Impact Assessment of Smart Meter Grouping on the Accuracy of Forecasting Algorithms

Dejan Ilić
SAP Research
Karlsruhe, Germany
dejan.ilic@sap.com

Stamatis Karnouskos
SAP Research
Karlsruhe, Germany
stamatis.karnouskos@sap.com

Per Goncalves da Silva
SAP Research
Karlsruhe, Germany
per.goncalves.da.silva@sap.com

Malte Jacobi
SAP Research
Karlsruhe, Germany
malte.jacobi@sap.com

ABSTRACT

The increased penetration of smart meters generates huge amounts of fine-grained data, which may empower a new generation of energy related applications and services. Significant research efforts focus on the usage of such data to mainly improve the business processes of the electrical grid operators and provide some value added services to the end-users. Forecasting has a prominent position as it is a crucial planning step, and is mostly used to predict the grid load through highly-aggregated data. However, with the dramatic increase on fine-grained data, new challenges arise as forecasting can now also be done on much shorter and detailed time-series data, which might provide new insights for future applications and services. For the smart grid era, being able to segment customers on highly predictable groups or identify highly volatile ones, is a key business advantage as more targeted offers can be made. This work focuses on the analysis and impact assessment of in the context of smart metering data aggregation. A system to measure the impact of aggregation is designed and its performance is assessed. We experiment with measuring of the forecast accuracy on various levels of individual load aggregation, and investigate the identification of highly predictable groups.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision support*; G.3 [**Probability and Statistics**]: Time series analysis; I.1.2 [**Algorithms**]: Analysis of algorithms

Keywords

smart metering, smart grid systems, meter reading analysis, forecast accuracy

1. INTRODUCTION

Electricity injected into the grid needs to be constantly balanced with the demand. An enabler towards achieving this goal is prediction of future demand, which today is done by aggregating large numbers of consumers and attempting to predict their future consumption. The incurred costs of these forecasting errors, e.g. balancing costs, are passed on to the consumers via the electricity bill, thus shared homogeneously among all customers. With the introduction of the Advanced Metering Infrastructure (AMI), better accuracy can be achieved as well as timely assessment of the end-user's production and/or consumption (prosumption). With this improved communication between the stakeholders, there are many efforts in the direction of Demand Side Management (DSM) to further assist towards energy balancing, especially when we consider highly distributed systems. However, is not always possible to eliminate or significantly limit the prediction error [1], resulting in a certain level of grid imbalance. This is a business opportunity for service providers that can help, e.g. the Distribution System Operator (DSO), with its grid balancing efforts. However, this is still a high-cost activity, and especially in the envisioned smart grid distributed systems, a more challenging issue to tackle.

The stochastic behaviour of some customers that can't be relatively accurately predicted, may result in some cases to extremely high balancing costs. As we drill down to smaller groups of customers or even individuals, the more difficult forecasting becomes when no additional information is provided. The prediction error varies for different load types, for instance some appliances are highly predictable while others have completely stochastic behaviour. The newly deployed AMI may assist in this direction by providing a better insight on the individuals in the context of both timing and quality of information [10]. Smart metering data allows new dimensions in analysis e.g. predictability level assessment for any customer, or a group of them. It is important to understand that the focus does not necessarily have to be on customers as such, but virtually any grouping (based on certain criteria e.g. location, economic, social etc.) of devices or users being connected to the grid [7].

In the Smart Grid era, devices may be consumers or producers of electrical energy (or both i.e. prosumer device). In any case, if the predictability of a device is within some limits, it is considered "good-enough", and one may cluster it as part of a virtual "predictable" group. The same holds true for the prosumers owning these devices. Having predictable groups is important especially for the construction of prosumer Virtual Power Plants (pVPP) [7], where their usefulness is directly bound to understanding their potential contribution to the grid and the ability to control it. If highly predictable, or highly volatile, groups are identifiable (preferably in real-time), new business opportunities for smart-grid stakeholders may arise. As an example, one might try to "normalize" the effects of volatile groups by combining them appropriately, or by prioritizing the deployment of demand-response programmes.

2. BUSINESS ASPECTS

Given the high unpredictability of individual customers, any forecasting algorithm would struggle to consistently meet high-precision goals. This may have a significant impact on the vision of the smart grid where individual prosumers buy and sell energy based on their own predicted needs [6]. The fear would be that any prediction error might result in false trades and financial penalties e.g. due to imbalance caused. The production side solves a similar problem through the creation of Virtual Power Plants (VPPs), where the aggregation of Distributed Energy Resources (DERs) into the virtual equivalent of a large power station [12]. Such a coalition allows them to potentially participate more cost-efficiently on an electricity market. This concept was further refined [7] for virtual communities at several layers e.g. per user, device etc. In any case constructing highly predictable groups or identifying stochastic prosumers (within pVPPs) is expected to result in new business opportunities for smart-grid stakeholders [8] and enable them to take better decisions.

Participating in a coalition is in the best interest of its members [2]. Therefore, due to the potential for cost reduction of participating in energy markets, highly predictable customers may join communities and participate on local markets by buying and selling energy. Such groups can only benefit most their users if they, as a total, have a highly predictable behaviour upon which strategies for trading can be build. That way users belonging to a highly-predictable group may benefit from lower (overall) prices – rather than always paying for individual imbalance costs. For instance, if highly predictable groups can be created, their low prediction errors may result in lower balancing costs than a retailer's (mostly static) service fees (as depicted in the Figure 1).

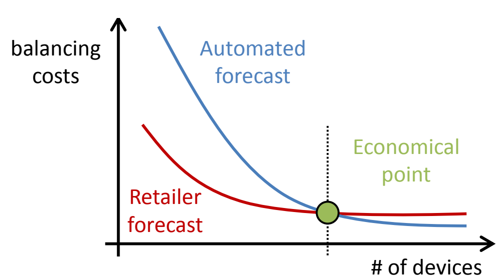

Figure 1: A predictable group over perform retailer's service costs

Additional to the algorithmic forecasting, it is expected that external factors, e.g. storage technologies, can assist towards improving the prediction accuracy by, for instance, adjusting the locally incurred imbalances to the expected behaviour. One can expect that this may be done on a building (or neighbourhood) level, which may assist towards more predictable behaviours. However, this combined effort is not further considered here and is left as future angle to explore. Finally, the existence of a highly predictable group is a key aspect to enable fully automatized brokerage agents that act on the behalf of such a group in local energy markets [6]. In such scenarios, the earliest the "economical point" (as depicted in Figure 1) is achieved, the better the benefits for the end-users might be.

The process of identifying highly predictable customers will also result in identifying the customers with a stochastic behaviour. Similarly to the highly predictable groups, the identification of highly stochastic customers and groups is also important. Once these are identified several actions can be realised e.g. try to move them towards a more predictable behaviour by deploying demand-response programmes. This is of key importance for an energy provider that is rolling out such schemes and wants to prioritize the target groups and his roadmap. Additionally, it may attempt to offer demand side management programmes or even penalize the users via more expensive tariffs. In this case, analytics can assist towards user segmentation not only on their location but on their actual behaviour as well.

3. SMART METER GROUPING

A traditional retailer's business and internal cost benefit analyses rely on the existence of large customer numbers, where individual effects are absorbed by the overall group behaviour. This is true due the mathematical behaviour of time series aggregation, especially if aggregated time series hold similar patterns. A smart meter is denoted with $m \in M$, where M is the set of N smart meters. If i is an interval (e.g. 15 minutes), actual consumption of a m inside an interval i is denoted as $y_i^m \geq 0$. The forecast energy load for the same interval is denoted as $\widehat{y}_i^m \geq 0$. The energy difference between forecast and actual consumption is calculated as $\widehat{y}_i^m - y_i^m$, having surplus if positive or shortage if negative.

Two types of aggregation are possible: One the one hand, if the prediction is calculated before the aggregation step, a perfect fit for aggregation of two meters m_w and m_z extracted from the set M, where $w \neq z$, if $\widehat{y}_i^{m_w} - y_i^{m_w} + \widehat{y}_i^{m_z} - y_i^{m_z} = 0$ or having no prediction error. A perfect example would be aggregating $\sin(t)$ and $\sin(t+\pi)$. On the other hand, energy of any meter y_i^m may be aggregated with any other $m \in M$ for each interval i. The resulting series can be further used for the calculation of the aggregated prediction. This step produces a subset denoted as $G \subseteq M$ of size $n \leq N$, where n represents the number of meters in the subset. The aggregation of any G for one instance results is denoted as $y_i^G = \sum_{m \in G} y_i^m$, while the prediction $\widehat{y}_i^G$ is calculated from the aggregated series $y^G = \{y_0^G, y_1^G, \ldots, y_x^G, \ldots, y_{x+s-1}^G\}$, where the series length is denoted as s. The resulting prediction within the time frame is denoted as $\widehat{y}_X^G$, where $X = \{x, \ldots, x+s-1\}$, same as for real data to compare y_X^G.

3.1 Applied Approach

The approach used in this work can be characterized as some kind of brute-force method; the computational cycles are used to build random groups, create forecasts for these groups and measure the resulting forecast accuracy. The steps in the grouping approach rely on random numbers. The Monte Carlo method is used to build a group of randomly chosen smart meters from the original set. The probability, independently of a group size, must be equally distributed in order to ensure comparability between all group sizes. Thus, all time series have the same probability to be chosen for a group.

For every experiment the series length s is fixed f.g. in this work is $s = 96$ and represents one day in 15 minutes intervals. Still, every smart meter $m \in M$ contains time series y and is indexed as m_j, where $j \in [1, N]$. Once the size $n \leq N$ of the subset $G \subseteq M$ is determined, G gets populated by random smart meters drawn of m_j out of M, without replacement. Then $m \in G$ are aggregated in y^G, which is used for calculating $\widehat{y}_X^G$. Finally (one or more) accuracy comparison measurements between the two time series, y_X^G and $\widehat{y}_X^G$, are stored as result of the experiment.

3.2 Forecasting Algorithms

Energy load forecasting is influenced by several factors, the most fundamental of which is the length of the prediction horizon. The focus of this work is forecasting the next day load, categorized as the short-term forecast [3, 13] as its horizon is between one and seven days. Besides the forecast horizon, methods can be additionally categorized by considering seasonality. We have used smart meter energy readings, and selected the time series forecasting methods as they use only historical data of a variable for prediction [4]. The approach is to reveal the internal structure (e.g. seasonality, trend) by using statistical properties of the time series. Due to their robustness and implementation simplicity, time series forecasting methods are popular in short-term load forecasting. The most commonly used approaches are auto-regression or exponential smoothing models [4].

For this work, the exponential smoothing forecasting method was chosen mainly for its robustness e.g. the method of Holt-Winters (HW). Exponential smoothing shows good forecast performance in empirical studies and outperforms more complex methods [15]. In order to compare experiments of forecast models, a naïve forecast method was used. Since energy load data is highly seasonal data, the Seasonal Naïve (SN) algorithm was chosen. The principle behind the SN method is the usage of values from the previous season (e.g. day, week) as forecast value for the current season [4]. For example, the forecast value for Monday is equal to the last observed value for Monday.

3.3 Accuracy Measurements

Using historical values (that were used to build the forecast model), historical data must be used and compared against the predicted values. This is important to evaluate the forecasting accuracy [4]. Therefore, the available historical observations are split into a training- and a test-set. The training-set (before y_x^G) is used to fit the forecast model, while the test-set (y_X^G) is used for comparison against the predicted values $\widehat{y}_X^G$. As forecasts of different scales must be compared, the Mean Absolute Percentage Error (MAPE) is chosen due its scale-independence. MAPE estimates the fit of a model by expressing its accuracy as a percentage, the advantage of which is that it is not fixed to a specific unit. Therefore, arbitrary models can be compared regardless of the unit of their values or their level. The MAPE is calculated as the sum of the absolute errors, normalized by the actual value [5] i.e.:

$$MAPE(G) = \frac{100\%}{s} \sum_{i \in X} \frac{|\widehat{y}_i^G - y_i^G|}{|y_i^G|}$$

for the number of intervals within the season s. The major disadvantage of this error metric is that the MAPE has no upper bound, as there is only a lower bound, which is zero. Due to this missing upper bound, extremely high values for certain time series distort the comparability of the MAPE. Especially for the case of a small denominator y_i, the MAPE tends to infinity.

4. SYSTEM DESIGN

4.1 System Requirements

Designing a system to deal with the envisioned processing may suffer from data exchange and data fetch latency (due to the large amount of data). In addition it must offer flexibility regarding the data sources, forecast algorithms and measurement components. The overall system performance is required to be in strong correlation with the data set size. For small number of smart meters (N), a high performance may not be of key importance. However, in real-world scenarios, demand for high performance is inevitable. Such scenarios tend to involve thousands (or millions) of smart meters in M [9]. Of course the frequency of meter readings $y^m, \forall\ m \in M$, is also a significant factor. Following the approach presented in the section 3.1, one can imagine how the computational requirements grow based on large value of N; The greater the N the longer it will take to fetch and aggregate meter readings of $G \subseteq M$. It is therefore of paramount importance to have efficient data acquisition mechanisms in order to retrieve all the data into the system and to write results back.

Other requirements we considered focused on engineering aspects e.g. building an extensible system that will enable future enhancements (extensions). For that purpose two key interfaces are defined: Firstly, an abstraction is made on the used forecasting methods, as is expected to have new algorithms in future, and secondly another abstraction is made on the measurement algorithms. As a result one can implement and plug-in other algorithms, extending existing ones or realizing new functionality. As an example, one may easily implement an accuracy metric to calculate the total cost of the prediction error. This can be done by calculating the energy error (in Wh) $\forall\ i \in X$ and multiplying with the contracted balancing prices (per Wh, $\forall\ i$) to calculate costs.

4.2 System Architecture

Based on the aforementioned requirements, an architecture is derived consisting of the components depicted in Figure 2 in Fundamental Modeling Concepts notation (`www.fmc-modeling.org`). The user interacts with the comparison application which includes functionalities such as time series comparison, accuracy metrics and group building. The metering data as well as the results are maintained in a DBMS (Database Management System). The main component is `TimeSeriesComparison` and it communicates (time-series) and coordinates all the other components.

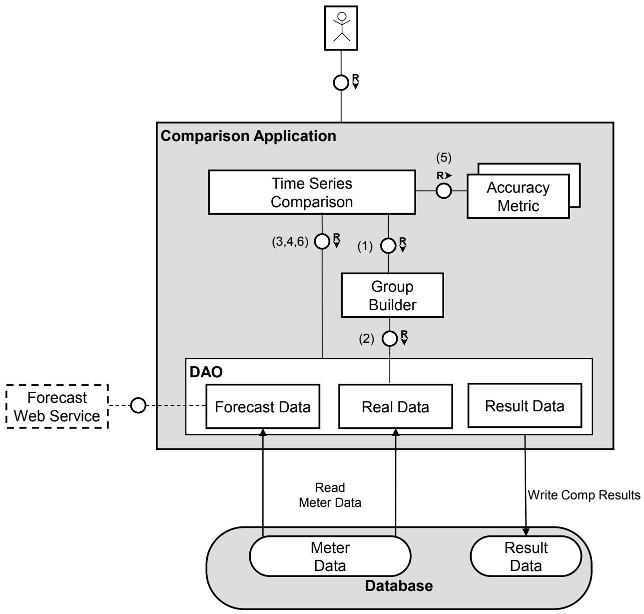

Figure 2: Components of the system architecture

The requirement concerning the flexibility of different data sources is addressed by using a designated layer for data reading/writing processes. As shown in Figure 2, the data layer consists of several Data Access Objects (DAO) components for data provision and storing. Three different DAO interfaces were designed i.e.: `RealDataDAO`, `ForecastDataDAO` and `ResultDataDAO`. The `RealDataDAO` interface is responsible for providing the real smart meter (aggregated) measurements to compare against the forecast. These forecasts, which are the results of forecasting algorithms over real metering data, are provided via the `ForecastDataDAO`. Its key purpose is to aggregate all the real metering data of a G and executes prediction algorithm in order to retrieve $\widehat{y}_X^G$ for entire X. Once time-series are available, their comparison calculation occurs within the component `AccuracyMetric` and its result (e.g. MAPE) is written over the `ResultDataDAO` interface. Every result contains the application configuration, the group constellation, the used forecast algorithm, the results of the accuracy metrics.

4.3 System Performance

Key performance aspects of the system were measured at selected measurement points that were chosen only for possible bottlenecks that could impact thus execution time for fetching the real and forecast data. While raw real data is directly fetched (and aggregated) from the storage, for prediction the *R environment* in combination with the `forecast` package was used. The package provides various implementations of forecasting methods, mainly from the domain of time series forecasting, out of which `HoltWinters` and `snaive` were used. For the random number generator the Mersenne-Twister algorithm was used [11]. This generator is used for drawing n smart meters from M for group G. Since there is no implementation of the Mersenne-Twister in standard *Java* libraries, the open source library *Colt* (`acs.lbl.gov/software/colt/`) was used.

One would expect to have bottlenecks on the data exchange, but all the processing (aggregation) of the real data and forecast data is made internally in the components. This results in exchanging only one final time-series of aggregated meter readings for a specific group. As all time-series aggregation is done inside the components, the data exchanged in between the components are y_X^G and $\widehat{y}_X^G$. Thus, size data exchanged is independently of the group size n, but only on X.

For the evaluation of the system performance, the HW algorithm was used. Since the SN algorithm uses previous day as prediction, it is expected to have twice the execution time for real data fetching. Its execution time resulted in linear growth in relation to group size n and it can be approximated as $0.02 * 2n$ seconds per one group. The HW forecast of one day $s = 96$ for any G require four weeks $(4 * 7 * s)$ of historical data (for every $m \in G$) to predict one day ahead (s). Fetching time for prediction data also resulted in linear growth with dependence on n. However, it is important to notice that HW algorithm requires fetching of an aggregated training-set (4 weeks) which is followed by execution of the actual forecast algorithm. Total execution can be approximated as $0.12 * n$, resulting in total $(0.02 + 0.12) * n$ seconds per one group. Due this limitation, the groups for this evaluation were limited to a group size of $n \leq 160$.

The MySQL implementation shows a linear growth in execution times in correlation to the size of a group. Although all measured time components increase linearly with increasing group sizes, the slope of the individual components differs. While the *Real Data* component reveals a small slope, the comparison one shows a higher slope. As an example, the average of ≈ 15 seconds is needed to perform a single comparison run for $n = 140$. As a result, one can use such a linear trend to determine approximated execution time of the complete experiment i.e. 100 comparison runs for $n = 150$ will take ≈ 30 minutes for the HW algorithm. Finally, this numbers are fully dependent on the data set size and is expected to grow if total number of meters or meter readings grow.

5. EVALUATION

The evaluation experiments reveal the grouping impact on the forecasting accuracy, and how the group accuracy depends on the accuracy of its individuals. We have used real-world data from the NOBEL (`www.ict-nobel.eu`) project which runs a trial with Spanish consumers. This original data set is filtered in order to acquire a high number of smart meters without any invalid measurements. The resulting set had $N = 1974$ smart meters without missing, or faulty, meter readings from 03 March 2011 to 09 June 2011 (98 days in total). As metering data was collected within the project trial, it was discovered that set had 2.8% of the 15 minute resolution and 97.2% had the 1 hour resolution of the metering data of $1kWh$ precision. In order to keep unique resolution, devices with resolution of 1 hour were linearly interpolated to 15 minutes.

The process described in the section 3.1 is repeated using this set. In the case of the HW method, the chosen seasonality was within-week seasonality. Using a weekly season achieved best forecast accuracy in preliminary experiments, which is also reported in [14]. For the SN forecast, a within-day seasonality was used, which means that the observations of the last day are the predicted values for the next day. This configuration depicted a superior forecast accuracy in preliminary experiments. Finally both algorithms use historical data to predict a specific date i.e. Tue 07 June 2011.

5.1 Grouping Impact on Accuracy

Grouping hides the stochastic behaviours and their impact. However, today with the fine-grained smart metering data offered, we can make detailed analysis on the impact of such stochastic behaviours on the overall accuracy. To understand their affect, an experiment was executed 100 times for every n in the spectrum $n = [1, 180]$, which was split into 4 sub-intervals. The first subinterval was $n = [1, 25]$ by an incremental step of 1 (25 groups steps in total); the second $n = (25, 50]$ by a step of 5; the third $n = (50, 100]$ by a step of 10; and the fourth $n = (100, 180]$ by a step of 20. Figure 3 shows the result of the experiment where the average MAPE (y-axis) per group size (x-axis) is shown for both HW and SN algorithms.

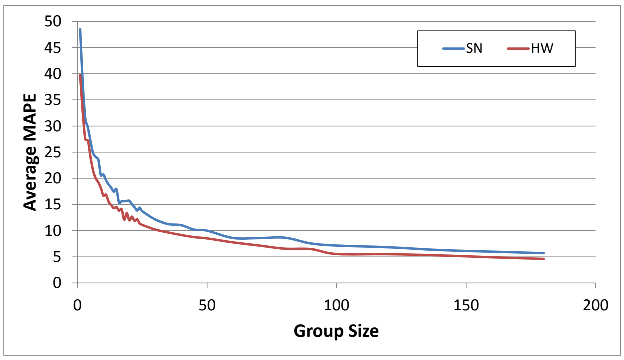

Figure 3: Grouping effect on the prediction accuracy (MAPE)

As expected the forecast accuracy increased with an increasing group size, showing bigger improvement for smaller group sizes while showing lower improvement for larger group sizes. Interestingly, the results also revealed that the simple forecast method (SN) performs almost identically to the more complex one (i.e. HW). However HW depicted a slightly better accuracy for all group sizes, having $\langle MAPE(G^{160})\rangle_M <$ 5% already at the group size of $n = 160$. Further experiments conducted revealed that a lower variance of series $y^G \forall G$ is the reason of the accuracy improvement rate. The same experiment within the winter season, where variance of the meter readings is higher, depicted slightly lower improvement rate. This experiment resulted in $\langle MAPE(G^{160})\rangle_M \approx$ 8% for the HW method. Still, both experiments depicted a similar trend in the increasing improvement rate.

Achieved results triggered further investigation for the competitiveness of the two selected algorithms. In order to validate the resulting behaviour, additional experiments were conducted. As shown in Figure 3, it was decided to fix n for comparison within a rolling-time window (other days of the week) to cross validate. Group size of $n = 50$ was chosen as greater n resulted in slighter accuracy improvement. Thus, one-day ahead was predicted 100 times (and therefore with different G) for every day of the week. The results for Tuesday 07 June 2011 are depicted in the Figure 3; however now a rolling-time window allow us to move along the rest of the weekdays, the results of which are shown in Figure 4.

The results depicted in Figure 4 show that the HW algorithm performed better for all days, as its $\langle MAPE(G^{50})\rangle_M$

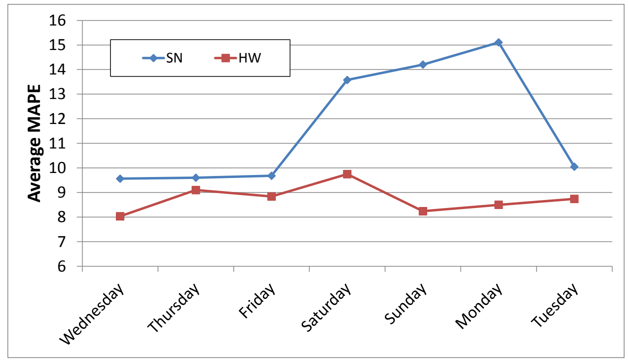

Figure 4: Example where SN algorithm fails to predict next day ($n = 50$)

range varies in between range of $8-10\%$ for all X. Although SN was comparable to HW from Tuesday to Friday, its forecast accuracy degraded for Saturday, Sunday and Monday. Such behaviour was expected due the fact that energy data actually contains two seasons, daily and weekly. Since SN predicts day-ahead (e.g. future 96 instances of 15 minutes) using data of one day-before (i.e. previous 96 instances), one can expect that customers (residential or commercial) behave differently on Saturdays in comparison to Fridays. As an example, load characteristics of a commercial customers usually change drastically over non-working weekend days.

The results of this experiment leads to the conclusion, that the forecast accuracy improvement by grouping is not a random effect. However, it is remarkable that the SN algorithm performed almost as good as HW for all the other weekdays (for this data set). Since the HW method depicted greater accuracy it was selected for identifying the key accuracy indicators.

5.2 Key Accuracy Indicators

We have shown that the forecast accuracy improvement by grouping is not a random effect. It was shown that the SN algorithm failed to predict correctly for Saturday, Sunday and Monday, even when they were aggregated within a group of 50. Obviously many devices within such group in 100 runs resulted in higher MAPE for Saturday than for Tuesday. However, one can generally expect that grouping impact is improved if every individual has a good prediction (or lower $MAPE(G^1)$) on its own. In other words, grouping two predictable smart meters will result into a lower MAPE than two unpredictable ones.

To confirm this assumption, an experiment is conducted where the MAPE for every smart meter is calculated individually using 4-weeks of historical data to predict Tuesday (07 June 2011). Figure 5 shows the cumulative density function of the HW method in dependency of the resulting MAPE values.

The median of this data set resulted to a MAPE of 36.06% and is assumed to be an indicator for creation of groups with greater and smaller forecast accuracy. If the hypothesis holds true, one would be able to create "good" and "bad" groups from the individual prediction accuracy. Using the median value from the figure 5, the time series with MAPE lower than 36.06% were considered as devices in the "good" predictability set ($A \in M$), while the remaining were con-

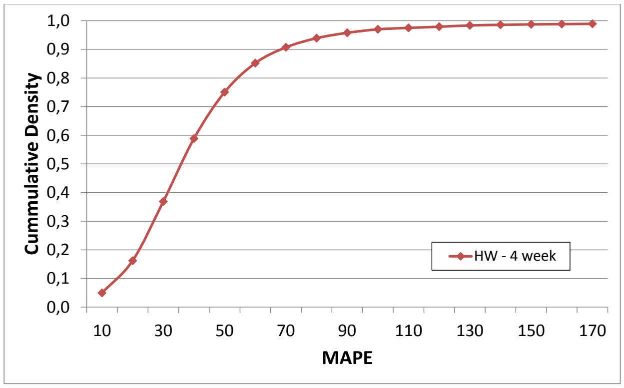

Figure 5: Cumulative density function built from every device in the set

sidered as devices for the "bad" predictability set ($B \in M$). To confirm the hypothesis, the same experiments from the section 5.1 were conducted for both sets (A and B) individually. Figure 6 shows progress of the original set M and the derived sets.

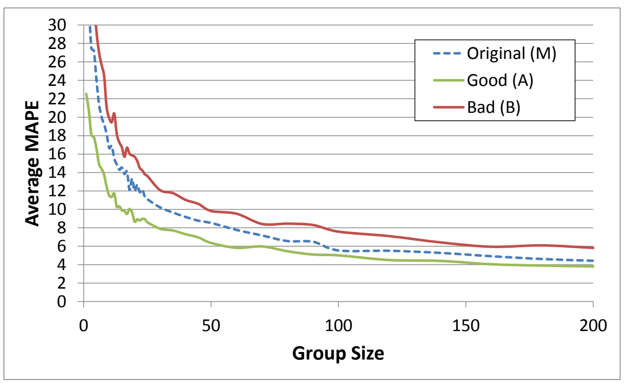

Figure 6: Impact of grouping using good(A) and bad(B) predictable sets

Noticeably, both sets followed the exponential improvement of the original set with a slight offset. However, the results of this experiment revealed that creating groups out of good/bad time series, based on the individual forecast accuracy, proved the prior hypothesis. The set A showed better accuracy ($\langle MAPE(G^{160})\rangle_A \approx 4,09\%$) than the set B ($\langle MAPE(G^{160})\rangle_B \approx 5,93\%$), concluding that the hypothesis was correct if empirical results of the depicted experiment are considered.

5.3 Summary

Two different forecast methods were used to demonstrate and to compare the impact on the forecast accuracy by time series aggregation. Interestingly, results showed competitiveness between simple and robust algorithms. The complex algorithm HW (requiring 4 weeks of historical data) used, slightly over-performed the simple one SN (that requires only the previous day). However, this was not always the case. Since SN showed much worse performance for some days of the week, HW method (with best overall forecast accuracy) was used in further experiments. The method was used to show that groups with better and worse prediction accuracy can be built out of the individual prediction results. Thus the empirical results shown that the individual forecast accuracy is an indicator of the group prediction accuracy. Furthermore, this assumption was proved to be true independently of the used forecast algorithm.

6. CONCLUSIONS

The experiments conducted in this work revealed that improvement of the prediction accuracy by smart metering data aggregation is not a random effect. The results of the experiments show how accuracy progresses along within larger group sizes, as well the impact of the individual prediction accuracy. High predictability might be of key importance in future energy systems and the realisation of prosumer virtual power plants, which effectively group different prosumers and based on their behaviour can derive market trading strategies that may lead to lower costs. If forecast algorithms can be used to accurately report electricity loads in advance, one may preform better production scheduling and thus avoid high balancing costs. Additional improvement of group predictability (e.g. by using an electricity storage system) may allow for the existence of fully automated energy brokering [6].

For the numerous experiments conducted, we hardly considered any system landscape options e.g. high performance machine clusters or advanced computational techniques e.g. parallelism, database calibration or data pre-processing. In the future one may expect that huge smart metering data sets will require greater performance and more advanced system landscapes may be required. Furthermore, data analysis can be used to predict the accuracy improvement rate (of a forecast method) by grouping, prior to building such groups. Such knowledge may reduce computation time needed to find groups constellations of certain predictability, without intensive data fetching. The business implications of these result can vary from a better understanding of clustered user behaviour, to decision making of prioritization in demand-response and demand side management programmes.

7. ACKNOWLEDGMENTS

The authors would like to thank the partners of European Commission co-funded projects NOBEL (`www.ict-nobel.eu`) and IMC-AESOP (`www.imc-aesop.eu`) for the fruitful discussions.

8. REFERENCES

[1] S. Aman, Y. Simmhan, and V. K. Prasanna. Improving energy use forecast for campus micro-grids using indirect indicators. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*, ICDMW '11, pages 389–397, Washington, DC, USA, 2011. IEEE Computer Society.

[2] G. Chalkiadakis, V. Robu, R. Kota, A. Rogers, and N. R. Jennings. Cooperatives of distributed energy resources for efficient virtual power plants. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11, pages 787–794, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.

[3] E. A. Feinberg and D. Genethliou. Load forecasting. In J. Chow, F. Wu, and J. Momoh, editors, *Applied*

Mathematics for Restructured Electric Power Systems: Optimization, Control, and Computational Intelligence, pages 269–285. Springer, 2005.

[4] R. J. Hyndman and G. Athanasopoulos. Forecasting: principles and practice, 2012.

[5] R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, Oct. 2006.

[6] D. Ilic, P. Goncalves Da Silva, S. Karnouskos, and M. Griesemer. An energy market for trading electricity in smart grid neighbourhoods. In *6th IEEE International Conference on Digital Ecosystem Technologies – Complex Environment Engineering (IEEE DEST-CEE), Campione d'Italia, Italy*, June 2012.

[7] S. Karnouskos. Communityware SmartGrid. In *21st International Conference and Exhibition on Electricity Distribution (CIRED 2011), Frankfurt, Germany*, 6-9 June 2011.

[8] S. Karnouskos. Demand side management via prosumer interactions in a smart city energy marketplace. In *IEEE International Conference on Innovative Smart Grid Technologies (ISGT 2011), Manchester, UK*, Dec. 5–7 2011.

[9] S. Karnouskos, P. G. da Silva, and D. Ilic. Assessment of high-performance smart metering for the web service enabled smart grid. In *Second ACM/SPEC International Conference on Performance Engineering (ICPE'11), Karlsruhe, Germany.*, Mar. 14–16 2011.

[10] R. Katz, D. Culler, S. Sanders, S. Alspaugh, Y. Chen, S. Dawson-Haggerty, P. Dutta, M. He, X. Jiang, L. Keys, A. Krioukov, K. Lutz, J. Ortiz, P. Mohan, E. Reutzel, J. Taneja, J. Hsu, and S. Shankar. An information-centric energy infrastructure: The berkeley view. *Sustainable Computing: Informatics and Systems*, 2011.

[11] M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, Jan. 1998.

[12] D. Pudjianto, C. Ramsay, and G. Strbac. Virtual power plant and system integration of distributed energy resources. *Renewable Power Generation, IET*, 1(1):10 –16, march 2007.

[13] L. J. Soares and M. C. Medeiros. Modeling and forecasting short-term electricity load: A comparison of methods with an application to brazilian data. *International Journal of Forecasting*, 24:630–644, 2008.

[14] J. W. Taylor. Short-Term Electricity Demand Forecasting Using Double Seasonal Exponential Smoothing. *Journal of the Operational Research Society*, 54(8):799–805, 2003.

[15] J. W. Taylor, L. M. de Menezes, and P. E. McSharry. A comparison of univariate methods for forecasting electricity demand up to a day ahead. *International Journal of Forecasting*, 22(1):1–16, Jan. 2006.