Efficient load profiling and forecasting in large electric power systems

Imre Lendák[1], Tomáš Horváth[1]

Data Science and Engineering Department, Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary,
lendak@inf.elte.hu,
WWW home page: http://t-labs.elte.hu

Abstract: The goal of this paper is to present an efficient load forecasting algorithm for large electric power systems. It uses a combination of nearest neighbor-based load profile clustering and rule-based load forecasting. The load data was sliced into daily load curves, which were K-Means-clustered, thereby compressing data and simplifying the solution. K-Means was chosen in the proof of concept phase and will be substituted with more precise solutions later. In the forecasting phase the daily load profile is predicted based on the forecast date, day type (e.g. weekday or weekend) and historical consumption data for similar days in the past. The solution was tested on a large dataset consisting of one year-long, 5-minute measurement data in a 1900-power-line system. The solution showed excellent performance in both the training and forecast phases. It produced meaningful forecasts even when the input data contained significant amounts of anomalies. An additional advantage of the presented solution is that it can be used for medium and long-term forecasting with limited and/or missing input data.

1 Introduction

The challenge to accurately predict the power flows in today's large electric power systems receives ample attention. Numerous papers are published in specialized smart grid journals with the promise of being able to predict the electricity consumption of single households or their groups. Others develop solutions which predict the power flows in electric power transmission systems, which span over large geographic regions, e.g. sizable parts of continental Europe or the USA. Yet another group of scientists works on data compression algorithms, with the intention to lower the communication and storage costs incurred in modern smart grids.

Within this setting, we start from the idea that the flows on the power lines in electric power transmission systems have some form of periodicity. More specifically, we will theorize that the configurations of these large systems does not change frequently, and under the same load conditions the flows will be similar on the power lines for similar days, e.g. for Wednesdays in July the load will most probably be very similar under the same loading conditions. Therefore we propose a 2-phase load forecasting algorithm, consisting of a daily load profile clustering and a load forecasting phase.

The following sections of this document contain more detailed description of each of the above steps.

2 State-of-the-art

The body of electric load forecasting knowledge is very large, with numerous papers published in all major domain-specific journals and conferences. As an extensive review of all relevant solutions would not be feasible due to the page limits, we will only refer to those research results, which specifically focus on time series clustering, smart meter big data management and the combination of solutions from these domains used in load forecasting.

2.1 Time series analysis

Reference [19] presents two novel time series clustering methods, namely k-shape and k-MultiShapes (k-MS), which rely on scalable iterative refinement procedures based on shape-based distances (SBD). The authors claim that their solution(s) achieve similar results to dynamic time warping, which at a lower computational cost. k-Shape is quoted as a suitable and novel solution for creating homogeneous and well-separated clusters of time series data. The positive characteristics of k-Shape are domain independence, accuracy and efficiency [20].

Reference [33] describes a convolutional neural network-based time series classification solution, in which the time series features are automatically learned instead of handpicking. The authors describe the process of data preparation, filtering, and the structure of the used network. The authors of reference claim that semi-supervision can boost time series clustering performance [7].

2.2 Data compression

Reference [29] contains an application-oriented review of smart meter data analysis solutions. Three main application areas are identified, namely load analysis, load forecasting, and load management. This is a rare reference which addresses the data privacy and security aspect of the analyzed solutions as well. The most important motivation behind data compression in smart metering are reduced congestion of communication channels used for data transmission, storage overhead, as well as improved data mining efficiency. Reference [30] presents a comprehensive study on smart meter big data compression solutions. The authors of reference [25] present a feature-based, load data compression method for smart metering infrastructures. The solution is not lossless. The authors claim it is efficient, with little reconstruction error. The solution was validated on the Irish Smart Metering Trial Data. The authors of reference [22] present lossless compression algorithms for power system operational data.

The authors of reference [28] use K-SVD sparse representation technique. In the dictionary learning phase, they decompose load profiles into linear combinations of several partial usage patterns (PUPs). In the sparse coding phase, a linear support vector machine (SVM) is used to classify load profiles as residential or small and medium-sized enterprises (SMEs). The authors claim that their solution outperforms k-means, the discrete wavelet transform (DWT), principal component analysis (PCA), as well as piecewise aggregate approximation (PAA).

The solution presented in reference [12] utilizes deep-stacked auto-encoders in electric load data compression and classification.

2.3 Load classification and forecasting

A more general review of smart meter data intelligence is provided in references [1] and [15]. The authors of references [10][21] and [24] explore state-of-the-art machine learning approaches in load forecasting. They review more than 50 research papers and group their contributions into single and hybrid computational intelligence-based approaches. They perform a qualitative analysis based on accuracy and prove the superiority of hybrid solutions. Various short-term load forecasting techniques were compared as early as 1989 in reference [17]. Various machine learning-based short-term load forecasting techniques ranging from moving averages to deep neural networks are addressed in references [2][5][6][8][11] [23][24]. Smart meter forecasting from one minute to one-year horizons is presented in reference [16]. Electricity price and demand forecasting is tackled by the authors in [18]. Bus load forecasting is addressed in reference [3]. Reference [27] presents a smart meter data characterization method based on the Gaussian mixture (GM) model. The authors claim that compared to other state-of-the-art solutions, theirs offers significantly better fitting for meter data. Reference [26] describes a hybrid clustering and classification technique in short-term energy consumption forecasting.

Reference [4] proposes to use clustering in bottom-up, short-term load forecasting. The authors cluster load curves by using wavelets to measure similarity and thereby create super-consumer profiles. The solution was implemented in R and is freely available. The authors of reference [9] analyze four years of measurements represented as time series collected at 245 HV/MV substations. They use the stationarity property of the estimated models to identify daily customer profiles.

The authors of reference [13] analyze annual load curves of households and create annual and weekly load profiles. They also show how additional features of household affect annual consumption and random variation in household energy consumption. Reference [32] presents an analysis of the daily consumption data of 300 residential customers in China. The authors identify four types of monthly usage patterns and 9 abnormal users, with significantly different electricity use patterns. They prove that more than 80% of households have a similar monthly electricity usage pattern.

The authors of references [31] used k-Shape for building energy usage pattern analysis and tested their solution on real-life data measured in ten institutional buildings. Reference [14] goes even further, by using ML techniques to guess the lifestyles of energy consumers based on their consumption patterns.

3 Problem definition

It is necessary to develop a load forecasting solution which is capable to predict loads in extremely large, Europe-wide electric power transmission systems consisting of thousands of power lines. The input data will consist of historical measured loads with a sampling rate of 5 minutes available for at least the last 1-year period. This data will be referred to as dynamic data, due to its frequency of change. Due to data privacy limitation, the input data will not contain the complete static data model of the system under consideration. This means that there will be no data provided to build a mathematical graph consisting of the busbars (vertices) and power lines (edges) connecting them.

It is expected that the prediction horizon will be 5 minutes ahead. The solution should be extensible and be capable to provide acceptable mid- and long-term forecasts (1 day or 1 week ahead) as well. Additionally, it is necessary for the solution to handle temporary unavailability of significant amounts of measurements, when those will not be provided in a timely manner by one or more countries and/or companies in the geographical area under consideration. Optionally, the solution should be able to incorporate weather forecast and other freely available 3rd party data and thereby increase the accuracy of its outputs. The relevance of such data might vary, as the extent of data anonymization required will not allow the forecasting tool access to the geographical location of system resources (i.e. power lines).

4 Solution

We suppose that the power flows measured on the power lines in large electric power systems show some level of regularity and can be therefore classified into load profiles. Based on this assumption, we propose to create a hybrid forecasting solution which consists of two phases. In the first phase the electric power flow data is clustered into daily load profiles. In the second phase we experiment with various forecasting algorithms to predict the daily load profile for each power line based on historical data. This means that instead of predicting the expected values of power flows 5 minutes in the future, we predict the load profile for an entire day in advance.

This solution addresses most of the more complex requirements listed above, namely it is expected that it can handle missing data and create forecasts for multiple days ahead. More specifically, it tolerates the absence of significant amounts of short-term historical data and still produces meaningful forecasts based on medium- or long-term historical data (e.g. data older than a week or month). Similarly, medium- or long-term (7 days or more ahead) forecasts are feasible with this type of solution.

In the following sections we propose the load profile generation and forecasting steps.

4.1 Load profile generation

The power flow data is sliced into daily (24h) load profiles consisting of measured flow values sampled every 5 minutes. The sliced daily load profiles are normalized. The amplitude of each daily load profile is memorized. The normalized daily flows are clustered, separately for each power line, i.e. a set of representative load profiles is calculated for each power line. For each year, month, day of the week and power line we memorize the load profile and amplitude.

The daily load profile clustering introduces some error, but significantly improves algorithm performance if it is not necessary to re-calculate the centroids too often. As the main idea is that the daily load profiles will be similar, this should not be an issue, i.e. we should re-calculate the representative daily load profiles relatively rarely, e.g. once in 15 or 30 days.

4.2 Load forecasting

Our improved baseline load forecasting solutions is rule-based. In the predict phase it looks up the most likely historical load profile and amplitude for each of the power lines. They are used to calculate the predicted load profile for the entire prediction day and extract as many samples as required. This means that we will predict for a whole day, i.e. 288 future values with a 5-minute sampling interval.

Predictions spanning two calendar days are somewhat challenging as in their case it is necessary to either select two load profiles and stitch them together; or to select the ‘end’ and the ‘start’ of the same load profile. In the actual solution we perform the latter, simpler solution, i.e. use a single daily profile to cover both calendar days before and after midnight.

Load profile prediction The load profile selection is rule-based, and it is performed in the following steps (listed by priority):

1. if there are (past) values for the same (year, weekday, power line) tuple within the last two months, then select one;

2. if there are (past) values for the same (year-1, month, weekday, power line) tuple, then select one;

3. if there are historical values for (year – (2:N), month, weekday, power line) tuples, then select the most likely one – N is configurable and defaults to 15; or

4. if none of the above is found, do a random load profile selection.

Amplitude calculus Amplitudes are chosen as averages of historical values for similar days in the past. This part of the solution is also rule-based, and its steps/choices are very similar to the curve selection algorithm presented above (also listed by priority of choice):

1. choose non-zero amps for the same weekday within the last two calendar months and average them;

2. average last year’s amplitudes for the same month and weekday combination;

3. average the values in the longer-term history up to M years in the past, where M is configurable and defaults to 15; or

4. choose a default amplitude, which was for simplicity set to a (configurable) scalar.

5 Experiments

The input dataset was loaded from Heterogeneous Data Format (HDF), version 5. We used HDFView version 3 for data exploration. The training data consisted of historical power flows for 1935 power lines over a July-to-July one-year period expanding over two calendar years. The training data consisted of 3-month-long power flows over a July-September period immediately after the training data. The training data was split into 68 data slices ranging in length between one hour and a couple of days-long. Both training and testing power flows were sampled every 5 minutes.

We implemented the solution in Python version 2. We used the scikit-learn library for K-Means and other necessary ML algorithms. Data visualization during experimentation was performed with matplotlib.

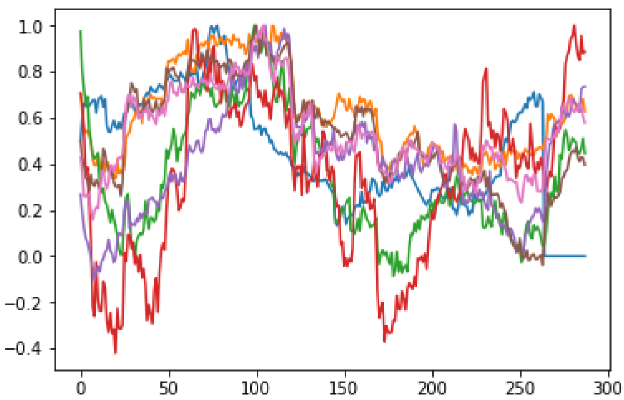
5.1 Load profiles

The number of representative load profiles (i.e. centroid count) was set to 10, which was shown to be a sufficient during data exploration. The clustering was performed by K-Means, which was chosen due to its efficiency. Each load profile was a curve consisting of 288 numerical values.

We re-trained the model if a sufficiently long time period expired since the last training performed. For simplicity we performed full re-train is runs from scratch. This design decision was acceptable as the clustering phase for the 1-year period and 1935 power lines took up to 20 minutes on a personal computer with an Intel i7 CPU, 8 GB of RAM and SSD.

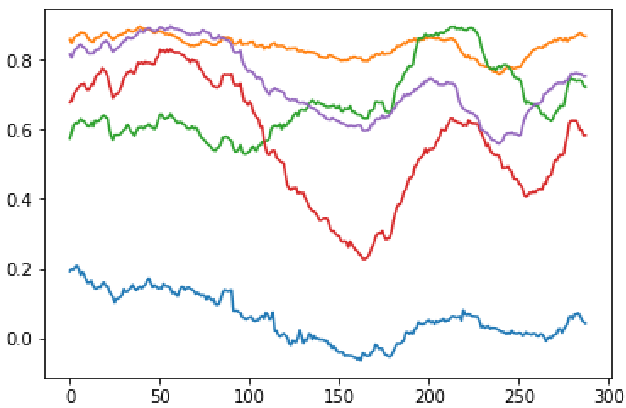
Example 1-week input load profiles are shown in Figure **??**. Example load profiles detected for the above sin-

Figure 1: Power flows for a selected power line



gle power in its corresponding 1-year-long flow data with K-Means and cluster number 5 are shown in Figure **??**. Note that the cluster number of five was used here only to illustrate the clustering results in a visually pleasing diagram. Otherwise, during most experiments cluster number 10 was used. The introduction of these load profiles

Figure 2: Load profiles for cluster number = 5

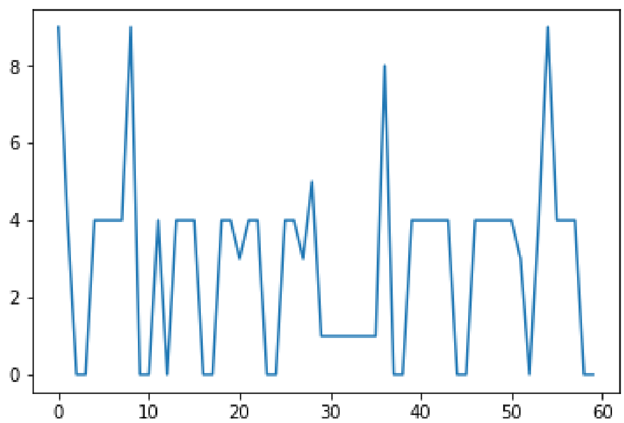


allowed us to significantly reduce the solution space, i.e. to replace 365 (or more if the dataset spans a longer time period) daily curves with a small set of load profiles. Daily loads in the historical data were essentially represented with tuples consisting of a load profile identifier, daily amplitude multiplier (as the curves were normalized in the (-1, 1) range), year, month and weekday. We used weekdays as based on past experience, and the related works, we theorized that load profiles will be quite similar in a certain day of the week in each month of the calendar year, e.g. customer electricity use is usually similar on each Saturday in July if the weather is good.

As explained earlier, we reduced the size of the historical dataset by introducing the load profiles and thereby simplified the forecasting task. The original, vast dataset consisted of 288 daily flow measurements with a 5-minute sampling rate, collected for 365 days and 1916 power lines (i.e. 288x365x1916 = 201,409,920 values). With the load profiles we reduced the dataset to a tuple consisting of the power line identifier, date, load profile identifier and amplitude, i.e. the multiplier with which the load profile is multiplied to obtain the 'original' load measurements. This meant that instead of 288 floating point values for each of the day and power line combination, we received a tuple consisting of the above four elements, i.e. 4x365x1916 = 2,797,360, which was a data reduction by 72 times, i.e. almost two orders of magnitude.

We randomly selected one power line and created a plot of the assigned load profile identifiers with K-Means over a 60-day long period, starting with a Thursday (i.e. weekday identifier 3). The resulting diagram can be seen in Figure **??**. The diagram covers a 60-day period during the

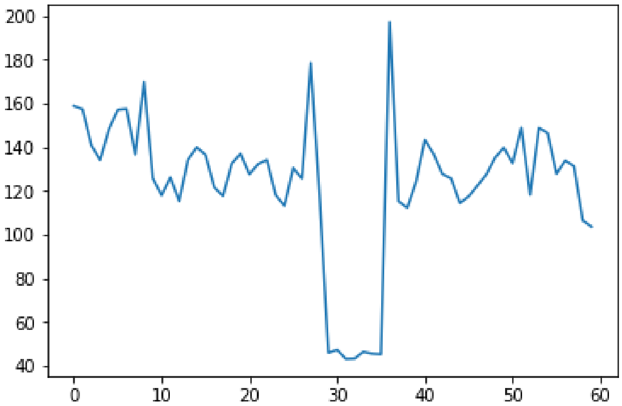
Figure 3: Assigned load profiles over 60 days



July-August period. We can see in the diagram that for almost all weekends (for Saturdays 2, 9, 16, and Sundays, 3, 10, 17, etc.) the representative load profile was with ID = 0. Additionally, we can see that load profile 4 was very frequent for weekdays.

The power flow amplitudes for the same power line and period is shown in Figure **??**. We can clearly identify an anomalous period around day 30, similarly as in the load

Figure 4: Calculated amplitudes over 60 days



profile diagram above. Such periods are usually related to periods with different weather conditions (e.g. colder, rainy days with less use or air-conditioning) and/or configuration changes in the power system. In this diagram we might identify dips in amplitudes during the weekends (day 0 is a Thursday), but there is no other clear regularity identifiable via visual inspection.

5.2 Load forecasting

We transformed the date information into a tuple of three values, namely year, month and weekday. With this modification, the inputs fed into the load forecasting code were tuples representing historical daily loads in the following format:

(line id, year, month, day of the week, profile id, amp)

We implemented the rule-based, baseline algorithm as described above. The forecasting code expected the following inputs:

- Forecast date, e.g. May 30th, 2019;
- Power line identifier.

The code transformed the dates into the (year, month, day of the week) sub-tuples and subsequently looked up the most similar historical data as explained in section IV/B. The load forecasting code returned two values, namely the 'expected' load profile identifier and amplitude for the prediction day.

Experiment I: Short-term forecasting We compared the results of our load forecasting solution to the persistence model, which simply predicts that the N future values will be equal to the N historical values preceding them. The persistence model implementation used for testing was limited to short-term forecasts for the next N values immediately following the last time period received in the training data, i.e. it did not support time gaps between the latest training data and the forecasting period.

Table 1: RMSE values - Short-term forecasting

Length	Persistence model	Our model
1 hour(s)	47.72	55.53
2 hour(s)	53.44	51.56
4 hour(s)	61.98	64.05

Table 2: RMSE values - Mid-term forecasting

Length	Our model
1 day	130.23
1 week	148.33
1 month(s)	137.38

We experimented with N=(12, 24, 48), i.e. a 1, 2 and 4 hour short-term prediction horizons. One execution of our prediction code took around 30 seconds to execute, regardless of the selected interval length. The time to execute the same prediction task on the persistence model was quite similar, which meant that most of the time was spent on creating the resulting datasets.

The RMSE errors calculated with the persistence and our model are shown in Table **??**. The error was calculated between the prediction values and measured values extracted from the adapt/test data.

Experiment II: Mid-term forecasting As explained above, the proposed solution is capable to produce meaningful mid and long-term forecasts with limited historical data availability. We tested the algorithm on the following three prediction tasks:

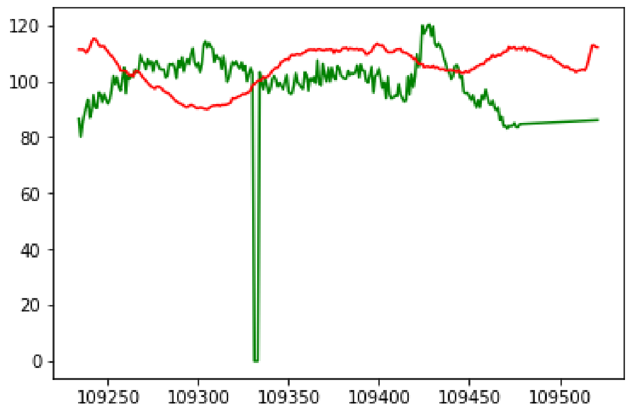
- 1 day ahead, i.e. predict loads for the next day.
- 1 week ahead, i.e. predict the power flows for the day one week in future compared to the last training data item.
- 1 month ahead.

As the persistence model used during this research did not have built-in support for these types of prediction tasks, we measured the RMSE for the proposed model only. Table **??** contains the results of our measurements.

We decided to further explore the resulting predictions (i.e. load forecasts) by visually comparing the predicted load curves to the actual measurements received as part of the test/adapt data. A randomly selected load flow prediction and real daily load values are shown in Figure **??**.

The values predicted one week in the future (in red color) are in a similar value range, i.e. they do not have an

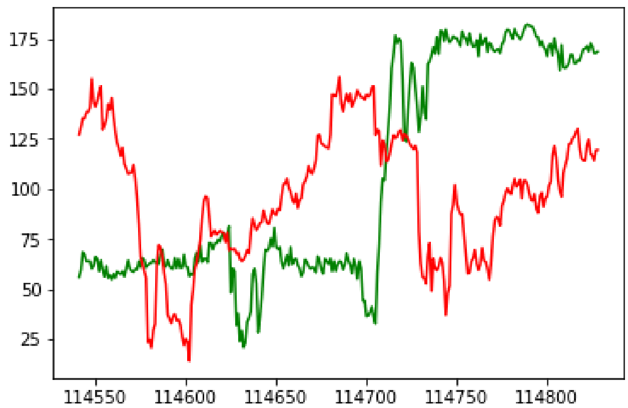
Figure 5: Predict one week ahead



inverted sign or exceedingly different amplitudes. Not surprisingly the anomalous zero value in the (real) measured value is not predicted by the presented load forecasting algorithm.

In Figure **??** we present the resulting daily load curve for the same power line as in the previous example.

Figure 6: Predict one month ahead



The relatively low accuracy levels can be improved by implementing a more accurate load profile clustering technique, instead of the relatively coarse K-Means, with a different distance measure. Such changes would allow the authors to find the most relevant daily load profiles.

6 Conclusion

This paper describes a power flow prediction algorithm, which relies on analyzing historical (power) flow information and creating a configurable number of representative daily load profiles for each power line. Predictions are based on high performance look-ups – a single load profile index is selected for the target prediction day, for which a (daily) load profile is calculated. The algorithm is rule-based as opposed to creating a neural network-based or other machine learning solution. It can be tweaked further, and one might expect from it to produce deterministic results for the target electric power system under consideration. It might be used as a baseline solution, against which less deterministic, machine-learning solutions can be compared and measured.

The main advantages of the presented algorithm are its training and prediction performance. It analyzes the historical flow information and creates configurable numbers of representative daily load profiles for each power line. Predictions are based on high performance look-ups – a single load profile index is selected for the prediction day, a predicted (daily) flow curve is calculated for the whole calendar day for which the prediction is initiated. The predicted values are 'stitched' (i.e. amplitudes are augmented) to the actual input flows and the requested number of predicted samples is returned. The solution can make predictions based on (very) limited information and handle gaps in input data. It is also capable to quickly predict the most likely and meaningful flows for extended future periods, i.e. instead of covering only a couple of hours immediately after the last input (training) data received, it is capable to predict a day, week or even longer periods ahead. Prediction accuracy will obviously vary as a function of the amount of historical data, i.e. if there are historical flow values measured multiple years in the past, then we expect to obtain higher accuracy. This was not shown in our experiments though, as the training data available covered only one calendar year.

The solution was tested on dataset consisting of power flows collected over a one year, July-to-July period. The test data covered the July to September period immediately following the training data. The system under consideration consisted of 1916 power lines. The accuracy of the proposed model was compared to the persistence model. We showed that the RMSE was quite similar in short-term forecasting tasks (1 to 4 hours) to the persistence model. We measured the accuracy of our algorithm in mid-term load forecasting scenarios, whose length was set to be 1 day, 1 week, as well as 1 and 3 months in the future.

The main disadvantage of the algorithm is its reliance on historical flow information only, i.e. it does not take auxiliary information into consideration. Additionally, the algorithm does not specifically cover national holidays, which often fall on weekdays and result in weekend-like load profiles - this missing element is relevant if the algorithm is used for state-level forecasting, but has lower relevance in continent-wide (e.g. Europe) load forecasting scenarios, in which the impact of national holidays is lower. Further tuning and optimization of the load profile classification and curve/amplitude selection algorithms might further improve performance and accuracy.

References

[1] Alahakoon, D., Yu, X.: Smart electricity meter data intelligence for future energy systems: A survey. IEEE Transactions on Industrial Informatics. **12(1)** (2016) 425–436.

[2] Amjady, N. (2001). Short-term hourly load forecasting using time-series modeling with peak load estimation capability. IEEE Transactions on Power Systems, 16(3), 498-505.

[3] Amjady, N.: Short-term bus load forecasting of power systems by a new hybrid method. IEEE Transactions on Power Systems. **22(1)** (2007) 333–341.

[4] Auder, B., Cugliari, J., Goude, Y., Poggi, J. M.: Scalable clustering of individual electrical curves for profiling and bottom-up forecasting. Energies. **11(7)** (2018) 1893.

[5] Chen, K., Chen, K., Wang, Q., He, Z., Hu, J., He, J.: Short-term load forecasting with deep residual networks. IEEE Transactions on Smart Grid. (2018) 1–1.

[6] Chen, J. F., Wang, W. M., Huang, C. M.: Analysis of an adaptive time-series autoregressive moving-average (ARMA) model for short-term load forecasting. Electric Power Systems Research. **34(3)** (1995) 187–196.

[7] Dau, H. A., Begum, N., Keogh, E.: Semi-supervision dramatically improves time series clustering under dynamic time warping. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. (2016, October) 999–1008.

[8] Ding, N., Benoit, C., Foggia, G., Bésanger, Y., Wurtz, F.: Neural network-based model design for short-term load forecast in distribution systems. IEEE Transactions on Power Systems. **31(1)** (2016) 72–81.

[9] Espinoza, M., Joye, C., Belmans, R., De Moor, B.: Short-term load forecasting, profile identification, and customer segmentation: a methodology based on periodic time series. IEEE Transactions on Power Systems. **20(3)** (2005) 1622–1630.

[10] Fallah, S., Deo, R., Shojafar, M., Conti, M., Shamshirband, S.: Computational intelligence approaches for energy load forecasting in smart energy management grids: state of the art, future challenges, and research directions. Energies. **11(3)** (2018) 596.

[11] Hagan, M. T., Behr, S. M.: The time series approach to short term load forecasting. IEEE Transactions on Power Systems. **2(3)** (1987) 785–791.

[12] Huang, X., Hu, T., Ye, C., Xu, G., Wang, X., Chen, L.: Electric Load Data Compression and Classification Based on Deep Stacked Auto-Encoders. Energies. **12(4)** (2019) 653.

[13] Kuusela, P., Norros, I., Reittu, H., Piira, K.: Hierarchical Multiplicative Model for Characterizing Residential Electricity Consumption. Journal of Energy Engineering. **144(3)** (2018) 04018023.

[14] Kwac, J., Flora, J., Rajagopal, R.: Lifestyle segmentation based on energy consumption data. IEEE Transactions on Smart Grid. **9(4)** (2018) 2409–2418.

[15] Liu, X., Golab, L., Golab, W., Ilyas, I. F., Jin, S.: Smart meter data analytics: Systems, algorithms, and benchmarking. ACM Transactions on Database Systems (TODS). **42(1)** (2017) 2.

[16] Massidda, L., Marrocu, M.: Smart meter forecasting from one minute to one year horizons. Energies. **11(12)** (2018) 3520.

[17] Moghram, I., Rahman, S.: Analysis and evaluation of five short-term load forecasting techniques. IEEE Transactions on power systems. **4(4)** (1989) 1484–1491.

[18] Motamedi, A., Zareipour, H., Rosehart, W. D.: Electricity price and demand forecasting in smart grids. IEEE Transactions on Smart Grid. **3(2)** (2012) 664–674.

[19] Paparrizos, J., Gravano, L.: Fast and accurate time-series clustering. ACM Transactions on Database Systems (TODS). **42(2)** (2017) 8.

[20] Paparrizos, J., Gravano, L.: k-shape: Efficient and accurate clustering of time series. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. (2015, May) 1855–1870.

[21] Raza, M. Q., Khosravi, A.: A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. Renewable and Sustainable Energy Reviews. **50** (2015) 1352–1372.

[22] Sarkar, S. J., Kundu, P. K., Sarkar, G.: Development of lossless compression algorithms for power system operational data. IET Generation, Transmission Distribution. **12(17)** (2018) 4045–4052.

[23] Sun, X., Luh, P. B., Cheung, K. W., Guan, W., Michel, L. D., Venkata, S. S., Miller, M. T.: An efficient approach to short-term load forecasting at the distribution level. IEEE Transactions on Power Systems. **31(4)** (2016) 2526–2537.

[24] Taylor, J. W., McSharry, P. E.: Short-term load forecasting methods: An evaluation based on european data. IEEE Transactions on Power Systems. **22(4)** (2007) 2213–2219.

[25] Tong, X., Kang, C., Xia, Q.: Smart metering load data compression based on load feature identification. IEEE Transactions on Smart Grid. **7(5)** (2016) 2414–2422.

[26] Torabi, M., Hashemi, S., Saybani, M. R., Shamshirband, S., Mosavi, A.: A Hybrid clustering and classification technique for forecasting short-term energy consumption. Environmental Progress Sustainable Energy. **38(1)** (2019) 66–76.

[27] Tripathi, S., De, S.: An efficient data characterization and reduction scheme for smart metering infrastructure. IEEE Transactions on Industrial Informatics. **14(10)** (2018) 4300–4308.

[28] Wang, Y., Chen, Q., Kang, C., Xia, Q., Luo, M.: Sparse and redundant representation-based smart meter data compression and pattern extraction. IEEE Transactions on Power Systems. **32(3)** (2017) 2142–2151.

[29] Wang, Y., Chen, Q., Hong, T., Kang, C.: Review of smart meter data analytics: Applications, methodologies, and challenges. IEEE Transactions on Smart Grid. **10(3)** (2019) 3125–3148

[30] Wen, L., Zhou, K., Yang, S., Li, L.: Compression of smart meter big data: A survey. Renewable and Sustainable Energy Reviews. **91** (2018) 59–69.

[31] Yang, J., Ning, C., Deb, C., Zhang, F., Cheong, D., Lee, S. E., ... Tham, K. W.: k-Shape clustering algorithm for building energy usage patterns analysis and forecasting model accuracy improvement. Energy and Buildings. **146** (2017) 27–37.

[32] Yang, T., Ren, M., Zhou, K.: Identifying household electricity consumption patterns: A case study of Kunshan, China. Renewable and Sustainable Energy Reviews. **91**

(2018) 861–868.

[33] Zębik, M., Korytkowski, M., Angryk, R., Scherer, R.: Convolutional Neural networks for time series classification. In International Conference on Artificial Intelligence and Soft Computing. Springer, Cham. (2017, June) 635–642.