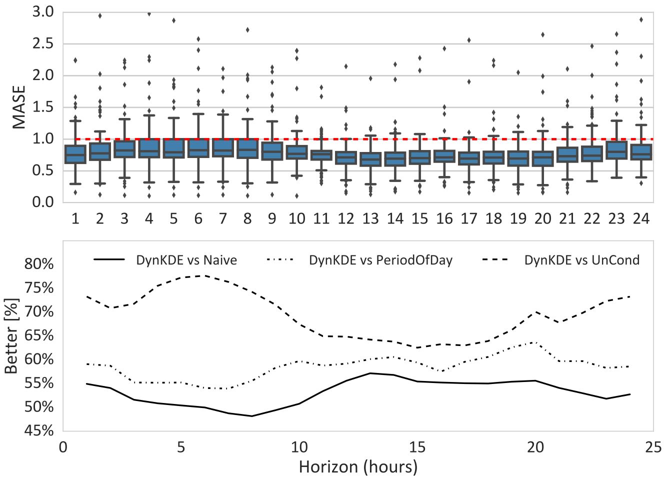

Fig. 4. Deterministic performance results for 24 hours-ahead forecasts. The top panel are the MASE outcomes using the metric defined in Eq.12. It compares the DYNKDE and NAIVE forecasts. The bottom panel compares the deterministic approach with respect to the metric *Better* defined in Eq. 13.

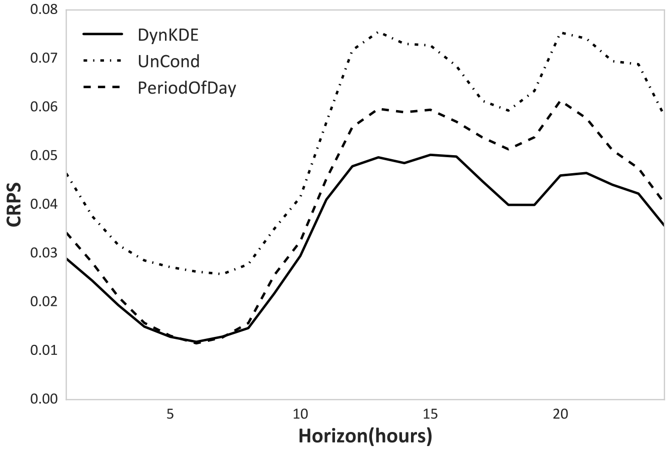

Fig. 5. CRPS mean results for 24 hours-ahead probabilistic forecasts.

## V. CONCLUSION

This papers proposed a method to obtain the forecast density estimation using a non-parametric approach that basically searches for analogs in the historical data.

A dynamic simplex algorithm was modified and employed for the parameter estimation, allowing continuous parameter tuning with few parsimonious function evaluations. The proposed algorithm is implemented in a distributed and scalable computational solution.

The proposed method was compared with three benchmark models: naive method where the forecast is equal to the observed consumption in the previous day; unconditional quantiles computed from the historical data; quantiles conditioned to the time-of-day. Different metrics were considered to evaluate the point and probabilistic forecast performance of the method.

The results showed that the proposed method was able to outperform the benchmark models in both point and probabilistic forecast. The forecasted conditional densities showed a superior calibration are more sharper than the other models, leading to a lower value of the continuous ranking probability score.

The simulation study also demonstrated the applicability of the distributed in-memory computing solution for a practical operational scenario. For instance, in a dataset of 840 LV clients the computational time required to generate a forecast for the next 24 hours is 336 seconds (considering 30 threads from 2 computers), a reasonable result that ensures scalability for a grow demand in number of clients (or smart meters).

## Acknowledgments

The research leading to this work is being carried out as a part of the UPGRID project (*Real proven solutions to enable active demand and distributed generation flexible integration, through a fully controllable LOW Voltage and medium voltage distribution grid*). UPGRID project has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under grant agreement No. 646531. The authors acknowledge IBERDROLA for providing the smart meter data.

## References

[1] P. Mirowski, S. Chen, M. Hill, T. K. Ho, and C.-N. Yu, "Demand forecasting in smart grids," *Bell Labs Technical Journal*, vol. 18, no. 4, pp. 135–158, March 2014.

[2] B. Hayes, J. Gruber, and M. Prodanovic, "Short-term load forecasting at the local level using smart meter data," in *2015 IEEE Eindhoven PowerTech*, 2015.

[3] S. B. Taieb, R. Huser, R. J. Hyndman, and M. G. Genton, "Forecasting uncertainty in electricity smart meter data by boosting additive quantile regression," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2448–2455, September 2016.

[4] M. G. Lobo and I. Sanchez, "Regional wind power forecasting based on smoothing techniques, with application to the spanish peninsular system," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 1990–1997, Nov 2012.

[5] M. G. Lobo, "Métodos de predicción de la generación agragada de energía eólica," Ph.D. dissertation, Universidad Carlos III de Madrid, 2010.

[6] S. X. Chen, "Beta kernel estimators for density functions," *Computational Statistics and Data Analysis*, vol. 31, no. 2, pp. 131 – 145, 1999.

[7] J. W. Arora, S.and Taylor, "Forecasting electricity smart meter data using conditional kernel density estimation," *Omega*, vol. 59, Part A, pp. 47 – 59, 2016.

[8] Q. Xiong and A. Jutan, "Continuous optimization using a dynamic simplex method," *Chemical Engineering Science*, vol. 58, no. 16, pp. 3817–3828, 2003.

[9] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679 – 688, 2006.

[10] T. Gneiting, F. Balabdaoui, and A. E. Raftery, "Probabilistic forecasts, calibration and sharpness," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 2, pp. 243 – 268, 2007.