Event Detection Based on Robust Random Cut Forest Algorithm for Non-intrusive Load Monitoring

Lingxia Lu, Ju-Song Kang, and Miao Yu

Abstract—**Non-intrusive load monitoring (NILM) can provide appliance-level power consumption information without deploying submeters for each load, in which load event detection is one of the crucial steps. However, the existing event detection methods do not efficiently detect both the starting time of an event (STE) and the ending time of an event (ETE), and their adaptability to scenarios with different sampling rates is limited. To address these problems, in this paper, an event detection method based on robust random cut forest (RRCF) algorithm, which is an unsupervised learning method for detecting anomalous data points within a dataset, is proposed. First, the mean-pooling preprocessing is applied to the aggregated load power series with a high sampling rate to minimize fluctuations. Then, the power differential series is obtained, and the anomaly score of each data point is calculated using the RRCF algorithm for preliminary detection. If an event has been preliminarily detected, misidentification caused by fluctuation will be further eliminated by using an adaptive power difference threshold approach. Finally, linear fitting is used to finely and accurately adjust the STE and ETE. The proposed method does not require any pretraining of the detection model and has been validated with both the BLUED dataset (with high and low sampling rates) and the REDD dataset (with low sampling rate). The experimental results demonstrate that the proposed method not only meets real-time requirements, but also exhibits strong adaptability across multiple scenarios. The precision is greater than 92% in distinct sampling rate scenarios, and the F1 score of phase B on the BLUED dataset reaches 94% in the scenario with a high sampling rate. These results indicate that the proposed method outperforms other state-of-the-art methods.**

Index Terms—**Non-intrusive load monitoring, event detection, robust random cut forest, adaptive threshold.**

I. Introduction

FEEDBACK information on energy usage can provide electricity consumers with a basis to better control their electricity utilization behaviours and ultimately save energy [1], [2]. Therefore, obtaining load usage information is highly important for energy management [3]-[5].

Non-intrusive load monitoring (NILM), which was initially proposed in [6], can provide individual electrical load usage information without deploying submeters for each load [7]-[9]. According to the information acquired from NILM, electricity consumers can determine and adjust their electricity utilization behaviours to save energy. Moreover, NILM can help both electricity consumers and providers locate devices with high power consumption at peak hours, identify malfunctioning devices, and forecast power demand [10], [11].

NILM can be divided into two main categories: nonevent-based NILM and event-based NILM. In NILM studies, a switch action or change in the working state of a load is called an event. An event-based NILM could have better performance than a nonevent-based NILM because the acquired rich load features can greatly enhance the identification accuracy [12].

The main purpose of event detection in NILM is to detect the starting time of an event (STE) and ending time of an event (ETE) when a state transition occurs from aggregated load measurements [13]. Accurate event detection is a prerequisite for precise NILM [14], [15]. For this reason, many studies have carried out in-depth investigations in this field. The existing event detection methods can be divided into three main categories according to their implementation principles: expert heuristic methods, probabilistic methods, and pattern matching methods.

Expert heuristic methods mainly utilize professional knowledge and propose a set of decision rules for event detection. Reference [16] utilized the variation in average active power in the pre-window and post window to detect events and provided exact time stamps of the events in the aggregated signal. Reference [13] proposed a computationally fast algorithm with low complexity that returns the time at which corresponding events occur by detecting the variance and mean absolute deviation of the aggregated active power. In [17], the median filter algorithm and ripple mitigation algorithm were used to remove unexpected disturbances in the aggregated load power series and extract the real signal of switching on/off events.

Probabilistic methods use the statistical probability distribution of aggregated load data to detect changes after an

Manuscript received: November 17, 2023; revised: January 5, 2024; accepted: February 29, 2024. Date of CrossCheck: February 29, 2024. Date of online publication: April 9, 2024.

This work was supported by the Zhejiang Provincial Natural Science Foundation of China (No. LGG22F030008) and Key R&D Program of Zhejiang Province (No. 2021C01113).

L. Lu, J.-S. Kang, and M. Yu (corresponding author) are with the College of Electrical Engineering, Zhejiang University, Hangzhou, China (e-mail: lulingxia@zju.edu.cn; 3170300663@zju.edu.cn; zjuyumiao@zju.edu.cn).

DOI: 10.35833/MPCE.2023.000901

event occurs. Typical probabilistic methods include the likelihood ratio and cumulative sum. A generalized event detection method based on likelihood ratio was proposed in [18], in which a sliding detection window was used to determine whether a state transition occurred. Reference [19] proposed an improved cumulative sum method for NILM. When there is no high-level noise at the common bus, the threshold is reduced to detect loads with low power; otherwise, the threshold is increased dynamically to reduce noise interference. Two robust algorithms were proposed in [20], including a modified version of the chi-square goodness-of-fit test and an event detection method based on cepstrum smoothing.

Pattern matching methods detect events by matching the sequence fragments corresponding to the event transient process with a known feature library. In [21], the transient load features were represented by the spectrogram of the derived root mean square (RMS) of current signal, and these spectrograms were used as inputs to the neural network to detect the event. Reference [22] proposed an unsupervised framework that includes an algorithm for characterizing transient load features in a given environment and a proximity-based motif matching algorithm for event detection.

In addition to the above three types of event detection methods, several other methods have been proposed. For instance, [23] proposed an event detection composition block by combining the individual detection results of event detection agents to produce a single output that identifies the instances with a high likelihood of a load event in time. In [24], a multivariate event detection algorithm was proposed that selected the optimal threshold by analysing the operating characteristic curves with three metrics, namely, the F-measure, the largest vertical distance from the receiver operating characteristic (ROC) curve to the main diagonal, and the closest point to the $(0,1)$ corner on the ROC curve. Another method that combined probabilistic methods and expert heuristic methods was proposed in [25]. This method included a voting-based improved isolated forest algorithm for highly sensitive event predetection and a time-shift down sampling matching algorithm for highly accurate event verification.

Although the above methods have made impressive progress, there is still space for improvement. First, although expert heuristic methods and pattern matching methods are excellent at detecting specific types of events, they cannot handle complex events. Decision rules must be set by developers manually, and it is challenging to select appropriate conditions to adapt the data. Thus, model accuracy is highly dependent on developer's expertise. With the growth of the dataset, both the predefined rules and patterns may not adapt to the new data, and the prior setting may limit the generalization of the methods.

Compared with expert heuristic methods, probabilistic methods are more flexible. The generalization performance of these methods is excellent because they are data driven. However, probabilistic methods have high requirements for data quality and rely on customized parameter settings at the beginning of the process. Consequently, they cannot maintain satisfactory performance in different scenarios. For instance, the method in [12] can only work at a high sampling rate, and the methods in [13] and [26] must be implemented at a low sampling rate. The initialized parameters of these methods cannot support both scenarios simultaneously, which impacts the adaptability of each method in practical situations.

Furthermore, most of the existing event detection methods [14] have focused only on detecting the STE while ignoring the ETE. However, both STE and ETE are crucial for subsequent load identification, as event-based NILM methods rely on extracting load features such as harmonic current and voltage-current (*V-I*) trajectories [27] by calculating the differences in steady-state voltage and current before and after the event. Although some methods have considered detecting both the STE and ETE [14], [26], they focused mainly on one or two specific types of events and have not generalized their methods to more complex events in actual scenarios.

To address these challenges, in this paper, an event detection method based on the robust random cut forest (RRCF) algorithm is proposed. This method, which can handle streaming data and offer precise STE and ETE information for subsequent load identification, can work in different scenarios. First, the STE and ETE are preliminarily detected by using the RRCF algorithm, and then the misidentification caused by fluctuations is further eliminated by using an adaptive power difference threshold, which can be adjusted in real time according to the standard deviation of the aggregated load power. Finally, the STE and ETE are finely adjusted by linear fitting. The proposed method can address challenging events such as repetitive events, high fluctuation events, long transient events, and near-simultaneous events, and improve event detection accuracy, as validated on the basis of the BLUED dataset [28] and REDD dataset [29].

The main contributions of this work are as follows.

1) The proposed method can detect both the STE and ETE with high accuracy. Thus, this method provides a good foundation for subsequent load identification.

2) The proposed method has high sensitivity in scenarios with high sampling rates, which means that it can detect events that occur within a short period of time.

3) The proposed method has high practicality and adaptability because it can meet real-time requirements and performs well in scenarios with high and low sampling rates.

This paper is organized as follows. Section II introduces the principle of the RRCF algorithm and the calculation of anomaly scores. Section III presents the architecture of the proposed method and the principles of each stage for event detection. Section IV analyses the performance of several challenging event detection methods, and the proposed method is verified on the BLUED and REDD datasets. Finally, Section V presents the main conclusions.

II. Principle of RRCF Algorithm and Calculation of Anomaly Scores

Before introduction of load event detection, in this section, we will first introduce the principle of the RRCF algorithm, which lays the foundation for the proposed method.

A. Principle of RRCF Algorithm

RRCF algorithm [30] is an outlier detection algorithm for dynamic data streams generated in real time. It has been applied in various scenarios such as in the real-time detection of abnormal wind power data [31].

The first step of the RRCF algorithm is to create a random forest of trees, where each tree is obtained by partitioning the sample data. The second step is to calculate the anomaly score for each data point in the trees, in which the anomaly score is defined as the expected change in the complexity of the tree as a result of adding or removing that data point from the tree. The random cut forest assigns an anomaly score by computing the average score from each constituent tree and scaling the result with respect to the sample size.

Anomaly scores can manifest during unexpected spikes in time series data or breaks in periodicity or with unclassifiable data points. Therefore, when viewed in a plot, data points with a high anomaly score are often easily distinguishable from "regular" datasets.

RRCF algorithm can be run in steaming data or batch processing mode, enabling the model to adapt to different data types and anomaly patterns. The computational complexity of the RRCF algorithm can be optimized by adjusting the parameters of the forest, namely, the number of trees and the size of the tree, which can maintain the balance between computational complexity and model accuracy.

B. Calculation of Anomaly Score

The procedure for calculating an anomaly score is as follows. Given a set of points Z and a point $y \in Z$, let $f(y, Z, T)$ be the depth of y in tree $T(Z)$. Consider the tree produced by deleting $\{x\}$ as $T(Z-\{x\})$. Let the depth of y in $T(Z-\{x\})$ be $f(y, Z-\{x\}, T)$. Figure 1 shows an example of deleting one data point based on the tree structure.

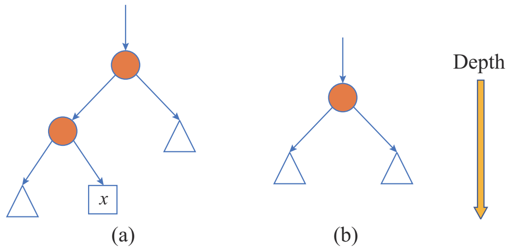

Fig. 1. Deleting one data point from tree $T(Z)$. (a) Tree $T(Z)$. (b) Tree $T(Z-\{x\})$.

In Fig. 1, the circle represents the parent node, the triangle represents the child leaf node, and the square represents the data point that needs to be removed. The anomaly score for data point $\{x\}$ is calculated as:

$$Score(x, Z) = \sum_{y \in Z-\{x\}} (f(y, Z, T) - f(y, Z-\{x\}, T)) \tag{1}$$

In the tree, the depth of an anomalous data point is usually much shallower than that of a normal data point. Thus, the anomaly score will increase when abnormal data points are added or deleted. Therefore, a low anomaly score means that the corresponding data point is "normal", and a high anomaly score means that the corresponding data point is "anomalous".

III. Architecture of Proposed Method and Principles of Each Stage for Event Detection

In this section, how to employ the RRCF algorithm in the event detection is presented. First, the procedure of the proposed method is introduced. Then, the principle of preliminary detection based on RRCF algorithm is elaborated, and after preliminary detection, the details of postprocessing is further described. Finally, the complete algorithm is displayed.

A. Procedure of Proposed Method

A flowchart of the proposed method is shown in Fig. 2.

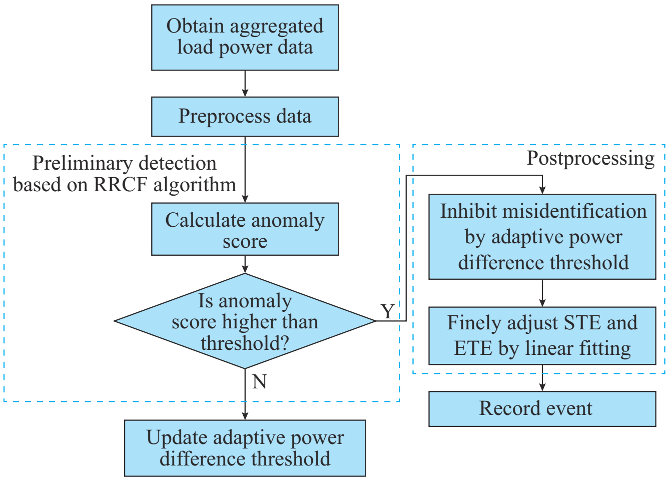
Obtain aggregated load power data
Preprocess data
Preliminary detection based on RRCF algorithm
Calculate anomaly score
Is anomaly score higher than threshold?
Y
N
Update adaptive power difference threshold
Postprocessing
Inhibit misidentification by adaptive power difference threshold
Finely adjust STE and ETE by linear fitting
Record event

Fig. 2. Flowchart of proposed method.

The functions of each module are as follows.

1) Data preprocessing: the mean-pooling preprocessing is applied to the aggregated load power series with a high sampling rate to eliminate fluctuations, which is unnecessary for series with a low sampling rate. Then, the power differential series is obtained via calculation.

2) Preliminary detection based on RRCF algorithm: the anomaly score of each data point in the power differential series is calculated, and the possible event is preliminarily detected.

3) Postprocessing: when a possible event is detected, the power difference threshold further inhibits the misidentification event. Then, the STE and ETE are finely adjusted by linear fitting.

4) Adaptive power difference threshold updating: the standard deviation of the aggregated load power data is calculated at each moment to update the adaptive power difference threshold.

B. Principle of Preliminary Detection Based on RRCF Algorithm

First, for data with a high sampling rate, the mean-pooling processing is applied to eliminate fluctuations and reduce frequency. Nevertheless, the mean-pooling processing is unnecessary for data with a low sampling rate.

Figure 3(a) shows an aggregated load power series consisting of two events, and the sampling frequency is 20 Hz. When an event occurs, the power value becomes an outlier relative to the power values in the previous steady state. Hence, these data points can be detected as outliers by the RRCF algorithm.

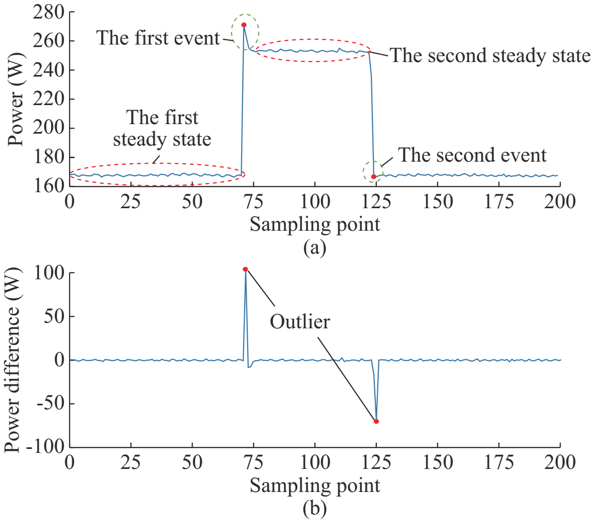

Fig. 3. Data series acquisition results. (a) Aggregated load power series. (b) Power differential series.

However, when the aggregated load power series are used directly, a potential problem could occur. As the RRCF algorithm determines whether the value of the current moment is an outlier according to the series data before this moment, if a certain load is switched on and off in a short time (e.g., the number of data points for the second steady state in Fig. 3(a) is not enough) and the power value after the event is the same as that of the previous steady state (e.g., the power value after the second event in Fig. 3(a) is the same as the power value of the first steady state), the data points after the event may not be identified as abnormal data points or they may have lower anomaly scores.

Therefore, the power differential series shown in Fig. 3(b) is used to calculate the anomaly score. Based on the aggregated load power series $\{P(t)|t=1,2,...,N\}$ with a length of N, the power differential series $\{\Delta P(t)|t=1,2,...,N-1\}$ is obtained as:

$$\Delta P(t)=P(t+1)-P(t) \tag{2}$$

As shown in Fig. 3(b), the power differential series is very close to zero in the steady state. When an event occurs, the power difference changes suddenly, and the resulting value becomes an outlier. Thus, the RRCF algorithm can be used to detect events based on the power differential series.

First, the random forest is initialized with 100 data points obeying a normal distribution $X \sim N(0,5)$. Then, the aggregated load power series can be obtained in real time, and the anomaly score is calculated for each point in the power differential series. The anomaly score threshold is set in advance and compared with the anomaly scores. When the current anomaly score exceeds the threshold, the power difference value changes, indicating that an event may occur. When the anomaly score converges to the threshold, it can be preliminarily deduced that the event is finished.

When the anomaly score threshold is set to be 20, the event detection results for Fig. 3 are shown in Fig. 4. As shown in Fig. 4, in addition to the two actual events, two falsely detected events are caused by power fluctuations. Therefore, it is necessary to determine whether the event is reasonable or unreasonable by postprocessing.

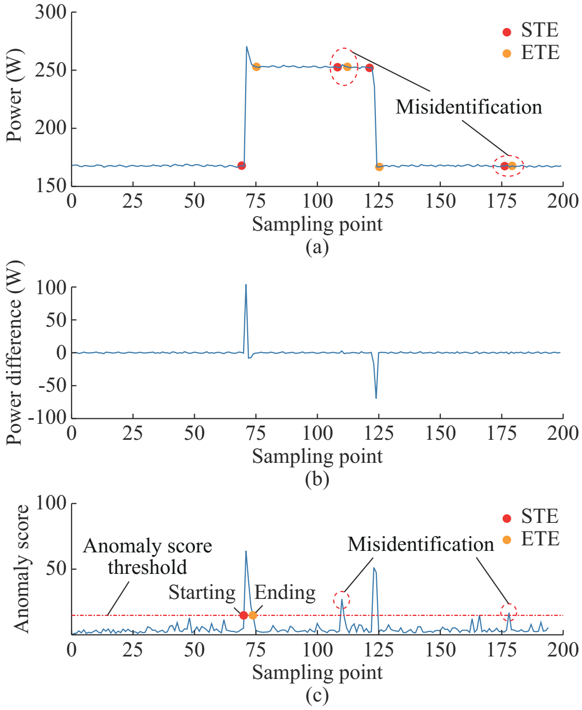

Fig. 4. Event detection results for Fig. 3. (a) Aggregated load power series. (b) Power difference. (c) Anomaly score.

C. Postprocessing

The primary purpose of postprocessing is to eliminate misidentifications caused by power fluctuations and to accurately locate the STE and ETE.

1) Inhibition of Misidentification Based on Adaptive Power Difference Threshold

Most event misidentifications are false-positive events, i.e., one event is falsely detected, although there is no actual event, which usually arises due to power fluctuations in high-power appliances [12]. These misidentifications can be eliminated by setting a power difference threshold and comparing it with the power difference between the STE and ETE. However, when power fluctuation is high, the performance of a fixed power difference threshold is not ideal. Load events with low power may be missed when the threshold is too large. When the threshold is too small, there are many misidentifications in the case of large fluctuations. Hence, the power difference threshold needs to be dynamically updated to adapt to fluctuations or noises in the aggregated power signal.

In this paper, the standard deviation of the aggregated power signal in the steady state is used to adjust the threshold, which is expressed as:

$$\Delta P_{thr}=\max\left(\Delta P_0, sd+\Delta P_0\cdot\arctan\left(\frac{sd}{\Delta P_0}\right)\frac{4}{\pi}\right) \tag{3}$$

where ΔP_{thr} is the adaptive power difference threshold; ΔP_0 is the preset threshold with the zero standard deviation; and sd is the standard deviation of the steady state before the latest event. As shown in (3), when sd is very small, ΔP_{thr} is equal to ΔP_0. With the increase of sd, the corresponding ΔP_{thr} also increases. The changing trend of ΔP_{thr} is shown in Fig. 5.

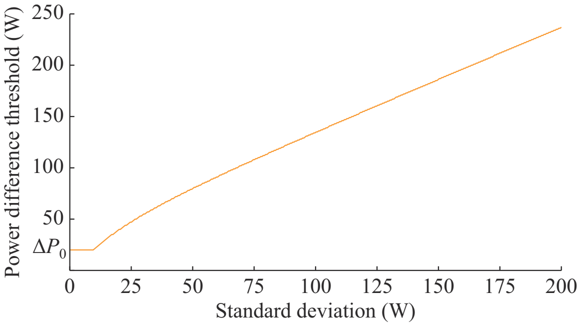

Fig. 5. Changing trend of ΔP_{thr}.

The value of ΔP_0 is defined by the electric customer. For example, if the customer pays attention only to high-power load events, the value of ΔP_0 can be set higher; if the customer also requires attention be paid to low-power load events, the value of ΔP_0 can be set lowered. When the absolute value of the power difference is greater than the threshold, the event is considered true; otherwise, it is deemed false and can be eliminated.

2) *Adjustment of STE and ETE*

The purpose of event detection is to extract the features of the load causing the event; thus, it is essential to locate the STE and ETE accurately. However, the STE and ETE found by the RRCF algorithm are only the starting and ending time of outliers in the aggregated load power series, as shown in Fig. 6, and these values are not accurate enough. Thus, the accurate STE and ETE in the steady state still need to be obtained.

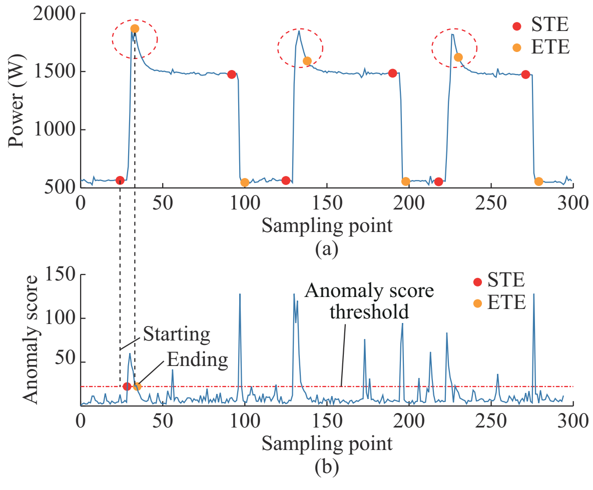

Fig. 6. Inaccurate STE and ETE. (a) Aggregated load power series. (b) Anomaly score.

As shown in Fig. 6, the ETE is detected by the RRCF algorithm, which usually occurs slightly earlier, as highlighted through the red dotted line circle; that is, the event is considered to have ended before the steady state is completely reached. To determine the accurate STE and ETE, several data points near the starting and ending points detected by the RRCF algorithm are selected and then the linear fitting is carried out to calculate the slope and goodness of fit. For example, as shown in Fig. 7, to obtain an accurate ETE, the algorithm selects some ETE points for the linear fitting process. These ETE points include both those that are in an unstable state and those that reach a steady state. Two ETE points are passed by the brown and green lines, which have steep slopes that indicate they are in an unstable state, and one ETE point is passed by the red line, which has gradual slope that represents it is located in a steady state.

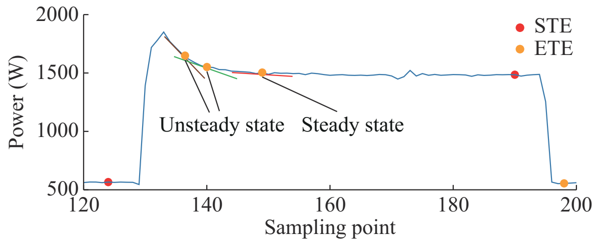

Fig. 7. Principle of accurate identification of event.

Linear fitting is implemented according to the following equations:

$$\bar{P}=ax+b \tag{4}$$

$$a=\frac{\dfrac{\sum_{i=1}^{N}P(t_0\pm i)\sum_{i=1}^{N}x(i)}{N}-\sum_{i=1}^{N}x(i)P(t_0\pm i)}{\dfrac{1}{N}\left(\sum_{i=1}^{N}x(i)\right)^2-\sum_{i=1}^{N}x^2(i)} \tag{5}$$

$$b=\frac{\sum_{i=1}^{N}P(t_0\pm i)-a\sum_{i=1}^{N}x(i)}{N} \tag{6}$$

$$r=1-\frac{\sum_{i=1}^{N}\left(P(t_0\pm i)-\bar{P}(i)\right)^2}{\sum_{i=1}^{N}\left(P(t_0\pm i)-\dfrac{1}{N}\sum_{i=1}^{N}P(t_0\pm i)\right)^2} \tag{7}$$

where N is the number of the selected data points; $P=\{P(t_0\pm i)|i=1,2,...,N\}$, and t_0 is the STE or ETE detected by the RRCF algorithm; + and – of ± are for the ETE and STE, respectively; $x=\{x(i)=i|i=1,2,\dots,N\}$; $\bar{P}$ is the power value after linear fitting; and a, b, and r are the slope, bias, and goodness of fit, respectively. The thresholds of slope and goodness of fit are set as a_{thr} and r_{thr}, respectively. When the slope and goodness of fit meet the criteria $a<a_{thr}$ and $r>r_{thr}$, the steady state is reached. No adjustment is needed if another event occurs before the above conditions are met.

D. *Detailed Algorithm*

The detailed algorithm for the proposed method is shown in Algorithm 1, where "*event flag*" indicates whether a possible event occurs, and "*adjust flag*" indicates whether the latest detected STE and ETE are adjusted properly.

```
Algorithm 1: detailed algorithm for proposed method
Input: aggregated load power data
Output: load event list
Step 1: initialize the random forest with 100 data points obeying normal
  distribution $X \sim N(0,5)$ and all thresholds in algorithm
Step 2: sample the aggregated load power series in real time
Step 3: apply mean-pooling processing to high-frequency data. Then, ob-
  tain the power differential series via calculation
Step 4: input power differential series into the random forest to calculate
  the anomaly score for each data point
Step 5: if anomaly score is larger than threshold $Score_{thr}$, then event flag
  is set to be 1 and return to Step 2 to continue sampling
      else
        if event flag = 1: event flag is reset to be 0 and go to Step 6
        else: go to Step 7
Step 6: calculate the power difference $\Delta P$ before and after the event
      if $\Delta P > \Delta P_{thr}$: adjust flag is set to be 0 and record event. Return to
        Step 2 to continue sampling
      else: go to Step 9
Step 7: if adjust flag = 1: go to Step 9
      else: go to Step 8
Step 8: calculate the slope $a$ and the goodness of fit $r$ by the linear fitting
  before the next event occurs
      if $a$ is smaller than threshold $a_{thr}$ and $r$ is larger than threshold
        $r_{thr}$: adjust the recorded event information in Step 6 and adjust
        flag is set to be 1. Return to Step 2 to continue sampling
      else: go to Step 2 to continue sampling
Step 9: an adaptive power difference threshold is calculated by the stan-
  dard deviation of the actual aggregated load power data and go to Step
  2 to continue sampling
```

IV. Case Study

In this section, the proposed method is validated on public datasets with different sampling rates. First, the parameters of the RRCF algorithm that can meet the real-time requirement are determined. Then, the detection results for several challenging events are discussed, and the proposed method is validated on both the BLUED dataset (with both high and low sampling rates) and the REDD dataset (with low sampling rates). Finally, the advantages of the proposed method are demonstrated by comparing it with other methods.

A. Parameter Determination of RRCF Algorithm

The RRCF algorithm has three parameters: the number of trees, the size of the tree, and the shingle size. The shingle size is set to be 1 because the power data are sampled once at a time. Since the event detection task needs to be run in real time, the other two parameters should be kept as small as possible to meet real-time requirements. Besides, the high identification accuracy should be maintained. After testing, when the number of trees is set to be 2 and the size of the tree is set to be 64, the time cost is approximately 0.33 s for 400 sampling points, which means that the proposed method can meet the real-time requirement while achieving satisfactory accuracy. All of the following tests are performed with these parameters.

B. Detection of Several Challenging Events in Scenario with a High Sampling Rate

In this subsection, the detection results for several challenging events on the BLUED dataset [28] are presented. This dataset includes the household-grade voltage and current data with a high sampling frequency (12 kHz) from one household in the United States over a period of approximately 8 days. This dataset contains aggregated load power series at a sampling rate of 60 Hz.

First, mean-pooling processing is employed to eliminate high fluctuations, taking the average value of every three data points. Therefore, the time interval between two adjacent data points is 0.05 s. Although mean-pooling processing is very simple, it effectively inhibits the periodic interference.

The parameter values used in event detection for the BLUED dataset are shown in Table I, and the length of the data points for linear fitting is set to be 6. ΔP_0 is set to be 30 W, which is the minimum power change for an individual load based on the BLUED dataset. After several validations, the optimal anomaly score threshold is turned to be 35.

TABLE I
PARAMETER VALUES USED IN EVENT DETECTION FOR BLUED DATASET

Anomaly score threshold	ΔP_0 (W)	Slope threshold a_{thr}	Goodness threshold r_{thr}
35	30	5	0.8

We then select several challenging events from the BLUED dataset and verify the detection performance of the proposed method.

1) Case 1: Repetitive Event Detection Within a Short Time

A repetitive event, as shown in Fig. 8, means that the steady state has many switching on/off events.

As observed from Fig. 8, the proposed method can accurately detect all repetitive events and adjust each STE and ETE. The time interval between the second event and the third event is only approximately 0.35 s, and this event can also be accurately detected. The results show that the proposed method has good sensitivity, which means that it can detect events that occur within a very short time.

2) Case 2: Large-fluctuation Event Detection

As shown in Fig. 9, a large-fluctuation event has serious fluctuation in the steady state.

As shown in Fig. 9(b), there are some misidentifications due to high fluctuations when the power difference threshold is fixed at 30 W. In Fig. 9(c), the red dotted line circle indicates the large fluctuations during this period. Thus, using a specified threshold, these misidentifications are detected as actual events. The adaptive power difference threshold can be updated according to the fluctuations, as shown in Fig. 9(d). Thus, the misidentifications caused by fluctuations can be effectively eliminated and the accuracy of event detection is improved.

3) Case 3: Long Transient Event Detection

For long transient events, it takes a long time to reach the steady state after the STE.

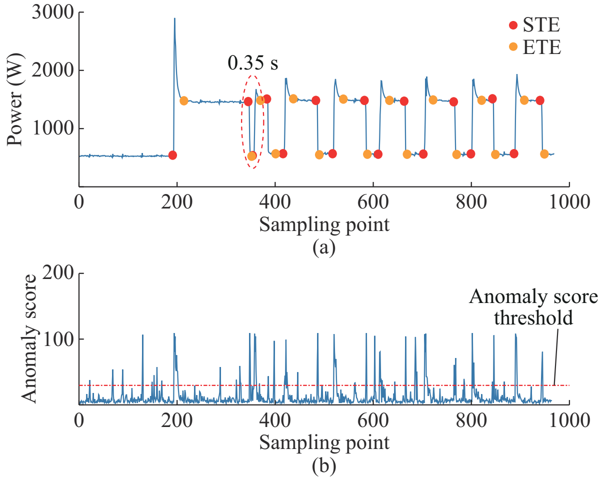

Fig. 8. Repetitive event detection within a short time. (a) Aggregated load power series. (b) Anomaly score.

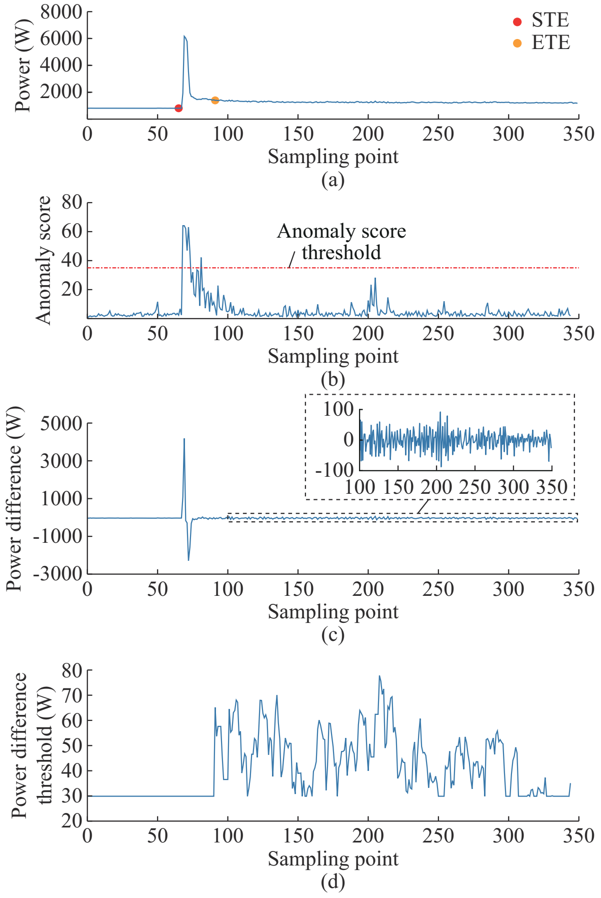

Fig. 9. Large-fluctuation event detection. (a) Aggregated load power series. (b) Anomaly score. (c) Power difference. (d) Power difference threshold.

There are two main types of long transient events in the scenario with a high sampling rate, i. e., medium-long transient events and ultra-long transient events (which may be several minutes), as shown in Fig. 10. The medium-long transient event takes 0.7 s to reach the steady state, and the ultra-long transient event will keep the transient process for almost 10 s.

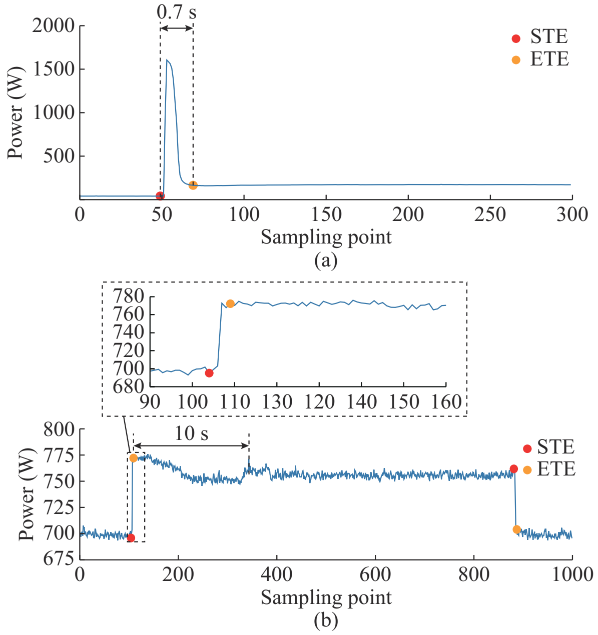

Fig. 10. Long transient event detection. (a) Medium-long transient event. (b) Ultra-long transient event.

Several studies have attempted to identify the entire transient process of an event [12], [26]; however, in practice, it is often difficult to obtain the whole transition process. This is because the load-switching event is irregular, which cannot guarantee that no other event occurs during the transient process. The purpose of event detection is to extract load features for load identification. Therefore, if enough load features can be extracted, event detection can be achieved. To meet this objective, the local steady state (the steady state of a load during a transient process) in the long transient event transition is sufficient to extract the needed load features.

For long transient events, the entire transition process of the medium-long transient event can be detected, as shown in Fig. 10(a), while the local steady state of the ultra-long transient event can be detected at the end of the event, as shown in Fig. 10(b).

The results show that the proposed method has high practicality, which means that it can detect another event that occurs when the load has not reached the steady state; this is also shown in the following case.

4) Case 4: Near-simultaneous Event Detection

When two events occur within a very short time, it is called a near-simultaneous event, as shown in Fig. 11(a). This event is similar to case 1, but the difference is that in this case the two events occur because of two different loads. In Fig. 11(a), the interval time between the first event and the second event is approximately 0.3 s, and when the second event occurs, the first event is still in the transient process, as indicated via the red dotted line circle.

Suppose that the information of the local steady state is not used. In this case, the near-simultaneous events that occur from different loads are detected as a single event, and it is difficult to extract the features of the load in the subsequent load identification stage.

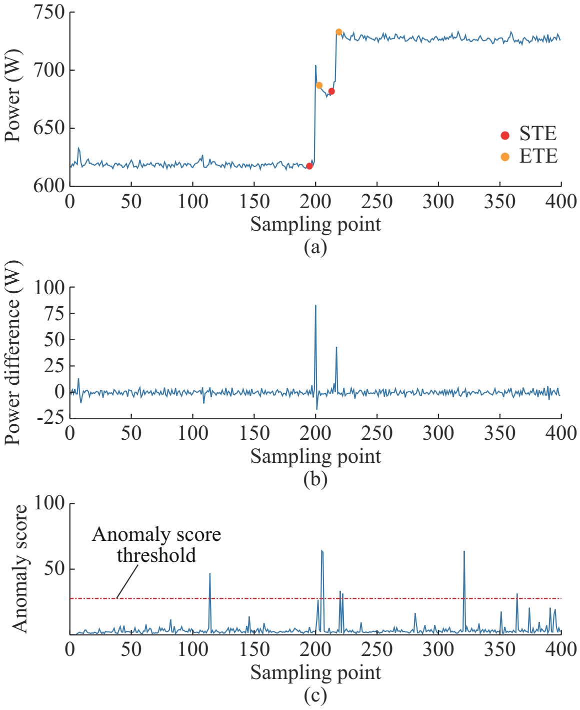

Fig. 11. Near-simultaneous event detection. (a) Aggregated load power series. (b) Power differential series. (c) Anomaly score.

The proposed method can accurately detect near-simultaneous events by using local steady-state data points during the transient process of the event.

C. Validation in Scenario with a High Sampling Rate

In this subsection, the proposed method is validated in the scenario with a high sampling rate with the BLUED dataset. The preprocessing procedure for data with a high sampling rate is the same as that in Section IV-B. Therefore, the final sampling frequency is 20 Hz. In the BLUED dataset, the appliances are connected to two phases: phase A and phase B. Phase A contains appliances with relatively stable power. In contrast, phase B has appliances with relatively large power fluctuations. This dataset also provides true labels about the occurrence times of events, with 904 events recorded in phase A and 1578 events recorded in phase B.

The event detection results during certain time periods, where load events are very frequent, are shown in Fig. 12. The aggregated load power series of phase B fluctuates substantially and is much more complex than that of phase A. In phase A, 940 events are finally detected by the proposed method, which is greater than the actual 904 events. In phase B, 2001 events are finally detected by the proposed method, which is greater than the actual 1578 events. The number of detected events is greater than that of actual events due to the separation of repetitive events in the running state of a particular load into several events, as shown in Fig. 12(c). In phase B, a few repetitive events are not recorded due to the short time period between adjacent events and the large power fluctuation, as shown by the red dotted line circle in Fig. 12(e).

The event detection results show that the proposed method can accurately detect the STE and ETE and effectively eliminate the misidentification caused by fluctuations. Although some events in phase B are not recorded, this problem is not severe because these events are repetitive events caused by the same load.

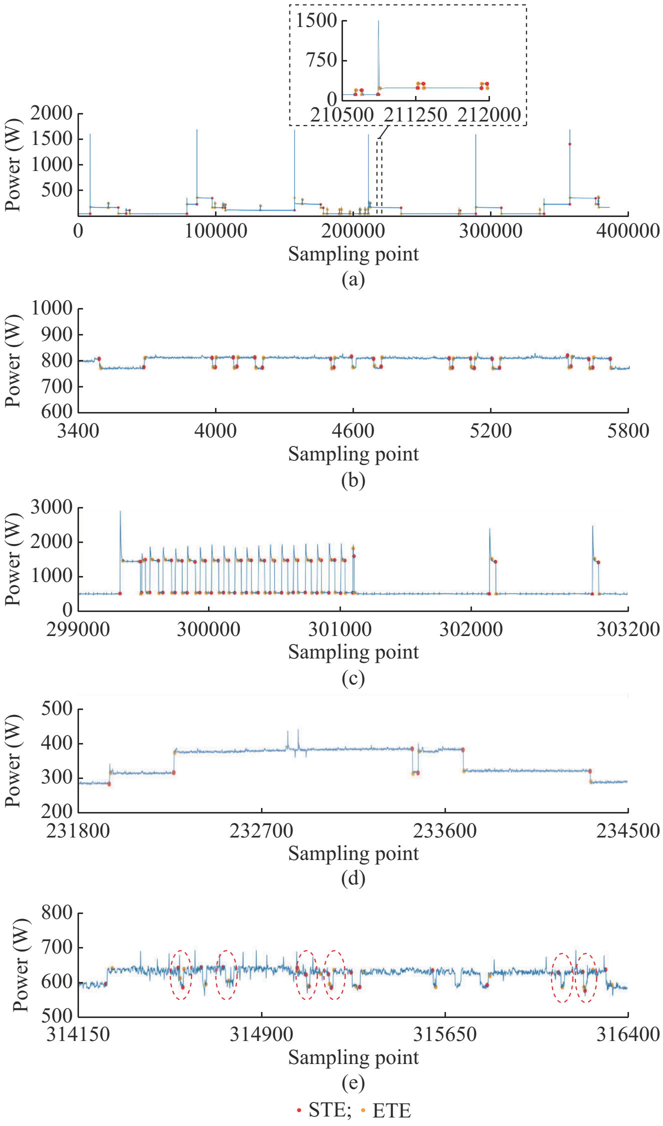

Fig. 12. Event detection results in BLUED dataset with a high sampling rate. (a) Phase A. (b) Phase B (from sampling points 3400 to 5800). (c) Phase B (from sampling points 299000 to 303200). (d) Phase B (from sampling points 231800 to 234500). (e) Phase B (from sampling points 314150 to 316400).

D. Validation in Scenario with a Low Sampling Rate

In this subsection, the proposed method is validated in the scenario with a low sampling rate with both BLUED and REDD datasets.

1) Event Detection Results in BLUED Dataset

First, the proposed method is validated with the BLUED dataset. To reduce the sampling rate to 1 Hz, the number of data points in the mean-pooling processing is set to be 60. Meanwhile, the slope threshold a_{thr} is set to be 10, which is a larger value than that in the scenario with a high sampling rate. This is because the time interval between adjacent data points of the aggregated load power series in the scenario with a low sampling rate is longer compared with that in the scenario with a high sampling rate. The other parameters are the same as those in the scenario with a high sampling rate

The event detection results for a selected day that is the most representative in a week are shown in Fig. 13. As shown in Fig. 13(a), the events in phase A can be accurately detected. However, the event detection results of phase B are not as good as those in the scenario with a high sampling rate, mainly because of the repetitive event. In the repetitive event, only one data point can be sampled in each repetition; thus, these data points in the aggregated load power series are closer to the impulse fluctuation, as shown in Fig. 13(c). This deficiency is due to the low sampling rate.

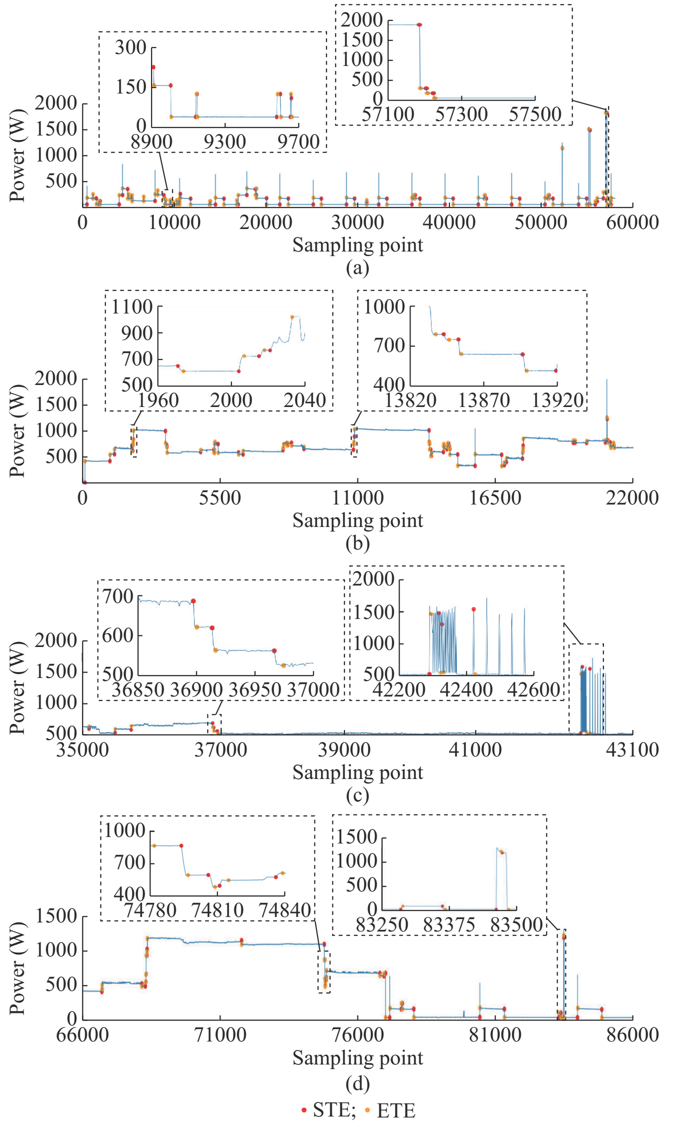

Fig. 13. Event detection results in BLUED dataset with a low sampling rate. (a) Phase A. (b) Phase B (from sampling points 0 to 22000). (c) Phase B (from sampling points 35000 to 43100). (d) Phase B (from sampling points 66000 to 86000).

2) *Event Detection Results in REDD Dataset*

The performance of the proposed method is validated in the REDD dataset as well. The REDD dataset provides the aggregated load power series that are recorded at 1 Hz and collected from six real houses. Considering the complexity of appliance composition, we select the aggregated load power series for approximately one day from House 1 to validate the proposed method.

The event detection results in the REDD dataset are shown in Fig. 14. As can be observed, the proposed method performs well in the scenario with a low sampling rate. The REDD dataset also contains many repetitive events, as shown in Fig. 14(b) - (d). However, the time period of the switching on/off state in Fig. 14 is longer than that in Fig. 13, with most lasting more than 1 s; therefore, the proposed method can accurately detect all repetitive events.

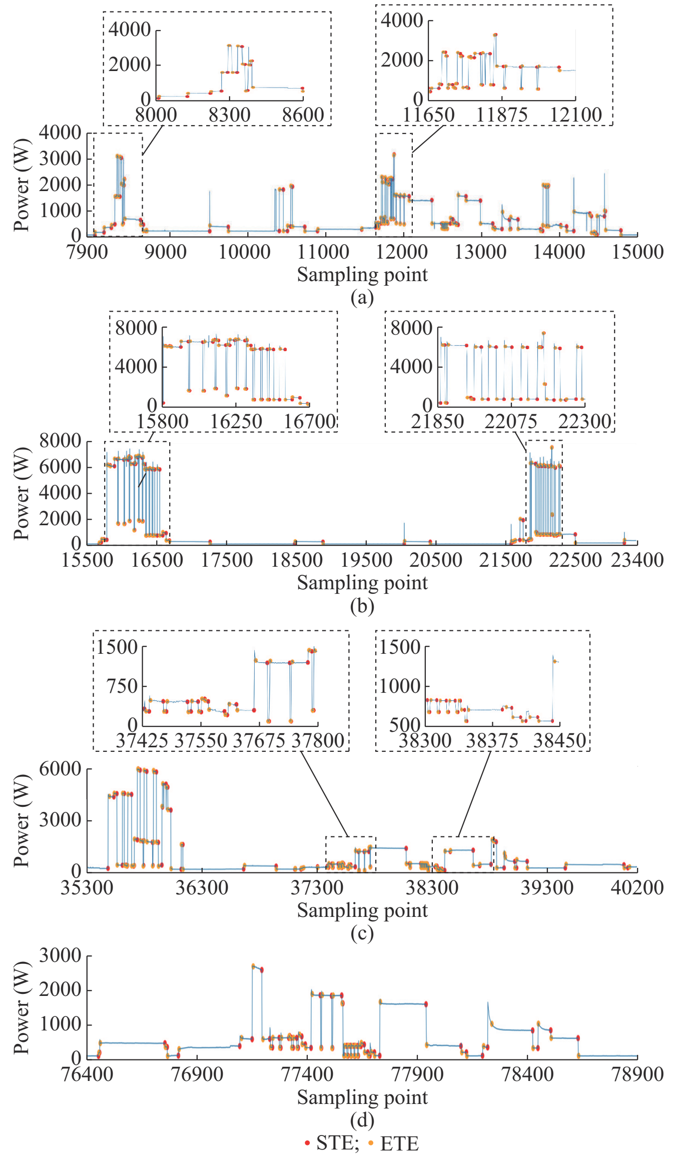

Fig. 14. Event detection results in REDD dataset. (a) Results from sampling points 7900 to 15000. (b) Results from sampling points 15500 to 23400. (c) Results from sampling points 35300 to 40200. (d) Results from sampling points 76400 to 78900.

The validation results for the REDD dataset show that the proposed method can perform well for different households, which means that it has high adaptability.

E. Comparison with Other Methods

In order to show the superiority of the proposed method, it is compared with several well-known methods for the BLUED dataset. The comparison results are shown in Table III in terms of three metrics, namely, the correct rate *Precision*, the recall rate *Recall*, and the F1 score F_1, which are calculated as:

$$F_1 = 2\frac{Precision \cdot Recall}{Precision + Recall} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

where *TP*, *FP*, and *FN* denote the true positive, false positive, and false negative cases, respectively.

TABLE III
COMPARISON WITH OTHER METHODS

Method	Aggregated signal	Sampling rate (Hz)	Phase	*Precision* (%)	*Recall* (%)	F_1 (%)
Proposed method	Active power	20	A	99.40	100.00	99.70
			B	92.60	95.50	94.03
		1	A	99.60	99.40	99.50
			B	94.60	74.20	83.17
[12]	Active power	60	A	99.20	99.20	99.20
[14]	Active power	1	A	99.30	81.01	89.22
			B	77.72	57.04	65.79
[25]	Current	1	A	98.96	99.48	99.22
[26]	Active power	1	A	98.87	100.00	99.43
			B	79.98	92.55	85.81

The comparison results show that the proposed method outperforms other state-of-the-art methods. The method in [14] focuses only on the STE. The methods in [25] and [26] try to detect the whole transition process of the event. Nevertheless, these methods do not allow another event to occur during the long transient event detection, which is not practical in reality.

In addition, the event detection methods in the references can only be applied to low-frequency or high-frequency power data, while the proposed method can be used in both scenarios. With respect to the scenario with a high sampling rate, the proposed method can accurately detect both the STE and ETE of all events; thus, it can provide a favourable foundation for subsequent load identification. Although repetitive event detection in scenarios with a low sampling rate is not as effective as that in scenarios with a high sampling rate, the first and last events are accurately detected, providing sufficient information to perform subsequent load identification. Other events, except for repetitive events in scenarios with low sampling rates, can be accurately detected.

V. CONCLUSION

In this paper, an event detection method based on the RRCF algorithm is proposed. The power differential series is input into the random forest, and the anomaly score is calculated for each data point to roughly determine the STE and ETE. Then, postprocessing is carried out to inhibit misidentifications by using an adaptive power difference threshold and accurately locating the STE and ETE. The proposed method is validated in the BLUED and REDD datasets. The results illustrate that the proposed method outperforms other state-of-the-art methods. The STE and ETE can be accurately detected by the proposed method, and the adaptive power difference threshold can enhance the accuracy by eliminating the misidentifications caused by fluctuations. In addition, the proposed method has high adaptability because it performs well in different scenarios with distinct sampling rates.

The proposed method offers a favourable foundation for NILM problems in different scenarios, but there are still some limitations that require further research. For example, with growth of the dataset and changes in data distribution, the accuracy might be influenced. Therefore, follow-up research could explore mechanisms for adaptation to ensure the continued accuracy of event detection.

In addition, the energy consumption modes related to NILM problems need to be considered. Both [32] and [33] improved the energy efficiency based on residential occupancy information via various probabilistic prediction methods. Additionally, [34] identified the periods when high demand occurs by building a closely tied relationship with residential occupancy patterns and consumer activities, catering to demand response and improving energy efficiency. In future work, the proposed method could be combined with the demand response capability identification to promote energy efficiency and sustainability in buildings.

REFERENCES

[1] A. Gabaldón, R. Molina, A. Marín-Parra *et al*., "Residential end-uses disaggregation and demand response evaluation using integral transforms," *Journal of Modern Power Systems and Clean Energy*, vol. 5, no. 1, pp. 91-104, Jan. 2017.

[2] E. Tabanelli, D. Brunelli, A. Acquaviva *et al*., "Trimming feature extraction and inference for MCU-based edge NILM: a systematic approach," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 943-952, Feb. 2022.

[3] S. Wang, H. Chen, L. Guo *et al*., "Non-intrusive load identification based on the improved voltage-current trajectory with discrete color encoding background and deep-forest classifier," *Energy and Buildings*, vol. 244, p. 111043, Aug. 2021.

[4] B. Buddhahai, W. Wongseree, and P. Rakkwamsuk, "An energy prediction approach for a nonintrusive load monitoring in home appliances," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 1, pp. 96-105, Feb. 2020.

[5] J. Zhang, X. Chen, W. W. Y. Ng *et al*., "New appliance detection for nonintrusive load monitoring," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4819-4829, Aug. 2019.

[6] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870-1891, Aug. 1992.

[7] J. S. Kang, M. Yu, L. Lu *et al*., "Adaptive non-intrusive load monitoring based on feature fusion," *IEEE Sensors Journal*, vol. 22, no. 7, pp. 6985-6994, Apr. 2022.

[8] A. U. Rehman, T. T. Lie, B. Vallès *et al*., "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, Sept. 2021.

[9] Z. Zhou, Y. Xiang, H. Xu *et al*., "Unsupervised learning for non-intrusive load monitoring in smart grid based on spiking deep neural network," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 3, pp. 606-616, May 2022.

[10] S. Dash and N. C. Sahoo, "Electric energy disaggregation via non-in-

trusive load monitoring: a state-of-the-art systematic review," *Electric Power Systems Research*, vol. 213, p. 108673, Dec. 2022.

[11] Y. Liu, L. Zhong, J. Qiu *et al*., "Unsupervised domain adaptation for nonintrusive load monitoring via adversarial and joint adaptation network," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 266-277, Jan. 2022.

[12] L. Yan, W. Tian, H. Wang *et al*., "Robust event detection for residential load disaggregation," *Applied Energy*, vol. 331, p. 120339, Feb. 2023.

[13] A. U. Rehman, T. T. Lie, B. Valles *et al*., "Event-detection algorithms for low sampling nonintrusive load monitoring systems based on low complexity statistical features," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 3, pp. 751-759, Mar. 2020.

[14] M. Lu and Z. Li, "A hybrid event detection approach for non-intrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 528-540, Jan. 2020.

[15] S. Kotsilitis, E. Kalligeros, E. C. Marcoulaki *et al*., "An efficient lightweight event detection algorithm for on-site non-intrusive load monitoring," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1-13, Dec. 2023.

[16] A. Yasin and S. A. Khan, "Unsupervised event detection and on-off pairing approach applied to NILM," in *Proceedings of 2018 International Conference on Frontiers of Information Technology*, Islamabad, Pakistan, Dec. 2018, pp. 123-128.

[17] M. Liu, J. Yong, X. Wang *et al*., "A new event detection technique for residential load monitoring," in *Proceedings of 2018 18th International Conference on Harmonics and Quality of Power*, Ljubljana, Slovenia, May 2018, pp. 1-6.

[18] B. Völker, P. M. Scholl, and B. Becker, "Semi-automatic generation and labeling of training data for non-intrusive load monitoring," in *Proceedings of the 10th ACM International Conference on Future Energy Systems*, Phoenix, USA, Jun. 2019, pp. 17-23.

[19] S. Zhang, Z. Zhu, B. Yin *et al*., "Event detection methods for nonintrusive load monitoring in smart metering: using the improved CUSUM algorithm," in *Proceedings of 2018 International Conference on Sensing*, *Diagnostics*, *Prognostics*, *and Control*, Xi'an, China, Aug. 2018, pp. 738-742.

[20] L. de Baets, J. Ruyssinck, C. Develder *et al*., "On the Bayesian optimization and robustness of event detection methods in NILM," *Energy and Buildings*, vol. 145, pp. 57-66, Jun. 2017.

[21] F. Ciancetta, G. Bucci, E. Fiorucci *et al*., "A new convolutional neural network-based system for NILM applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-12, Nov. 2021.

[22] M. Afzalan, F. Jazizadeh, and J. Wang, "Self-configuring event detection in electricity monitoring for human-building interaction," *Energy and Buildings*, vol. 187, pp. 95-109, Mar. 2019.

[23] A. E. Lazzaretti, D. P. B. Renaux, C. R. E. Lima *et al*., "A multiagent NILM architecture for event detection and load classification," *Energies*, vol. 13, no. 17, p. 4396, Aug. 2020.

[24] S. Houidi, F. Auger, H. B. A. Sethom *et al*., "Multivariate event detection methods for non-intrusive load monitoring in smart homes and residential buildings," *Energy and Buildings*, vol. 208, p. 109624, Feb. 2020.

[25] F. Zhang, L. Qu, W. Dong *et al*., "A novel NILM event detection algorithm based on different frequency scales," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1-11, Jun. 2022.

[26] W. Luan, Z. Liu, B. Liu *et al*., "An adaptive two-stage load event detection method for nonintrusive load monitoring," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1-14, Dec. 2022.

[27] A. Wang, B. Chen, C. Wang *et al*., "Non-intrusive load monitoring algorithm based on features of *V-I* trajectory," *Electric Power Systems Research*, vol. 157, pp. 134-144, Apr. 2018.

[28] K. Anderson, A. Ocneanu, D. Benitez *et al.*, "BLUED: a fully labeled public dataset for event-based non-intrusive load monitoring research," in *Proceedings of 2nd Workshop on Data Mining Applications in Sustainability*, Beijing, China, Aug. 2012, pp. 1-5.

[29] J. Z. Kolter and M. J. Johnson, "REDD: a public data set for energy disaggregation research," in *Proceedings of Workshop on Data Mining Applications in Sustainability*, San Diego, USA, Aug. 2011, pp. 59-62.

[30] S. Guha, N. Mishra, G. Roy *et al.*, "Robust random cut forest based anomaly detection on streams," in *Proceedings of the 33rd International Conference on Machine Learning*, New York, USA, Jun. 2016, pp. 2712-2721.

[31] M. Dong, M. Sun, D. Song *et al*., "Real-time detection of wind power abnormal data based on semi-supervised learning robust random cut forest," *Energy*, vol. 257, p. 124761, Oct. 2022.

[32] J. Chaney, E. H. Owens, and A. D. Peacock, "An evidence based approach to determining residential occupancy and its role in demand response management," *Energy and Buildings*, vol. 125, pp. 254-266, Aug. 2016.

[33] L. He, Y. Liu, and J. Zhang, "An occupancy-informed customized price design for consumers: a Stackelberg game approach," *IEEE Transactions on Smart Grid*, vol. 13, no. 3, pp. 1988-1999, May 2022.

[34] G. Tang, Z. Ling, F. Li *et al*., "Occupancy-aided energy disaggregation," *Computer Networks*, vol. 117, pp. 42-51, Apr. 2017.

Lingxia Lu received the B.S. degree in automation from Northeast University, China, in 2005, and the M.S. degree in control theory and control engineering from Zhejiang University, Hangzhou, China, in 2007. Afterwards, she worked as a Lecturer at the College of Electrical Engineering, Zhejiang University. Her research interests include embedded systems, smart energy system, and information security in Internet of Things (IoT).

Ju-Song Kang received the B.S. degree in automation from Zhejiang University, Hangzhou, China, in 2021. His main research interests include embedded system and machine learning.

Miao Yu received the B.S. degree in automation from Zhejiang University, Hangzhou, China, in 2007, and the Ph.D. degree in control theory and control engineering from Zhejiang University, in 2012. From 2013 to 2015, he worked in Aalto University, Espoo, Finland, as a Postdoctoral Researcher. Since 2016, he has been working in College of Electrical Engineering, Zhejiang University, where he is now a Professor. His current research interests include renewable power generation, control strategy in the microgrid, and active distribution networks.