Leveraging Apache Spark for Appliance-Level Load Monitoring with Ensemble Learning Techniques

Iosif Arvanitis[1], Menelaos Panagiotis Papastergiou[1], Agorakis Bompotas[1], Evangelos Gerardidis[1], Ioanna Giannoukou[2], Ioannis Karydis[3], and Spyros Sioutas[1]

[1] Dept. of Computer Engineering and Informatics, University of Patras, 26 504 Patras, Greece
[2] Dept. of Management Science and Technology, University of Patras, 26 334 Patras, Greece
[3] Dept. of Informatics, Ionian University, Kerkyra 49 132, Greece

Abstract. Appliance load monitoring measures the electricity consumption of household appliances, driven by motivations such as financial savings, reducing environmental impact, and predictive maintenance. It provides users with detailed energy usage information, leading to informed decisions about appliance operation and replacement, significant savings, and reduced carbon footprints. Traditionally relying on individual smart meters, a costly and complex method, load monitoring has evolved with non-intrusive load monitoring (NILM), which estimates individual appliance power use by analyzing overall household consumption. This shift, coupled with the exponential data growth from widespread smart meter adoption, necessitates scalable processing techniques. Apache Spark, a distributed processing framework, offers a solution with its scalability and flexibility, handling expanding workloads effectively. However, due to Spark's lack of built-in support for deep learning and specialized NILM algorithms, in this paper, we explore whether simpler solutions like Random Forests and Gradient-Boosted trees within Spark's MLlib are able to produce satisfactory results. Our experimental evaluation demonstrated that these algorithms perform well, especially for devices with smoother consumption patterns, highlighting the potential of shallow machine learning techniques in NILM applications, making them practical for real metering systems where data volume and speed rule out more complex solutions.

Keywords: Machine Learning · Distributed Systems · Energy consumption · Non-Intrusive Load Monitoring · Energy Disaggregation

1 Introduction

Appliance load monitoring is the practice of measuring the electricity consumption of household appliances. There are various motivations for its development, of which the main ones are financial savings, reducing environmental impact and predictive maintenance. Information regarding how much energy each appliance consumes, may empower the users of load monitoring to make informed decisions regarding equipment operation, purchase and replacement. Indeed, such feedback has been shown to result in significant savings for domestic applications [9]. In addition to savings, utilizing appliances in an energy efficient manner, reduces a household's carbon footprint. Moreover, load monitoring allows the users to identify or even predict potential appliance failures, resulting in increase reliability of these devices, and reduce related costs. Predictive maintenance techniques have also been shown to result in substantial savings for the users [14,7].

Traditionally, load monitoring relied on installing smart meters for each appliance. While this method provides highly accurate results, it suffers from several drawbacks. For instance, installing individual smart meters can be time-consuming, costly and inconvenient. Non-intrusive load monitoring (NILM) offers a promising alternative by analyzing the whole household's electricity consumption. NILM disaggregates this signal to estimate individual appliance power use, achieving the benefits of traditional methods without the high cost and installation complexity. However, NILM is not without limitations. The accuracy of disaggregating specific device readings can be affected by various factors, including appliances with similar power signatures, low-power devices, and dynamic loads with fluctuating power draw [19].

A model suitable for NILM application takes as the aggregate power as the input signal, and aims to infer the individual power consumption of each appliance. We can formulate the problem as an inverse summation problem. For a given household with n appliances, at each given moment P_{total} represents the total power draw of all appliances, plus an error term e which accounts for noise, unidentified or very low power devices. NILM aims to recover the individual appliance power draws, P_i such that:

$$P_{total}(t) = \sum_{i=1}^{n} P_i(t) + e(t)$$

By modeling each appliance as a finite state machine, we can identify changes in power draw and infer the operating state of each device. The most common assumption is that appliances have two states (ON/OFF). A classification problem can emerge from the regression problem we formulated: the model predicts the state of each appliance at a given moment [22].

With the widespread adoption of smart meters, the volume of collected data has grown exponentially and high frequency measurements only intensify this issue. For instance, prominent datasets REDD and UK-Dale [18,16], measured electrical data for 6 and 5 households respectively, whereas the Pecan Street's Dataport [20], includes measurements to more than 600 households across the US. This transition propels NILM into the big data domain, demanding the development of scalable techniques to handle the complexities of processing and analyzing vast amounts of data for appliance-level recognition.

The objective of this work is to take steps in addressing this shift. We propose Apache Spark, a distributed processing framework as a suitable solution for tackling NILM at scale. We specifically evaluate the effectiveness of Spark's ensemble learning algorithms, Gradient Boosted Trees (GBTs) and Random Forests, for appliance-level regression. To this end, we use a subset of the ENERTALK dataset [25]. This initial exploration focuses on shallow learning ensemble methods, paving the way for further investigation into more complex models suited for big data applications in NILM. Our findings aim to contribute to the development of distributed machine learning solutions that can handle the ever-growing volume of data associated with smart meter deployments.

This paper is structured as follows. In Section 2, we provide a brief overview of methods and datasets used in NILM. In Section 3, we demonstrate the experimental setup used for the assessment of the models discussed. In Section 4 the results of our work are presented. Closing, Section 5 briefly summarizes our observations and conclusions.

2 Related Work

The field of NILM was pioneered by G. W. Hart's seminal work during the 80s and 90 [12]. Hart's approach involved pattern recognition in power draw signals and combinatorial optimization to identify individual appliance states that collectively explain the total power consumption. Since then, a wealth of algorithms and techniques have been used for NILM. Several surveys provide a comprehensive overview of the current state of the art [11,2,27,24].

As the surveys indicate, deep learning methods have become increasingly more popular in recent years. Whoever, there are several well-established techniques predating deep learning methods. We mention some of the most widely applied approaches. Hidden Markov Models (HMMs) statistically analyze power sequences, identifying hidden states that correspond to appliance activity. Factorial HMMs (FHMMs) extend this concept to handle multiple appliances simultaneously [17,21]. Optimization techniques search for the closest combination of appliance states to approximate the total power draw. Genetic algorithms and integer programming are two such approaches [6,3]. Shallow learning has also been applied to this problem, most often via Support Vector Machines, Decision Trees, Random Forests and Gradient Boosted Trees [8,13]. Other techniques include signal processing and graph signal processing methods, (fuzzy) clustering, Bayesian procedures and others [23,29,10].

Artificial neural networks have proven their effectiveness on this problem due to their ability to both recognize power consumption patterns and accurately identify devices [26]. Convolutional Neural Networks (CNNs) can effectively capture the spatial patterns within the data, useful for identifying appliances with distinct power signatures [28]. NILM datasets are comprised by time series data, which lends itself naturally to Recurrent Neural Networks (RNNs). Particularly Long Short-Term Memory (LSTMs) excel at handling the sequential nature of power measurements, allowing them to learn temporal dependencies within the power draws [15]. Gated Recurrent Units (GRUs) are used to replicate this attribute of LSTMs, with lower computational cost. Autoencoders offer another deep learning approach, where the network learns a compressed representation of the power signal, potentially highlighting key features that differentiate appliance usage. These deep learning methods have demonstrated impressive performance in NILM, receiving the attention of researchers as Angelis et. al. [2] highlight.

This advancement in disaggregation techniques could not have taken place without datasets which made a twofold contribution to the field of NILM. Quality datasets enable training and evaluation of sophisticated models while also providing a baseline for method comparison. The most prominent datasets within the literature are REDD, UK-DALE, BLUED, ECO among many others [18,16,5,1]. The present work, has been based on the ENERTALK [25] dataset. Also providing meaningful comparisons, NILMTK is a library of utilities and algorithms, aiming to provide benchmark algorithms for NILM, and aides in working with various datasets [4].

3 Experimental Setup

We make use of on an Apache Spark cluster using the Spark MLlib API for execution of our experiments. More specifically, we used *RandomForestRegressor* and *GBTRegressor* from MLlib for our experiments. Those regression algorithms are optimized to work on the distributed Spark environment and provide a user-friendly API for users integrating similar models from non-distributed source code to Spark.

The ENERTALK dataset was used to train the models. Aggregate and per-appliance measurements were collected at a sampling rate of 15 Hz for 22 houses. Data were collected for 29 to 122 days, depending on the house, and for up to 7 appliances in each house. AMI technology was used instead of conventional meters to enhance responsiveness and sampling rates. For our work, data from house 00 was used for training. We trained models for appliances in the house including a refrigerator, a rice cooker, a washing machine, a TV, a microwave oven, a water heater and a kimchi refrigerator.

Training was performed on the active power of appliances and the aggregated signal. *Active power (P)* refers to the usable or consumed energy of an AC circuit measured in Watts (W).

$$P = V \cdot I \cdot \cos(\theta)$$

where V is the RMS (Root Mean Square) voltage of the house, I is the RMS current derived from the AC waveform, and θ is the phase angle between the two.

Initially, each dataset was uploaded and cached into Spark dataframes in the cluster for efficient computations. We consider only the active power for each appliance and the aggregated measurements so the rest of the time series was dropped. Based on the active power of the house sensor at a specific timestamp, we make a prediction for each appliance in the house at that exact timestamp using each one of the models. Then, we compare the predicted value with the actual measurement of the sensor at that moment.

We trained a *Random Forest* and a *Gradient Boosted Tree* for each appliance. Those ensemble shallow learning models were selected both for their ability to capture non-linear correlations in the data, which are prevalent in NILM, and their relatively fast training speed, compared to other more complex models. The models are required to perform well to be useful for various applications, but their training should also be relatively fast to allow for real time retraining based on new supplied data. A case can be made for edge computation capability, where the processing power is also constrained, further diminishing the speed at which models can retrain. Those two conflicting factors influenced the majority of design decisions we took on our approach.

We performed our training using 80% of the measured values and evaluated the performance of the models using the remaining 20% of the values. Before training, a normalization phase took place, where each value was scaled to reside in range [0, 1] using a min-max scaler. Parameter selection for the models was influenced by the following two factors. We tried to maximize the size of our models in hope of greater performance, while maintaining good overall speed of training. Random Forests are parameterized mostly on the maximum number of trees allowed and the maximum depth of each tree. To keep our models efficient, we used up to 100 trees per model and each tree had at most 20 levels.

The Gradient Boosted Tree models are characterized by their maximum depth. Another parameter that is specified during training is the number of iterations or stages in which the weak learner of the previous stage is improved in order to minimize the residuals left over. The number of iterations forms an important factor when estimating the training speed of the algorithm and thus was kept relatively small, at 80 iterations per model. The maximum depth was also reduced to 10 to allow greater performance and flexibility of the models so as to not overfit.

4 Results

In order to evaluate the performance of our models we employed a variety of different performance metrics. More specifically we present the *root mean squared error (RMSE)*, the *mean average error (MAE)* and the *R2* metrics of each of the trained models. The performance on each appliance based on those metrics is presented below.

In general, the performance of both the random forest and the gradient boosted tree models on each appliance was similar. Gradient boosted trees performed slightly better, but the difference in performance is negligible. This implies that ensemble shallow learning models tend to have similar performance on NILM and greater performance should be expected by models of greater learning capacity, like deep learning models. An argument for gradient boosted trees can be made as they tend to have a smaller memory footprint than random forests, due to their iterative approach to learning.

Appliance	RMSE	MAE	R^2
Refrigerator	29.786	8.043	0.509
Rice Cooker	0.058	0.046	0.004
Washing Machine	108.675	24.271	0.905
TV	13.9	7.148	0.907
Microwave Oven	12.26	0.709	0.645
Kimchi Refrigerator	29.786	8.043	0.509
Water Heater	24.529	5.745	0.849

Table 1. Evaluation metrics for the Random Forest models

Appliance	RMSE	MAE	R^2
Refrigerator	28.737	4.92	0.543
Rice Cooker	0.058	0.046	0.001
Washing Machine	94.77	10.556	0.927
TV	12.699	4.393	0.923
Microwave Oven	11.797	0.447	0.671
Kimchi Refrigerator	28.737	4.92	0.543
Water Heater	25.21	3.875	0.841

Table 2. Evaluation metrics for the Gradient Boosted Tree models

Depending on the appliance type, the models performance varied significantly, as one can observe from the evaluation metrics. The worst performing model was that of the rice cooker, where R^2 was close to 0. The active power of the rice cooker was very low, compared to the aggregate measurements, as this appliance was mostly idle. Such small measurements are hard to predict accurately and relations cannot be extracted easily from that dataset. If we were concerned with classification of the state of operation, on the other hand, both models would accurately predict the correct (idle) state, as they both predicted very small active power for the rice cooker.

The refrigerator and kimchi refrigerator, for which the models performed almost identical, also gave models that perform weakly. Those devices are always open and their active power consumption rarely has spikes during their operation. Because of this, it is hard to predict when those devices will show power spikes, which, in turn, does not allow for easy event detection. The models did fit well on the data, concerning the always open operation of the refrigerators.

Overall, for the rest of the appliances, both models performed well and fit the data properly. Overfitting, which was accounted for in the selection of smaller parameters for our models, was not really observed in any case. Those two observations imply that by constraining the size of ensemble shallow learning models, to allow for fast training and smaller memory footprint, adequate performance for active power prediction of common appliances can be obtained. It is worth mentioning that the large size of the dataset did contribute to the accuracy of our models and allowed for smaller models to perform well.

5 Discussion

IoT devices generate vast amounts of data at high velocity, necessitating systems or frameworks that can handle real-time stream processing and large-scale data analytics. Apache Spark, with its in-memory computing and parallel processing abilities, excels in processing data streams from various IoT devices efficiently. Its integrated libraries like Spark Streaming allow for seamless data ingestion and processing, making it possible to perform complex event processing and real-time analytics. Moreover, Apache Spark's scalability and flexibility make it ideal for IoT ecosystems. As IoT deployments grow, the data volume increases exponentially, requiring a scalable solution that can handle expanding workloads without compromising performance. Spark's distributed computing model allows it to scale out across clusters, ensuring that large datasets from numerous IoT sensors can be processed timely and efficiently. Furthermore, Spark provides excellent support for varied data sources, along with built-in machine learning and graph processing libraries. The aforementioned capabilities are crucial when dealing with IoT data and make Apache Spark essential in processing such data.

Nevertheless, Apache Spark, despite its strengths, does not come without limitations. More specifically it lacks innate support for deep learning algorithms. While Spark can integrate with external deep learning frameworks like TensorFlow or PyTorch, this can add significant complexity and often leads to suboptimal performance because of the data that needs to be exchanged among the cluster's nodes. Thus, for applications that depend upon sophisticated neural network models, Spark's current ecosystem may not be the most efficient choice as it requires additional tools and infrastructure to achieve the desired outcomes. Similarly, Spark does not have dedicated libraries or APIs for NILM. NILM involves specialized algorithms that require precise and domain-specific processing capabilities, which are not directly addressed by Spark's general-purpose data processing framework. While Spark's machine learning library, MLlib, offers a range of standard algorithms, it does not cater specifically to NILM's unique requirements.

However, the nature of the problem necessitates the use of big data frameworks, as the sheer volume and velocity of data demands robust and scalable processing capabilities. This gap in capabilities means that, while we can handle the data volume and processing speed, we are often left without the precise analytical tools required to extract the most valuable insights from our data. In an attempt to overcome these limitations, we explored whether simpler solutions might yield adequate results. This involved testing more straightforward algorithms and methods that are inherently supported by Apache Spark's MLlib. While these simpler solutions are easier to implement and integrate, they might be deficient in terms of accuracy and depth of analysis as the complexity of the problem's domain often requires advanced techniques that go beyond the capabilities of these basic approaches. Thankfully, as the results presented in this paper suggest this was not the case here.

In summary, it was observed that the algorithms of random forests and Gradient-Boosted trees used produced quite satisfactory results. Their metrics were quite similar, with the gradient-boosted trees algorithm appearing to have a slight edge on average across the devices. At the level of each device, the gradient-boosted trees continue to outperform the random forests, with the exception of a few devices where the random forests seem to provide a slightly better interpretation of their consumption. Finally, it was noted that for devices with smoother consumption over time, such as refrigerators, the produced results are of higher quality. The experimental evaluation that wasw performed demonstrated the potential of even simpler, shallow Machine Learning techniques in solving the NILM problem. This result is particularly important because these techniques are immediately applicable in real metering systems, where the volume of data and the speed at which it is generated make more complex solutions practically impossible.

Acknowledgments. This work has been co-financed by the European Regional Development Fund of the European Union and Greek National funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH–CREATE–INNOVATE (project code: T2EΔK-00127). This work has also received funding by the European Union for the project REMARKABLE (project code: GA101086387). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Anderson, K.D., Ocneanu, A., Carlson, D.R., Rowe, A.G., Berges, M.E.: Blued : A fully labeled public dataset for event-based non-intrusive load monitoring research (2012), https://api.semanticscholar.org/CorpusID:25397318
2. Angelis, G.F., Timplalexis, C., Krinidis, S., Ioannidis, D., Tzovaras, D.: Nilm applications: Literature review of learning approaches, recent developments and challenges. Energy and Buildings **261**, 111951 (02 2022). https://doi.org/10.1016/j.enbuild.2022.111951
3. Baranski, M., Voss, J.: Genetic algorithm for pattern detection in nialm systems. In: 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583). vol. 4, pp. 3462–3468 vol.4 (Oct 2004). https://doi.org/10.1109/ICSMC.2004.1400878
4. Batra, N., Kelly, J., Parson, O., Dutta, H., Knottenbelt, W., Rogers, A., Singh, A., Srivastava, M.: Nilmtk: an open source toolkit for non-intrusive load monitoring. In: Proceedings of the 5th International Conference on Future Energy Systems. p. 265–276. e-Energy '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2602044.2602051, https://doi.org/10.1145/2602044.2602051
5. Beckel, C., Kleiminger, W., Cicchetti, R., Staake, T., Santini, S.: The eco data set and the performance of non-intrusive load monitoring algorithms. In: Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings.

p. 80–89. BuildSys '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2674061.2674064, https://doi.org/10.1145/2674061.2674064

6. Bhotto, M.Z.A., Makonin, S., Bajić, I.V.: Load disaggregation based on aided linear integer programming. IEEE Transactions on Circuits and Systems II: Express Briefs **64**(7), 792–796 (July 2017). https://doi.org/10.1109/TCSII.2016.2603479
7. Bradbury, S., Carpizo, B., Getnzel, M., Horah, D., Thibert, J.: Digitally enabled reliability: Beyond predictive maintenance. Company Report (10 2018)
8. Chen, Z., Chen, J., Xu, X., Peng, S., Xiao, J., Qiao, H.: Non-intrusive load monitoring based on feature extraction of change-point and xgboost classifier. In: 2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2). pp. 2652–2656 (Oct 2020). https://doi.org/10.1109/EI250167.2020.9347014
9. Darby, S.: The effectiveness of feedback on energy consumption. A Review for DEFRA of the Literature on Metering, Billing and direct Displays **486** (01 2006)
10. De Baets, L., Ruyssinck, J., Develder, C., Dhaene, T., Deschrijver, D.: On the bayesian optimization and robustness of event detection methods in nilm. Energy and Buildings **145**, 57–66 (2017). https://doi.org/https://doi.org/10.1016/j.enbuild.2017.03.061, https://www.sciencedirect.com/science/article/pii/S037877881631605X
11. Faustine, A., Mvungi, N.H., Kaijage, S., Michael, K.: A survey on non-intrusive load monitoring methodies and techniques for energy disaggregation problem (2017), https://arxiv.org/abs/1703.00785
12. Hart, G.: Nonintrusive appliance load monitoring. Proceedings of the IEEE **80**(12), 1870–1891 (Dec 1992). https://doi.org/10.1109/5.192069
13. Jain, A.K., Ahmed, S.S., Sundaramoorthy, P., Thiruvengadam, R., Vijayaraghavan, V.: Current peak based device classification in nilm on a low-cost embedded platform using extra-trees. In: 2017 IEEE MIT Undergraduate Research Technology Conference (URTC). pp. 1–4 (Nov 2017). https://doi.org/10.1109/URTC.2017.8284200
14. Jiang, L., Li, J., Luo, S., West, S., Platt, G.: Power load event detection and classification based on edge symbol analysis and support vector machine. Applied Computational Intelligence and Soft Computing **2012**(1), 742461 (2012). https://doi.org/https://doi.org/10.1155/2012/742461, https://onlinelibrary.wiley.com/doi/abs/10.1155/2012/742461
15. Kelly, J., Knottenbelt, W.: Neural nilm: Deep neural networks applied to energy disaggregation. In: Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments. BuildSys '15, ACM (Nov 2015). https://doi.org/10.1145/2821650.2821672, http://dx.doi.org/10.1145/2821650.2821672
16. Kelly, J., Knottenbelt, W.: The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes. Scientific Data **2**(1), 150007 (Mar 2015). https://doi.org/10.1038/sdata.2015.7, https://doi.org/10.1038/sdata.2015.7
17. Kim, H., Marwah, M., Arlitt, M., Lyon, G., Han, J.: Unsupervised Disaggregation of Low Frequency Power Measurements, pp. 747–758 (2011). https://doi.org/10.1137/1.9781611972818.64, https://epubs.siam.org/doi/abs/10.1137/1.9781611972818.64
18. Kolter, J., Johnson, M.: Redd: A public data set for energy disaggregation research. Artif. Intell. **25** (01 2011)
19. Meier, A., Cautley, D.: Practical limits to the use of non-intrusive load monitoring in commercial buildings. Energy and Buildings **251**, 111308 (2021)

20. Parson, O., Fisher, G., Hersey, A., Batra, N., Kelly, J., Singh, A., Knottenbelt, W., Rogers, A.: Dataport and nilmtk: A building data set designed for non-intrusive load monitoring. In: 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP). pp. 210–214 (Dec 2015). https://doi.org/10.1109/GlobalSIP.2015.7418187
21. Parson, O., Ghosh, S., Weal, M., Rogers, A.: Non-intrusive load monitoring using prior models of general appliance types. Proceedings of the AAAI Conference on Artificial Intelligence **26**(1), 356–362 (Sep 2021). https://doi.org/10.1609/aaai.v26i1.8162, https://ojs.aaai.org/index.php/AAAI/article/view/8162
22. Precioso, D., Gómez-Ullate, D.: Thresholding methods in non-intrusive load monitoring. The Journal of Supercomputing **79**(13), 14039–14062 (Apr 2023). https://doi.org/10.1007/s11227-023-05149-8, http://dx.doi.org/10.1007/s11227-023-05149-8
23. Puente, C., Palacios, R., González-Arechavala, Y., Sánchez-Úbeda, E.F.: Non-intrusive load monitoring (nilm) for energy disaggregation using soft computing techniques. Energies **13**(12) (2020). https://doi.org/10.3390/en13123117, https://www.mdpi.com/1996-1073/13/12/3117
24. Revuelta Herrero, J., Lozano Murciego, A., Barriuso, A., Hernandez de la Iglesia, D., Villarrubia, G., Corchado Rodríguez, J., Carreira, R.: Non intrusive load monitoring (nilm): A state of the art. pp. 125–138 (06 2018). https://doi.org/10.1007/978-3-319-61578-3_12
25. Shin, C., Lee, E., Han, J., Yim, J., Rhee, W., Lee, H.: The enertalk dataset, 15 hz electricity consumption data from 22 houses in korea. Scientific Data **6**(1), 193 (Oct 2019). https://doi.org/10.1038/s41597-019-0212-5, https://doi.org/10.1038/s41597-019-0212-5
26. Srinivasan, D., Ng, W., Liew, A.: Neural-network-based signature recognition for harmonic source identification. IEEE Transactions on Power Delivery **21**(1), 398–405 (Jan 2006). https://doi.org/10.1109/TPWRD.2005.852370
27. Verma, A., Anwar, A., Mahmud, M.A.P., Ahmed, M., Kouzani, A.: A comprehensive review on the nilm algorithms for energy disaggregation (2021), https://arxiv.org/abs/2102.12578
28. Zhang, C., Zhong, M., Wang, Z., Goddard, N., Sutton, C.: Sequence-to-point learning with neural networks for nonintrusive load monitoring (2017), https://arxiv.org/abs/1612.09106
29. Zhao, B., Stankovic, L., Stankovic, V.: Blind non-intrusive appliance load monitoring using graph-based signal processing. In: 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP). pp. 68–72 (Dec 2015). https://doi.org/10.1109/GlobalSIP.2015.7418158