# Self-configuring Event Detection for Electricity Disaggregation

Farrokh Jazizadeh
Via Department of Civil and Environmental Engineering
Virginia Tech
Blacksburg, VA, USA
jazizade@vt.edu

Milad Afzalan
Via Department of Civil and Environmental Engineering
Virginia Tech
Blacksburg, VA, USA
afzalan@vt.edu

Jue Wang
Via Department of Civil and Environmental Engineering
Virginia Tech
Blacksburg, VA, USA
juew@vt.edu

***Abstract*—Event-based non-intrusive load monitoring (NILM) is a major class of techniques that are used for electricity disaggregation in buildings. Event detection algorithms play a critical role in the performance of these techniques. There are different deterministic and probabilistic algorithms for detection of the events that commonly call for a number of parameters that need to be tuned as these algorithms are applied to a new dataset. The need for tuning could be a barrier to wide adoption of these algorithms. Therefore, in this study, we are proposing a self-configuring event detection framework that integrates conventional event detectors, automated clustering algorithms, and data-driven parameter optimization to enable self-configuration. The framework has been evaluated on real-world data and demonstrated promising results in high-quality detection of events.**

***Keywords—Electricity disaggregation, event detection, self-configuring algorithms, unsupervised methods.***

## I. Introduction

The techniques of non-intrusive load monitoring (NILM) for electricity disaggregation are categorized into two main classes of event-based and non-event based. Event-based techniques track individual events on the aggregate electricity time series and associate them with the operation of individual appliances. Events are associated with the change in operational states of appliances and are reflected on the time series as sudden changes in the average value of aggregate electricity consumption. Upon detection of the events, the characteristics of the signal in the vicinity of events are extracted as feature vectors to feed the classification algorithms and infer the appliance label. A critical step in this process is the detection of events that represent the changes of the operational states in the physical space. Depending on the resolution of electricity consumption metric, different algorithms could be used for event detection. Increasing the resolution of the data helps improve the accuracy of the classification algorithms in inferring load identities. This improvement is due to the application of transient feature vectors, which contain more information about the dynamics of the loads. However, increasing the resolution brings about increase in noise interference, which results in challenges for event detection algorithm, such as an increase in false positive detection.

To tackle the challenges in detecting events on high-resolution data, advanced event detection techniques were adopted and devised. Generalized likelihood ratio (GLR) test was introduced and adopted by Luo et al. [1] to improve the performance in the presence of noise. In this method, events are detected by measuring if samples before and after an event are coming from two different probabilistic distributions. Further research efforts have been conducted to introduce different event detection algorithms and improve detection accuracy. Some of the example efforts include using a goodness-of-fit $\chi$^2 test-based algorithm [2], Window-with-Margins event detection method [3], and adaptive event detection [4] that detects the time limits of each transition interval.

As a common attribute of these algorithms, the performance depends on a number of parameters that need to be tuned for different environments. These parameters include the window sizes for calculation of statistical metrics as well as various thresholds for detection statistics. Although the tuning could be carried out in field experimental efforts and across different environments, diversity in the appliances' technologies could pose a challenge in achieving scalable performance in different environments. Therefore, there will a need for reconfiguration of the algorithm to ensure that the better set of parameters for each environment is set.

In order to contribute to the scalability of the electricity disaggregation solutions, in this study, we have proposed a framework to move towards self-configuring event detection techniques. The proposed framework is centered around enabling event detectors to learn from the data in an environment and adjust their performance according to the characteristics of an environment and its associated appliances. The framework is built on an existing event detection approach to create the initial dataset for learning and using shapelet-based techniques for event detection.

## II. Framework Description

The proposed framework seeks to shift the detection logic from searching for abrupt changes on a time series to a shapelet-based detection approach. Therefore, the framework promotes searching for the most probable transient shapes on an electricity time-series that are associated with state transitions. A key component of this framework is the identification of critical shapelets in a given environment. Critical shapelets are obtained by averaging similar feature vectors (each representing an appliance state). As noted earlier and the logic implies, the framework is designed around high-resolution data acquisition systems and their associated time series. In this study, our focus is on the application of power time series as the representative metric of aggregate electricity consumption. Therefore, the key steps in the framework include:

- Detecting events on buffered data in an environment using conventional event detectors (we used GLR in this study)
- Identifying the critical shapelets (e.g., Figure 1) through unsupervised clustering on detected events in buffered data
- Shifting event detection to shapelet-based: events will be identified, where the power time series shape matches one of the shapelets

Figure 2 illustrates the expanded version of the framework.

### A. *GLR Event Detection*

The GLR event detection algorithm uses a statistical test to identify events. The algorithm evaluates likelihood ratio (detection statistics) between Gaussian distributions ($S \sim N(\mu, \sigma^2)$), assigned to samples of data before and after each data point on the power time series:

$$L(n) = \ln \frac{P(s_i|\mu_1, \sigma_1^2)}{P(s_i|\mu_0, \sigma_0^2)} \tag{1}$$

where $s_i$ is the $i$-th signal sample point, and $\mu_0$, $\sigma_0^2$, $\mu_1$ and $\sigma_1^2$ are mean and standard deviation in two windows before and after each data point. Accordingly, the algorithm calls for setting the size of two windows before and after each sample point for estimating parameters of the distributions. Time series sample points with a likelihood ratio higher than a predefined threshold are marked as events. In addition to the abovementioned parameters, in some implementations, a threshold for minimum consecutive power variations is also added to avoid excessive false positives due to noise.

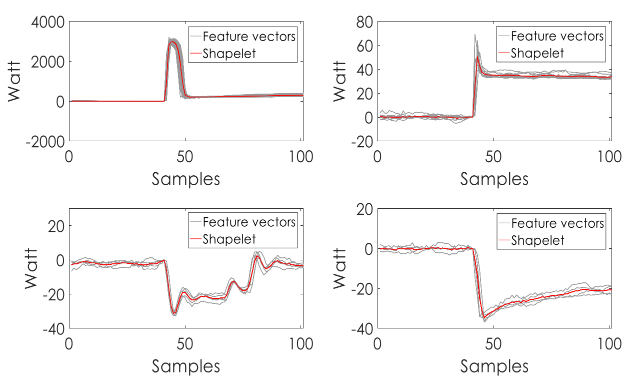


Fig. 1. Examples of feature vectors, clusters, and their associated shapelets (real power component only). Each feature vector contains 101 samples (corresponding to a duration of ~1.7s).

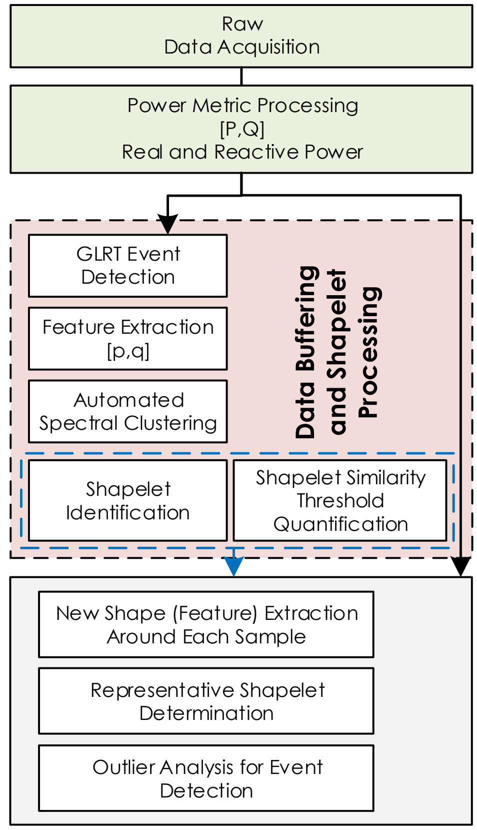


Fig. 2. The framework for shapelet-based event detection

### B. *Shapelet Characterization*

Shapelets are characterized in a similar way that feature vectors are extracted for classification. In our implementation, the feature vectors are extracted as power samples in the vicinity of an event. Two windows of pre-event samples ($w_{pre}$) and post-event samples ($w_{post}$) are used to extract the features. In the NILM field, these feature vectors are also known as signatures. The signatures are represented as follows (see examples in Figure 1):

$$\boldsymbol{x_n} = \{p[n], q[n]\} \tag{2}$$

where $p$ and $q$ are real and reactive power components, respectively. Identifying the critical shapelets calls for grouping similar feature vectors that represent the same appliance states. Critical shapelets are the ones that play an important role in the quantification of energy performance in a building. These shapeletes are obtained in the environment. Therefore, their number is *not* fixed and is updated as the buffered data increases. In a previous study, the feasibility of identifying critical signatures through clustering has been explored [5].

In order to form the shapelets, an automated clustering algorithm is required. Although clustering algorithms are categorized as unsupervised learning techniques, they commonly call for *a priori* parameters to determine the partition in a feature space. Application of algorithms that need additional input parameters contradicts the objective of self-configuration. Therefore, we have developed autonomous clustering algorithms that obviate the need for input parameters (e.g., [6]). In this study, we have used our proposed heuristic spectral clustering algorithm [7] that iteratively partitions a feature space in a search tree structure revealing eigengaps at different scales of the feature space. Generic spectral clustering algorithms call for two input parameters: number of clusters and a scaling factor. For the latter, our approach uses a PCA-based quantification of scaling factor at different scales of the feature space. The outcome of the clustering process is the groups of feature vectors that represent different operational states of appliances in the target environment. Depending on the configuration of the appliances and their frequency of use, these clusters could contain an unbalanced number of data points. As illustrated in Figure 1, the centroid vector of each cluster will be used as the shapelets in the next steps of event detection. Therefore, the window sizes that are used for feature extraction will be also used for characterization of shapelets.

### C. *Proximity-based Event Detection*

As the description of the framework implies, this approach uses a proximity-based technique such as one nearest neighbor (1NN) for identification of events. Therefore the approach is a semi-supervised approach in nature. The main difference with classification is the need for an outlier detection stage for accepting or rejecting an event as a critical one. Therefore, the shapelet-based event detection process is as follows:

1. For each pint on the power time series, extract a feature vector following Equation 2
2. Identify the closest shapelet to the feature vector using 1NN algorithm
3. Run an outlier detection to accept or reject the feature vector as a critical event.

As the process shows, the proposed framework could combine the process of event detection and classification. Therefore, upon detection of an event, if the clusters are labeled (with the name of appliances in the physical environment), the load identity is revealed as well.

## III. Outlier Implementation

This study presents an implementation of the framework as a proof of concept. An important component of the framework is the approach for outlier detection. Formation of the clustered data enables us to use a data-driven approach in identifying context-aware thresholds for outlier testing – a critical step in enabling the framework to be self-configuring. Given that the clustered feature vectors representing a dense set of data points, the clustered shapelets will not include outliers. This fact enables us to identify a range of acceptable thresholds for each cluster. A variety of outlier algorithms could be used. In this study, we have implemented two threshold identification approaches as follows

- **Mahalanobis distance**: a threshold identification approach that uses Mahalanobis distance [8] is defined as follows:

$$d_{M_i} = d_M(\boldsymbol{s}_i, \boldsymbol{\mu}_c) = \sqrt{(\boldsymbol{s}_i - \boldsymbol{\mu}_c)\boldsymbol{\Sigma}_c^{-1}(\boldsymbol{s}_i - \boldsymbol{\mu}_c)^T} \quad (3)$$

where, $\boldsymbol{s}_i$ is a shapelet extracted from the power time series, $\boldsymbol{\mu}_c$ is the centroid of the closest cluster to the shapelet ($\boldsymbol{s}_i$), and $\boldsymbol{\Sigma}_c$ is the covariance matrix of the cluster. Mahalanobis distance accounts for correlation between the features in a clustered data and therefore, the transient shape of the signatures plays a role in identifying the distance. Therefore, it will be a more effective metric compared to Euclidean distance.

In order to obtain the acceptable range of distance for each cluster, in a preprocessing step, as it depicted in Figure 2, the distance of all shapelets in a cluster with respect to its centroid is calculated. In the real-time event detection stage, upon extracting the shapelet for each data point, the Mahalanobis distance is calculated between the extracted shapelet and the cluster shapelet (i.e., its centroid). Then a similarity measure is compared with an asymptotic threshold for each cluster for detection as below

$$\frac{1}{1+d_{M_i}} < \frac{1}{1+\mu_i+\sigma_i} \quad (4)$$

- **Cross-correlation:** Cross-correlation is a commonly used metric in signal processing for measuring the similarity of two time- series. The cross-correlation (with zero lag) of an extracted shapelet from the power time-series and the shapelet of the closest cluster is as follows

$$Cc_i = (\boldsymbol{s}_i * \boldsymbol{\mu}_c)[0] = \sum_{\forall m} \boldsymbol{s}_i[m]\boldsymbol{\mu}_c[m] \quad (5)$$

In order to define the thresholds for outlier detection, for each cluster, the autocorrelation (i.e., cross-correlation of a signal with itself) for every feature vector within that cluster is calculated. Then, the lowest and highest values are considered as the lower bound and upper bound threshold for each cluster. By this means, an event is detected if the $Cc_i$ lies between the lower and upper bound thresholds. In order to make the feature vectors more generalized for facilitating event detection, we have generated synthetic observations for each cluster as well. Each feature of the synthetic observation is selected by random sampling from a kernel density estimator, which is obtained from the real observation (50:50 ratio between real and synthetic observations). By this means, the lower and upper bound thresholds for each cluster become less tight for outlier detection.

Using the abovementioned approaches, the detection statistic is shifted to be a shapelet-based metric that does not require input from a calibration process.

## IV. Performance Evaluation

The framework was evaluated on a data set, collected from an apartment unit over two weeks as occupants interacted with the environment. We have used the data from one phase that fed 10 appliances. These appliances had a variety of different power range. For example, light and TV were among the lowest power draw appliances (~40-200 Watt) and air conditioning and refrigerator were among the highest power draw appliances (~1000-2000 Watt). In the US, the electricity infrastructure uses a split-phase system that feeds appliances on two major circuits (i.e., phases). Both on and off state transitions were taken into account in the evaluation. Signatures from on and off events were put together for the analysis. The data has been fully labeled by leveraging ground truth sensors (both electricity and light) that were installed at the consumption point.

In order to simulate the data buffering process in forming the benchmark shapelets, we have divided the data into two benchmark and test subsets using a 70-30 ratio, respectively (the

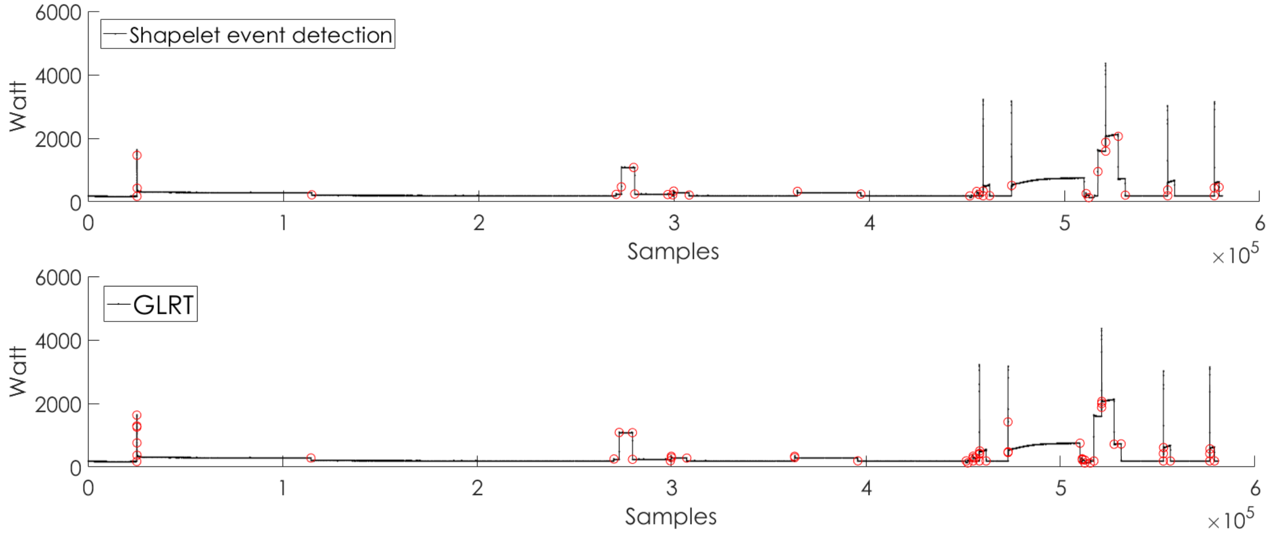

Fig. 3. Detected events on a sample power time series using (a) Shapelet-based event detection and (b) GLR algorithm

test subset includes ~17 million data points). The GLR event detection algorithm was used for detecting events and extracting the feature vectors in the data buffering stage. A pre-event window of size 40 (i.e., two third of a second) and a post-window of size 60 (i.e. one second) were used in feature extraction. The data in the benchmark subset was passed through the clustering algorithm to create the clustered shapelets. No further processing on the clustered signatures was conducted. Using the clustered data, a threshold range was identified for each cluster. The proposed framework and the conventional GLR algorithm were used on the test data subset. Leveraging previous observations, a general set of parameters for the GLR algorithm was considered. No specific effort was made to ensure that the GLR algorithm is tuned to its better level of performance.

According to the manually labeled data set, the actual number of events on the power time series was 250. Therefore, the total number of detected events as well as the counts for true positive (TP) events, false positive (FP) events, and false negative (FN) cases were used for evaluation of the performance. In order to quantify these metrics, a ±3 sample points was considered as tolerance in the comparison between ground truth and predicted events. Table 1 shows the results of the evaluation.

As the result shows, the proposed shapelet-based event detection with both outlier detection approaches had a promising performance without the need for input information or tuning any parameter. Although the number of true positives for GLR algorithm was higher, that came with the cost of having a higher number of false positives. The shapelet approach was able to keep a better balance in the trade-off between the TP and FP detection. It must also be noted that the buffered data in the environment for collecting the shapelets was relatively short. As the buffered data in the environment increases, the collection of shapelets would be more generalized, which could further improve the results on new datasets. In order to provide a visual comparison between these two methods, the detected events for the described methodologies have been illustrated in Figure 3.

The only parameters that were needed to be set by a user for shapelet-based event detection are the window sizes for feature extraction. It has been observed in classification studies that the detection performance is not highly affected by the size of the windows. A general rule of thumb in identifying the size is to ensure that signatures are not overlapping and the transient information is persevered. However, a quantitative analysis of the window size effect will be part of the future directions to identify the sensitivity of the approach. Moreover, considering the fact that the size of window for both benchmark shapelets identification and event detection is the same, the window size effect will be minimized.

An important feature in the shapelet-based event detection is the quality of the benchmark shapelets. The performance of clustering and initial event detection approach is critical to the quality of shapelets. Although it is not in the scope of this study, heuristic methods, such as rule-checking, could be integrated in the framework for improving the quality of the benchmark

TABLE I
EVALUATION RESULTS FOR EVENT DETECTION

| Event Detection Approach | Outlier detection | Actual events | Detected events | TP | FP | FN | Precision | Recall |
|---|---|---|---|---|---|---|---|---|
| Shapelet-Based Event Detection | Mahalanobis distance | 250 | 255 | 179 | 76 | 71 | 0.70 | 0.72 |
| | Cross-correlation | 250 | 241 | 183 | 58 | 67 | 0.76 | 0.73 |
| GLR Test | - | 250 | 418 | 202 | 216 | 48 | 0.48 | 0.81 |

shapelets. In a previous study, it was shown that data-driven techniques could also be used to evaluate the quality of benchmark clusters [5].

## V. Conclusion

This study presents the proof of concept for a shift from event detection based on abrupt changes in time series of representative power metrics to a shapelet-based approach. Shapelets are represented by clustered signatures of appliances' operational states in an environment to facilitate self-configuration of the algorithms. The realization of the proposed approach calls for data-driven techniques that automatically identify shapelet groups in an unsupervised manner and a similarity threshold for comparison between clustered shapelets and the shapelet of a new observation. An implementation of the proposed framework in this study used a combination of automated spectral clustering, 1NN classification, and an outlier detection. The evaluation of the framework on a two-week dataset demonstrated its promising performance in high-quality event detection.

Evaluation of more sophisticated outlier detection for improved performance of the approach, extended assessment of performance on other datasets, comprehensive analysis of feature representation, and enhancing shapelet identification process are among the future directions of this research.

## VI. Acknowledgement

The data set used in this study was collected under the support of the National Science Foundation under Grant #0930868. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

[1] D. Luo, L. K. Norford, S. R. Shaw, and S. B. Leeb, "Monitoring HVAC equipment electrical loads from a centralized location--methods and field test results/Discussion," *ASHRAE Transactions*, vol. 108, p. 841, 2002.

[2] Y. Jin, E. Tebekaemi, M. Berges, and L. Soibelman, "Robust adaptive event detection in non-intrusive load monitoring for energy aware smart facilities," in *ICASSP*, 2011, pp. 4340-4343.

[3] H. Azzini, R. Torquato, and L. C. da Silva, "Event detection methods for nonintrusive load monitoring," in *PES General Meeting| Conference & Exposition, 2014 IEEE*, 2014, pp. 1-5: IEEE.

[4] K. Barsim, R. Streubel, and B. Yang, "Unsupervised Adaptive Event Detection for Building-Level Energy Disaggregation," *Proceedings of power and energy student summt (PESS), Stuttgart, Germany*, 2014.

[5] F. Jazizadeh, "Building Energy Monitoring Realization: Context-Aware Event Detection Algorithms for Non-Intrusive Electricity Disaggregation," in *Construction Research Congress 2016*, 2016, pp. 839-848.

[6] F. Jazizadeh, B. Becerik-Gerber, M. Berges, and L. Soibelman, "An unsupervised hierarchical clustering based heuristic algorithm for facilitated training of electricity consumption disaggregation systems," *Advanced Engineering Informatics*, vol. 28, no. 4, pp. 311-326, 2014.

[7] M. Afzalan and F. Jazizadeh, "Automated spectral clustering for multi-scale data," In Review, 2018.

[8] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1-18, 2000.