SVM-Based Segmentation of Home Appliance Energy Measurements

Marc Wenninger, Dominik Stecher, Jochen Schmidt
Department of Computer Science, Rosenheim Technical University of Applied Sciences, Germany
Email: marc.wenninger@th-rosenheim.de

***Abstract*—Generating a more detailed understanding of domestic electricity demand is a major topic for energy suppliers and householders in times of climate change. Over the years there have been many studies on consumption feedback systems to inform householders, disaggregation algorithms for Non-Intrusive-Load-Monitoring (NILM), Real-Time-Pricing (RTP) to promote supply aware behavior through monetary incentives and appliance usage prediction algorithms. While these studies are vital steps towards energy awareness, one of the most fundamental challenges has not yet been tackled: Automated detection of start and stop of usage cycles of household appliances. We argue that most research efforts in this area will benefit from a reliable segmentation method to provide accurate usage information. We propose a SVM-based segmentation method for home appliances such as dishwashers and washing machines. The method is evaluated using manually annotated electricity measurements of five different appliances recorded over two years in multiple households.**

I. Introduction

In a time where the global electricity demand is constantly increasing although the electricity consumed by each device is decreasing, the electricity usage attribution becomes more and more difficult. There have been many studies that investigate the impact of electricity information systems that provide feedback on the consumption behavior to the householders. Some studies claim that information systems help to decrease consumption between 5% and 15% [1]. The authors of these studies mostly relate the reduction to the increased awareness of the actual consumption leading to a consumption cut. This is supported by studies on Real-Time-Pricing (RTP) tariffs that intend to provide an incentive to shift load to different time periods, but users respond with reducing their overall electricity consumption and simply do not use the device instead of using it at a different time [2]. RTP tariffs are pricing models where the electricity price varies over time. A field study by [3] monitored the habits regarding resource management of 15 households over a period of 3 months concludes that householders desire visibility of their behavior on resource usage, especially in real-time. Over the years, in domestic households, there has been an increasing effort to enable power suppliers as well as the users to gain more insight into a household's electricity usage pattern by installing smart-meters that are able to collect electricity usage data on a much higher frequency than annual manual readings. The installation of smart-meters will, in a first step only, allow more frequent usage information of the full household to be recorded but by it self will not help the householders to identify energy intensive devices and the related behaviors. This monitoring is known as Non-Intrusive-Load-Monitoring (NILM) in contrast to intrusive load monitoring (ILM), where each device or socket in the household is measured.

A. Disaggreation

In order to enable NILM to provide the detailed information of intrusive monitoring, researchers have proposed algorithms to disaggregate the load into the individual loads. For example, [4] use a Gaussian distribution to model each application and further use a clustering algorithm to identify the number of devices in use; [5] implement disaggregation using sparse coding. While the problem of comparability of such publications has been addressed by [6], demanding the community to use more robust evaluation metrics such as F1-score or RMSE, we find that disaggregation is trying to solve the most important goal of them all while neglecting some fundamental task required to provide accurate information to householders, or even further, enabling researchers to develop such algorithms.

B. Load Patterns

Most devices in a household that are interesting for customer feedback are those which require user interaction to run, such as dishwashers or washing machines. These devices have a multi-pattern load characteristic, where each state of the washing process has a very characteristic load pattern. A multi-stage pattern load can therefore be described by a composition of patterns such as the washing start-up, heating or spinning: Figure 1. Single-pattern devices, such as a refrigerator, have a continuous, nearly binary load pattern, where during compression there is a nearly constant power draw and otherwise a near zero power draw (see Figure 2). In both cases the load curve characteristic from cycle to cycle is very similar and therefore the energy demand as well. Being able to describe the loads by a composition of patterns, segmenting start-stop events may be sufficient information to determine the energy load. This also enables energy disaggregation to reduce the complexity of identifying the start-stop event, or even just some pattern within a cycle and derive the full usage from previously learned patterns.

C. Usage and Load Prediction

Another field of active research is the prediction of device usages and the related energy demand. [7] build a non-homogeneous Markov Chain to model end-use energy profiles

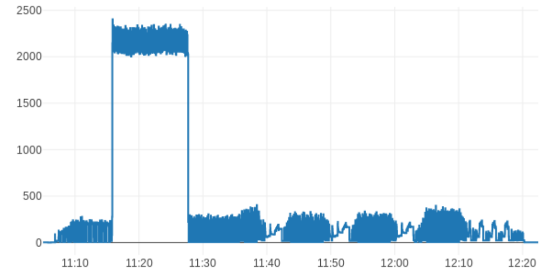

Fig. 1. Multi-pattern energy load of a washing machine

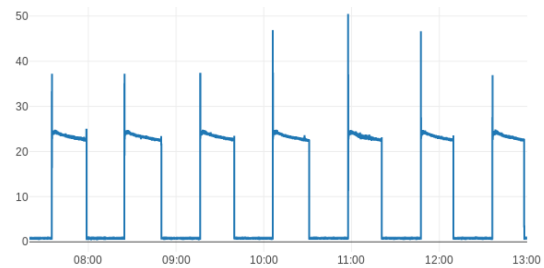

Fig. 2. Single-pattern energy load of a fridge

on appliance level, [8] propose a daily pattern-based probability model, [9] describe a method using decision trees and Bayesian networks, [10] also use a bayesian network, and [11] evaluate the performance of many classifiers such as bayesian networks and decision trees. In [12] a probabilistic model is described that combines three probabilities: The time of the day, the elapsed time since the appliance was last used and a factor for unusual inactivity.

In order to identify the usage of such devices in the first place, we need to be able to segment the cycles of such devices, meaning we need to identify start and stop timestamps in the measurement time-series. This fundamental task has so far not been addressed by many researchers and was usually tried to solve by using a simple thresholding algorithm. Thresholding is done by defining a device as started in case a certain load threshold is reached and stopped in case it drops below the threshold. As shown in Figure 1 and 2, estimating such a threshold needs to be done for each device individually and might fall short in case a device drops below the threshold during a usage cycle.

We are not aware of a published household energy dataset that contains start-stop annotations, hence the evaluation of such prediction algorithms rely on unverified heuristics such as thresholding for (semi-)unsupervised segmentation. Manual annotation is only feasible for very small datasets such as the widely used REDD [13], which only has recordings up to 19 days. Assuming a device was used once per day, the dataset only provides 19 cycles which are not enough samples for an evaluation.

D. Data Annotation

We therefore conclude that the area of appliance usage prediction, energy demand predictions and disaggregation will benefit heavily from a robust and reliable method to annotate and segment energy data. The annotation can also directly be used for householder feedback systems since it enables new statistics on usage behavior. [14] describe an annotation system where they identify load patterns using a DBSCAN algorithm to cluster different power states to overcome manual threshold parameter tuning. They also describe a feedback loop to end-users using a smartphone-app and push notifications and ask the user to validate segmentation and gather data to retrain. [15] describes a collaborative framework in form of a web-based annotation system. The platform intends to crowdsource the manual annotation task using gamification techniques to encourage the users to contribute.

In this publication we present a method to identify the start and stop event of such devices automatically using supervised machine learning.

II. Method

We are aiming for a method that can rely on a few annotation samples, hence we chose to incorporate Support-Vector-Machines (SVM) to classify sliding windows of the time-series. Using a sliding window helps to overcome the fact that load measurements are a continuous time series that contain too much data to be classified all at once, as well as achieving the required default series length. The use of a sliding window will also help to further develop such methods into a streaming method, where just recorded data is classified on-the-fly. Our method is a classic classification pipeline, separated into multiple stages: data annotation, preprocessing and classification.

A. Data Annotation

In order to train an SVM we need annotated training and test data. We therefore manually annotated some of your appliance measurements, saving the start and end timestamp. This was done using a custom web application, allowing multiple people to participate in the data annotation. In total, we created 777 manual annotations for 5 appliances as shown in Table I.

B. Data Preprocessing

A sliding window is used to convert this continuous time series to individual feature vectors. To avoid near-identical feature vectors for large window sizes and reduce training time, the step parameter was introduced, defining the number of seconds the window is shifted each time. Each sliding window of the data is preprocessed using a discrete wavelet transformation (DWT). We only keep the approximate wavelet coefficients in order to obtain a low-pass representation of the signal. We found that using the Daubechies wavelet with 30 coefficients (db30) leads to the most robust results. The wavelet transformation was done using the PyWavelets[1] python package.

[1] https://github.com/PyWavelets/pywt

C. Classification

For devices such as a washing machine, the start and end timestamps are very rare events. The maximum number of usages per day is given by the usage cycle length, meaning if a cycle takes 3 hours, the maximum number of start events is given by $24h/3h = 8$. Measuring data at 1Hz will lead to 86400 measuring points per day, meaning the events looked for are at maximum 8 timestamps out of 86400, thus extremely rare. Since we are using a sliding window, we define the classification task to identify whether a given window contains a start or stop event. We also experimented changing the labels so that the event must be within a defined area of the window to avoid having too little characteristic of the event within the window. Since it did not lead to a significant improvement we abandoned this hypothesis.

SVMs use a hyperplane to linearly separate the classes in a dataset. The hyperplane not only separates two classes, it also maximizes the distance between the hyperplane and each class, making SVMs a large margin classifier [16]. The hyperplane is thus defined by the closest data points from each class, which in turn become the support vectors. The benefit of using an SVM over other classifiers such as Neural Networks is, that it is not so easily prone to a sampling selection bias as it does class separation through a hyperplane instead of class-conditional probabilities. Because the problem is most probably not linearly separable, we use a non-linear polynomial kernel function to transform the data into a higher-dimensional space. SVMs support classification of more than two classes or binary problems called multi-class SVM. It is important to note that SVMs are inherently capable of separating only two classes since a single hyperplane cannot separate more than two classes, thus they have to be implemented using workarounds such as multiple one-vs-rest classifiers. We decided to not incorporate multi-class SVMs, but to train two completely independent SVMs in order to classify start and stop events as simple heuristics can be used to connect the two events.

D. Extracting Exact Timestamp

Using a sliding window of e.g. 512 seconds with a single label per window will lead to an approximated event where we only know that the event lies within the window. The larger the window, the less exact the event classification. Since we are classifying successive sliding windows, we can calculate the number of windows the event is present in using:

$$eventWindowCount = windowLength/stepSize$$

In case all windows are classified correctly, the event is the last value of the first window and the first value of the last window. This allows for improvements of the classification by first calculating a probability on the found event by comparing the number of positive labeled successive windows to the expected window count. Second, we can estimate the exact timestamp by averaging the last value of the first window and the first value of the last window. In case all windows were classified correctly, we are able to calculate the exact second the event occurred, thus overcoming the problem of increasing window size.

TABLE I
APPLIANCES MANUAL USAGE ANNOTATIONS

ID	Type	Usage Count	Duration	Avg. Usage
#1	Washing Machine	82	259 days	1h 27m
#2	Washing Machine	100	204 days	2h 30m
#3	Dishwasher	143	515 days	2h 23m
#4	Dishwasher	224	843 days	2h 57m
#5	Dishwasher	228	367 days	1h 27m

III. Data

The data used for evaluation was collected in private households in Germany between 2016 and 2018. Electric power is measured at a rate of 1Hz using commercial plugs placed between the outlet and appliance that can be accessed using a local WiFi network. The measurements are collected in each home using a Raspberry Pi running hypriot OS[2], an operating system designed to run Docker containers. An Eclipse Smarthome[3] instance with a custom developed binding to communicate with the plugs is used to collect and persist the data in a PostgreSQL database. Locally persisted data is uploaded hourly to a central database via the internet using a secure SSH connection. In order to gain ground truth for training and evaluation, we manually annotated 5 of the collected devices (see Table I) using a custom developed web application. The annotated appliances are normal consumer washing machines and dishwashers, as these have a more complex multi-pattern load compared to e.g. fridges as described in Section I.

IV. Experiments

A. Data Preprocessing

In our experiment, we evaluated the method using window sizes of (32, 64, 128, 256, 512) seconds, a step size of 5 seconds and the db30 wavelets transformation. The classifier is based on an SVM implemented in Python using the Scikit-learn toolkit[4]. The Radial Basis Function (RBF) and the Polynomial Kernel (PK) both produced equally good results with the PK offering more parameters for fine-tuning.

The best performing SVM configuration for our dataset was using a 5 degree polynomial kernel with a coefficient of 10. The detailed results can be found in Table II. Changes in step size did not affect the results significantly and are therefore not listed.

B. Test Procedure and Performance Measures

As the data is highly unbalanced with 2-3% of the total data being events, accuracy is not a suitable metric. Thus the performance of the classifier was primarily evaluated using F1-score. For testing, three dishwashers and two washing machines were used, as these had a significant amount of manual annotations available. Using these, we only extracted

[2]https://blog.hypriot.com
[3]https://www.eclipse.org/smarthome/
[4]https://scikit-learn.org

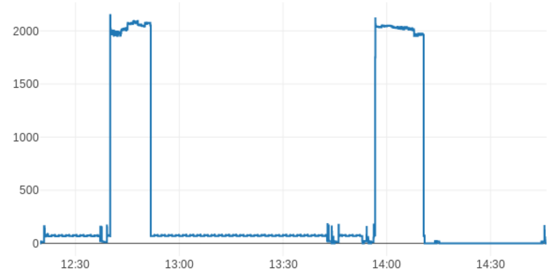

Fig. 3. Washing cycle of appliance #3

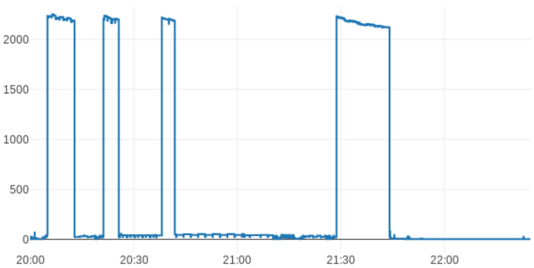

Fig. 4. Washing cycle of appliance #4

data from the given annotation adding 5 minutes before and after a cycle. As the power consumption in between usages is zero, this helped to increase the number of positive feature vectors. The converted datasets are then split into 60% training and 40% test data.

C. Results

The most obvious finding is, that the classification improves with increasing window size, which can be attributed to the fact that a larger window size also provides more context for the classifier, further improving results at the cost of a longer delay in real-time applications. The overall classification results for the dishwashers (#3, #4, #5) are promising. For appliance #3 and #4 the classification of start events using a window size of 512 are with a F1-score of 0.98 and 0.95 very good and the average time error is only 7.63 and 28.93 seconds. The stop event for appliance #3 provides a satisfying F1-score of 0.90, but on the other hand the stop event classification for appliance #4 has a F1-score of 0.0, thus completely failing the classification task. #5 performs well for both start and stop events. For start events and a large window size there is a very distinct value change, the heating phase, where the power consumption increases to 2kW. For stop events there is no such significant event. Examining the data shows that the stop event has, in general, a far less distinct pattern compared to the start event, and comparing appliance #3 and #4, the latter has only few recurring characteristics in between cycles and no distinct time between the last significant pattern und the actual stop event (see Figure 3 and 4). Appliance #3 has such a distinct recurring pattern and therefore the time error for stop events is 0.0 seconds, thus the exact second of each stop event can be found.

TABLE II
EXPERIMENTAL RESULTS ON APPLIANCES LISTED IN TABLE I SHOWING F1-SCORE, ERROR MATRIX AND AVERAGE TIME ERROR IN SECONDS.

ID	Windows-Size	Type	F1	TP	TN	FN	FP	Avg. Time Error
#1	32	start	0.0906	240	34431	2	4817	115.33
		stop	0.0529	196	32277	22	6995	178.04
	64	start	0.1396	451	33271	4	5554	103.79
		stop	0.1150	382	33016	41	5841	174.79
	128	start	0.3788	881	35091	28	2862	114.38
		stop	0.2676	765	33909	63	4125	144.82
	256	start	0.7243	1609	35188	206	1019	107.97
		stop	0.5740	1440	34445	134	2003	127.55
	512	start	**0.7736**	1782	33516	415	628	112.52
		stop	**0.8474**	1649	34098	206	388	95.83
#2	32	start	0.0286	284	58980	66	19235	96.26
		stop	0.0104	92	60945	17	17511	167.49
	64	start	0.0762	601	63143	72	14494	97.81
		stop	0.0197	173	60960	39	17138	182.61
	128	start	0.2550	1187	69675	147	6787	64.62
		stop	0.0379	296	62487	113	14900	173.41
	256	start	0.5192	2192	70524	395	3664	61.66
		stop	0.0606	415	63488	294	12578	106.27
	512	start	**0.5207**	1569	70268	1531	1357	132.78
		stop	**0.0498**	236	65488	595	8406	115.46
#3	32	start	0.1766	398	100903	1	3710	24.09
		stop	0.2105	161	103643	204	1004	22.57
	64	start	0.2996	734	100482	7	3425	6.21
		stop	0.3404	523	102098	208	1819	29.51
	128	start	0.1137	1478	79390	4	23041	3.42
		stop	0.5260	1230	100466	215	2002	28.09
	256	start	0.8336	2941	98336	23	1151	4.65
		stop	0.6232	2633	96634	237	2947	27.83
	512	start	**0.9863**	3424	96002	53	42	7.63
		stop	**0.9018**	3256	95556	69	640	0.00
#4	32	start	0.1347	641	194266	10	8227	89.56
		stop	0.0014	42	142072	0	61030	138.75
	64	start	0.4060	1176	197953	33	3408	80.82
		stop	0.0025	73	144499	0	57998	129.44
	128	start	0.4711	2365	193746	53	5257	91.38
		stop	0.0046	126	147267	0	54028	75.43
	256	start	0.7815	4720	191773	99	2540	99.22
		stop	0.0073	188	147790	0	51154	45.58
	512	start	**0.9590**	5525	188546	87	386	28.93
		stop	**0.0063**	148	147770	0	46626	0.00
#5	32	start	0.4054	540	106542	82	1502	72.69
		stop	0.0577	572	89418	0	18676	207.18
	64	start	0.5197	1034	105137	123	1788	131.29
		stop	0.1196	1144	90095	7	16836	189.82
	128	start	0.3209	2246	95161	68	9440	108.71
		stop	0.2729	2283	92467	0	12165	169.70
	256	start	0.4782	4550	90098	76	9855	62.51
		stop	0.6539	4567	95177	0	4835	43.56
	512	start	**0.8401**	5334	92546	96	1935	54.10
		stop	**0.8889**	5318	93263	1	1329	0.00

The results for the washing machine #1 are also promising with an F1-score of 0.77 and 0.84 for the start and stop event. For appliance #2 the results are poor and the classification task completely fails for the stop events.

Overall it can be said that the classification works well for start events and only for some appliances on stop events. The most significant part being that the start event provides much more distinct features.

V. CONCLUSION

The segmentation of energy loads is a not yet solved problem and the proposed method based on supervised segmentation using an SVM shows promising results. While the method performs very well for the segmentation of the start events, the stop events very much depend on a rich pattern at the end of the cycle. The results also show that there needs to be a sufficient amount of pattern present within a window in order to classify correctly. Even on large window sizes the method is able to identify the exact second of an event. The cost of a large window size only plays a vital role in a real-time scenario as the device would have to run at least for the time of the window-size in order to identify the event. In terms of annotating datasets for further research this aspect is negligible. While the method does not incorporate any heuristics to further improve the results, especially for the off event these may just make the difference as a stop event always has a start event before and a cycle usually will only take a certain time, narrowing down the part of a signal where a stop event may occur.

ACKNOWLEDGEMENT

This work is supported by a research grant of the Bayerische Wissenschaftsforum (BayWISS).

REFERENCES

[1] T. Hargreaves, M. Nye, and J. Burgess, "Making energy visible: A qualitative field study of how householders interact with feedback from smart energy monitors," *Energy Policy*, vol. 38, no. 10, pp. 6111 – 6119, 2010, the socio-economic transition towards a hydrogen economy - findings from European research, with regular papers. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S030142151000460X

[2] J. Schleich and M. Klobasa, "How much shift in demand? Findings from a field experiment in Germany," in *Rethink, renew, restart. ECEEE 2013 Summer Study. Proc.*, T. Lindström, Ed. European Council for an Energy-Efficient Economy, 2013, pp. 1919–1925.

[3] M. Chetty, D. Tran, and R. E. Grinter, "Getting to green: Understanding resource consumption in the home," in *Proceedings of the 10th International Conference on Ubiquitous Computing*, ser. UbiComp '08. New York, NY, USA: ACM, 2008, pp. 242–251. [Online]. Available: http://doi.acm.org/10.1145/1409635.1409668

[4] T. Liu, X. Ding, and N. Gu, "A generic energy disaggregation approach: What and when electrical appliances are used," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, Nov 2015, pp. 389–397.

[5] J. Z. Kolter, S. Batra, and A. Y. Ng, "Energy disaggregation via discriminative sparse coding," in *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'10. USA: Curran Associates Inc., 2010, pp. 1153–1161. [Online]. Available: http://dl.acm.org/citation.cfm?id=2997189.2997318

[6] S. Makonin and F. Popowich, "Nonintrusive load monitoring (nilm) performance evaluation," *Energy Efficiency*, vol. 8, pp. 809–814, 12 2014.

[7] Z. Kang, M. Jin, and C. J. Spanos, "Modeling of end-use energy profile: An appliance-data-driven stochastic approach," in *The 40th Annual Conf. of the IEEE Industrial Electronics Society, Dallas, TX, USA*, 2014, pp. 5382–5388.

[8] C. Chang, P.-A. Verhaegen, J. R. Duflou, M. M. Drugan, and A. Nowe, "Finding days-of-week representation for intelligent machine usage profiling," *Journ. of Industrial and Intelligent Information*, vol. 1, no. 3, pp. 148–154, 2013.

[9] N. Arghira, L. Hawarah, S. Ploix, and M. Jacomino, "Prediction of appliances energy use in smart homes," *Energy*, vol. 48, no. 1, pp. 128–134, 2012, 6th SDEWES Dubrovnik Conference SDEWES 2011.

[10] L. Hawarah, S. Ploix, and M. Jacomino, "User behavior prediction in energy consumption in housing using bayesian networks," in *Proc. of the 10th ICAISC*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 372–379. [Online]. Available: http://dl.acm.org/citation.cfm?id=1894214.1894263

[11] K. Basu, L. Hawarah, N. Arghira, H. Joumaa, and S. Ploix, "A prediction system for home appliance usage," *Energy and Buildings*, vol. 67, pp. 668 – 679, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378778813000789

[12] M. Wenninger, J. Schmidt, and T. Goeller, "Appliance Usage Prediction for the Smart Home with an Application to Energy Demand Side Management – And Why Accuracy is not a Good Performance Metric for this Problem," in *6th Int. Conf. on Smart Cities and Green ICT Systems (SMARTGREENS)*, Porto, Portugal, 2017, pp. 143–150.

[13] Z. Kolter and M. J. Johnson, "REDD: A public data set for energy disaggregation research," in *Proc. of SustKDD*, 2011.

[14] S. Rollins, N. Banerjee, L. Choudhury, and D. Lachut, "A system for collecting activity annotations for home energy management," *Pervasive and Mobile Computing*, vol. 15, 06 2014.

[15] H.-A. Sandlin, T. Kurniawan Wijaya, K. Aberer, and N. Nunes, "A collaborative framework for annotating energy datasets," 11 2015.

[16] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ser. COLT '92. New York, NY, USA: ACM, 1992, pp. 144–152. [Online]. Available: http://doi.acm.org/10.1145/130385.130401