

Article

# Data Requirements for Applying Machine Learning to Energy Disaggregation

**Changho Shin** <sup>1,†</sup>, **Seungeun Rho** <sup>2,†</sup>, **Hyoseop Lee** <sup>1</sup> and **Wonjong Rhee** <sup>3,\*</sup> <sup>1</sup> Encored Technologies; Seoul 06109, Korea; chshin@encoredtech.com (C.S.); hslee@encoredtech.com (H.L.)<sup>2</sup> NC Soft Inc., Seongnam 13494, Korea; seungeun07@snu.ac.kr<sup>3</sup> Department of Transdisciplinary Studies, Seoul National University, Seoul 08826, Korea

\* Correspondence: wrhee@snu.ac.kr

† Contributed equally.

Received: 18 January 2019; Accepted: 30 April 2019; Published: 5 May 2019

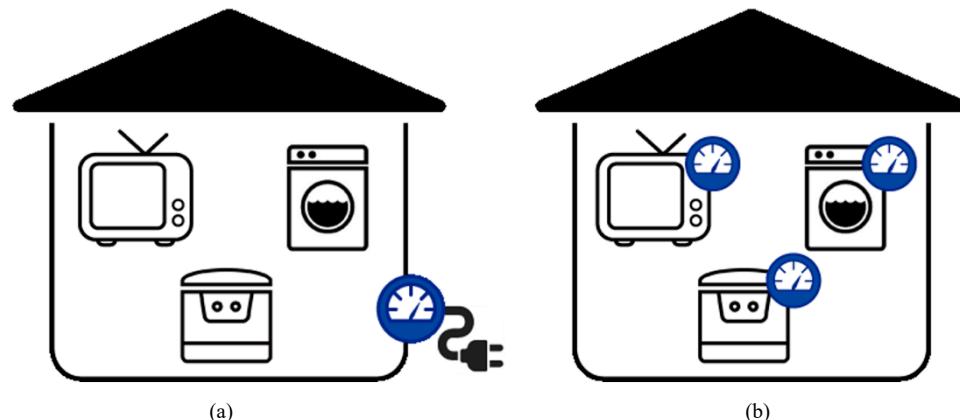


**Abstract:** Energy disaggregation, or nonintrusive load monitoring (NILM), is a technology for separating a household's aggregate electricity consumption information. Although this technology was developed in 1992, its practical usage and mass deployment have been rather limited, possibly because the commonly used datasets are not adequate for NILM research. In this study, we report the findings from a newly collected dataset that contains 10 Hz sampling data for 58 houses. The dataset not only contains the aggregate measurements, but also individual appliance measurements for three types of appliances. By applying three classification algorithms (vanilla DNN (Deep Neural Network), ML (Machine Learning) with feature engineering, and CNN (Convolutional Neural Network) with hyper-parameter tuning) and a recent regression algorithm (Subtask Gated Network) to the new dataset, we show that NILM performance can be significantly limited when the data sampling rate is too low or when the number of distinct houses in the dataset is too small. The well-known NILM datasets that are popular in the research community do not meet these requirements. Our results indicate that higher quality datasets should be used to expedite the progress of NILM research.

**Keywords:** energy disaggregation; nonintrusive load monitoring (NILM); machine learning; data requirements

## 1. Introduction

Disaggregating individual appliance usage from the aggregate electricity data, without extra per-appliance measurements, is referred to as nonintrusive load monitoring (NILM) [1]. As shown in Figure 1, NILM aims to disaggregate a single point measurement of total consumption, as shown in (a), into each appliance's energy consumption. Typically, the single point corresponds to the power distribution board where the outside power source and inside power lines interface, and a reliable NILM solution can help avoid the cost and trouble of installing numerous measurement devices over numerous household appliances. Disaggregated energy consumption can be used for providing feedback to consumers in order to modify their energy consumption behavior. For instance, Neenan & Robinson have shown that energy breakdown information can lead consumers to energy-saving behavior that improves user efficiency by 15% [2]. Moreover, it can be used for detecting malfunctioning appliances, designing energy incentives, managing demand-response [3], etc. Thus, NILM can be an attractive solution in that it provides energy breakdown information without the need for a measurement device for each appliance.



**Figure 1.** Collection of electricity consumption data. (a) For collecting aggregate data, only one measurement device is needed; (b) For collecting individual appliance data, one measurement device per appliance is needed.

Most of the studies and services that are related to NILM focus on using the disaggregation information as feedback to humans for improving energy efficiency [2,4]. However, many other services are possible [5]; Table 1 shows four different application areas together with the required algorithm type, examples of analysis outputs from algorithms, and examples of real-world services that can be offered. The table was constructed based on the opinions of energy platform operators and field engineers.

**Table 1.** Application areas of nonintrusive load monitoring (NILM).

Application Area	Algorithm	Examples of Analysis Output	Examples of Real-world Services
Energy Cost Saving	Regression	<ul style="list-style-type: none"> <li>Identification of large consumption appliance</li> <li>Identification of large consumption appliance compared to neighbors</li> </ul>	<ul style="list-style-type: none"> <li>Targeted marketing of energy-efficient refrigerator</li> <li>Simulation of cost saving by changing AC usage patterns, followed by automated advices</li> </ul>
Smart Home	Real-time Detection	<ul style="list-style-type: none"> <li>Real-time on/off monitoring</li> <li>Appliance operation pattern monitoring</li> </ul>	<ul style="list-style-type: none"> <li>Real-time monitoring and on/off control</li> <li>Real-time notice and alarm</li> </ul>
House Categorization	Classification	<ul style="list-style-type: none"> <li>Identification of houses that have           <ul style="list-style-type: none"> <li>TV on until late night</li> <li>A high level usage of washer</li> <li>A high level usage of rice-cooker</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Video-on-demand service promotion</li> <li>Targeted marketing for home pick-up laundry services</li> <li>Targeted marketing for home rice delivery service (Japan)</li> </ul>
Life Coaching	Detection, Recommendation	<ul style="list-style-type: none"> <li>Anomaly detection of TV watching pattern</li> <li>Monitoring of on/off events that indicate human activity</li> <li>AC on-off time recommendation</li> <li>'Do not run laundry' recommendation</li> </ul>	<ul style="list-style-type: none"> <li>Family/people monitoring service – one-person household, elderly parents, etc.</li> <li>Recommendations on when to turn on AC (now or save cost for other days with worse weather; weather prediction used together)</li> </ul>

In a real-world service, the required type of algorithm can be identified according to the characteristics of the service to be provided. Energy cost saving, which is arguably the most widely recognized application of NILM, requires accurate regression. While energy saving is an important case for justifying the value of NILM, accurate regression is technically the most challenging and there are many other real-world services that can be implemented without regression. For instance, the 'family/people monitoring service' in life coaching can be a valuable service to single-elderly-person

households, and a service can be provided as long as an algorithm can confidently detect ‘no human action for a long while’. Another exemplary service is classifying if a house has an old refrigerator with excessive energy consumption. With this information, targeted marketing becomes possible and only a binary classification of each house over a long time period is sufficient for providing the service. This is a considerably less challenging task than generating a continuous output of the refrigerator’s electricity consumption. Smart homes are yet another application that are attracting tremendous industry interest. For a basic function of reporting if an appliance is on or off, one only needs to develop an algorithm that can reliably detect on/off events in real-time. In Table 2, the algorithm and service requirements for four application areas are shown.

**Table 2.** Requirements for providing useful services.

Application Area	Algorithm	Analysis Output’s Time Resolution			Analysis Period		Algorithm Accuracy		Output Information		
		1-sec	15-min	day	1-s	15-min	day	High	Median	Low	Usage amount(kWh)
Energy Cost Saving	Regression	▲	◎				◎	◎			◎
T Smart Home	Real-time Detection	◎			◎		◎				◎
Home Categorization	Classification	▲	◎		▲	◎	▲	◎			◎
Life Coaching	Detection, Prediction	▲	◎		▲	◎	▲	◎			◎

◎: Required; ▲: Optional.

It is evident from Tables 1 and 2 that there are several services that require a wide range of specifications, including algorithm accuracy. In fact, energy IoT platforms are being used not only for NILM-relevant services, but also for other energy and life-related services such as demand response, real-time monitoring of aggregate electricity, energy bill calculation and prediction, and malfunctioning appliance or improper electric wiring detection [4,5]. Therefore, NILM should be viewed as a library that provides many different functions relevant to disaggregation, and not as a stand-alone application.

In addition to basic energy saving, NILM has the potential to be utilized for a variety of services. Despite the importance and long history, NILM is still in the early stage of adoption. Part of the reason for this slow adoption may be attributed to the limitations of the datasets that have been used by the research community. As will be explained in Section 2.2, there have been only a handful of public datasets that have been repeatedly used for numerous studies and these datasets are rather limited in terms of size and information. To understand the impact of these limitations on algorithm performance, we collected data from 58 households using sensing devices with 10 Hz sampling rate. Additional sensing devices were deployed to independently and simultaneously collect electricity consumption of a set of selected appliances, which enabled an evaluation of NILM performance. Using the newly collected dataset, we studied the sampling rate of sensor data and the number of households that need to be included in a dataset for performing reliable NILM research.

Recently, a variety of problem frameworks have been investigated for developing NILM algorithms. For instance, the possibility of utilizing NILM for anomalous behavior detection has been addressed in [6] and adoption of denoising autoencoders is considered in [7]. In this study, however, we focused on the basic classification and regression algorithms because our goal is to understand the data

requirements. The classification algorithms in this paper were developed with the goal of encompassing the most representative approaches, namely, feature engineering-based models, basic deep learning models, and deep learning models with extra parameter tuning. For regression, we adopted a state-of-the-art deep learning algorithm, as described in [8].

In Section 2, previous works on NILM algorithms and the popular datasets used for algorithm research have been explained. In Section 3, we explain the research dataset used in this study. In Sections 4 and 5, we present the frameworks that we used for the data requirement studies and their basic results. In Section 6, we explore the conditions that the energy dataset must satisfy in terms of sampling rate and the number of houses when developing supervised NILM algorithms. In Section 7, the limitations of our study are discussed.

## 2. Backgrounds

In this section, NILM algorithms and the datasets used in the previous studies have been explained. As for the datasets, the data quality issue has been addressed as well.

### 2.1. NILM Algorithms

Energy disaggregation is made possible by the signatures of each appliance that can be detected in the aggregated power consumption. Signatures are the patterns that contain information on the appliance activity, such as its on/off status. Appliances have their own power consumption patterns depending on electrical and mechanical inner components. Practical challenges, however, exist for detecting such patterns. For instance, the signatures of a particular type of appliance can vary depending on the manufacturer and product model—not all TVs will show the same fingerprints, but will fall into a few groups of fingerprints instead. Furthermore, some appliances possess multiple signatures since they have multiple operation modes. A good example is a rice cooker that has a ‘cook’ mode and a ‘keep warm’ mode. Another challenge is that the signatures of an appliance are usually distorted and occluded by those of other appliances, and this is the most challenging part of energy disaggregation.

Many studies have been conducted to overcome the aforementioned difficulties. Two main approaches among them are event detection and decomposition. An ‘event’ refers to a transition of state for an appliance (e.g., on to off, off to on). The event detection approach focuses on identifying and classifying an event from the aggregate data. Hart first suggested this approach [1] in 1992. More advanced techniques with load signature feature extraction [9] and unsupervised NILM framework with event detection method [10] have been proposed as well. Contrary to the event detection approach, the decomposition approach directly addresses energy disaggregation by separating the entire aggregate signal into its component appliance signals. Such methods can be grouped into supervised and unsupervised. Sparse coding [11] might be the most representative approach in supervised methods. The algorithm learns the dictionary of each appliance using the signals of appliances in the training datasets and models the aggregate signal as a sparse linear combination of the components from the dictionary. Sparse subset selection technique was suggested for composing the dictionary of each appliance [12]. Singh et al. (2016) combined deep learning with sparse coding, which used multiple layers with sparse coding [13]. On the other hand, a frequently used algorithm among unsupervised methods is the factorial hidden Markov model (FHMM) [14]. Unsupervised energy disaggregation using FHMM and variants of FHMM can be found in [15], and improvement of FHMM with prior models is suggested in [16]. Zhong et al. incorporated signal aggregate constraints (SACs) into an additive factorial hidden Markov model, which significantly improved FHMM [17]. Shaloudegi et al. enhanced Zhong et al.’s algorithms by combining convex semidefinite relaxations randomized rounding [18]. However, decomposition algorithms require the total number of appliances to be fixed and known, which is an unrealistic assumption.

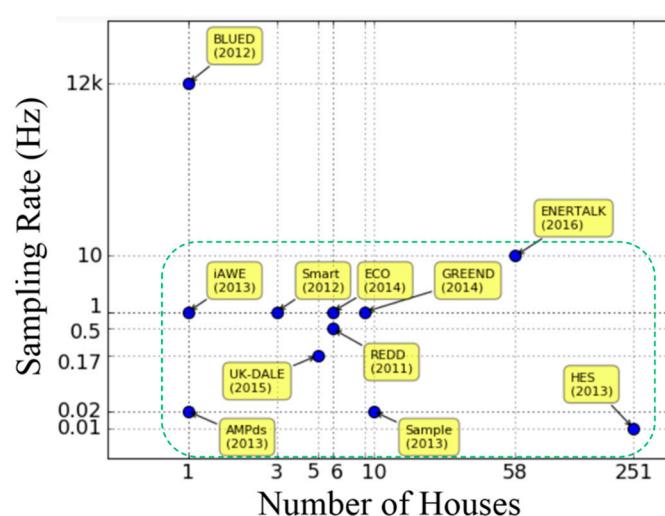
Recently, state-of-the-art performances have been achieved by using deep learning techniques. Kelly et al. applied recurrent neural network and denoising autoencoders on the UK-DALE dataset [19].

Huss proposed a hybrid energy disaggregation algorithm based on CNNs and a hidden semi-Markov model [20]. Zhang et al. showed sequence-to-point learning with CNN wherein the treatment of the single midpoint of a time window as the output of the network instead of the whole sequence of the window was beneficial [21]. Chen et al. applied deep residual networks for convolutional sequence to sequence learning of NILM, which also improved the performance [22]. Most recently, Shin et al. proposed subtask gated networks that incorporate on-off classification information in addition to the original regression information to outperform the previous best regression result [8].

## 2.2. Datasets and Data Quality

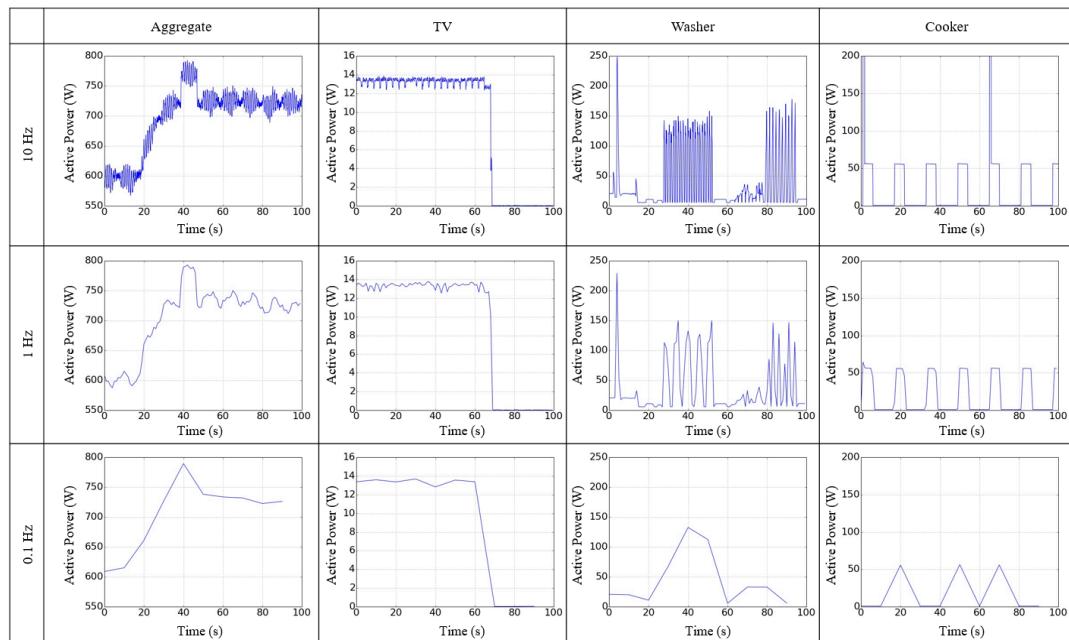
Data generation ideally requires a large-scale deployment, but there is a cost issue because dedicated hardware and software need to be developed and deployed. As for the hardware, typically three tiers of products can be manufactured depending on the cost flexibility and the data type/sampling-speed requirements. The first tier can sample electricity waveforms a few million times per second; therefore, the high-frequency signature of each appliance can be used for disaggregation. While this approach provides ‘naming each’ capability, the type of signatures can be highly irregular. For instance, the signatures might look quite different even among the TVs with the same display technology, and consequently, a manual process for matching each ‘signature’ to an appliance in a particular house might be needed before disaggregation can be used for the house. Furthermore, the cost of these devices is high for the first tier. The other two tiers of products collect fewer samples per second—a few thousand per second for the second tier and at most tens of samples per second for the third tier. While the data quality is the worst for the third tier, the manufacturing cost is the cheapest (usually several times cheaper than the first tier) and the collected data provides a significant amount of information for disaggregation anyway. The data fields of the third tier usually include active power, reactive power, and voltage. In our study, we focus on the third tier because of its competitive cost for mass deployment.

The popular public datasets from [23–28] and the ENERTALK dataset introduced in this article are shown together in Figure 2. Because of the apparent difficulty of collecting and handling big data, many of the existing studies in the literature have utilized the popular public datasets. The quality and quantity of the datasets, however, are limited. As we can see in Figure 2, most of the datasets have sampling rates of 1 Hz or under. Furthermore, each dataset typically contains the data collected from less than ten houses.



**Figure 2.** Summary of publicly available NILM datasets and the ENERTALK dataset used in this study. Datasets within the green box are the ones that require the lowest level of cost for hardware. Some of the datasets partly contain aggregate data with higher sampling rates, but only those sampling rates that are used for the supervised NILM tasks have been shown.

Among the many data requirements of NILM, sampling rate has been found to be the most critical factor in such studies according to Armel et al. [4]. In their study, they perform a metastudy of the existing works and showed that the identifiability of appliances depended on the sampling rate. Many real-life modern-day appliances need to be disaggregated using edge shapes of on/off events or repeated signatures during the on-state. The sampling rates of public datasets might not be sufficiently high for capturing the critical information. In Figure 3, we have plotted the exemplary electricity shapes of aggregate, TV, washer, and rice cooker signals. While the regularity of repeated shapes during the on-state is obvious for 10 Hz sampling, the shapes start to become ambiguous as the sampling rate is reduced to 1 Hz and then to 0.1 Hz. In the case of a rice cooker, the heating is done by a heating component that consumes electricity in a train of pulse shapes. The amplitude and duty cycle decide how much power the appliance consumes. In the top-right of Figure 3, the train of the pulse can be clearly observed. When the sampling rate is reduced to 1 Hz, however, the pulse shape starts to be distorted and the two ‘overshooting’ signatures, which were observed with 10 Hz sampling, disappear. In the bottom-right of Figure 3, the sampling frequency is 0.1 Hz and it is impossible to confidently declare that these signatures correspond to a rice cooker. Similar behaviors can be observed for aggregate, TV, and washer signals.



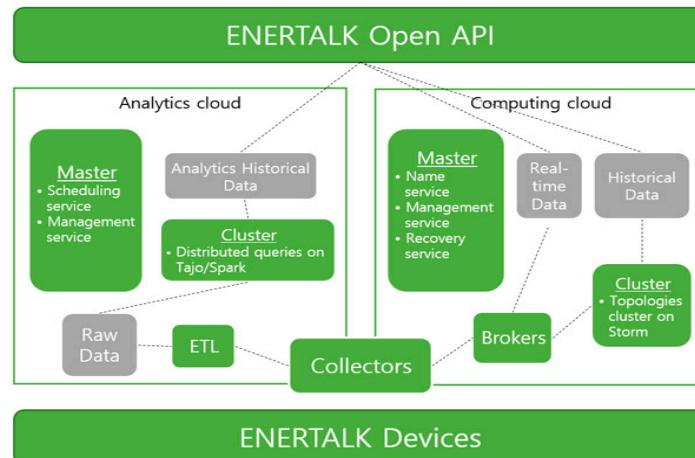
**Figure 3.** Electricity signal shapes (signatures) of aggregate, TV, washer, and rice cooker for 10, 1, and 0.1 Hz sampling.

Besides the sampling rate issue, many public datasets contain only a small number of houses. The importance of the number of houses can be explained in two ways. First, machine learning approaches for NILM can have an overfitting problem when the number of houses is not large enough. Data acquired from many houses can be crucial for a better generalization of NILM algorithms. As the number of houses increases, the number of combinations of appliances covered by the algorithm also increases, which makes NILM algorithms applicable to new houses. Secondly, the diversity of models for the same appliance type cannot be addressed by the limited datasets. As an example, one can focus on a particular model of TV by studying the dataset from a single house, but there is no guarantee that the findings will generalize to all TV models. In fact, the core display technology for TV has been rapidly changing, and there are easily five types of core technologies such as CRT (Cathode Ray Tube), plasma, LCD (Liquid Crystal Display), LED (Light-Emitting Diode), and OLED (Organic Light-Emitting Diode). Therefore, to develop a ‘TV disaggregation algorithm’, one needs

to have access to the data collected from at least tens of houses such that all types of TVs can have a reasonable chance of being included in the dataset.

### 3. ENERTALK Dataset

In our study, we used the ENERTALK dataset collected through a commercial energy IoT platform called 'ENERTALK'. In Figure 4, the system diagram of ENERTALK is shown. It is a general IoT platform for collecting, storing, and analyzing data, and NILM is one of the analysis functions in the platform. ENERTALK platform can be seen as a data intelligence platform based on smart meters [29].



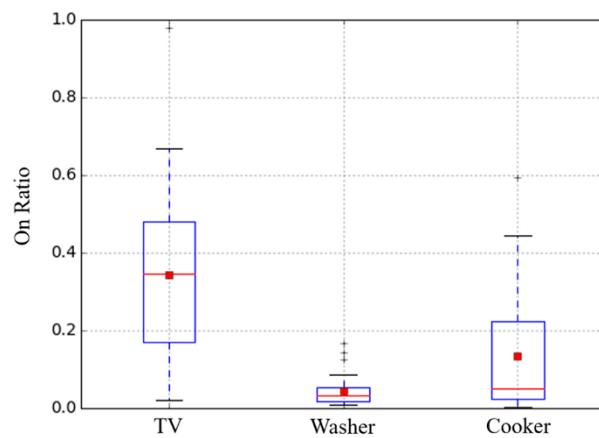
**Figure 4.** System diagram of the ENERTALK platform.

The ENERTALK dataset (unfortunately, the ENERTALK dataset used for this study cannot be made public due to privacy policy issues. Instead, a comparable dataset sampled at 15 Hz is scheduled to be released soon) contains 10 Hz energy readings from 58 homes in Japan, collected between July and August of 2016. The exact measurement period varies by the house, with 67% of houses measured for two months, 24% for one week, and the rest in between. The data provides readings of active power and reactive power, along with timestamps. Both aggregate data and individual appliance data were collected simultaneously for the developing the algorithm. Individual appliance data, which is used to construct the true answer to the model, consists of TVs, washers, and rice cookers, and additional per-appliance data collection devices were also deployed. Although data was collected for other appliances as well, per-appliance measurement devices were deployed only partially for some appliances. Therefore, our study focuses on the aforementioned three devices only. Among the 58 houses considered in this study, we collected data for 40 houses for TVs, 50 houses for washers, and 22 houses for rice cookers.

The actual usage time of an appliance was widespread and depended on the household. For example, a house had a TV on for more than 16 hours in a typical day, while another had a TV on for less than half an hour. Furthermore, rice cookers had multiple states of 'on'. For the households that usually use the keep-warm mode of the rice cooker, a certain level of power was always in use and a large portion of cooker data was labeled as on. In contrast, some households hardly used the keep-warm mode, and the rice cooker was measured to be off most of the time. This on-time variation is shown in Figure 5. The resulting imbalance in the class label ratio makes the training of the model more difficult.

As can be seen in Figure 2, the ENERTALK dataset contains 58 houses measured at 10 Hz sampling rate. Compared to the existing public datasets, this dataset has a relatively higher sampling rate and a larger number of houses. Dataset BLUED offers higher frequency, but contains only one house. REDD and part of UK-DALE also offer higher frequency data, but are applicable only for aggregate data,

not appliance data. The HES offers data from 251 houses, but the sampling rate is only one sample every two minutes, and thus 1200 times slower than 10 Hz.



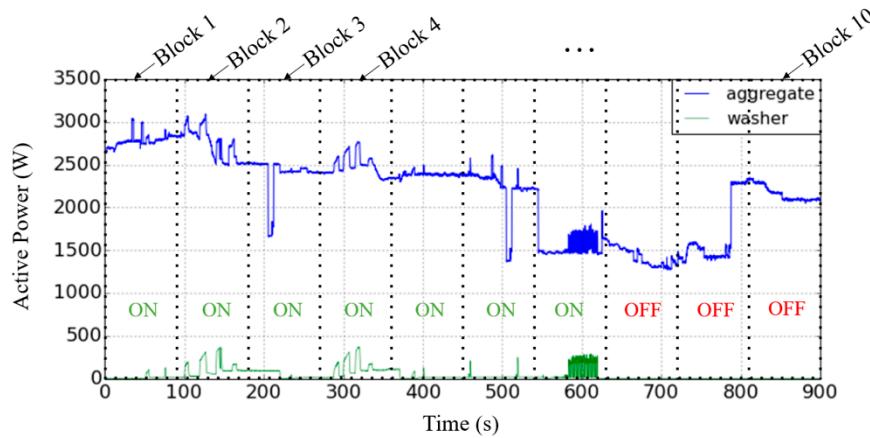
**Figure 5.** On ratio of appliances in the ENERTALK dataset.

#### 4. NILM Algorithms Considered in This Study

As discussed in the introduction, all regression, classification, detection, and recommendation algorithms are relevant to utilizing NILM for real-world services. In this study, we focused on the most basic modeling frameworks—a binary classification framework and a power usage regression framework. The binary classification framework can be the simplest choice for the sensitivity study, but it is closely related to many fundamental NILM services and thus an important one to consider. Power usage regression framework is the traditional framework for studying NILM algorithms [6,11–22].

For the classification framework, the raw measurements are used to construct 90-second blocks, as shown in Figure 6. Each block contains 900 samples of aggregate electricity measurements and three labels on whether TVs, washers, and rice cookers were on. Here, the block size of 90 s was chosen because it is long enough to accommodate the inherent patterns that were found in the feature engineering study in the binary classification framework. Per-appliance measurements were used for creating the on-off labels and calculating the sum of power consumption. Because the data collection periods vary over the houses and each appliance's chance of being on or off varies (see Figure 5), we balanced the training dataset to have all houses contribute with equal importance and to have each of the on and off samples occupy 50% of the training samples. At the end, we randomly selected 1000 on blocks and 1000 off blocks from the dataset of each house and constructed the training data for each appliance. For the 58 houses, a total of 116,000 blocks were used as the training dataset. We have confirmed that the equal on/off ratio in the training data is very helpful for improving the classification performance. For validation and testing, we used the true on/off ratio in the raw data instead of the balanced on-off ratio in order that our evaluations would represent realistic scenarios.

For the regression framework, the input's window size was fixed to 90 s as in the classification framework, and the target to estimate was chosen as the appliance's power usage over the entire 90 s. This was necessary because our dataset contained only the 90 s aggregate power usage value for each appliance's power usage. For the services that utilize NILM regression results, one estimation every 90 s can be slower than what is ideally desired but most of the services, including the most important energy-saving service, can be reliably offered anyway. As in the classification, the training dataset was balanced over the houses, and the on-off states and validation and testing data were not balanced.



**Figure 6.** Defining 90-s blocks for binary classification and power usage regression.

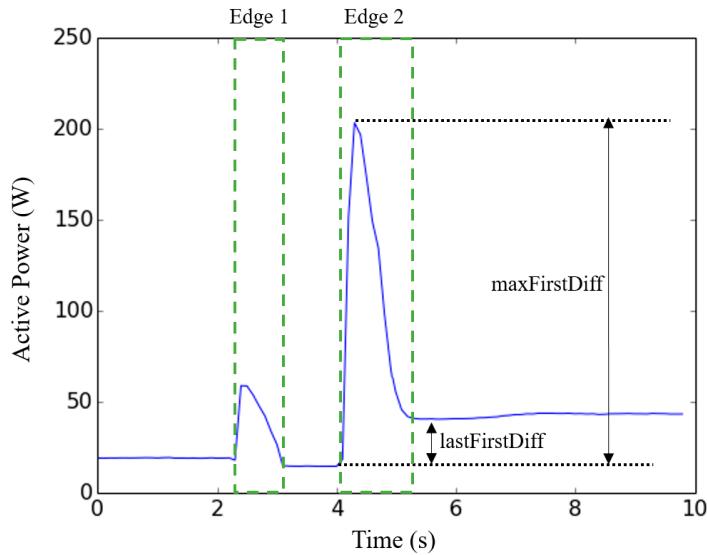
#### 4.1. Classification Algorithms

##### 4.1.1. Vanilla DNN

Vanilla CNN and LSTM (Long Short-Term Memory) [30], the two representative models of deep neural networks, were used as the basic benchmark. In the case of CNN, one-layer CNN was applied instead of the usual two-layer CNN, where active-power (AP) 900 points and reactive-power (RP) 900 points were placed side by side in a block. The architecture of CNN was adopted from LeNet-5 [31] that uses two convolutional layers with successive pooling layers, and two fully connected layers at the end. Filters moved in one dimension from left to right to perform convolution with the signals. The number of convolution filters was 32 for the first layer and 64 for the second layer, and the size of pooling was two. Dropout was used in the fully connected layer. In LSTM, AP and RP were used to create a  $2 \times 900$  matrix data as the input to the hidden layer, and a total of 900 steps were passed. The output of the last step was then used to decide whether an appliance was on or off for the block. As a preprocessing step, we standardized the input data (AP and RP) house by house. For each house, every single data point was subtracted by the average of the full data from the house, and divided by the standard deviation.

##### 4.1.2. Machine Learning with Feature Engineering

Electricity data contains a large number of signatures that are closely related to the underlying components within the appliance. Therefore, feature engineering, especially based on signal processing techniques, is a natural way to approach NILM. For this group of algorithms, we applied two traditional ML algorithms, Logistic Regression, and Random Forest, after creating 59 features from the 90-s raw data blocks. The number of trees was 100, the minimum leaf size was 1, and the number of predictors to sample was 8 for Random Forest, which was selected through a simple grid search. For feature engineering that created 59 features, we carefully selected features from many more features that were investigated in an in-depth study. While it is not possible to explain these features in detail, we have provided an explanation of a set of features that were finally included and we hope to provide a general ideas on how feature engineering was performed in this work. In Figure 7, an exemplary aggregate data of a house over a 10-s period is shown. The 0–2 s period is flat, and suddenly there is an event a little after 2 s. It becomes flat again at ~3 s, and then another event is observed between 4 and 5 s. Clearly, there are ‘flat’ regions and varying ‘edge’ regions, and one can create attributes based on these characteristics. In Table 3, eight of such attributes are listed. Once the attributes are defined, actual features can be defined as functions that use attribute values as the inputs. Eight of such features are listed in Table 4. These handcrafted features exploit the signatures of each appliance’s signal.



**Figure 7.** Illustration of edges and a few edge attributes.

**Table 3.** List of edge attributes.

Attribute Name	Description
Type	Edge type
firstIdx	Final index of edge
lastIdx	Last index of edge
lastFirstDiff	Difference between first point last point
maxFirstDiff	Difference between maximum point and first point
minFirstDiff	Difference between minimum point and first point
maxSlope	Slope between maximum point and first point
minSlope	Slope between minimum point and first point

**Table 4.** List of Features.

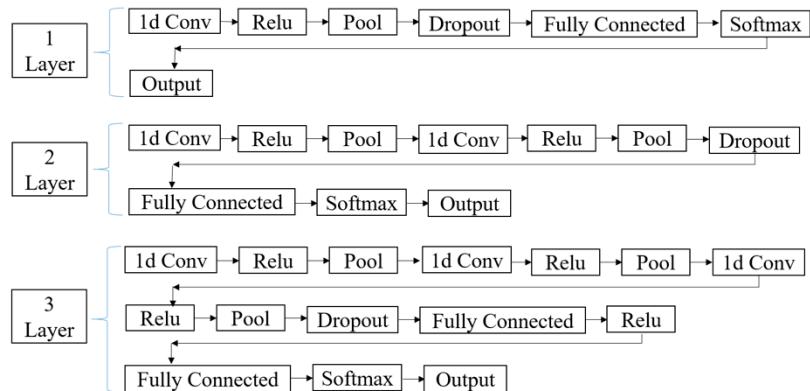
Feature Name	Description
edgeCount	Number of edges in 90 secs box
maxSlopeMean	Mean value of maxSlope attributes of edges
edgeLengthMax	Maximum length of edges
smallRisingCount	Number of rising edges with $\text{maxFirstDiff} \leq 40$
smallFallingCount	Number of falling edges with $\text{minFirstDiff} \geq -40$
flatCount	Number of flat regions
flatLengthMax	Maximum length of flat regions
betweenFlatMax	Maximum length between pairs of flat regions

#### 4.1.3. CNN Optimized by HPO

DNN is an efficient solution for avoiding feature engineering, which is time-consuming and difficult. However, as will be seen in Section 5, using vanilla DNN does not guarantee good performance partly because, typically, DNN performance is heavily dependent on the values of the hyperparameters [32]. We can conduct a grid search as for Random Forest, but DNN has a huge search space and the optimization can take a long time. To overcome this problem, we adopted an automatic hyperparameter optimization (HPO) method that was described in [33] and is based on a Bayesian optimization method to optimize the architecture of DNN. Rather than attempting to try multiple points in the hyperparameter space randomly, it assumes a Gaussian process prior and updates the prior based on the points the HPO algorithm have measured, evaluating the next points that maximize the expected improvement. HPO is effective when evaluating performance is expensive in terms of time consumption and computation. Training a deep neural network undoubtedly fits into this

category. Bayesian optimization has shown many promising results in the HPO field. It even surpasses the expert human model on image classification in 2012 [34].

Because CNN tends to perform better than LSTM for our tasks, we limited the scope of this study to CNN only and optimized the hyperparameters, including the number of layers, neurons, and epochs. Before running HPO, the list of hyperparameters and range of each hyperparameter need to be determined. We ran HPO on three different CNN models as shown in Figure 8, where each model's number of convolution (with pooling) layers was 1–3. The maximum step size of Bayesian optimization was chosen to be 20 for all three models. Hence, we used CNN architectures shown in Table 5.



**Figure 8.** Architecture of a convolutional neural network (CNN) with 1–3 layers.

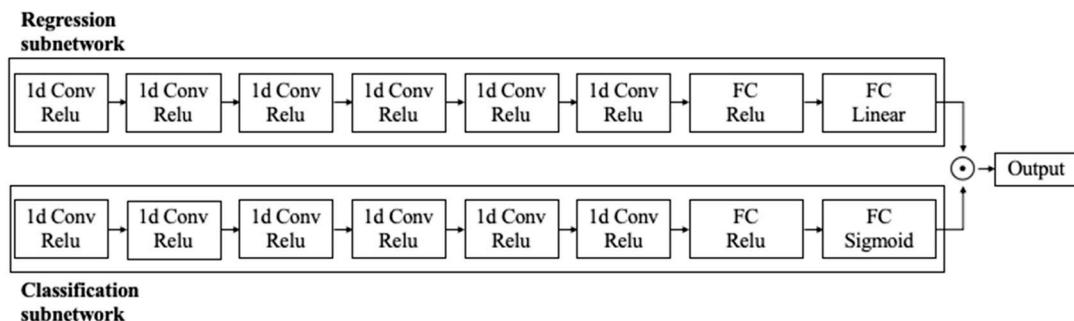
**Table 5.** CNN with hyperparameter optimization (HPO): Range setting and optimized parameters.

	Range Setting	Optimized Parameters		
		TV	Washer	Cooker
1 layer HPO	$E : 1 \sim 10$	$E : 4$	$E : 7$	$E : 8$
	$F_{1,\text{size}} : 2 \sim 50$	$F_{1,\text{size}} : 2$	$F_{1,\text{size}} : 27$	$F_{1,\text{size}} : 35$
	$F_{1,\text{num}} : 5 \sim 30$	$F_{1,\text{num}} : 30$	$F_{1,\text{num}} : 21$	$F_{1,\text{num}} : 5$
	$P_{1,\text{size}} : 2 \sim 50$	$P_{1,\text{size}} : 19$	$P_{1,\text{size}} : 42$	$P_{1,\text{size}} : 38$
2 layer HPO	$E : 1 \sim 10$	$E : 5$	$E : 9$	$E : 10$
	$F_{1,\text{size}} : 2 \sim 12$	$F_{1,\text{size}} : 2$	$F_{1,\text{size}} : 10$	$F_{1,\text{size}} : 12$
	$F_{1,\text{num}} : 5 \sim 30$	$F_{1,\text{num}} : 5$	$F_{1,\text{num}} : 7$	$F_{1,\text{num}} : 5$
	$P_{1,\text{size}} : 2 \sim 20$	$P_{1,\text{size}} : 19$	$P_{1,\text{size}} : 19$	$P_{1,\text{size}} : 20$
	$F_{2,\text{size}} : 2 \sim 12$	$F_{2,\text{size}} : 2$	$F_{2,\text{size}} : 8$	$F_{2,\text{size}} : 4$
	$F_{2,\text{num}} : 5 \sim 40$	$F_{2,\text{num}} : 40$	$F_{2,\text{num}} : 36$	$F_{2,\text{num}} : 5$
	$P_{2,\text{size}} : 2 \sim 20$	$P_{2,\text{size}} : 10$	$P_{2,\text{size}} : 12$	$P_{2,\text{size}} : 2$
3 layer HPO	$E : 1 \sim 10$	$E : 1$	$E : 10$	$E : 10$
	$F_{1,\text{size}} : 2 \sim 12$	$F_{1,\text{size}} : 2$	$F_{1,\text{size}} : 12$	$F_{1,\text{size}} : 10$
	$F_{1,\text{num}} : 5 \sim 30$	$F_{1,\text{num}} : 24$	$F_{1,\text{num}} : 30$	$F_{1,\text{num}} : 13$
	$P_{1,\text{size}} : 2 \sim 20$	$P_{1,\text{size}} : 7$	$P_{1,\text{size}} : 11$	$P_{1,\text{size}} : 3$
	$F_{2,\text{size}} : 2 \sim 12$	$F_{2,\text{size}} : 9$	$F_{2,\text{size}} : 10$	$F_{2,\text{size}} : 8$
	$F_{2,\text{num}} : 5 \sim 40$	$F_{2,\text{num}} : 5$	$F_{2,\text{num}} : 24$	$F_{2,\text{num}} : 27$
	$P_{2,\text{size}} : 2 \sim 20$	$P_{2,\text{size}} : 20$	$P_{2,\text{size}} : 4$	$P_{2,\text{size}} : 10$
	$F_{3,\text{size}} : 2 \sim 12$	$F_{3,\text{size}} : 2$	$F_{3,\text{size}} : 9$	$F_{3,\text{size}} : 6$
	$F_{3,\text{num}} : 5 \sim 40$	$F_{3,\text{num}} : 5$	$F_{3,\text{num}} : 14$	$F_{3,\text{num}} : 40$
	$P_{3,\text{size}} : 2$	$P_{3,\text{size}} : 2$	$P_{3,\text{size}} : 2$	$P_{3,\text{size}} : 2$

#### 4.2. Regression Algorithm

For the regression algorithm, a state-of-the-art NILM algorithm called subtask gated networks (SGN) was used [8]. It is a sequence-to-point algorithm, and Figure 9 shows the architecture of SGN, in which the output of the regression subnetwork is gated by the output of the classification subnetwork.

Details of the algorithm can be found in [8], and we too used the exact same algorithm, except for applying it to the ENERTALK dataset.



**Figure 9.** Architecture of subtask gated networks (SGN). ‘FC’ represents a fully connected layer.

As with the Vanilla DNN in the binary classification task, the active power and reactive power of aggregated power consumption were used to create a  $2 \times 900$  matrix data as the input. We used the per-appliance measurements of the true regression outputs, and the same labels as used in the classification task for calculating the loss function during training and evaluating performance during testing. As a preprocessing step, we standardized the input data (AP and RP) as in the previous research [8].

## 5. Basic Performance Results

In this section, the basic performance results of running the algorithms introduced in Section 4 over the ENERTALK dataset are provided. These results serve as the baseline performance for the sensitivity study of data sampling rate and the number of houses in Section 6.

### 5.1. Performance Metric

For evaluation, a five-fold cross-validation was used. Unlike how training data were generated (1000 on blocks and 1000 off blocks sampled from each house in the training folds), all blocks were available for the test houses so that the test result would be as realistic as possible. After testing five-fold in this manner, the average area under receiver operating characteristic curve (AUROC) over five-fold results were calculated as the final performance metric of classification. For the regression, we calculated the average mean absolute error (MAE) over the five-fold results as the performance metric.

### 5.2. Experimental Results

The results of binary classification experiments are shown in Table 6. In the case of TVs, ML with feature engineering performed so poorly that even vanilla DNN achieved better performance. Both ML algorithms with feature engineering showed AUROC values that were too low for practical use. This result implies that handcrafted features failed to be sufficiently related to TVs. This can be explained by the characteristics of TV signatures. Most of the TV models show a fast fluctuation in electricity usage depending on what colors are dominantly displayed at each moment. When the screen scene is dark, less power is used. When the screen scene is bright, more power is used. Therefore, the patterns of TV are not very regular, making it difficult for handcrafted feature engineering to work well. Vanilla DNN and CNN with HPO, however, performed better. After a deep dive to investigate how they did better, it was found that the two algorithms utilized the correlation where TV tends to be on when many other appliances are on (e.g., evening time). There was a sufficient correlation between the usages of TV and all the other appliances together, and deep learning framework was able to exploit it. The best performing algorithm was one-layer CNN with HPO, and the result is consistent with our hypothesis that TV signals are irregular and using more layers would not be very helpful.

**Table 6.** Summary of classification performance (area under receiver operating characteristic curve (AUROC) values).

Model	Vanilla DNN		ML with Feature Engineering		CNN with HPO		
	LeNet-5	LSTM	Logistic Regression	Random Forest	1 layer	2 layer	3 layer
TV	0.8219	0.8341	0.6924	0.6910	<b>0.8427</b>	0.8354	0.8391
Washer	0.8438	0.7514	0.8974	<b>0.9121</b>	0.8658	0.8909	0.9017
Cooker	0.7653	0.6689	0.7734	0.7871	0.8487	0.8529	<b>0.8826</b>

In the case of washers, an opposite result was obtained, and Random Forest with feature engineering performed the best. The AUROC value was 0.9121, which is very high for a real-world scenario. As can be seen in Figure 3, signals from washers have very strong and distinguishable shapes, and they were well captured by feature engineering. CNN with HPO also worked well, and it outperformed vanilla DNN by a large margin. As for the number of layers, CNN with HPO worked the best when three layers were used. It can be speculated that the structures of the washers' signatures were complicated enough to cause the three-layer model to work the best. Perhaps even better performance would have been obtained by stacking more layers. In the case of rice cookers, CNN with HPO performed well. The AUROC value was very high at 0.8826, indicating that real-world services can be reliably built over the NILM algorithm. For washers and cookers, ML with feature engineering worked better for one (washers), and CNN with HPO worked better for the other (cookers). The performance difference, however, was sometimes marginal and the results could have been different with extra effort in either of the algorithm groups. While there is enough room for improving the performance, the performance results showed enough diversity for us to continue with the sensitivity analysis.

The results of the regression experiments are shown in Table 7. Note that the MAE metric was calculated based on power usage (Wh), unlike the Watt-based calculations as in the previous research [8]. From these results, it is evident that regression of TVs is much more difficult than for washers or cookers. As TV signatures are known to be irregular, their regression tends to perform poorly even with the 10 Hz sampling that provides a better resolution on the raw data patterns. As we will see in Section 6, this fundamental aspect makes the performance of TVs less dependent on the sampling rate.

**Table 7.** Summary of regression performance (mean absolute error (MAE) values).

Appliances			
Model	TV	Washer	Cooker
SGN	1.1476	0.3704	0.2668

## 6. Sensitivity Analysis Results for Sampling Rate and Number of Houses

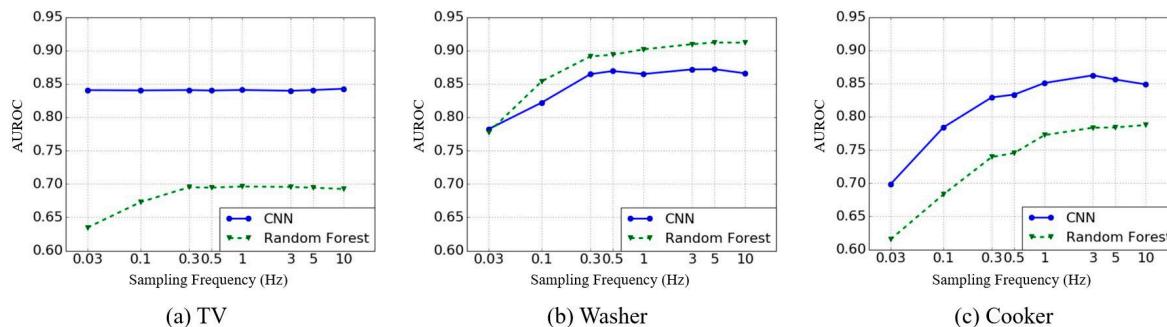
The key requirements of datasets are the sampling rate and number of houses, as we have discussed in the Introduction and Section 2. In this section, we describe the NILM's performance sensitivity to the data sampling rate and the number of houses.

### 6.1. Sensitivity to Sampling Rate

When generating energy IoT data, the sampling rate is an important parameter for design. Apparently, hardware cost can go up if a certain threshold is passed and high-end components need to be integrated in the data collection device. Furthermore, the cost of the data platform, where storage, analytics, and other functions need to be performed, is obviously affected by the sampling rate because the data size is closely related. Therefore, understanding the performance and cost trade-off of the sampling rate is important. The sampling rate of many of the public datasets given in Figure 2, however, might have been determined without such a trade-off consideration. As can be seen in

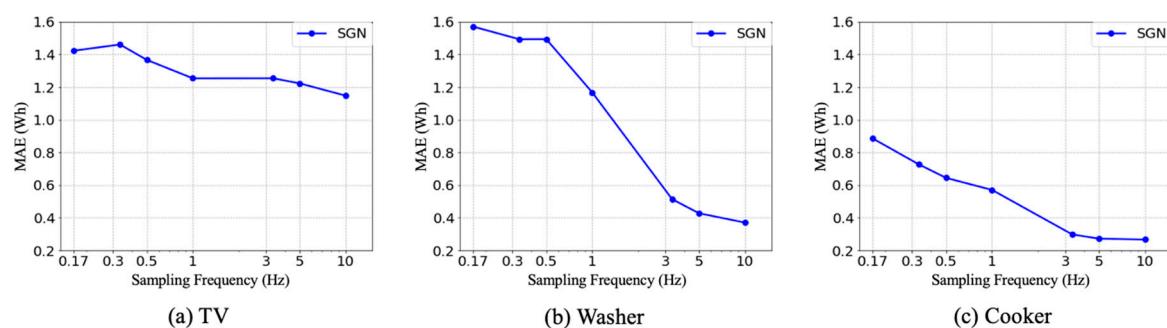
Figure 3, the signatures are significantly distorted as the sampling rate decreases. These plots indicate that NILM performance can be significantly affected, and we attempted to confirm this hypothesis in this study.

In Figure 10, binary classification performance is shown as the sampling rate is reduced from 10 Hz to 0.03 Hz. To be precise, the data blocks with 900 samples were downsampled such that 3, 9, 30, 45, 90, 300, 450, and 900 samples existed in the 90-s period. Three samples per 90-s corresponds to a single sample in 30 s. For each sampling rate, Random Forest and an optimized one-layer CNN was trained to find the AUROC value. Here, one layer was used because a very small sample size cannot be tested for CNNs with more layers that utilize pooling multiple times. In Figure 10a, the NILM performance for TV is affected by the sampling rate for Random Forest, but there is hardly any performance loss for CNN. As discussed in 5.3, the TV signatures are very simple and weak, and that must have resulted in the insensitivity of CNN. In Figure 10b,c, it can be clearly observed that the performances of the washers and cookers were seriously impaired as the sampling rate was reduced. In fact, a sampling rate of at least 1–10 Hz is desired to prevent performance loss, and a sampling rate higher than 10 Hz might be helpful as well. This observation is consistent with Figure 3, where original signatures can be barely identified at 0.1 Hz sampling rate.



**Figure 10.** Sensitivity of classification algorithms to sampling rate. Sensitivities of CNN and Random Forest are shown where a larger value is desired for AUROC.

In Figure 11, the regression performance is shown. Note that the sampling frequency starts with 0.17 Hz because of the minimal number of samples required for SGN architecture. In the regression task, the data blocks with 900 samples were down-sampled such that 15, 30, 45, 90, 300, 450, and 900 samples existed in the 90-s period. The regression results followed similar trends to those of classification performance. In Figure 11a, NILM regression performance for TVs was relatively less affected by the sampling rate than the other appliances. For washers and cookers, the MAE curves dropped sharply as the sampling rate increased to 3 Hz or above. After the sampling rate reached 3 Hz, the performance curves tended to improve minimally.



**Figure 11.** Sensitivity of the regression algorithm to the sampling rate. Sensitivity of SGN is shown where a smaller value is desired for MAE.

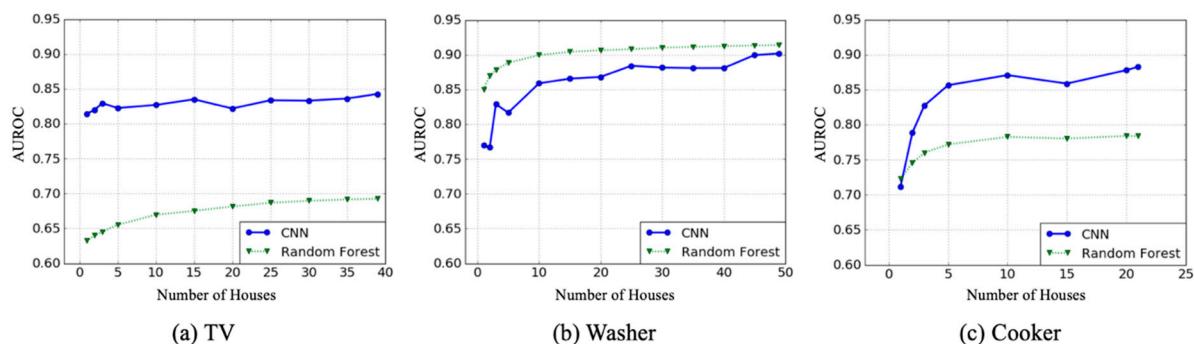
In summary, we have found that both classification and regression suffer as the sampling rate decreases, except for TVs, for which the algorithms generally perform poorly. Therefore, it is crucial to use datasets with a proper sampling rate when running or developing NILM algorithms. For the simple tasks that we have investigated, at least 1 Hz and 3 Hz sampling rates are required for classification and regression, respectively.

## 6.2. Sensitivity to Number of Houses

Another important aspect of data quality is the number of individual houses that are used for training. We conducted experiments to find out how the performance of the NILM algorithm changed with the changes in the number of houses.

For the binary classification task, we used the leave-one-out evaluation method to use the data maximally, where the data of one house was left out for testing while the rest were used for training for each fold. We used the same method as in the previous section to check the change in AUROC as a function of the number of distinct houses in the training dataset. We first trained NILM algorithms using the maximum number of houses (as described in Section 5), and then gradually reduced the number of distinct houses. House selection for inclusion in the training dataset was conducted randomly. Since the performance was dependent on the actual selection of the individual houses in the training set, we repeated these experiments 20 times, for which the houses selected for training were randomly shuffled, and averaged the results.

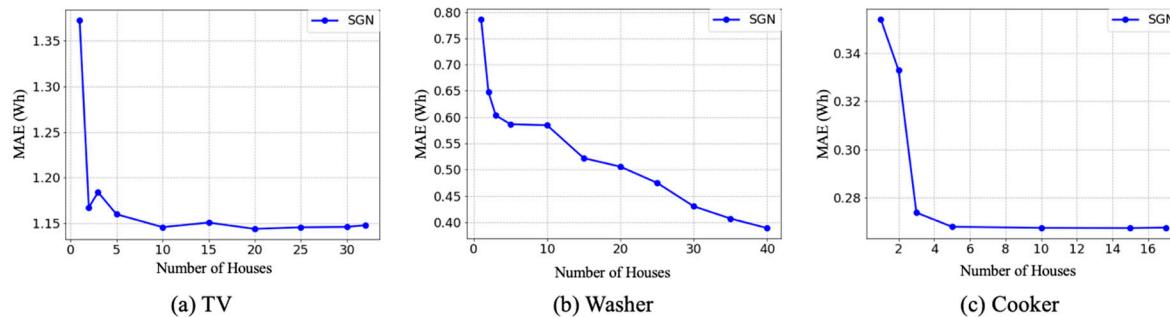
The classification results are shown in Figure 12. In general, AUROC improved as the number of houses increased. For the Random Forest algorithm, all TVs, washers, and rice cookers showed a monotonic improvement as the number of houses increased. The main improvements, however, were observed to occur as the number of houses increased to 10–30. Among the three, rice cookers showed the fastest ramp up, and the additional gain after 5–10 houses was small. For TVs, the improvement was slow and steady, and continued all the way to 30–40 houses. For one-layer CNN, the general trend of improved performance for a larger number of houses was the same, but the shape of the curves was different. The performance of TVs did not improve much for a larger number of houses. Washers showed improvement all the way up to 40–50 houses. The performance of rice cookers improved sharply as the number of houses increased to 5–10, as in the Random Forest algorithm.



**Figure 12.** Sensitivity of classification algorithms to the number of distinct houses used in the training data. As the number of houses increases, the test performance tends to improve. Owing to the limitation in the dataset, up to 39, 49, and 21 houses were investigated for TVs, washers, and rice cookers, respectively.

The regression results are shown in Figure 13. For the regression task, experiment conditions were the same as in the binary classification task except that we used five-fold cross validation instead of leave-one-out. For an obvious reason, the houses selected for the test were excluded from the training dataset, as in the classification experiments. Further, the averaging over 20 repeated experiments was performed for regression as well. These results are shown in Figure 13. Overall, similar trends as in the classification were observed. For the three appliances, the figures showed a sharp drop of MAE when

the number of houses was increased to 3–5. For washers, a steady improvement continued until the maximum number of 40 houses were included in the training. The signatures of washers are known to be quite recognizable, but they vary over different manufacturers and models. Therefore, including more houses in training implies a higher chance of having the signatures of the target washer being included in the training data.



**Figure 13.** Sensitivity of the regression algorithm to the number of distinct houses used in the training data. As the number of houses increases, the test performance tends to improve. Owing to the limitation in the dataset and five-fold cross validation setup, up to 32, 40, and 17 houses were investigated for TVs, washers, and rice cookers, respectively.

The experiment results of the two tasks show that it is very important to include data from a sufficiently large number of distinct houses as in the case of sampling rate. Overall, it can be concluded that the number of houses needs to be at least in the range of five to ten for TVs and cookers, and at least tens of houses for washers. For a general study where many appliance types are studied, it would be best to include at least tens of houses. In Figure 2, note that the UK-DALE and REDD datasets have only five and six houses each.

## 7. Limitations of this Study

Nowadays, uncountably many types of appliances exist. Furthermore, each type of appliance (e.g., TV) can be manufactured with a variety of base technologies, where the characteristic of the electric signature is significantly affected by the choice of the base technology. The increasing diversity is making NILM an increasingly challenging problem. In our work, we have investigated the data requirements in terms of sampling rate and number of houses using the ENERTALK dataset. Compared to the traditionally popular datasets, the ENERTALK dataset is undoubtedly much richer in information thanks to its higher sampling rate and larger number of houses. Nonetheless, the ENERTALK dataset is still limited in size, information, and labeling, and thus our study ends up with its own limitations.

First, we were able to study only three types of appliances. The characteristics of signatures can widely vary for other appliance types, and a further study is needed if a more general conclusion on the sampling rate and number of house requirements is desired. Secondly, the three types of appliances happen to have signatures that can be recognized by a human when a 90-s block of data is presented. This might not be true for other appliance types, and a longer block size might need to be used for studying such appliance types. Thirdly, the dataset was collected only from Japan. For another country or a region, the popular appliances will be different and furthermore the usage patterns of the local population will be different, too. Therefore, a minimum requirement for drawing a general conclusion is to study datasets collected from at least a few or possibly several different countries. The last limitation that we would like to address is the set of algorithms that we have used in this study. We have done our best to cover the most representative categories of the NILM algorithms. However, the NILM algorithm is continuously evolving, especially with the recent developments based on deep learning, and the data requirements can be dependent on the choice of algorithm.

Despite the limitations discussed above, our case study based on the ENERTALK dataset clearly points out a few important insights. The sensitivity study results clearly indicate that many of the

existing studies could have produced much better performance results if better datasets had been used. In fact, it is unclear which algorithms would be the best performing ones when a higher quality dataset is used. Therefore, we believe it would be prudent for the research community to establish a common understanding on what the exact data requirements for studying NILM algorithms are, and to create several datasets that meet the requirements. Ideally, the datasets would be collected from different continents. Our study cannot pinpoint the data requirements, but we believe it is one of the first attempts, if not the first, for establishing sound guidelines on the data requirements for studying NILM.

## 8. Conclusions

The potential and benefit of a real-world NILM service are known, but its real-world deployment has been limited so far. We first summarized the possible NILM services that were compiled by interacting with business experts and users of an energy IoT platform. The wide spectrum of real-world services indicates that NILM research should not be limited to the original regression framework, and other frameworks such as classification, detection, and recommendation should be considered as well. It is widely known that NILM is a very difficult problem, but some of the real-world services can be much easier to provide because the underlying algorithms and performance goals are less challenging. Then, using a new dataset called ENERTALK, we investigated the data quality requirements for developing NILM algorithms. For the case we have investigated, our study on data sampling rate showed that at least 1–3 Hz sampling rate is required to prevent NILM performance from deteriorating significantly in the case of the ENERTALK dataset. This is because of how signatures of real-world appliances appear—when the sampling rate is too low, the signatures are destroyed. The study conducted on the number of distinct houses in the training data set indicated that at least tens of distinct houses need to be included in the training dataset in the case of the ENERTALK dataset. Otherwise, the NILM performance started to deteriorate substantially. This might be natural, because there are many manufacturers and product models for each appliance type (e.g., TV), and our goal is to find common traits or an exhaustive list of traits of all within an appliance category. The sensitivity studies indicate that the existing public NILM datasets might need to be used with caution because of the limited sampling rate and house counts.

**Author Contributions:** Conceptualization, C.S., S.R., H.L., and W.R.; methodology, W.R. and S.R.; software, C.S. and S.R.; validation, S.R., C.S., and W.R.; formal analysis, S.R., C.S., and W.R.; investigation, S.R., C.S. and W.R.; resources, H.L. and W.R.; data curation, H.L., C.S., and S.R.; writing—original draft preparation, S.R., C.S., and W.R.; writing—review and editing, C.S., S.R., and W.R.; visualization, S.R. and C.S.; supervision, W.R.; project administration, S.R. and C.S.; funding acquisition, H.L. and W.R.

**Funding:** This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20151210200080) and by the National Research Foundation of Korea (NRF) (No. NRF-2017R1E1A1A03070560).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Hart, G.W. Nonintrusive appliance load monitoring. *Proc. IEEE* **1992**, *80*, 1870–1891. [[CrossRef](#)]
2. Neenan, B.; Robinson, J.; Boisvert, R. *Residential Electricity Use Feedback: A Research Synthesis and Economic Framework*; Electric Power Research Institute: Palo Alto, CL, USA, 2009.
3. Froehlich, J.; Larson, E.; Gupta, S.; Cohn, G.; Reynolds, M.S.; Patel, S.N. Disaggregated end-use energy sensing for the smart grid. *IEEE Pervasive Comput.* **2010**, *1*, 28–39. [[CrossRef](#)]
4. Armel, K.C.; Gupta, A.; Shrimali, G.; Albert, A. Is disaggregation the holy grail of energy efficiency? The case of electricity. *Energy Policy* **2013**, *52*, 213–234. [[CrossRef](#)]
5. Baechler, M.C.; Hao, H. *Business Case for Nonintrusive Load Monitoring*; No. PNNL-25425; Pacific Northwest National Laboratory (PNNL): Richland, WA, USA, 2016.
6. Rashid, H.; Singh, P.; Stankovic, V.; Stankovic, L. Can non-intrusive load monitoring be used for identifying an appliance’s anomalous behaviour? *Appl. Energy* **2019**, *238*, 796–805. [[CrossRef](#)]

7. Bonfigli, R.; Felicetti, A.; Principi, E.; Fagiani, M.; Squartini, S.; Piazza, F. Denoising autoencoders for non-intrusive load monitoring: improvements and comparative evaluation. *Energy Build.* **2018**, *158*, 1461–1474. [[CrossRef](#)]
8. Shin, C.; Joo, S.; Yim, J.; Lee, H.; Moon, T.; Rhee, W. Subtask Gated Networks for Non-Intrusive Load Monitoring. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019.
9. Liang, J.; Ng, S.K.; Kendall, G.; Cheng, J.W. Load signature study—Part I: Basic concept, structure, and methodology. *IEEE Trans. Power Del.* **2010**, *25*, 551–560. [[CrossRef](#)]
10. Dong, M.; Meira, P.C.; Xu, W.; Chung, C. Non-intrusive signature extraction for major residential loads. *IEEE Trans. Smart Grid* **2013**, *4*, 1421–1430. [[CrossRef](#)]
11. Kolter, J.Z.; Batra, S.; Ng, A.Y. Energy disaggregation via discriminative sparse coding. In Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 6–11 December 2010; pp. 1153–1161.
12. Elhamifar, E.; Sastry, S. Energy disaggregation via learning powerlets and sparse coding. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15), Austin, TX, USA, 25–30 January 2015; pp. 629–635.
13. Singh, S.; Gulati, M.; Majumdar, A. Greedy deep disaggregating sparse coding. In Proceedings of the 3rd International Workshop on Non-Intrusive Load Monitoring, Vancouver, BC, Canada, 14–15 May 2016.
14. Ghahramani, Z.; Jordan, M.I. Factorial hidden Markov models. *Mach. Learn.* **1997**, *29*, 245–273. [[CrossRef](#)]
15. Kim, H.; Marwah, M.; Arlitt, M.F.; Lyon, G.; Han, J. Unsupervised disaggregation of low frequency power measurements. In Proceedings of the 2011 SIAM International Conference on Data Mining, Mesa, AZ, USA, 28–30 April 2011; pp. 747–758.
16. Parson, O.; Ghosh, S.; Weal, M.; Rogers, A. Non-intrusive load monitoring using prior models of general appliance types. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012; pp. 356–362.
17. Zhong, M.; Goddard, N.; Sutton, C. Signal aggregate constraints in additive factorial HMMs, with application to energy disaggregation. In Proceedings of the Twenty-Eighth Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 3590–3598.
18. Shaloudegi, K.; György, A.; Szepesvari, C.; Xu, W. SDP relaxation with randomized rounding for energy disaggregation. In Proceedings of the Thirtieth Annual Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; pp. 4978–4986.
19. 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, Seoul, Korea, 4–5 November 2015; pp. 55–64.
20. Huss, A. Hybrid Model Approach to Appliance Load Disaggregation: Expressive Appliance Modelling by Combining Convolutional Neural Networks and Hidden Semi Markov Models. Master’s Thesis, KTH Royal Institute of Technology School of Computer Science and Communication, Stockholm, Sweden, 2015.
21. Zhang, C.; Zhong, M.; Wang, Z.; Goddard, N.; Sutton, C. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
22. Chen, K.; Wang, Q.; He, Z.; Chen, K.; Hu, J.; He, J. Convolutional Sequence to Sequence Non-intrusive Load Monitoring. *arXiv* **2018**, arXiv:1806.02078. [[CrossRef](#)]
23. Kolter, J.Z.; Johnson, M.J. REDD: A public data set for energy disaggregation research. In Proceedings of the Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA, USA, 21 August 2011; pp. 59–62.
24. Anderson, K.; Ocneanu, A.; Benitez, D.; Carlson, D.; Rowe, A.; Berges, M. Blued: A fully labeled public dataset for event-based nonintrusive load monitoring research. In Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD), Beijing, China, 12 August 2012; pp. 1–5.
25. Monacchi, A.; Egarter, D.; Elmenreich, W.; D’Alessandro, S.; Tonello, A.M. Greend: An energy consumption dataset of households in Italy and Austria. In Proceedings of the IEEE International Conference on Smart Grid Communications (SmartGridComm), Venice, Italy, 3–6 November 2014; pp. 511–516.

26. 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, Cambridge, UK, 11–13 June 2014; pp. 265–276.
27. 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, Memphis, TN, USA, 5–6 November 2014; pp. 80–89.
28. Kelly, J.; Knottenbelt, W. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Sci. Data* **2015**, *2*, 150007. [CrossRef] [PubMed]
29. Alahakoon, D.; Yu, X.-H. Smart electricity meter data intelligence for future energy systems: A survey. *IEEE Trans. Ind. Inform.* **2016**, *12*, 425–436. [CrossRef]
30. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
31. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
32. Bergstra, J.S.; Remi, R.; Bengio, Y.; Kegl, B. Algorithms for hyper-parameter optimization. In Proceedings of the Twenty-Fifth Annual Conference on Neural Information Processing Systems, Granada, Spain, 12–17 December 2011; pp. 2546–2554.
33. Brochu, E.; Cora, V.M.; Freitas, N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv* **2010**, arXiv:1012.2599.
34. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. In Proceedings of the Twenty-sixth Annual Conference on Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–8 December 2012; pp. 2951–2959.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).