

Predictive management of low-voltage grids

Marisa Reis¹, André Garcia¹, Ricardo Bessa¹ ✉, Luís Seca¹,
Clara Gouveia¹, Jorge Moreira², Pedro Nunes², Pedro G. Matos²,
Fernando Carvalho³, Pedro Carvalho³

¹INESC TEC, Porto, Portugal

²EDP Distribuição, Lisbon, Portugal

³AmberTREE, Lisbon, Portugal

✉ E-mail: ricardo.j.bessa@inesctec.pt

Abstract: This study presents the functional model that provides net-load forecasts for each low-voltage (LV) node (including PV generation and self-consumption), developed for the UPGRID (real proven solutions to enable active demand and distributed generation flexible integration, through a fully controllable low-voltage and medium-voltage distribution grid) framework project. Several tests scenarios were simulated and the results regarding forecast accuracy and computational performance are given. Results demonstrate the applicability of the distribution in memory solution in a practical operational scenario, offering a highly scalable forecasting system for LV networks. Based on forecasts and available real-time information, an architecture for preventive control of LV grids is built upon chronological analysis capabilities of DPlan. An illustration on how such capabilities are used in the context of the foreseen UPGRID preventive control framework is provided.

1 Introduction

The installation of smart grid equipment, such as smart meters (SM) and data concentrators, contributes to increase the monitoring and control capabilities of low-voltage (LV) grids. However, advanced software functions 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 is used to run power flows to detect potential technical problems in the pre-defined time horizon (e.g. 24 h ahead). When technical problems are detected, a set of control rules and/or automatic optimisation algorithms can be applied to derive preventive control actions that mitigate the foreseen technical problems. This approach can be complementary to the real-time control and avoid expensive control actions such as load or renewable energy curtailment.

The UPGRID project developed this framework to be tested in the Portuguese demo [1].

2 Net-load forecasting tool

The net-load and generation forecasting tool, architecture depicted in Fig. 1, generates forecasts for each LV node [including photovoltaic (PV) generation and self-consumption] and secondary MV/LV substation for the next hours and days.

The main features of this component are as follows: (i) scalability – parallel computing techniques combined with appropriate data structures and statistical algorithms; (ii) information about forecast uncertainty – probabilistic forecasts for the load that particularly relevant for the LV level.

The core modules of this component are as follows:

- *KDE forecast model* – statistical method that combines conditional kernel density estimation with locally learning methods to produces point and probabilistic forecasts (represented by probability density functions and/or a set of quantiles) for a time

horizon constrained by the time horizon of the weather prediction data. More details can be found in [2].

- *In-memory and distributed computing tasks* – combination of Gearman (gearman.org) to distribute tasks to multiple processes/computers and Memcached (memcached.org) to store time series data in cache for quick access by KDE, without going through layers of parsing or disk I/O.

The statistical forecast method is non-parametric and essentially operates considering the historical data for analogue past situations, combining and weighting them based on the new measurements [2, 3]. When establishing a similarity criterion (a distance function), only a percentage (p_r) of the total historical data are used for the density estimation.

A weighting function with parameter α is applied, regulating how local the model is. These parameters – p_r and α – are included and benefit the forecasting accuracy. A dynamic simplex algorithm [4] was modified to fit these parameters, taking into consideration the online architecture of the load forecasting tool – it ensures the continuous tracking of the optimum (which might be changing with time).

The parameters are estimated separately for each of the LV clients, making it suitable for distributed calling of the algorithm. Once new measurements arrive, the coefficients are updated, with several parsimonious function evaluations. This is because the dynamic simplex algorithm takes a greedy path to the optimal solution. At the end, the coefficients are stored, ready to be called once the forecasting schedule is triggered.

Once it is triggered, the application sends the workload to a Gearman-Job-Server node, responsible for distributing the tasks to available registered workers that will perform the task independently of each other. In this way, it is possible to scale the problem as it grows to multiple machines, adding more workers to the process making it a good solution for big volumes of LV clients and data. Besides being able to allow work to be done in parallel, Gearman is also a framework designed for load balancing and peer-to-peer communication, essential for stability and availability of the system.

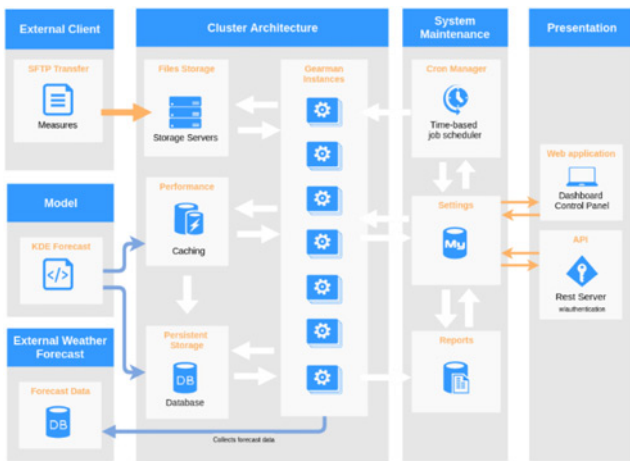


Fig. 1 Detailed architecture of the net-load and generation forecasting module

A simple example of the communication between client, Gearman worker and Gearman server can be summed to the following:

- *client to server*: ask server to perform a task. The server acknowledges the request and assigns an identification to the request;
- *server to all workers*: tell workers registered for the task that there is work to be performed;
- *worker to server*: I will perform the task you just told us about;
- *server to worker*: ok, go ahead, here is the information about the task;
- *worker to server*: here is the result of the task performed;
- *server to client*: here's the result of the task you asked me to get someone to do for you.

3 Case study description

This section describes the SM dataset used in this validation phase, as well as the validation environment of the component. The main goal of this validation phase was to assess the computational performance of the component and the forecasting accuracy, considering the key performance indicators (KPI).

The dataset used to evaluate the component's performance has a historical data load consumption of 255 LV clients, collected by the smart metering infrastructure from the Portuguese demonstrator. The load measurements are from 2016 with 15 min resolution, covering the months May, June, August and September. After aggregation of the data into hourly measurements, the training set is used to feed the statistical model, while the test set is used to evaluate the model's performance. We have divided May, June and August for training and September for testing.

The validation occurs offline and includes forecast accuracy and data management reliability indicators. The metrics – mean absolute scaled error (MASE) [5] and continuous ranked probability score (CRPS) [6] – are used as reference to define performance indicators related to forecasting in a smart grid context.

To illustrate the complexity of processing multiple LV forecasts and the advantage of using a distributed system to sustain the computational effort, we also simulate an environment with 1020 unique clients, each one containing the power consumption information from 1 May to 31 August 2016 in an hourly interval. This information is persistently stored in Cassandra database and cached in memory through Memcached for quicker access by the client application.

The validation process respects the following pattern:

- i A query to Cassandra database selects all the available LV clients.
- ii For each LV client, a request is issued to Gearman-Job-Server to execute 'KDE coefficients update' function for that specific client. Gearman queues each task until receives an instruction to run all the tasks.

(iii) iii Once instructed to run the tasks, Gearman issues a command to all the workers registered over the cluster to perform the work. The application enters in an asynchronous execution where processes operates independently from other processes:

- (a) the workers assigned to this task will update the KDE coefficients and generate a new forecast for each requested client. The results are stored in Cassandra database and in the file system;
- (b) program finishes execution. The execution time is saved for benchmark and systems evaluation.

4 Simulation results

In this section, the results obtained in the simulations will be resumed and commented, highlighting the overall performance and the obtained KPI results.

4.1 Forecasting skill evaluation

The MASE results in Fig. 2 (top plot) provide a direct comparison between the deterministic solution versus the naive approach, where the overall errors (MAE) are compared. Values below one (red line) indicate better performance of the forecast model. In general, the adopted solution outperforms the naive approach, more clear for the time of the day where exists more consumption variability.

In Fig. 2 (bottom plot), the probabilistic version of the algorithm is evaluated based on the CRPS. To compare the CRPS values across all LV series, the consumption values were standardised to lie between 0 and 1. The probabilistic evaluation shows that the quantile prediction has an error of ~4%. As expected, the CRPS is higher during the periods of the day where large consumption values occur.

4.2 Computational performance evaluation

To illustrate the relation between time complexity and the number of LV forecasts processed, we simulate different scenarios aimed to reproduce a production/operational environment, where the algorithm updates its coefficients and a forecast is generated for the next 24 h. This procedure also evaluates the scalability of the proposed solution.

The scenarios are defined as synchronous or asynchronous executions, varying in the number of workers (parallel processes) from 2, 4, 8, 16 or 32 workers. The simulation results given in Fig. 3 show a clear difference between synchronous (1 worker) and asynchronous execution times, where the multiple workers executions significantly exceed in performance the single process.

By examining the results (best to worse), we realise that for each 255 batch of clients the application takes near 0.7 min to compute the results in the best scenario using 32 workers from 2 computers. In the

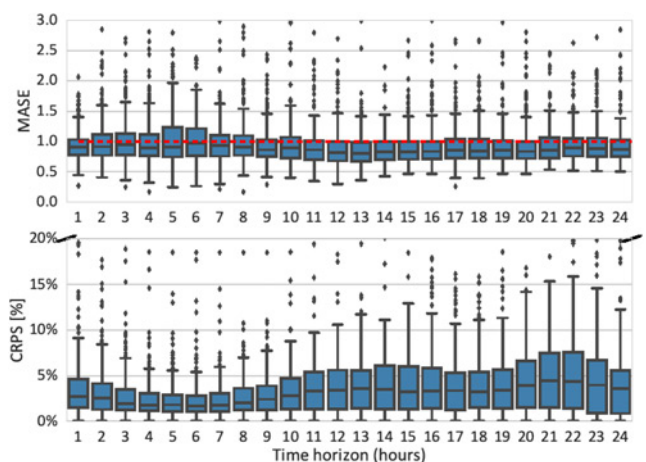


Fig. 2 CRPS distribution results for all LV clients for each time horizon

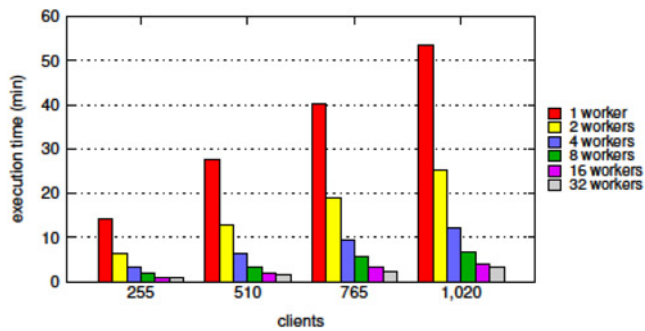


Fig. 3 Computational performance results

worst scenario using only one worker, the synchronous execution takes up to 14 min to process the same task, revealing a linear progression between the time to finish the execution and the number of LV clients to forecast. The simulation also demonstrates that better results can be achieved by increasing the number of available workers to execute the same task, which offers scalability for the future in case of higher number of LV clients to forecast. This result shows that the usage of the multiple thread and distributed architecture, improves the time required to complete the same tasks by 55–95%, compared to the traditional single thread method.

It also reveals a linear progression between the time to finish the execution and the number of LV clients to forecast. The simulation even demonstrates that better results can be achieved by increasing the number of available workers to execute the same task, which offers scalability for the future in case of higher number of LV clients to forecast.

From the simulation results in Fig. 3, it is also possible to model the relationship between the number of LV clients and the number of workers required to execute a higher number of tasks. For that purpose, the execution time was modelled as a $1/(\alpha \cdot x)$ function, where α is the parameter to be estimated and x the number of workers.

Fig. 4 depicts the scalability projection results with each line representing the estimation for a set of LV clients, making it possible to relate the estimated complexity between time and computational effort at larger scales. Considering 250 000 clients (green line) from our function, we can speculate that it would be necessary at least 150 workers to keep the execution time below 100 min, meaning that for a normal workstation it would represent somewhere between 9 and 10 computers (considering 16 workers for each machine) required to reach the number of 150 workers. The number could easily drop to a lower cluster depending on the machine characteristics.

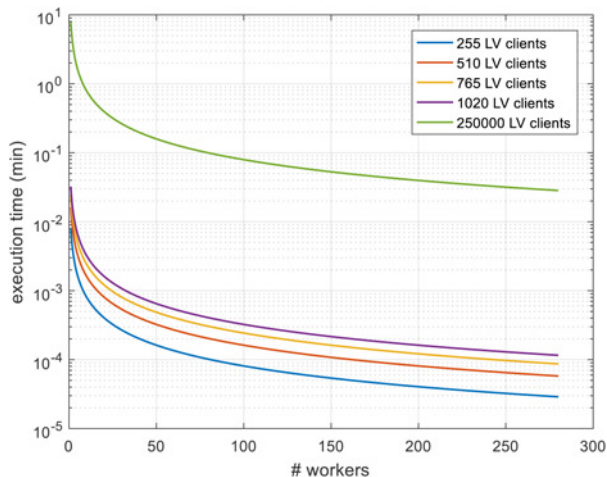


Fig. 4 Horizontal scalability projections

5 Predictive power flow

Preventive control of LV distribution grids requires net-load forecast information on each node a few hours ahead. With such information, chronological power-flow analysis can be used to provide a new level of awareness to the LV dispatch operators, while allowing a proactive management of potential grid problems. See Fig. 4 where load-forecast results are used to anticipate under-voltage and cable congestion problems.

To build such awareness, net-load forecast results are sent on a periodical basis to a DPlan server, in which a virtual client automatically analyses future impacts of such loads for an updated grid topology. When such impacts are expected to be severe, DPlan notifies the dispatch operators by triggering pre-defined alarms. After being notified, operators use DPlan to support grid mitigation actions such as reconfiguration, transformer tap changes, demand response or load/generation curtailment actions [7]. The (optimum) reconfiguration action found by DPlan to solve the problems identified in Fig. 5 eliminates the congestion and leads to the changes in the voltage profile illustrated in Fig. 6. The switching operations are illustrated in Fig. 7.



Fig. 5 Predictive power flow results: red coloured lines identify overloaded feeder sections (over 110% of their rated capacity) and red coloured circles identify bus voltages below acceptable levels (below 92% of the nominal value)

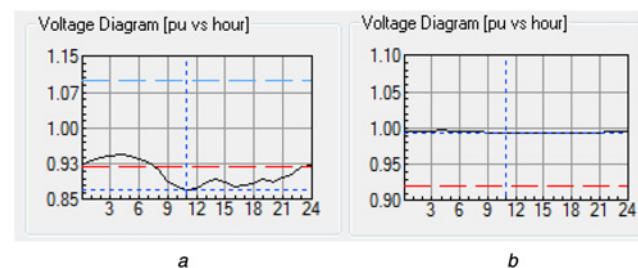


Fig. 6 Voltage profile at the grid worst-voltage customer (a) Before optimal predictive control reconfiguration, (b) After optimal predictive control reconfiguration

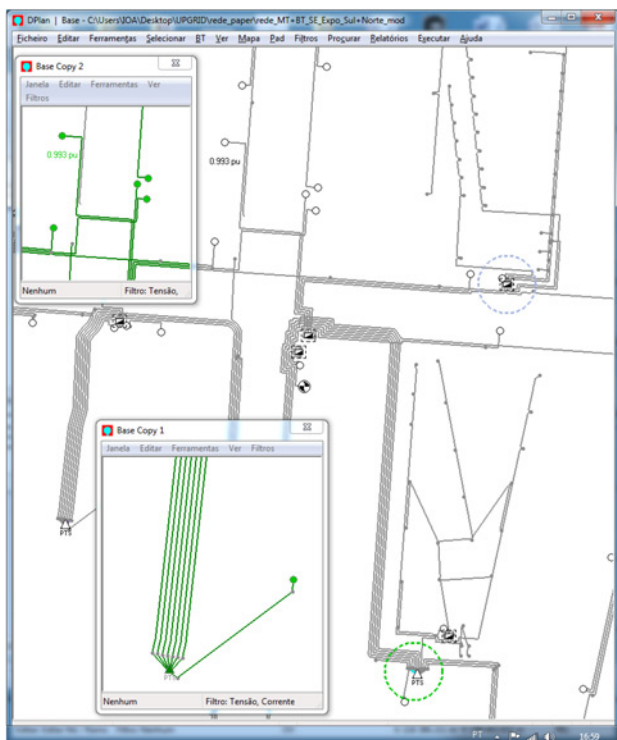


Fig. 7 Predictive power flow results after optimum reconfiguration. The dashed-line circles on the RHS of the picture identify the switching sites: blue circle for the switching OFF and the green circle for the switching ON operation

Control actions such as reconfiguration should be planned ahead but need to be confirmed for necessity and sufficiency before being undertaken in the field. Such confirmation requires near real-time information on grid topology and loading state.

6 Near real-time support

Near real-time analysis requires access to SMs up-to-date information. Considering possible constraints in obtaining updated SMs data, DPlan estimates SM net-loads based on near real-time transformer measurements gathered at data concentrators. Load estimation tools take data initially assigned to SMs (forecast, historical or typical profiles) and adjust such load data iteratively so that the results of the power-flow match the near real-time measurements at the sites of data concentrators (typically, transformer sites).

Once adjusted, SM loads can be used to run power flow and validate the planned ahead control actions against the newly updated loading situation. If such actions become unnecessary under the new conditions, the operator discards the actions. If they keep being necessary but become insufficient, then the operator uses DPlan to simulate the network under the new conditions to find new optimal actions to mitigate the new problems identified [8].

For illustration of one insufficient control action, take the reconfiguration example used before. The planned ahead switching action, temporarily transfers some load from one secondary substation to another (the receiving-load substation is identified by the green circle in Fig. 6). Such load transfer is seen as adequate beforehand as the load-receiving transformer is expected to operate at 90% of its rated capacity after the switching action (Fig. 8a). However, when DPlan estimates the SM loads based on the secondary substations near real-time metering data and simulate the planned switching action, then the load of the receiving-load substation rises (2%) above the transformer rated capacity

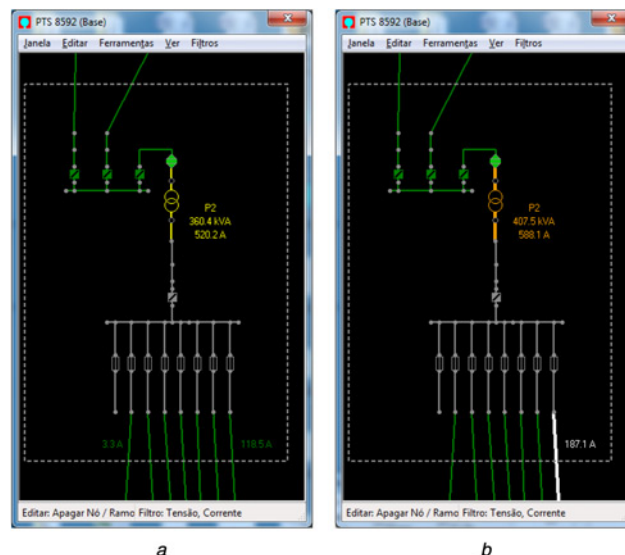


Fig. 8 Comparison between planned loading and near real time loading of the load receiving transformer after reconfiguration action

(Fig. 8b). That requires (i) further analysis to decide if the overload is acceptable or not and, if not, (ii) further utilisation of DPlan to support alternative grid mitigation actions to solve the problem in real-time.

7 Conclusions

This paper described a net-load forecasting algorithm for LV grids, whose main goal was to ensure high scalability. The proposed solution is simple and combines a NoSQL database with a job-server node responsible for distributing the tasks. The developed statistical model, based on analogues search and kernel density estimation, outperforms the naive approach (i.e. net-load profile equal to the previous day). The usage of the multiple thread and distributed architecture improves the time required to complete the same tasks by at least 80% compared to the traditional single thread method.

With forecasted information, chronological power-flow analysis can be used to provide a new level of awareness to the LV dispatch operators (generate alarms), while allowing a proactive management of potential grid problems, such as reconfiguration, transformer tap changes, demand response or load/generation curtailment actions. In real-time, SM measurements can be used to run power flow and validate the planned control actions against the newly updated loading situation.

8 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), funded by the European Union's Horizon 2020 Framework Programme under grant agreement No. 646.531.

9 References

- 1 EU Project UPGRID, website: upgrid.eu
- 2 Reis, M., Garcia, A., Bessa, R. J.: 'A scalable load forecasting system for low voltage grids', Proc. of the IEEE PES Powertech 2017, Manchester, UK, 2017
- 3 Lobo, M.G., Sanchez, I.: 'Regional wind power forecasting based on smoothing techniques, with application to the Spanish peninsular system', *IEEE Trans. Power Syst.*, 2012, 27, (4), pp. 1990–1997

- 4 Xiong, Q., Jutan, A.: 'Continuous optimization using a dynamic simplex method', *Chem. Eng. Sci.*, 2003, **58**, (16), pp. 3817–3828
- 5 Hyndman, R. J., Koehler, A. B.: 'Another look at measures of forecast accuracy', *Int. J. Forecast.*, 2006, **22**, pp. 679–688
- 6 Gneiting, T., Balabdaoui, F., Raftery, A. E.: 'Probabilistic forecasts, calibration and sharpness', *J. R. Stat. Soc.*, 2007, **69**, (2), pp. 243–268
- 7 Carvalho, P.M., Ferreira, L.M., Almeida, B., *et al.*: 'Improved demand controllability by grid reconfiguration for congestion management'. Proc. of the 18th Power Systems Computation Conf. (PSCC'14), Poland, 2014
- 8 Kostalova, A., Carvalho, P.M.: 'Towards self-healing in distribution networks operation: bipartite graph modelling for automated switching', *Electr. Power Syst. Res.*, 2011, **81**, (1), pp. 51–56