Exploiting Breadth in Energy Datasets for Automated Device Identification

Sean Barker, Kyle Morrison, and Tucker Williams
Bowdoin College
Brunswick, ME, USA

***Abstract*—The recent explosion of interest in smart building energy-efficiency has led to a proliferation of public energy datasets. Most of these datasets focus on depth (i.e., many devices in a few buildings) as opposed to breadth (e.g., a few devices in many buildings), and thus most smart building algorithms are evaluated on depth-oriented datasets. We argue that increasing data breadth conveys important benefits that are not easily achieved by even a large quantity of deep data. As an illustrative case study, we consider the problem of classifying previously unseen appliances using an off-the-shelf classifier trained on known instances of other devices. Our experiments on multiple real-world datasets (both depth- and breadth-oriented) demonstrate significant and sustained benefits from increased data breadth, and point to the importance of incorporating greater breadth into similar techniques that rely on generalized electrical load models.**

I. Introduction

Nearly all techniques for smart and energy-efficient buildings rely on high-quality data for experimentation and validation. Over the past several years, the research community has responded vigorously to this need through the release of real-world energy datasets. Today there are over 15 such datasets publicly available [1], most of which are collected from residential homes (though some are also collected from larger buildings, such as university dormitories).

A typical energy data collection deployment consists of a small set of homes in which sensing and data collection is ubiquitous. A representative example of this approach is the REDD dataset [2], which monitored data from a wide range of electrical appliances within six residential homes. Other well-known examples of such datasets include UK-DALE [3] and BLUED [4]. These datasets and many others share the same core characteristics: a small number of homes and a high degree of instrumentation. We refer to such datasets as *deep* datasets. Deep datasets provide detailed but inherently narrow profiles of energy usage – i.e., the consumption patterns of a relatively small set of specific devices as used by a particular group of building occupants.

An alternate approach is a dataset collected from a large number of homes, but possibly with a lesser degree of instrumentation (e.g., monitoring a few large appliances from 100 homes). We refer to such datasets as *broad* datasets. A prominent example of a broad dataset is Dataport [5], which contains appliance-level data from over 1000 homes. While broad datasets may provide a less complete picture of any particular home or group of occupants, they are also less subject to the idiosyncrasies of the specific devices or users under observation. Unfortunately, broader data also generally comes with compromises, such as limited data resolution (e.g., 1/60 Hz in Dataport) or a lack of device-specific data. Owing to such limitations, as well as to the logistical challenges of collecting broad, real-world data [6], depth-oriented datasets remain the default for most researchers.

A notable limitation of deep datasets, however, is that while most common types of devices may be represented, there are generally few instances of any particular type. For example, while a typical home is likely to have all the usual appliances – refrigerator, dishwasher, and so forth – it is not likely to have more than one or two of each. As a result, while a typical deep dataset may contain hundreds of monitored devices from a few homes, there may be little variation among specific device types within the data. This implicit homogeneity may be problematic for smart building applications that rely on generalized models of devices. Algorithms for varied problems such as identification, forecasting, and disaggregation are often developed using depth-oriented datasets, but are designed to be effective in more general environments. To ensure generality, it is often necessary to assume implicitly that devices are representative of their respective types – e.g., one refrigerator behaves mostly like another, and thus good performance on a few refrigerators is indicative of good performance more broadly. However, there are multiple reasons this assumption might not hold, including unpredictable activity patterns of "smart" appliances, unusual usage patterns by owners, or simple variations between manufacturers. A simple example is shown in Figure 1, which depicts the power consumption of two refrigerators. While there are many similarities in their usage profiles (e.g., similar power consumption, cyclic behavior, and "inrush" current at the start of each cycle), their actual cycle patterns are quite different, with the former displaying an oscillating pattern while the latter displays a flatter, more consistent pattern. Broader datasets include more of these device variations, but minimal attention has been given to whether these variations are significant and how much breadth is needed to account for most 'typical' variations.

As an exploratory case study of this question, we consider the problem of automatically identifying electrical devices based on their consumption. Automatic device identification is a well-studied problem, particularly within the context of

978-1-5386-8099-5/19/$31.00 ©2019 IEEE

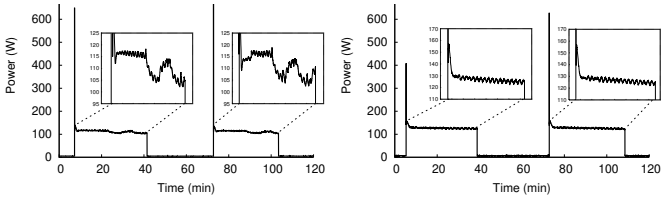

Fig. 1. The power usage of two refrigerators (left and right), demonstrating variations in cycle behavior.

non-intrusive load monitoring (NILM) [7]–[9], which aims to disaggregate whole-house usage into individual devices. We focus on the simpler problem of identifying single devices using device-level data (e.g., from a smart outlet), which has been termed *non-intrusive load identification* (NILI) [10]. Unlike in prior work, however, we explicitly consider breadth by measuring performance on devices not previously observed during training. For example, the system might be trained using a particular LG refrigerator, but then used to identify a different Maytag refrigerator. Ideally, such an identification system should be able to identify both devices using a general model of a refrigerator, without respect to any particular refrigerator within the source dataset.

We conduct experiments using an off-the-shelf classifier and two public datasets: Tracebase [11] and Dataport [5]. We find that the classifier improves significantly as the number of devices within a given device class (i.e., exploiting data breadth) is increased, including (and especially) beyond the size of most popular datasets. Our results point to the potential of applying broader datasets to many types of energy analytics.

II. Related Work

Prior work on outlet-level device identification has considered a variety of machine learning based approaches, but most either requires user-driven training [12] or does not consider the unseen device problem in detail (e.g., [10], [13], [14]). For example, [12] requires the manual operation of a device during an explicit training phase. Support vector machines are used in [15] for automated identification using both 1 Hz data and higher resolutions, but even these resolutions are higher than in many existing datasets.

Much work relating to device identification exists within the space of non-intrusive load monitoring (NILM) research, which has been extensively studied (e.g., overviews in [7], [9]). Since the NILM disaggregation problem was originally investigated using simple edge detection techniques [8], newer techniques have included Hidden Markov Models [2], Viterbi's algorithm [16], and deep learning [17]. Most of this work centers on lower-resolution data that is typical of smart meters; more accurate identification has been accomplished using high-frequency data from specialized meters [18]. NILM studies have noted various difficulties in disaggregating unknown houses; e.g., low accuracy [2] or a lack of homes on which to train [17]. Other recent work [19] has highlighted the particular importance of broader datasets to further NILM. Here, we investigate the significance of data breadth from the perspective of unseen device identification, and further consider the tradeoff of breadth vs. data resolution that often exists in real-world datasets.

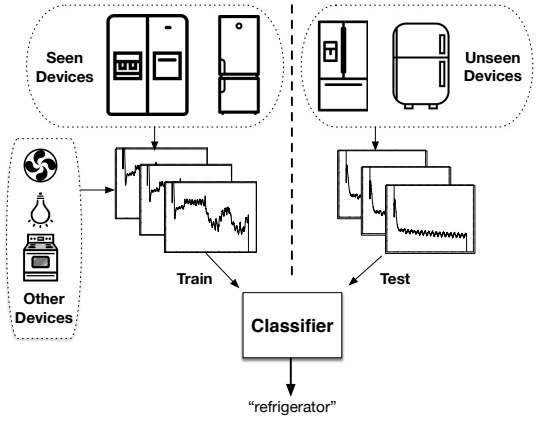

Fig. 2. Data flow in the unseen device classification problem.

III. Unseen Device Classification

We formulate the problem of identifying unseen devices as a timeseries classification problem. Specifically, consider a device connected to a smart outlet that measures its real energy consumption (e.g., once per second). For simplicity, we consider energy data in 24-hour chunks, and assume that the same device is connected (but not necessarily active) for the entire 24-hour period. Our goal is to automatically identify the category of the attached device (e.g, "dishwasher", "microwave", or similar) using only the energy timeseries data collected from the outlet, even if we are not able to identify the device more specifically (e.g., a specific manufacturer and model) due to never having encountered that particular device.

A straightforward approach is to take historical data from devices with known (labeled) categories, split the data into 24-hour segments, and then build a classifier to label new timeseries. Such a system could be used to automatically label devices in a home without human intervention, and could also adapt to movements of devices from one outlet to another. This basic approach was demonstrated in [10] and shown to result in over 90% classification accuracy over 15 device categories using standard cross-validation.

In the problem of unseen device classification, however, we make a key distinction, and focus our attention on devices that are completely absent from training data – we call these *unseen* devices. For example, suppose that there are four microwaves labeled m_1 through m_4. We might only have data from m_1 and m_2 available during training (the *seen* devices), but would still like to be able to identify m_3 and m_4 (the *unseen* devices) using the trained classifier. This basic data flow is illustrated in Figure 2. Note that in addition to the seen devices, any number of devices of other classes may be present in training as well (e.g., ovens, lights, etc), but in contrast to standard cross-validation, here we are interested specifically in performance on the unseen devices within the category of interest. In effect,

other device categories exist solely to force the classifier to discriminate between the shared device category of the seen and unseen devices and all other device categories learned during training. Testing will then only be successful if the learned model truly generalizes across many devices (seen or unseen) within the category.

Identifying unseen devices is more difficult than the simpler identification problem (using cross-validation) for several reasons. First, multiple devices of the same device class might have different characteristics, such as several microwaves with different wattages or several dishwashers with different cycle behaviors. Second, owners of devices might also exhibit significantly different usage patterns. User behavior could even result in substantial differences among identical devices – for example, one owner of a washing machine might always use the default cycle settings, while another owner of the exact same washing machine might use the special cycle settings for different loads (which could result in more varied energy consumption). As another example, a freezer that is opened frequently would display a less regular compressor cycle than a freezer that is rarely opened (such as a basement freezer).

Despite these challenges, unseen device classification is a more realistic (and useful) formulation of the problem. In a real-world deployment of an automated identification system, a limited number of devices would be used to train a classifier, but a potentially limitless number of future devices could then be presented for identification. For a large or long-running deployment, the number of unseen devices would likely dwarf the number of seen devices. Hence, performance on unseen devices is arguably the most appropriate metric with which to judge the effectiveness of such a system.

Features. Classification is performed on features computed over each 24-hour chunk of raw timeseries data. Our current system uses a simple set of statistics-based features: average power and variance, maximum power, the percentage of time the device is active (defined as power consumption above a baseline threshold), and a set of multiple features capturing the magnitude of energy deltas. Each of these delta features is defined as the number of observed power deltas (i.e., steps) within a certain range of magnitudes. For example, the 10-20W delta feature would consist of the number of power steps of magnitude between 10W and 20W over the course of the 24-hour observation period. Due to the tendency of larger steps to be split across multiple measurements (since power changes may occur in the middle of a meter measurement interval), back-to-back power steps in the same direction are aggregated as a preprocessing step before computing delta features.

Importantly, we note that extensive feature engineering is not our primary goal here. Instead, we are interested primarily in the classifier's ability to generalize across a device class (i.e., exploiting data breadth) using simple features; we would expect a more sophisticated classifier to be able to generalize at least as well as our simple model.

Device Type	1 Hz #	1/60 Hz #	Tracebase	Dataport
Air Conditioner	18	50		✓
Coffeemaker	6	–	✓	
Dishwasher	12	50	✓	✓
Dryer	11	50		✓
Electric Car	4	50		✓
Furnace	8	50		✓
Lamp	9	–	✓	
Laptop	16	–	✓	
Lights	9	50		✓
Microwave	13	50	✓	✓
Monitor	12	–	✓	
Oven	7	50		✓
Refrigerator	19	50	✓	✓
TV	10	–	✓	
Washing Machine	16	50	✓	✓

TABLE I
DEVICE TYPES AND DEVICE COUNTS IN SECOND- AND MINUTE-LEVEL TEST DATASETS.

IV. EVALUATION

We implemented the approach described in Section III using the J48 decision tree classifier, which implements the C4.5 algorithm [20]. Intuitively, C4.5 builds a decision tree by repeatedly splitting the data using the attribute that most effectively discriminates between the remaining training instances (as measured by information gain). The classifier was trained and tested on data from the Tracebase [11] and Dataport [5] datasets. Tracebase is intended as a repository of device data and contains second-resolution (1 Hz) readings from roughly 10 instances of many major appliance types. Dataport is a much broader dataset containing minute-level readings from several hundred homes. Dataport has also recently added second-level data from a smaller subset of homes. For our experiments, we consider two separate collections of test data:

- **Minute-level dataset**: a contiguous month of minute-level data from Dataport.
- **Second-level dataset**: second-level data from 40 homes in Dataport taken over the same month, combined with second-level data from Tracebase for a greater variety of devices and larger device count.

In short, the minute-level dataset is quite broad, but with lower data quality than is typically found in most popular datasets. The second-level dataset is more representative of other datasets, but retains a reasonable degree of breadth due to the use of multiple datasets.

Each device type in the data (e.g., "microwave") consists of a number of device instances (e.g., m_1 and m_2) as well as a number of timeseries per device instance (each corresponding to one day of usage data where there was at least some activity). Importantly, we note that not every device type is present (or monitored) in every home; thus, the number of device instances of any given type is substantially lower in most cases than the number of homes present. Supplementing the second-level Dataport data with Tracebase devices partially

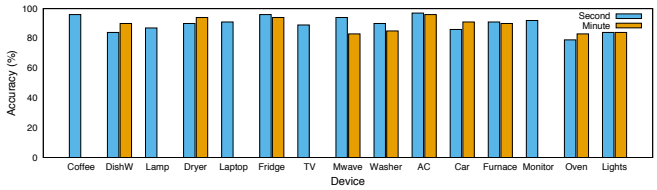

Fig. 3. Baseline classification accuracy on full datasets using 10-fold cross-validation.

counteracts this limitation. For the minute-level dataset, the much larger number of homes present in Dataport largely addresses this problem; here, we randomly select 50 devices out of the hundreds available for each device type. Table I lists the specific device types we consider, their source dataset(s), and how many device instances are considered within each dataset. Note that one consequence of our dataset construction is that the second-level dataset has 15 device types, whereas the minute-level dataset has only 10 distinct types due to the limited number of Tracebase device types.

A. Baseline Performance

To establish a baseline, we first train the classifier using each complete dataset (i.e., without excluding any devices) and test using standard 10-fold cross validation. This approach represents the conventional case where all devices are likely to be represented in the training data. Classification accuracy is shown in Figure 3, broken down by device class for both the second-level and minute-level dataset (devices with only one bar have only second-level data). Accuracy is high for all device classes in both datasets; the worst single result is 79% accuracy when identifying ovens in the second-level dataset. Note that while minute-level data is normally expected to result in lower accuracy relative to the second-level data, the smaller number of classes in the minute-level dataset sometimes results in modestly higher accuracy. Overall accuracy across all classes is 92% for the second-level dataset and 89% for the minute-level dataset. In short, when all devices are present during training, even a simple classifier such as ours is able to distinguish many device classes with high accuracy.

B. Unseen Device Performance

We now consider classification performance on devices not present during training. To do so, we partition the devices of a given type into *seen* and *unseen*, and exclude all data from unseen devices during training. We then evaluate the classifier *only* on data from unseen devices. This procedure measures the ability of the classifier to recognize generalized appliance types (e.g., "refrigerator") instead of any specific devices (e.g., a specific LG refrigerator model) present in the data.

Suppose we are considering a particular device class C (e.g., microwave) containing n_c distinct devices (i.e., the number of microwaves from which at least one day of known activity exists). To evaluate performance on class C, we first choose k devices of class C (where $k < n_c$) and designate those k devices as *seen*. The remaining $(n_c - k)$ devices of class C are designated as *unseen*. The classifier is then trained using the complete dataset *except for* these unseen devices. In short, the difference from the baseline classification experiment is that a set of unseen devices is designated, and then all timeseries produced by these devices are held back during training.

During testing, we present the classifier *only* with timeseries from the unseen devices (i.e., all presented timeseries are of class C) and define accuracy as the percentage of these timeseries that were correctly labeled as type C.

Note that n_c is effectively a measurement of the breadth of the dataset (the instance counts shown in Table I). The value of k (which is bounded by n_c) determines both the number of devices from which to generalize the class, as well as the number of devices with which to evaluate the generalized model. A depth-oriented dataset would typically require a small k, while a breadth-oriented dataset would support a much larger k.

Since we are interested in the impact of varying breadth, for every C and corresponding n_c we train and test the classifier for every possible k from 1 to $n_c - 1$. In the $k = 1$ case, one representative device is used and the classifier attempts to identify every other device of that class, while in the $k = n_c - 1$ case, every device of the class except for one is used to train the model, which is then tested on the sole holdout.

However, even for a given choice of k, the specific choice of which k devices are designated as seen is highly significant. For example, consider the case of $k = 1$, in which only one device is observed to learn the category. If the one chosen device happens to be poorly representative of the class as a whole (e.g., an unusually energy-efficient device or a device used in an atypical manner), then the resulting accuracy will likely suffer versus a more typical seen device used for training. To avoid unpredictably skewing results based on device variations, we repeat the training and testing process for every possible k-sized subset of devices. Thus, for every possible choice of C and k, we run a total of $\binom{n_c}{k}$ trials and average over the results. In the case of the minute-level data (where $n_c = 50$), running all possible trials is intractable; thus, for a given setting of k, we instead average over 100 independent trials with randomly chosen devices.

Classification accuracy on unseen devices for several settings of k is shown in Figures 4 and 5 for second-level and minute-level data, respectively. Figure 4 shows results for $k = 1$, $k = 3$, and $k = 0.75n_c$, reflecting the fact that n_c is both variable across classes and as low as 4 depending on the class. With a larger (and constant) setting of $n_c = 50$, Figure 4 shows results for $k = 1$, $k = 20$, and $k = 40$, leaving at least 10 unseen devices in all cases for testing.

Device-level accuracy rarely exceeds 80% in both cases, and is substantially less in both cases than their respective baselines shown in Figure 3 (which only once falls below 80%). Differences across classes are largely explained by the degree of variability within a class; e.g., air conditioners tend to be more consistent and predictable than refrigerators; hence, identification of unknown A/Cs is more accurate than unknown refrigerators both for small and large values of k.

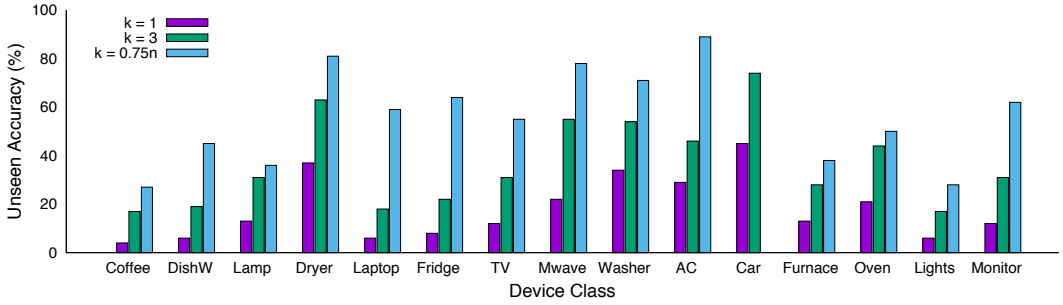

Fig. 4. Classification accuracy on unseen devices using second-level dataset.

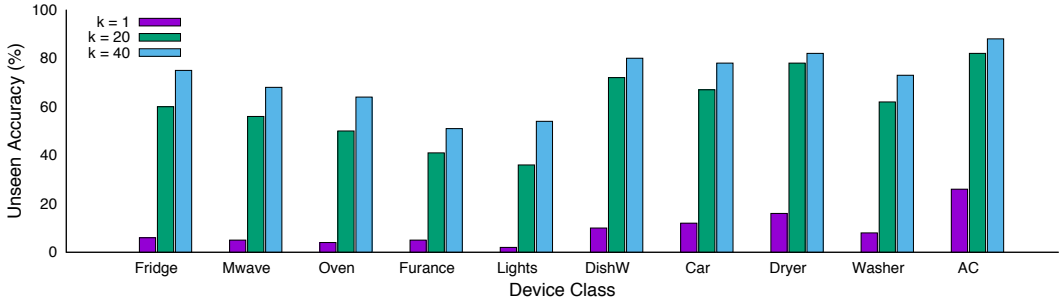

Fig. 5. Classification accuracy on unseen devices using minute-level dataset.

More significant, however, are the results across different values of k. Using only a single device for training ($k = 1$), performance is universally poor, indicating that a single device instance cannot reliably capture the behavior of the class. This effect is most pronounced in the case of the minute-level dataset, where a single seen device results in an average accuracy of only 9%. Increasing k, however, has a significant positive impact on accuracy in all cases. In the second-level case, most devices achieve at least a 2x increase in their accuracy through increasing breadth; in the minute-level case, the average improvement is nearly 8x.

It is fairly intuitive that increasing k beyond 1 leads to improvements; given that no two devices are likely to be perfectly alike, having more than one training device results in a more general (and accurate) model. Less obvious, however, is the continued improvement that we observe in Figure 5 when increasing k from an already sizable 20 to 40. One might expect a "leveling-off" as breadth is increased past a certain point once the model is as generalized as possible. However, while we do see diminishing returns as k increases further, notable gains continue beyond $k = 20$ as we move to $k = 40$. Some of these gains are substantial, e.g., a 50% increase in the case of lights. In short, even the considerable breadth exercised at the $k = 20$ level is not enough to fully generalize the model. Furthermore, we note that even $k = 20$ is an unattainable level of breadth for most energy datasets that are widely-used in the community.

Given the importance of the specific devices involved, we also consider the effect of varying the device instances for a given setting of k. Figure 6 shows the second-level classifier's

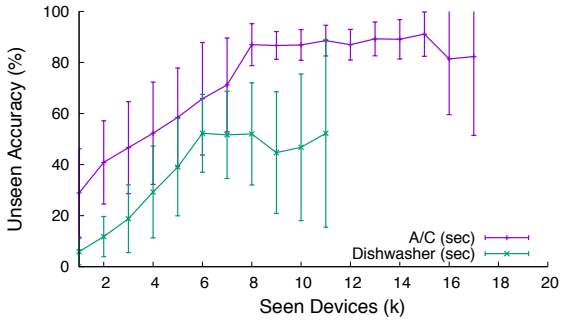

Fig. 6. Training or testing on only a few devices produces instability in either the model itself or in observed results.

average performance and variance across all trials on two typical device types (A/Cs and dishwashers) for their entire range of k values. The variance is particularly notable here, exhibiting a valley-like shape – higher for small and large values of k, but lower for values in between. This result underscores the point that a model learned from only a few devices (even if those few devices produce a substantial amount of data) is likely to result in an overly specific model that exhibits unstable behavior when applied more generally. The opposite case (i.e., k close to n_c) also speaks to the diversity of devices within the class; e.g., even with a general model trained on 19 of 20 devices, the 20th device may differ substantially and be difficult to identify (which also explains why the steady upward trend of the mean accuracy in Figure 6 stops at the largest values of k). In these cases, our model features may be inadequate to effectively characterize some

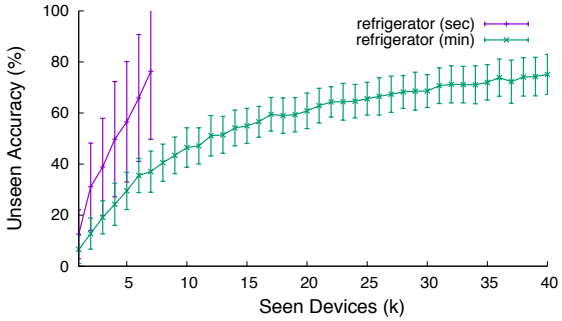

Fig. 7. Comparison of classification accuracy using second-level vs. minute-level data.

of the outlier devices, leading to poor performance in cases where outliers comprise most or all of the test devices.

Given that broader datasets such as Dataport exist, a natural question to ask is whether the benefits of broader data are worth the tradeoffs of less detailed data (e.g., 1/60 Hz versus 1 Hz resolution). The results shown in Figures 4 and 5 are not directly comparable, due to the greater number of classes in the second-level case (which degrades relative performance). For a more reliable test, we repeat the classification results on the refrigerator class for both second and minute-level data using only Dataport devices. The results of this experiment for varied k are shown in Figure 7. Initially, it is clear that the accuracy of the second-level case is both absolutely higher (for a given k) as well as faster-growing as k is increased, demonstrating that the finer resolution is a significant advantage to the classifier. However, a different message may also be taken from this result – that the benefits of greater breadth may counterbalance some data quality shortcomings. For example, in the case of Figure 7, it is preferable to use the lower-resolution dataset containing 20 device instances over the the higher-resolution dataset containing 5 device instances. Furthermore, as discussed previously, we see small but consistent benefits as breadth increases; in this experiment, starting at $k = 42$, the minute-level classifier surpasses the average accuracy of the second-level classifier at $k = 7$ (the highest setting we are able to evaluate given our dataset), and with substantially less variance.

Finally, we note that superior accuracy results could very likely be achieved by more sophisticated classifiers, either using more advanced features or different techniques altogether (e.g., deep learning). Our goal here is not to produce the best absolute results, but to demonstrate that greater data breadth can be effectively exploited even in a simple classifier and can even compensate for lower quality data. The same would presumably hold of more advanced classifiers as well.

V. Conclusion

In this work, we argue that a gap exists in most popular and publicly-available energy datasets: while most datasets focus on deeply-instrumenting a relatively small number of buildings, fewer datasets focus on a broader collection spanning a larger variety of devices. Moreover, datasets lacking depth may limit the applicability of experimental results that were collected using such datasets. As a motivating case study, we consider the problem of identifying the types of specific devices using an off-the-shelf classifier without having seen those devices beforehand. By varying breadth itself in experiments on real-world data, we find that a level of breadth substantially beyond that found in most public datasets is highly beneficial in distilling generalized device models, and can even compensate for reduced data quality (e.g., lower resolutions) versus conventionally 'deep' data. These results underscore the value of considering breadth in curating future energy datasets and of employing such datasets in smart building research.

References

[1] O. Parsons, "Public data sets for nialm," http://blog.oliverparson.co.uk/2012/06/public-data-sets-for-nialm.html, accessed June 2019.
[2] J. Z. Kolter and M. Johnson, "Redd: A Public Data Set for energy disaggregation research," in *SustKDD*, 2011.
[3] J. Kelly and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Scientific Data*, vol. 2, no. 150007, 2015.
[4] K. Anderson, A. Ocneanu, D. Benitez, D. Carlson, A. Rowe, and M. Berges, "BLUED: a fully labeled public dataset for Event-Based Non-Intrusive load monitoring research," in *SustKDD*, Beijing, China, August 2012.
[5] P. Street, "Dataport," https://dataport.cloud/, accessed June 2019.
[6] T. Hnat, V. Srinivasan, J. Lu, T. Sookoor, R. Dawson, J. Stankovic, and K. Whitehouse, "The Hitchhiker's Guide to Successful Residential Sensing Deployments," in *SenSys*, November 2011.
[7] K. Armel, A. Gupta, G. Shrimali, and A. Albert, "Is Disaggregation the Holy Grail of Energy Efficiency? the Case of Electricity," *Energy Policy*, vol. 52, no. 1, January 2013.
[8] G. Hart, "Nonintrusive Appliance Load Monitoring," *IEEE*, vol. 80, no. 12, December 1992.
[9] M. Zeifman and K. Roth, "Nonintrusive Appliance Load Monitoring: Review and Outlook," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, February 2011.
[10] S. Barker, M. Musthag, D. Irwin, and P. Shenoy, "Non-intrusive load identification for smart outlets," in *SmartGridComm*, 2014.
[11] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, and R. Steinmetz, "On the accuracy of appliance identification based on distributed load metering data," in *SustainIT*, 2012.
[12] A. Ruzzelli, C. Nicolas, A. Schoofs, and G. M. P. O'Hare, "Real-time recognition and profiling of appliances through a single electricity sensor," in *SECON*, 2010.
[13] A. Ridi, C. Gisler, and J. Hennebert, "Automatic identification of electrical appliances using smart plugs," in *WoSSPA*, May 2013.
[14] D. Zufferey, C. Gisler, O. A. Khaled, and J. Hennebert, "Machine learning approaches for electric appliance classification," in *ISSPA*, July 2012.
[15] M. Mittelsdorf, A. Hüwel, T. Klingenberg, and M. Sonnenschein, "Submeter based training of multi-class support vector machines for appliance recognition in home electricity consumption data," in *SMARTGREENS*, May 2013.
[16] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised Disaggregation of Low Frequency Power Measurements," in *SDM*, April 2011.
[17] J. Kelly and W. J. Knottenbelt, "Neural NILM: Deep neural networks applied to energy disaggregation," in *BuildSys*, November 2015.
[18] S. Patel, T. Robertson, J. Kientz, M. Reynolds, and G. Abowd, "At the Flick of a Switch: Detecting and Classifying Unique Electrical Events on the Residential Power Line," in *Ubicomp*, September 2007.
[19] C. Shin, S. Rho, H. Lee, and W. Rhee, "Data requirements for applying machine learning to energy disaggregation," *Energies*, vol. 12, May 2019.
[20] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.