Non-Intrusive Load Monitoring: Comparative Analysis of Transient State Clustering Methods

Mozaffar Etezadifar, Houshang Karimi, *Senior Member, IEEE,* and Jean Mahseredjian, *Fellow, IEEE*

***Abstract*—Non-intrusive load monitoring is one of the key tools in demand-side management (DSM). Recent advancements in the computational power of processors have accentuated the role of machine learning algorithms e.g., clustering, as a key function in the NILM solutions applied on power grids. In event-based NILM methods, the algorithm detects the transient states (load events) and clusters them based on the similarity of different features of the transient state. In this study, the performances of eight clustering algorithms are comprehensively investigated and the impact of choosing different input signals, e.g., P, Q, and I, on transient states clustering is analyzed. Various input signals from the BLUED dataset are fed to the clustering algorithms. By comparing the evaluation metrics including shape-based and ground-truth-based metrics, it is observed that the OPTICS algorithm fed by dual-stream input streams outperformed the rest of the investigated clustering algorithms and input sets. OPTICS algorithm groups load events based on their density in multi-dimensional space, using a dynamic radius. The OPTICS algorithm, as the best-performing transient state clustering algorithm for the low-frequency NILM purpose, is then tested with the downsampled input data in a wide frequency range, to observe the impact of the data-sampling frequency on the results, which simplifies the use of clustering algorithms in future studies.**

***Keywords*—Non-intrusive load monitoring, unsupervised learning, clustering, transient state, load disaggregation, machine learning, NILM**

NOMENCLATURE

KM	K-means
AP	Affinity propagation
MS	Mean shift
SC	Spectral clustering
HC	Hierarchical clustering
DB	DBSCAN
OP	OPTICS
BI	BIRCH
Sil	Silhouette
ARI	Adjusted rand index
V	V-measure
FM	Fowlkes-Mallow

I. INTRODUCTION

LOAD Monitoring (LM) plays an important role in smart grids. Having an effective LM in the grid provides opportunities to implement more advanced load forecasting, fault detection, and various DSM techniques [1].

Generally, LM approaches are categorized into two categories: intrusive (ILM) and non-intrusive (NILM). In the NILM method, which was first introduced by George Hart in 1992 [2], aggregated load measurements from a single-entry point are collected and load disaggregation is performed via different algorithms.

The majority of NILM studies involve key modules e.g., data acquisition, extraction of appliance features, and appliance classification [3]. In the data acquisition as the first step, the sampling frequency is a key factor. It can determine the feature extraction possibilities. The high-frequency sampling allows us to investigate the NILM methods that need transient state analysis, e.g., frequency responses [4], voltage noise [5], or harmonics [6]. In general, high-frequency NILM methods result in more precise load disaggregation over different load types [7], however, it comes with its costs e.g., larger data storage, initial investment on high-frequency samplers, and more complex hardware. Low-frequency NILM methods have attracted more attention recently, due to the fact that the majority of installed smart meters around the world generate low-frequency output [8]. Consequently, they do not impose expensive initial investments on the grid.

In the feature extraction step, we try to transform input data into a unique set of variables with which the NILM algorithm can identify different appliances. After extracting the features from the raw input data, a NILM algorithm should categorize the appliances based on their load signature features. Several artificial-intelligence-based algorithms have been employed by researchers for this step. Generally, they can fall into two groups: supervised classification and unsupervised clustering methods [8]. Supervised learning performs a good load disaggregation on the load profiles with which it has been trained [4]. Nevertheless, it is difficult, if not impossible, to create a labeled library of load signatures for every household, given the wide range of appliance manufacturers and types [9]. In contrast, unsupervised learning does not need labeled training datasets, which makes unsupervised NILM methods suitable candidates for real-world NILM applications [10].

In the existing literature, numerous studies have used machine learning (ML) algorithms as a part of a NILM solution [11]. However, a few papers have focused on the performance of these ML algorithms as an independent step in a NILM process. Because of the popularity of

This work was supported by NSERC, Hydro-Québec, RTE, EDF, and OPAL-RT as a part of the industrial chair "Multi time-frame simulation of transients for large scale power systems." (Corresponding author: Mozaffar Etezadifar). M. Etezadifar, H. Karimi and J. Mahseredjian are with the Department of Electrical Engineering, Polytechnique Montreal, Montreal QC H3T 1J4, Canada (e-mail: mozaffar.etezadifar@polymtl.ca; houshang.karimi@polymtl.ca; jean.mahseredjian@polymtl.ca)

Paper submitted to the International Conference on Power Systems Transients (IPST2023) in Thessaloniki, Greece, June 12-15, 2023.

supervised algorithms, some researchers have evaluated the performance of supervised algorithms in the event detection and classification step of a NILM solution. Authors in [12] compare five supervised algorithms for the load classification step with high-frequency data (30 kHz). In [13] authors compared the performance of ten supervised classification algorithms on low-frequency P data. Unsupervised clustering algorithms have been used in many NILM studies for load clustering, however, there is a lack of information on comparing unsupervised clustering algorithms' performance as a part of a NILM solution in the literature. In [14] authors used DBSCAN to cluster load events of a few high-consumption appliances. In [15] the authors used the K-means algorithm to cluster different power levels, despite the fact that K-means needs the number of clusters. Hierarchical clustering is used in [16] to cluster high-frequency load profiles by clustering harmonics and electromagnetic induction signals. In [17] HDBSCAN (a variation of DBSCAN) is used to cluster segments of the time series of a load profile. Authors in [18] used OPTICS and DBSCAN to cluster low-frequency load transitions. Despite the variety of unsupervised clustering algorithms that are used as an important step in NILM solutions, none of these studies explain why they have chosen a specific clustering algorithm for their load clustering purpose. There is a lack of metric-based comparison for choosing each of these unsupervised algorithms in every condition.

To fill this information gap, this paper proposes a comparative evaluation of eight unsupervised clustering algorithms for the NILM purpose fed by different input signals (P, Q, I). Different metrics are calculated to evaluate each algorithm's performance on clustering of load events. The best-performing algorithm is further tested by clustering load events in a frequency range (60 Hz to 1/600 Hz).

The main contributions of this study are summarized as follows:

- Comparing and analysis of the performance of the most commonly used unsupervised clustering algorithms for the NILM purpose under different input conditions;
- Investigating the performance of unsupervised clustering algorithms facing different load transient cases;
- Identifying and analysis of the best-performing clustering algorithm for the NILM purpose by comparing several effective metrics; and
- Examining the flexibility and tolerance of the best-performing clustering algorithm in a non-ideal frequency range.

II. Background

A. The clustering process

In NILM studies the first step is to collect the input data. The input data, e.g., active or reactive power, harmonics, or EMI (electromagnetic interference), then are processed to fit the requirements of the event detection algorithm. After determining the time steps in which the events have occurred, the clustering algorithm puts similar load transient states into the same clusters. Based on the similarities between each cluster and the load transients in the appliances database, the NILM algorithm assigns a name to each cluster. Finally, the power and energy estimation for every load is calculated. Fig. 1 illustrates the above-mentioned process. As it is observed, the clustering algorithm plays a key role in the NILM solution. Let's introduce each clustering algorithm briefly.

K-means (KM) is one of the most popular unsupervised clustering algorithms which scales well to a large number (N) of samples (X) and divides them into K disjoint clusters (C). Each cluster is represented by the mean of its sample (μ_j) which is called a centroid. The K-means algorithm chooses the centroids that minimize the inertia term as follows:

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2) \qquad (1)$$

Despite the acceptable performance of the K-means algorithm on datasets with even-shaped and convex clusters, the necessity of predetermining the number of clusters can be an obstacle for using K-means in NILM problems with numerous appliances (i.e., clusters). It is noteworthy that in the presence of few clusters, the true quantity of clusters can be found by running the K-means over a range of clusters number and choosing the elbow point.

Affinity propagation (AP) distinguishes clusters by analyzing the responsibility (r(i,k)), availability (a(i,k)), and similarity (s(i,k)) values between all sample pairs (i, k). The responsibility of sample k to be in the same cluster as sample i is given by:

$$r(i,k) \leftarrow s(i,k) - max[a(i,k') + s(i,k') \, \forall \, k' \neq k] \qquad (2)$$

and the availability of sample k to be in the same cluster as sample i is given by:

$$a(i,k) \leftarrow min[0, r(k,k) + \sum_{i' s.t \, i' \notin (i,k)} r(i',k)] \qquad (3)$$

as this operation is repeated for all sample pairs, affinity propagation is of high time and memory complexity, $O(N^2T)$ and $O(N^2)$ respectively, where N is the number of samples and T is the number of iterations.

Mean shift (MS), similar to K-means, is a centroid-based algorithm. Centroid x_i is initialized randomly, and in every iteration, it is updated by the mean of the samples in a certain bandwidth as follows:

$$x_i^{t+1} \leftarrow m(x_i^t) \qquad (4)$$

$$m(x_i) = \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i)x_j}{\sum_{x_j \in N(x_i)} K(x_j - x_i)} \qquad (5)$$

where K is a kernel function, and $N(x_i)$ is the neighborhood of samples in a given bandwidth around x_i.

Spectral clustering (SC), requires the number of clusters to be specified in advance, as in K-means. Spectral clustering uses eigenvectors of the affinity matrix between samples and performs clustering on its components. This algorithm is not advised for problems with a high number of clusters.

Hierarchical clustering (HC) is a general family of clustering algorithms. Agglomerative clustering is a specific type of hierarchical algorithm that uses a bottom-up approach. Each sample starts in its own cluster and clusters merge together successively based on the linkage criteria.

DBSCAN (DB), unlike K-means which focuses on the average of clusters to find centroids, calculates the density of the clusters. Therefore, DBSCAN is a suitable algorithm for non-convex clusters. DBSCAN considers a sample a "core sample" if there is a minimum number of samples in a distance less than a bandwidth around that. In contrast, there are "border" or "non-core" samples which are samples in the bandwidth range of the core samples, but without the minimum number of samples around them. Any non-core sample that is farther than bandwidth from the core samples is considered an outlier.

OPTICS (OP) is the general form of the DBSCAN algorithm. In DBSCAN the bandwidth is a fixed value while in OPTICS the bandwidth is a range that helps the OPTICS algorithm find clusters in a more flexible way.

BIRCH (BI), works efficiently on large datasets as it generates a compact summary of the dataset in its clusters called clustering feature (CF). The clustering feature consists of a number of data points (N), linear sum of data points (LS), and squared sum of data points. Each new sample entering the clustering feature tree joins the leaf (node) with which it has the highest similarity.

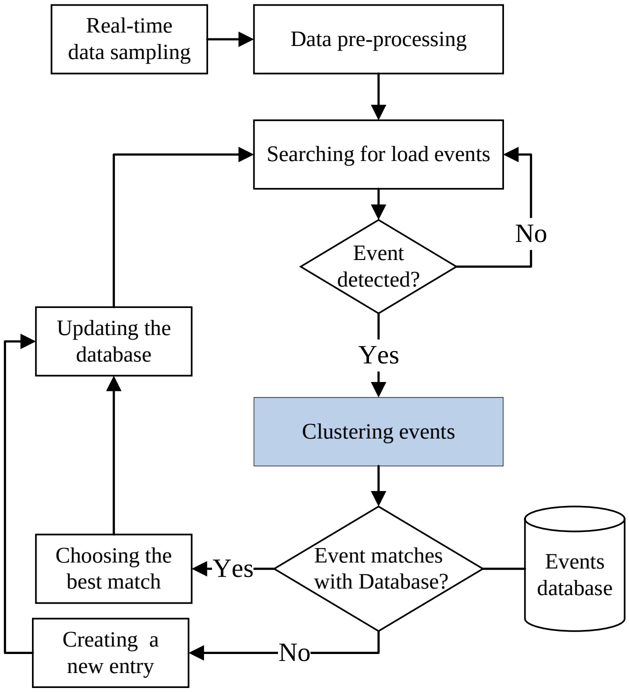

Fig. 1. The steps of the NILM algorithm using clustering

B. Clustering algorithms evaluation

Evaluating the performance of clustering algorithms is not the same as supervised classification algorithms due to the absence of ground truth data in real-world problems. Therefore, metrics for evaluating clustering algorithms are based on analyzing the quality of the clusters. Denser and better-separated clusters mean the clustering algorithm is performing better. However, as in this study we have access to the ground truth data, we can employ more accurate evaluation methods on clustering algorithms. The following metrics have been used in this study. Adjusted Rand Index (ARI) measures the similarity of the ground truth label and clustering assignment, ignoring the permutations. ARI ranges from -1 to 1, and a score of 1 represents perfect labeling while -1 indicates random labeling. ARI is calculated as follows:

$$RI = \frac{a+b}{C_2^{n_{samples}}} \tag{6}$$

Given C is the ground truth class assignment and K is the clustering, a and b can be defined as follows: a is the number of pairs of samples that are in the same sets in both C and K. b is the number of pairs of samples that are in different sets in both C and K. $C_2^{n_{samples}}$ is the total number of possible pairs in the entire dataset. To eliminate the effect of random labeling, expected RI ($E[RI]$) is used to calculate ARI.

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]} \tag{7}$$

V-measure (V) is the next important metric that needs ground truth labels. *V-measure* is a compound metric made of *homogeneity* and *completeness*. Homogeneity means that each cluster contains only members of a single class. Completeness means that all members of a class are assigned to the same cluster. It ranges from 0 to 1, and 1 represents perfect clustering.

$$V = \frac{(1+\beta) \times homogeneity \times completeness}{(\beta \times homogeneity + completeness)} \tag{8}$$

Fowlkes-Mallow (FM) score is another metric that is based on having ground truth data. It is defined as below:

$$FM = \frac{TP}{\sqrt{(TP+FP)(TP+FN)}} \tag{9}$$

where TP, FP, and FN stand for true positive, false positive, and false negative respectively. FM ranges from 0 to 1 and a high value indicates good similarity between two clusters.

Silhouette (Sil) coefficient, unlike previous metrics, does not need ground truth labels to evaluate the performance of a clustering algorithm. A higher silhouette coefficient indicates better-defined clusters. For each sample, the silhouette coefficient is calculated as below:

$$Sil = \frac{b-a}{max(a,b)} \tag{10}$$

where *a* is the mean distance between a sample and all other points in the same class, and *b* is the mean distance between a sample and all other points in the next nearest class. The total silhouette coefficient for a dataset is the mean of all silhouette coefficients of samples. This metric ranges from -1 to 1, and 1 indicates very dense clusters.

III. Research Methodology

In this study, eight different clustering algorithms have been evaluated by different metrics in the context of the NILM. The algorithms, process, and evaluation metrics are briefly explained in section II. The algorithms are fed by various pre-processed electrical signals. Then, the best-performing method is chosen to be tested in a range of data-sampling frequencies to further investigate the effect of sampling frequency on the clustering algorithms' performance. Fig. 2 illustrates the flow of the research methodology.

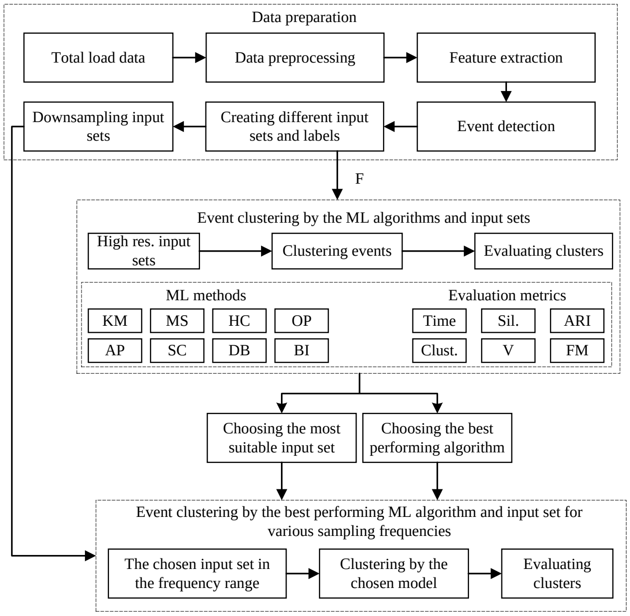
Data preparation
Total load data
Data preprocessing
Feature extraction
Event detection
Creating different input sets and labels
Downsampling input sets
F
Event clustering by the ML algorithms and input sets
High res. input sets
Clustering events
Evaluating clusters
ML methods
KM
MS
HC
OP
AP
SC
DB
BI
Evaluation metrics
Time
Sil.
ARI
Clust.
V
FM
Choosing the most suitable input set
Choosing the best performing algorithm
Event clustering by the best performing ML algorithm and input set for various sampling frequencies
The chosen input set in the frequency range
Clustering by the chosen model
Evaluating clusters

Fig. 2. The research flow

The dataset for this study is the BLUED dataset presented in [19]. This dataset contains the power consumption data of a house in Pennsylvania, US for 8 days. Electrical signals e.g., voltage (V), and current (I) have been sampled at the rate of 12 KHz [19]. However, corresponding active power (P) and reactive power (Q) are sampled at 60 Hz. Different home appliances were sampled in this dataset from which we chose four appliances. The air conditioner, fridge, iron, and wall socket (the same appliance is connected to this socket during the measurements) are chosen due to their load signature and frequent usage in the dataset. Two load signatures have significant transient states and the two others have symmetric shapes. Fig. 3 illustrates one cycle of load signatures of these four appliances, sampled at 1 Hz frequency.

The collected data is pre-processed (cleaned and normalized). Then, it is fed to the event detection function which is supposed to find the time step t in which the ON or OFF event has happened. In this study, the event detection algorithm is based on the Log Likelihood Ratio detector by Voting (LLR voting) [20]. As the focus of this paper is comparing the clustering algorithms' performance in the context of the NILM, and not the event detection process, the LLR Voting method is not detailed here. However, for the reproduction purpose, it is worth mentioning that for 1Hz frequency, the power threshold is set to 20 Watt, post and pre-window ($\omega 0 = \omega 1$) lengths are 5, voting window ω_v is 10, and voting threshold θ_v is 4. After the event detection algorithm identified the timestamps of the load events (transitions), a window of 5 samples before and after the timestamp is extracted as the transient state of that load event. For each extracted transient state a feature set is defined as below:

$$F = \{\sigma\{P_t, Q_t, I_t\}, \Delta\{P_t, Q_t, I_t\}\ \forall t = \text{load event}\} \quad (11)$$

where σ is the standard deviation and Δ represents the difference between the first and last data point of P, Q, and I transient state. In this study, different combinations of these three variables are fed to the clustering algorithms to investigate their effect on the clustering results.

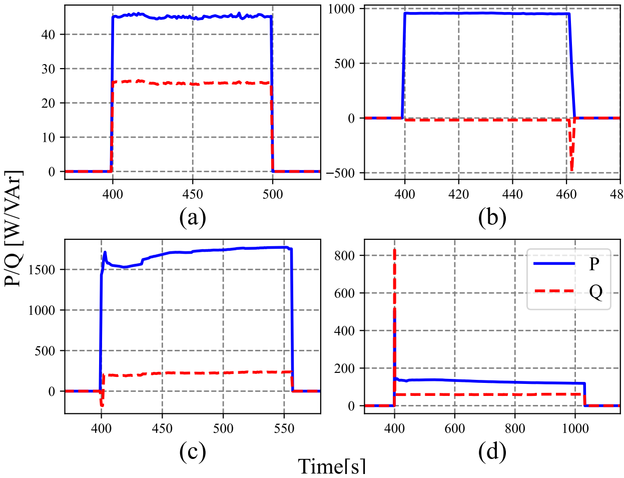

Fig. 3. Load signatures of the appliances of the BLUED dataset: a) wall socket, b) iron, c) AC, and d) fridge

IV. Simulation on Real-World Data and Discussion

Based on the methodology explained in section III, a comprehensive simulation is carried out. This simulation was run on a Lenovo T590 laptop with 16 Gigabytes of RAM and Intel core-i7 CPU @ 1.9 GHz. Python 3.8 was the main coding tool.

Forty-eight hours of data from BLUED dataset was cleaned and normalized to be fed to the clustering algorithms. Selected clustering algorithms' parameters were optimized to have the best performance on the input data sampled at 1 Hz frequency. Table I presents their parameters setting.

To investigate the impact of input electrical signal type on the clustering of the load transient states, the following input sets were chosen: I, P, Q, PI, QI, PQ, and PQI. Each of these input sets is processed by clustering algorithms, and their output labels are evaluated by the metrics introduced in section II. Some metrics do not need ground truth data. Thus, they only measure the quality of the clusters which is dependent on how dense and well-separated the clusters are. Silhouette metric is of this type. In contrast, some other metrics, e.g., ARI or V-measure, need ground truth labels to

TABLE I
CLUSTERING ALGORITHMS PARAMETERS

Method	Parameters
K-means	init='k-means++', n-clusters=9
Affinity propagation	max-iter=200000
Mean shift	min-bin-freq=3, max-iter = 5000, bandwidth=auto
Spectral clustering	n-clusters=9
Hierarchical clustering	distance-threshold=0.35,linkage = 'ward'
DBSCAN	eps=0.01, min-samples=15
OPTICS	min-samples=15, xi = 0.1
BIRCH	threshold=0.05,branching-factor = 50

determine how accurate the clustering is. However, the ground truth labels do not exist in real-world clustering problems. Thus, the combination of these two types of metrics provides us with proper judgment on the performance of the clustering algorithms in each scenario. Table II presents how every clustering method has performed given different input sets.

In Table II, column "Time" is the time that the clustering algorithm took to perform the clustering and it is in seconds. The next column is the number of identified clusters. The Silhouette, ARI, V-measure, and FM are calculated in the next columns. The last column is the average of the last three metrics (ARI, V-measure, and Fowlkes-Mallows). The last row for every method represents the average of that column. The number of clusters is not averaged as it does not convey meaningful information about the performance of the clustering method.

Since in real-world problems we deal with abundant data samples, the speed and consumed time of the clustering algorithm are of great importance. Affinity propagation (AP) takes 84 seconds on average to do the clustering on this dataset. K-means and Spectral clustering do the clustering faster, however, it should be noted that they need to find the optimal cluster numbers using the elbow method. For this dataset with a limited amount of events, this process should be run 15 times (15 clusters). Thus, it is fair to multiply their taken time by 10 which makes them a very time-consuming method same as the affinity propagation. Among other methods, the OPTICS (OP) has the highest mean performance which is 78%. With a clustering time of 0.16 s, it is the slowest method among other available methods, however, its high score in both types of metrics (shape-based and ground-truth-based) makes it a suitable method for the clustering purpose. Although Mean shift (MS), DBSCAN (DB), and BIRCH (BI) are fast clustering methods, none of them was able to identify the true number of clusters. Moreover, their mean performance score is also low (46%, 56%, and 47% respectively) compared to other methods which shows the lack of accuracy in their clustering process. Hierarchical clustering (HC) is the only fast method among these clustering methods which has an acceptable mean performance score (61%) compared to that of the OPTICS, and was able to determine the true number of clusters in most cases. Considering the above-mentioned comparison, OPTICS seems to be the best candidate for being tested with lower data sampling frequencies. Fig. 4 illustrates how different methods have performed with different input signals for the clustering purpose.

TABLE II
CLUSTERING ALGORITHMS PERFORMANCE

Method	Signal	Time	Clust.	Sil	ARI	V	FM	Mean
KM	I	0.58	9	72	59	83	66	69
	P	0.06	9	74	58	81	65	68
	Q	0.09	9	78	56	74	62	64
	PQ	0.04	9	72	62	81	68	70
	PI	0.05	9	71	64	82	69	71
	QI	0.045	9	73	64	82	69	71
	PQI	0.04	9	71	63	82	68	71
	Mean	0.13	NA	73	61	80	66	69
AP	I	365	5	66	49	75	62	62
	P	2.8	5	86	33	70	53	52
	Q	14	15	73	52	71	58	60
	PQ	1.71	8	69	48	75	58	60
	PI	183	6	67	55	79	65	66
	QI	4.75	8	69	48	75	58	60
	PQI	21	8	69	48	75	58	60
	Mean	84	NA	71	47	74	58	60
MS	I	0.04	5	86	33	71	53	52
	P	0.034	5	86	33	70	53	52
	Q	0.04	5	79	9	43	37	29
	PQ	0.03	7	83	28	67	47	47
	PI	0.03	3	84	28	58	49	45
	QI	0.03	7	83	28	67	47	55
	PQI	0.03	5	81	32	69	52	51
	Mean	0.033	NA	83	27	63	48	46
SC	I	0.96	9	19	36	48	44	42
	P	1.2	9	9	41	65	55	53
	Q	3.3	9	78	56	74	62	64
	PQ	3.1	9	72	62	81	68	70
	PI	0.64	9	20	37	71	53	53
	QI	2	9	73	63	82	69	71
	PQI	2.29	9	68	46	74	56	58
	Mean	1.92	NA	48	48	70	58	58
HC	I	0.009	5	86	33	71	53	52
	P	0.009	5	86	33	70	53	52
	Q	0.006	6	65	29	60	47	45
	PQ	0.004	9	72	62	81	68	70
	PI	0.012	6	67	55	80	65	66
	QI	0.007	9	73	64	82	69	71
	PQI	0.01	9	71	63	82	68	71
	Mean	0.008	NA	74	48	75	60	61
DB	I	0.004	5	32	47	73	59	59
	P	0.008	4	22	32	61	49	47
	Q	0.007	3	38	42	63	55	53
	PQ	0.0001	7	35	45	70	54	56
	PI	0.001	7	38	55	76	62	64
	QI	0.004	7	38	50	73	58	60
	PQI	0.003	7	32	44	70	54	56
	Mean	0.003	NA	33	45	69	55	56
OP	I	0.17	9	86	85	91	87	87
	P	0.15	9	82	81	87	83	83
	Q	0.21	6	24	30	57	42	43
	PQ	0.18	9	72	76	84	78	79
	PI	0.15	9	83	85	90	86	87
	QI	0.16	9	75	83	90	85	87
	PQI	0.15	9	76	82	89	84	85
	Mean	0.16	NA	71	74	84	77	78
BI	I	0.008	5	86	33	71	53	52
	P	0.009	5	86	33	70	53	52
	Q	0.007	5	79	9	43	37	29
	PQ	0.009	8	81	28	66	47	47
	PI	0.009	5	86	33	70	53	52
	QI	0.009	7	83	28	67	47	47
	PQI	0.009	10	64	40	74	53	55
	Mean	0.008	NA	80	29	65	49	47

Sil, ARI, V, and FM are in percent (%)

According to Fig. 4, a single Q stream is the worst input for these clustering algorithms in most cases. A single P

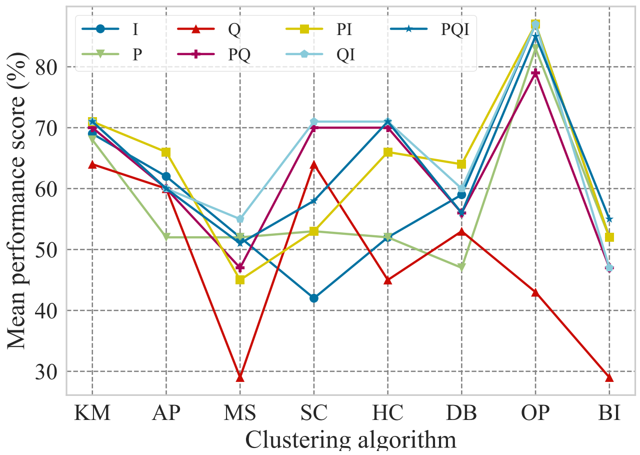

Fig. 4. Mean performance of different clustering methods for different inputs

input is also a bad candidate for clustering purposes, however, in three algorithms it has led to scores as high as the best inputs. A single current (I) input has led to high scores in half the algorithms. Dual-stream inputs e.g., PQ, PI, and QI have almost the same results in most algorithms, however, all of them have higher scores than single inputs in most cases. When it comes to triple-stream inputs, (PQI), the initial expectation is to have the best results compared to previously mentioned inputs. However, it is shown that the higher dimensionality has adversely affected its results in most methods. In best cases, three signal input (PQI) is as good as the best lower-dimension input. By looking at the Time column in Table II, it is observed that three signal input (PQI) has increased the computation time too without any improvement of the clustering scores. It can be concluded that in the context of the NILM, adding more dimensions to the input stream can unfavorably affect the clustering process. Considering the above, and averaging the score of each input over all methods, QI can be considered the best-performing input for this dataset. However, as the performance scores of dual-stream inputs are not significantly different, for the generality purpose, we can conclude that dual-stream inputs perform better than single or triple-stream inputs for event clustering purposes. To verify this, we added a limited extension to this part, by testing this process on iAWE dataset [21] where again dual-stream inputs obtained the best results and PQ was the best among dual-stream inputs.

Fig. 5 illustrates the normalized ΔP and ΔQ of the ground truth load events of the used dataset in this paper. By looking at Fig. 5 it is observed that AC ON and AC OFF events are far from other load events due to the AC's high power consumption. It is easily remarked that AC load events (both ON and OFF) are separated into two different groups due to the different power consumption at different ON/OFF events. It makes the clustering task challenging for the algorithms as they should still be clustered as the same load events. Iron ON and OFF events are dense and well-separated from other load events, however, they seem to be separated into two groups too, same as the case with the AC load events. In the middle, there is congestion of low-power load events and no event samples. To better illustrate the situation, the middle part of Fig. 5 is magnified. It is observed that the fridge and socket ON and OFF events are at a short distance from each other, and no-event data samples. However, their high density will help the clustering algorithms to better identify them.

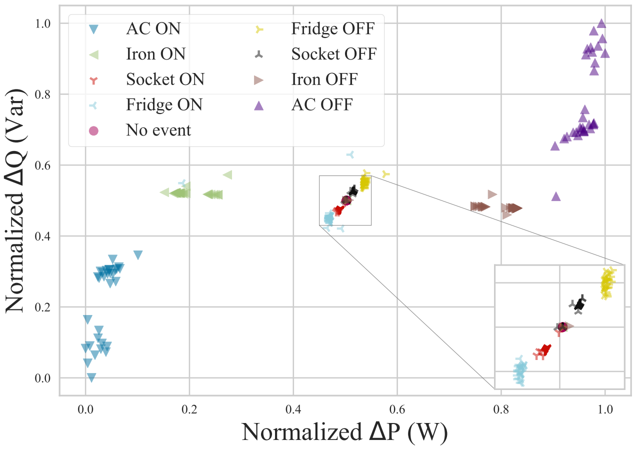

Fig. 5. Ground truth labeling of the dataset

Fig. 6 shows the clustering output of the investigated algorithms fed by QI streams. The number of identified clusters is written in front of each algorithm's name. The color of clusters differs in every algorithm as the labels are assigned in different orders. Some algorithms e.g., BIRCH, Affinity propagation, and Mean shift have identified most of the load events in the middle of the diagram as the same cluster which shows their inability in distinguishing close clusters. In contrast, DBSCAN has found so many clusters in that central area while ignoring most of the data samples as outliers (black circles). It shows DBSCAN shortcomings when facing dense and sparse clusters at the same time. Many of the algorithms have difficulty distinguishing AC load events (both ON and OFF) and identify them as 4 different clusters instead of 2. The only algorithm that has successfully identified AC load events is OPTICS. OPTICS uses variable bandwidth to calculate the areas with enough density, and for this reason, it is capable of identifying both separate and congested clusters.

Although the OPTICS clustering algorithm fed by a dual input stream showed the best performance among all investigated combinations, there is still an important factor in NILM which needs to be analyzed, and that is the sampling frequency. We want to test the flexibility and tolerance of the OPTICS algorithm when it faces non-idea frequency ranges. The BLUED dataset which is used in this study is sampled at 60 Hz frequency. To have a lower-resolution dataset, we downsampled the dataset to the frequencies of {30, 1, 1/10, 1/30, 1/60, 1/120, 1/300, 1/600 Hz}. Fig. 7 illustrates the same power profile for different sampling frequencies. Even though the load profile sampled at 1/600 Hz seems to be an easier case for the clustering algorithms due to the lack of transients, load signatures with small wattage (e.g., wall socket) have been eliminated in the sampling process. In this situation, the clustering algorithm is not able to identify all load events, i.e., lower clusters number.

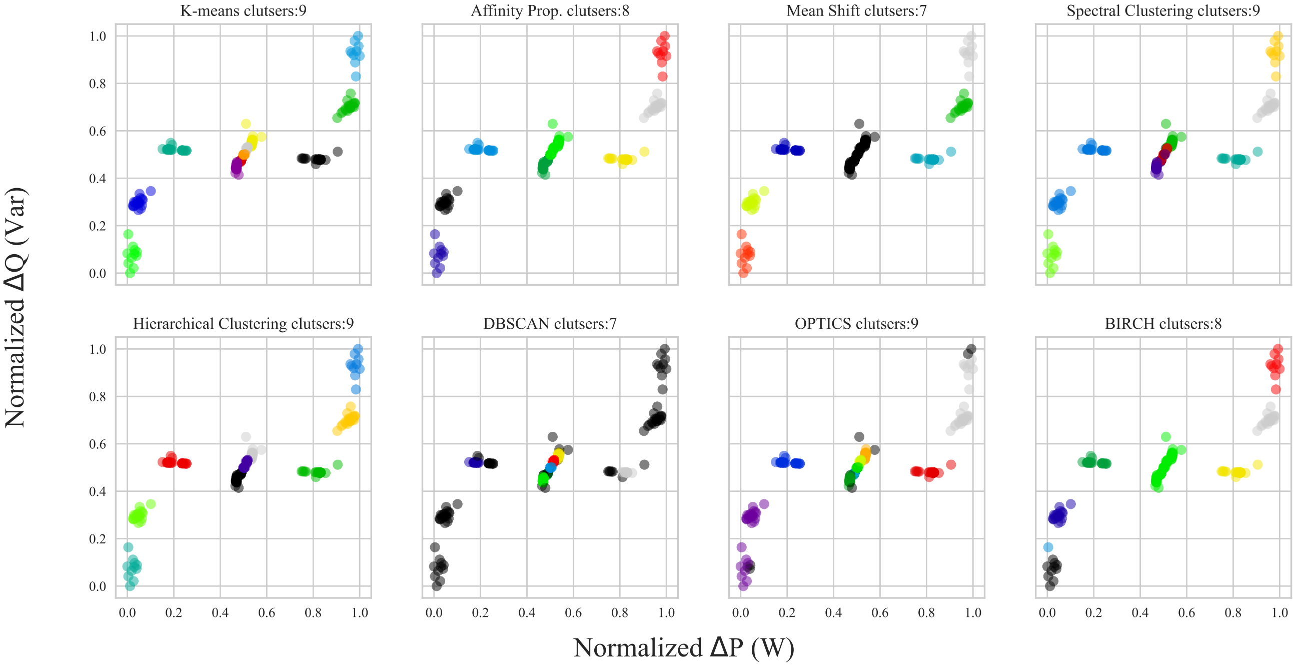

Fig. 6. All algorithms performance on the QI input

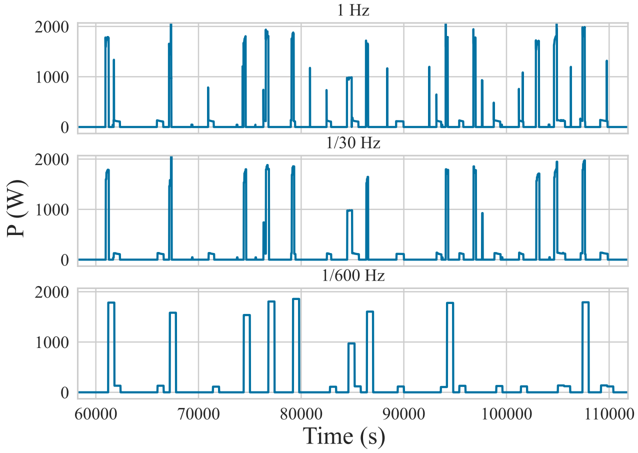

Fig. 7. The test Load profile of BLUED dataset sampled at different frequencies

By applying the OPTICS clustering algorithm on the QI input stream sampled in different time periods, Table III is achieved.

TABLE III
THE PERFORMANCE OF OPTICS CLUSTERING ALGORITHM

Method	freq.	Clust.	Sil	ARI	V	FM	Mean	Events
OPTICS	60	14	80	63	76	55	64	612
	30	12	77	75	73	61	69	437
	1	9	76	82	89	84	86	373
	1/10	9	54	76	86	79	81	358
	1/30	9	63	88	92	89	90	329
	1/60	7	67	88	90	89	89	302
	1/120	5	69	82	82	85	83	254
	1/300	5	80	93	91	95	93	212
	1/600	5	81	97	95	98	96	175

Sil, ARI, V, and FM are in percent (%)

It is observed that by lowering the sampling frequency to 1/60 Hz, the OPTICS algorithm is not able to correctly distinguish the number of clusters which is an important task. However, the performance metrics have generally risen by decreasing the sampling frequency. For example, 1/600 Hz sampling frequency has a mean performance metric of 96% and silhouette factor of 81% while 1 Hz period has 86% and 76% respectively.

It may seem counter-intuitive, as we expect the lower resolution to exacerbate the clustering performance not improve the metrics. It should be noted that the number of identified clusters is reduced in lower sampling frequencies. In particular, lower resolution eliminates the smaller load events which are reflected in the "Events" column of Table III. Note that the ground truth number of events is 364. Thus, the clustering algorithm identifies high wattage load events e.g., AC, more easily, and as a result, its performance score improves. However, this seemingly better performance comes at the cost of losing many smaller load events and reduced clusters number.

Considering the above-mentioned points, 1/30 Hz frequency is the lowest frequency with which the OPTICS algorithm is able to operate without sacrificing the load event detection and having good performance metrics at the same time.

High frequencies (60 and 30 Hz), have too many fluctuations in their transient states for the LLR voting event detection algorithm. Because of that, the LLR finds too many False Positives (FP) which increases the number of clusters and worsens the clustering scores.

With these results in mind, for the clustering purpose in the NILM context, it is shown that OPTICS is the best option among all eight investigated algorithms. While having single stream inputs e.g., P, Q, and I does not lead to the best performance of the clustering algorithms, having more complex inputs e.g., PQI confuses the algorithms in some cases. It is observed that dual-stream inputs are the best choice for the input in most clustering algorithms.

V. Conclusion

This paper presents a comparative evaluation of unsupervised clustering algorithms applied on different load transient states in the context of the NILM in the residential sector of power grids. Eight clustering algorithms are fed by different combinations of three electrical signals, i.e., P, Q, and I from BLUED dataset, as the input. Clustering algorithms' performance is evaluated by shape-based and ground-truth-based metrics. Different input sets are tested and among them, dual-stream inputs turned out to be the best. Regarding clustering methods, OPTICS outperformed other algorithms. To further investigate the effect of the input signal on the clustering algorithms in the load disaggregation and examine the flexibility of the OPTICS algorithm under non-ideal conditions, the input data were down-sampled to various frequencies. The OPTICS algorithm was able to correctly predict the number of load event clusters with a sampling frequency as low as 1/30 Hz. It is shown that decreasing sampling frequency leads to missed load events and clusters, however, the clustering scores improve and the clustering operation becomes faster. Moreover, higher sampling frequencies (60 and 30 Hz) exacerbated the clustering scores due to FP load event detection. The results of this study facilitate the process of choosing the best clustering algorithms, input signals, and sampling frequency for unsupervised NILM studies in the residential sector of power grids.

References

[1] S. S. Hosseini, K. Agbossou, S. Kelouwani, and A. Cardenas, “Non-intrusive load monitoring through home energy management systems: A comprehensive review,” *Renewable and Sustainable Energy Reviews*, vol. 79, pp. 1266–1274, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032117307359

[2] G. Hart, “Nonintrusive appliance load monitoring,” *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[3] P. Ducange, F. Marcelloni, and M. Antonelli, “A novel approach based on finite-state machines with fuzzy transitions for nonintrusive home appliance monitoring,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1185–1197, 2014.

[4] H.-H. Chang, K.-L. Chen, Y.-P. Tsai, and W.-J. Lee, “A new measurement method for power signatures of nonintrusive demand monitoring and load identification,” *IEEE Transactions on Industry Applications*, vol. 48, no. 2, pp. 764–771, 2012.

[5] S. Gupta, M. S. Reynolds, and S. N. Patel, “Electrisense: Single-point sensing using emi for electrical event detection and classification in the home,” in *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, ser. UbiComp ’10. New York, NY, USA: ACM, 2010, pp. 139–148. [Online]. Available: http://doi.acm.org/10.1145/1864349.1864375

[6] L. Xinwei, R. Zhiren, T. Bo, L. Hui, Y. Rui, and W. Haiping, “Hybrid load identification model based on grey wolf optimization algorithm,” in *2019 25th International Conference on Automation and Computing (ICAC)*, 2019, pp. 1–5.

[7] Hernández, A. Ruano, J. Ureña, M. Ruano, and J. Garcia, “Applications of nilm techniques to energy management and assisted living,” *IFAC-PapersOnLine*, vol. 52, no. 11, pp. 164–171, 2019, 5th IFAC Conference on Intelligent Control and Automation Sciences ICONS 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896319307657

[8] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, “Improving nonintrusive load monitoring efficiency via a hybrid programing method,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2148–2157, 2016.

[9] X. Wu, D. Jiao, and L. You, “Nonintrusive on-site load-monitoring method with self-adaption,” *International Journal of Electrical Power and Energy Systems*, vol. 119, p. 105934, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061519328911

[10] Y. Yang, J. Zhong, W. Li, T. A. Gulliver, and S. Li, “Semisupervised multilabel deep learning based nonintrusive load monitoring in smart grids,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 6892–6902, 2020.

[11] A. Singh, N. Thakur, and A. Sharma, “A review of supervised machine learning algorithms,” *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 1310–1315, 2016.

[12] M. Azaza and F. Wallin, “Evaluation of classification methodologies and features selection from smart meter data,” *Energy Procedia*, vol. 142, pp. 2250–2256, 2017, proceedings of the 9th International Conference on Applied Energy. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1876610217363890

[13] A. U. Rehman, T. T. Lie, B. Vallès, and S. R. Tito, “Comparative evaluation of machine learning models and input feature space for non-intrusive load monitoring,” *Journal of Modern Power Systems and Clean Energy*, vol. 9, no. 5, pp. 1161–1171, 2021.

[14] G. Jacobs and P. Henneaux, “Specific and generic unsupervised algorithms for nilm applications,” in *2020 International Conference on Smart Energy Systems and Technologies (SEST)*, 2020, pp. 1–6.

[15] A. Yasin and S. A. Khan, “Unsupervised event detection and on-off pairing approach applied to nilm,” in *2018 International Conference on Frontiers of Information Technology (FIT)*, 2018, pp. 123–128.

[16] T. Bernard, M. Verbunt, G. v. Bögel, and T. Wellmann, “Non-intrusive load monitoring (nilm): Unsupervised machine learning and feature fusion : Energy management for private and industrial applications,” in *2018 International Conference on Smart Grid and Clean Energy Technologies (ICSGCE)*, 2018, pp. 174–180.

[17] J.-P. Seevers, J. Johst, T. Weiß, H. Meschede, and J. Hesselbach, “Automatic time series segmentation as the basis for unsupervised, non-intrusive load monitoring of machine tools,” *Procedia CIRP*, vol. 81, pp. 695–700, 2019, 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827119304834

[18] S. S. Hosseini, B. Delcroix, N. Henao, K. Agbossou, and S. Kelouwani, “A case study on obstacles to feasible nilm solutions for energy disaggregation in quebec residences,” in *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 363–367. [Online]. Available: https://doi.org/10.1145/3563357.3566151

[19] K. D. Anderson, A. Ocneanu, D. R. Carlson, A. G. Rowe, and M. Bergés, “Blued : A fully labeled public dataset for event-based non-intrusive load monitoring research,” 2012.

[20] K. D. Anderson, M. E. Bergés, A. Ocneanu, D. Benitez, and J. M. Moura, “Event detection for non intrusive load monitoring,” in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, 2012, pp. 3312–3317.

[21] N. Batra, M. Gulati, A. Singh, and M. B. Srivastava, “It’s different: Insights into home energy consumption in india,” ser. BuildSys’13. New York, NY, USA: Association for Computing Machinery, 2013, p. 1–8. [Online]. Available: https://doi.org/10.1145/2528282.2528293