Practical Insights from Implementing Event-Based NILM Systems

Justus Breyer
RWTH Aachen University
Aachen, Germany
justus.breyer@comsys.rwth-aachen.de

Jonas Koerhuis
RWTH Aachen University
Aachen, Germany
jonas.koerhuis@rwth-aachen.de

Muhammad Hamad Alizai
LUMS
Lahore, Pakistan
hamad.alizai@lums.edu.pk

Klaus Wehrle
RWTH Aachen University
Aachen, Germany
wehrle@comsys.rwth-aachen.de

Abstract

Improving residential energy efficiency is essential for sustainability, and Non-Intrusive Load Monitoring (NILM) is a promising technology that provides detailed insights into energy consumption without requiring individual appliance monitoring. However, creating accurate NILM models using publicly available datasets involves substantial challenges. This paper identifies critical pitfalls in dataset handling, event detection accuracy, and feature extraction, aspects often undiscussed in prior literature. We propose novel algorithms to rectify event timestamp inaccuracies and effectively extract transient signals associated with appliance state changes. We rigorously evaluate our pipeline on multiple public datasets, analyzing feature stability and assessing the impact of aggregated versus isolated data. The insights from our practical implementation and evaluation aim to assist researchers in overcoming common early-stage obstacles in developing robust NILM systems and enhancing their applicability in real-world scenarios.

CCS Concepts

• **Hardware** → **Energy metering**; • **Computing methodologies** → *Supervised learning*.

Keywords

machine learning, non-intrusive load monitoring (NILM), energy disaggregation, load signatures

ACM Reference Format:
Justus Breyer, Jonas Koerhuis, Muhammad Hamad Alizai, and Klaus Wehrle. 2025. Practical Insights from Implementing Event-Based NILM Systems. In *The 16th ACM International Conference on Future and Sustainable Energy Systems (E-ENERGY '25), June 17–20, 2025, Rotterdam, Netherlands.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3679240.3734643

1 Introduction

Climate change necessitates substantial reductions in greenhouse gas emissions, which requires lowering energy consumption across multiple sectors. Detailed insights into individual energy usage patterns have been shown to motivate consumers towards energy-saving behaviors [5, 8]. Such insights typically rely on installing dedicated meters on every appliance; however, this approach places substantial installation and maintenance burdens on users.

Non-intrusive Load Monitoring (NILM) [9] offers a compelling alternative by analyzing aggregate energy consumption data to infer appliance-specific usage patterns. NILM employs machine learning (ML) models to disaggregate total household energy consumption into detailed appliance-level information. Despite its promise, NILM presents several technical and practical challenges, including the diversity of household appliances and the extensive data requirements for training ML models. Furthermore, the inherent complexity of NILM systems demands multiple fully-functional components to achieve effective performance, making implementation challenging for developers lacking substantial external support.

While extensive research has been conducted on NILM and the benchmarking of various ML models (e.g.,[7]), existing literature often overlooks critical early-stage processing details, favoring instead comprehensive discussions of model architectures and evaluation metrics. Publications focusing explicitly on event detection (e.g.,[1]) typically emphasize detection performance—i.e., whether events are successfully identified—rather than exploring event timing accuracy and its subsequent impact on downstream tasks. Similarly, studies addressing the calculation and performance evaluation of features [11, 16] typically provide theoretical formulas without adequately detailing their practical implementation on real-world datasets, especially concerning the handling of background noise.

To bridge these gaps, this study provides practical guidance for researchers focusing on classification and disaggregation tasks by outlining critical preprocessing steps on publicly available datasets. Specifically, we make the following contributions:

Contributions.

- We propose a straightforward yet effective method for adjusting event timestamps produced by standard detection algorithms.
- We introduce a simple algorithm for extracting transient signals around event timestamps, facilitating more effective downstream processing.
- We implement and systematically evaluate our timestamp adjustment and transient extraction techniques on publicly available datasets, utilizing publicly accessible event detection code.
- We investigate feature calculation and stability in isolated and aggregated energy data, highlighting noise introduced by simultaneous device activations and emphasizing the importance of carefully selecting areas of interest for feature extraction.

E-ENERGY '25, Rotterdam, Netherlands

ACM ISBN 979-8-4007-1125-1/25/06
https://doi.org/10.1145/3679240.3734643

This paper thoroughly documents practical strategies for addressing frequent early-stage challenges in NILM pipelines, aiming to support future research and innovation. These shared insights are intended to simplify the development of novel ML methods and improve the effectiveness of NILM solutions overall.

2 General NILM Pipeline

To deploy an NILM setup, multiple steps need to be considered. There have been two main approaches to this problem: *event-based* and *eventless* NILM [7], differing in the necessary components. Whilst eventless NILM systems continuously monitor the stream of power measurements, event-based systems on the other hand rely on event detection mechanisms to associate changes of device behavior via classifiers to the respective devices. Since our contributions lean heavily towards the event-based approach, the following paragraph focuses on the fundamental complexity of this category.

Data collection is crucial for all NILM approaches, leveraging sources like smart meters to gather consumption data. This data supports model training pre-deployment and informs energy disaggregation post-deployment. The research community has over the years established a variety of publicly available datasets [12, 15, 19] to spare the effort of data collection for further research on model development and enable a common ground for benchmarking. **Event detection**, specific to event-based NILM, identifies appliance state changes (e.g., on/off) through variations in power consumption. Event detection employs various methods such as expert heuristics (e.g., threshold detection), probabilistic models, or matched filters [1], grounded in the Switch Continuity Principle (SCP) [9]. SCP asserts that only one appliance alters its state in a specific time frame, requiring a particular sampling rate for reliability, as it becomes less dependable over intervals of several seconds [14]. Often, public datasets come with published metadata containing event information [15, 19]; the research community can also make use of open-source tools for event detection [2, 18]. **Feature calculation** in event-based NILM involves analyzing data around detected events to differentiate appliances or states, often using data transformations such as Fourier or wavelet transforms, which has been extensively reviewed [10, 11, 16]. **Classification** uses the extracted features to identify the appliance or its state using ML models, with event-based NILM having the advantage to draw from simple algorithms like RF or SVM [11]. **Energy disaggregation** in event-based NILM matches events to appliance states and their expected consumption for power estimation.

The following sections detail challenges that can arise, insights we gathered and solutions we developed to successfully navigate through the complexities of event-based NILM. These apply regardless of whether we relied on publicly available metadata and code or using our own implementations.

3 Timestamp Misalignment

One of the initial practical challenges in implementing a NILM pipeline is managing timestamp misalignments between events detected using low-frequency data and the corresponding high-frequency aggregated data from public datasets. This may be unavoidable to correctly assign a detected event to its corresponding appliance - older datasets [12, 13] only contain low-frequency measurements of isolated appliances but are popular for performance evaluation, besides others due to the extensive research conducted on them. For instance, the UK-DALE dataset [12], widely adopted in the NILM community due to its high sampling frequency of aggregated data, extensive coverage of diverse household scenarios, and prolonged recording periods, exemplifies this issue clearly. Specifically, UK-DALE provides appliance-level data at a frequency of $\frac{1}{6}Hz$, whereas the aggregated data is recorded at $16kHz$. Consequently, utilizing timestamps directly from event detection tools, such as NilmTK [2], for high-frequency data analysis may result in significant inaccuracies, as illustrated in Figure 1. Such misalignments undermine the accuracy of feature extraction from short intervals around the identified events, resulting in unreliable feature values.

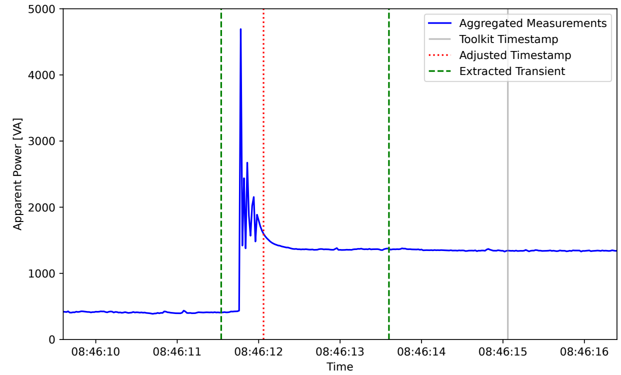

Figure 1: An event of a printer on March 2, 2017, in house 1 of UK-DALE [12] with detected [2] and adjusted timestamp. Extracting the transient is only possible after recalculating the timestamp.

To address this challenge, we propose a straightforward yet effective timestamp adjustment methodology. Initially, a segment spanning $3s$ before and after the detected event (equivalent to half the event detection sampling interval) is extracted from the high-frequency aggregated data and resampled to power values at line frequency, i.e., $50Hz$, the highest rate usable for power values, as they are calculated using root mean squares (RMS) over the periods of consumption. Given the assumption that the original timestamps closely approximate actual event occurrences, we refine them by identifying the inflection point corresponding to the maximum gradient change. As events (in the sense of a change in power consumption) typically do not correspond to just one point in time but rather a small window during which the new power level is reached, the inflection point is usually a timestamp somewhere within this window. A Savitzky-Golay filter [17] with a polynomial order of 2 and a window size of 15, both parameters resulting from optimization, effectively smooths the extracted power curve while preserving the essential structure of the signal. Subsequently, a median filter with a kernel size of 5 eliminates remaining outliers and abrupt signal fluctuations. Finally, by computing the second derivative, we precisely pinpoint the inflection point associated with the highest gradient, thus obtaining a refined event timestamp. Intuitively, choosing the whole area between the preceding and subsequent timestamp of the low-frequency data (e.g., in the case of UK-DALE, an area of $12s$ in total) for finding the correct timestamp would be the safe approach. However, in this specific case, half the window size proved already enough, as throughout our experiments, no inflection points were found beyond the initial $3s$ extension to the left and right of the timestamp. Additionally, multiple detections corresponding to the same event were effectively merged into a single timestamp through this approach, eliminating duplicates. The refined timestamp, exemplified in Figure 1, significantly enhances precision and ensures robust downstream feature calculation. A comparison of feature values in Figure 2 reveals noticeably clearer decision boundaries for respective devices, easing classification.

4 Transient Discovery

One fundamental aspect of NILM is the additive nature of electrical power consumption. NILM systems rely on identifying the unique power consumption signature exhibited by each appliance, enabling recognition from aggregate power readings. Due to power additivity, combined appliance signatures create a complex optimization scenario to reconstruct the individual appliance consumption accurately. Event detectors, as introduced in Section 2, primarily aim to identify timestamps at which appliances transition between operational states. These state transitions rarely occur instantaneously; instead, appliances typically experience a transient period characterized by fluctuations before settling into a stable power state.

Signature characteristics can be described by two categories of features: transient and steady-state. Transient features should be calculated exclusively within the transient interval, whereas steady-state features require a defined region of interest (ROI) post-transient, particularly when using aggregated data, and should be compared against pre-event ROIs to mitigate transient influence on statistical measures such as mean power consumption. However, literature has largely overlooked detailed transient extraction methodologies, with sparse discussions found even in extensive feature studies [11, 16] and specific transient-focused analyses [6].

We introduce a robust yet straightforward method for determining the start and end points of transients, given an initial approximate event timestamp within the transient (e.g., refined via Section 3). This method employs a bidirectional sliding window of 5 periods in size around the timestamp, evaluating the absolute gradient within each window segment. When *all* gradient values in a segment fall below a specified threshold, it indicates the detection of a plateau. To confirm the plateau marks the transient's actual end rather than an intermediate stability, we apply a short forward look-ahead, where $0.5s$ worked most effectively. If the gradient exceeds a higher threshold during this look-ahead, the transient is considered ongoing, prompting further iterations of the sliding window approach from the identified new point.

Empirical evaluations on optimized events from the UK-DALE, FIRED [19], and SustDataED2 [15] datasets demonstrated optimal gradient thresholding at 0.5. Additionally, the look-ahead validation approach was most effective with a window size of 4 and a gradient threshold of 1.8. Figure 1 provides an exemplary visualization of the extracted transient timestamps. Clearly, it would be possible to trim the transient of this example even further, as its elongation might affect transient feature calculation. However, our tests showed that using smaller thresholds resulted in a large increase of false negatives and instances, where the transient was preemptively cut. Reducing the transient range even further without compromising accuracy in an automated way is subject of our ongoing research. Nevertheless, we performed a comparative study, using provided events of the toolkit or metadata respectively as baseline, and were able to achieve a non-negligible improvement of at least 20% on the F1-score on each of the datasets [12, 15, 19] respectively. The details of this study are provided in Appendix A.

5 Feature Calculation & Stability

Even after correcting event detection timestamps and successfully extracting transients, significant challenges remain when processing aggregated electricity data into robust features for NILM classification tasks.

5.1 Handling Aggregated Data

Although prior work provides extensive overviews on feature engineering for NILM [11, 16], we observed a notable gap in the literature: there is little to no discussion on how feature behavior changes when calculated on isolated appliance signals versus aggregated household data.

Our experience suggests that this distinction is not negligible. Background noise and simultaneous device activity can distort certain features that are otherwise highly discriminative in isolated settings, leading to considerable fluctuations and diminished utility.

In our analysis, we identified two key challenges when working with aggregated measurements:

First, many features lack an additive property analogous to active power. That is, it is generally not valid to compute a feature f at an event time t_{event} as $f(t_{\text{event}}) = f(w_{\text{post}}) - f(w_{\text{pre}})$, where w_{pre} and w_{post} are windows before and after the event, respectively. This issue is particularly acute for transient features, which cannot be computed in isolation from other ongoing device activity. Using $f(t_{\text{event}}) = f(w_{\text{transient}})$ alone fails to capture the actual contribution of a single device's state change to the aggregate power signal. We outline a set of feature-specific adaptations in Appendix B to address this issue. Since all features are to some degree dependent on the current of the system, which holds the additive property, the idea behind these adaptations is to adjust the feature calculation in a way that integrates the additivity of current.

Second, even features theoretically assumed to be additive (e.g., active and apparent power) exhibit systematic offsets when computed from aggregated measurements. Using the FIRED dataset [19] — which provides both high-frequency isolated and aggregated data — we computed features across both modes at identical event timestamps. As illustrated in Figure 2, we found noticeable discrepancies: aggregated features tended to yield different values after removing noise than their isolated counterparts, e.g., in the case of FIRED's fridge especially detrimental in the areas above $300W$.

While we did not find a closed-form correction for these offsets, two important conclusions emerged from large-scale analysis:

- Features that are discriminative in isolated contexts generally remain so under aggregation, albeit with shifted distributions. Hence, general feature selection might be unaffected.

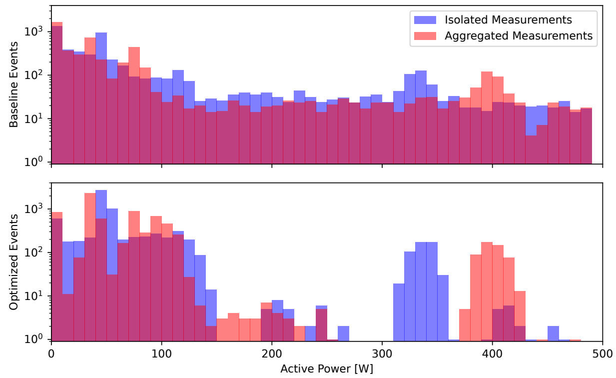

Figure 2: The power distribution of FIRED's fridge, calculated on isolated and aggregated measurements for baseline detected events and after timestamp correction.

- To minimize model confusion, it is critical that features used for training be computed directly from aggregated data. Training models on isolated features and deploying them on aggregated signals may lead to systematic misclassifications.

The role of simultaneous device activity in these shifts remains an open question and warrants further investigation.

5.2 Choosing the ROI

The selection of the ROI for calculating pre- and post-event features remains an underexplored area in the NILM literature. While it is tempting to assume that a sufficiently small ROI suffices—especially if the transient extraction is accurate—this assumption fails in practice. Devices can exhibit unstable behavior even after entering their steady-state. Hence, small ROIs risk capturing transient fluctuations or background noise, yielding unreliable estimates. Conversely, overly large ROIs may inadvertently include subsequent events.

To the best of our knowledge, there exists no systematic investigation of optimal ROIs for feature computation. Few works, such as [3], report ROI parameters, and even then without elaborating on how they were chosen.

We thus conducted an exhaustive analysis over twelve days of the FIRED dataset to assess the impact of ROI size on feature stability. Specifically, we computed a number of prominent features [11] over various ROI combinations, ranging from 4 to 24 periods, for both pre-event and (for steady-state features) post-event windows. We then evaluated the variance of each feature per configuration to quantify stability, with results depicted in Table 1.

Our findings suggest that steady-state features generally benefit from larger ROIs, with 20 periods often yielding the most stable results. In contrast, transient features are more robust with smaller pre-event windows (around 10 periods), which better capture background noise without incorporating unrelated fluctuations.

However, optimal parameter settings only tell part of the story. Many features exhibited low variance across ROI configurations, indicating inherent robustness. Notable exceptions included:

- **Harmonic Energy Distribution (HED):** Highly unstable below 10 periods, but stable with mid-range ROIs.

Table 1: Optimal number of periods for feature stability

Feature	Pre-Event	Post-Event
Active Power	20	24
Admittance	22	18
Apparent Power	10	18
Current Over Time	8	-
Form Factor	14	-
Harmonic Energy Distribution	10	-
Max-Min-Ratio	10	-
Mean-Variance-Ratio	8	-
Odd-Even-Ratio	10	-
Peak-Mean-Ratio	12	-
Resistance	12	16
Total Harmonic Distortion	10	-
Tristimulus	8	-

- **Mean Variance Ratio (MVR):** Performed best at lower ROIs (around 6–8 periods), but became unstable as ROI increased.
- **Resistance:** Exhibited high variance across all configurations, likely due to its inverse relation to admittance and sensitivity to minor signal variations.

To balance performance and simplicity, we recommend a default ROI of 10 periods for estimating background noise and pre-event behavior. For steady-state features, a post-event ROI of 20 periods is advisable. This configuration offers robust performance across a wide range of features without requiring feature-specific tuning.

6 Conclusion

This work highlights critical, often overlooked challenges in implementing event-based NILM systems, offering practical solutions and insights that bridge the gap between theoretical models and real-world deployment.

We began by addressing the common issue of event timestamp misalignment in high-frequency aggregated datasets, proposing a lightweight gradient-based refinement method. We then introduced a transient extraction technique that improves the accuracy of downstream feature computation—a step largely neglected in prior work. A combination of both timestamp adjustment and transient extraction offers a straightforward method for computing reliable ground-truth for the implementation and evaluation of NILM solutions. Our subsequent analysis of feature behavior revealed that assumptions like feature additivity often break down in aggregated contexts, leading to systematic discrepancies. We showed that training models on features derived from aggregated data is essential for robust performance and provided adjustments for commonly used features to support this. Finally, we conducted a comprehensive investigation of ROI configurations for feature extraction, yielding actionable guidelines that improve stability without requiring feature-specific tuning, facilitating future optimization of NILM systems. Together, these insights aim to reduce barriers to effective NILM research and development, enabling more reproducible, comparable and deployment-ready solutions.

References

[1] Kyle D Anderson, Mario E Berges, Adrian Ocneanu, Diego Benitez, and Jose MF Moura. 2012. Event Detection for Non Intrusive Load Monitoring. In *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 3312–3317.

[2] Nipun Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. 2014. NILMTK: An Open Source Toolkit for Non-intrusive Load Monitoring. In *Proceedings of the 5th international conference on Future energy systems*. 265–276.

[3] Justus Breyer, Muhammad Hamad Alizai, Sandeep Samant, and Klaus Wehrle. 2024. Advanced Filtering of Unknown Devices in Event-Based NILM. *ACM SIGENERGY Energy Informatics Review* 4, 5 (2024), 8–16.

[4] Justus Breyer, Sparsh Jauhari, René Glebke, Muhammad Hamad Alizai, Markus Stroot, and Klaus Wehrle. 2024. Investigating Domain Bias in NILM. In *Proceedings of the 11th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 333–336.

[5] K Carrie Armel, Abhay Gupta, Gireesh Shrimali, and Adrian Albert. 2013. Is Disaggregation the Holy Grail of Energy Efficiency? The Case of Electricity. *Energy Policy* 52 (2013).

[6] Agnim I Cole and Alexander Albicki. 1998. Data Extraction for Effective Non-intrusive Identification of Residential Power Loads. In *IMTC/98 Conference Proceedings. IEEE Instrumentation and Measurement Technology Conference. Where Instrumentation is Going (Cat. No. 98CH36222)*, Vol. 2. IEEE, 812–815.

[7] Suryalok Dash and NC Sahoo. 2022. Electric Energy Disaggregation via Non-intrusive Load Monitoring: A State-of-the-Art Systematic Review. *Electric Power Systems Research* 213 (2022), 108673.

[8] Karen Ehrhardt-Martinez, Kat A Donnelly, Skip Laitner, et al. 2010. Advanced Metering Initiatives and Residential Feedback Programs: A Meta-Review for Household Electricity-Saving Opportunities. American Council for an Energy-Efficient Economy Washington, DC.

[9] George William Hart. 1992. Nonintrusive Appliance Load Monitoring. *Proc. IEEE* 80, 12 (1992).

[10] Matthias Kahl, Thomas Kriechbaumer, Anwar Ul Haq, and Hans-Arno Jacobsen. 2017. Appliance Classification Across Multiple High Frequency Energy Datasets. In *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 147–152.

[11] Matthias Kahl, Anwar Ul Haq, Thomas Kriechbaumer, and Hans-Arno Jacobsen. 2017. A Comprehensive Feature Study for Appliance Recognition on High Frequency Energy Data. In *Proceedings of the Eighth International Conference on Future Energy Systems*. 121–131.

[12] Jack Kelly and William Knottenbelt. 2015. The UK-DALE Dataset, Domestic Appliance-Level Electricity Demand and Whole-House Demand from Five UK Homes. *Scientific Data* 2, 1 (2015).

[13] J Zico Kolter and Matthew J Johnson. 2011. REDD: A Public Data Set for Energy Disaggregation Research. In *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, Vol. 25. Citeseer, 59–62.

[14] Stephen Makonin. 2016. Investigating the Switch Continuity Principle Assumed in Non-Intrusive Load Monitoring (NILM). In *2016 IEEE Canadian conference on electrical and computer engineering (CCECE)*. IEEE, 1–4.

[15] Lucas Pereira, Donovan Costa, and Miguel Ribeiro. 2022. A Residential Labeled Dataset for Smart Meter Data Analytics. *Scientific Data* 9, 1 (2022).

[16] Nasrin Sadeghianpourhamami, Joeri Ruyssinck, Dirk Deschrijver, Tom Dhaene, and Chris Develder. 2017. Comprehensive Feature Selection for Appliance Classification in NILM. *Energy and Buildings* 151 (2017), 98–106.

[17] Abraham Savitzky and Marcel JE Golay. 1964. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical chemistry* 36, 8 (1964), 1627–1639.

[18] Benjamin Völker, Marc Pfeifer, Philipp M Scholl, and Bernd Becker. 2020. Annoticity: A Smart Annotation Tool and Data Browser for Electricity Datasets. In *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring*. 1–5.

[19] Benjamin Völker, Marc Pfeifer, P M Scholl, and B Becker. 2020. FIRED: A Fully-labeled hIgh-fRequency Electricity Disaggregation Dataset. In *ACM BuildSys*.

A Comparative Study

In order to quantify the impact that adjusting the timestamp and extracting an event's transient has on the overall performance of an event-based NILM system, we performed a comparative evaluation. For this purpose, common simple classifiers (kNN, SVM and Random Forest) were chosen, as they have been used in event-based NILM before (e.g., [4, 10]). As datasets, UK-DALE [12], FIRED [19] and SustDataED2 [15] were used. The events of the datasets were taken either from metadata (in the case of FIRED and SustDataED2) or generated via the NilmTK [2] (in the case of UK-DALE). Subsequently, to avoid heavy class imbalance, all appliances with a number of events below 60% of the respective median number of events per dataset were eliminated. The remaining imbalance was addressed using the `RandomUnderSampler` of Python's `imblearn` library. Subsequently, the resampled events were split into training and test data, using an 80-20 split.

Table 2: Random Forest performance using provided and optimized events

Dataset	Baseline Events		Optimized Events	
	Accuracy	F1-Score	Accuracy	F1-Score
FIRED	0.47	0.54	0.79	0.81
UK-DALE	0.07	0.09	0.26	0.30
SustDataED2	0.18	0.2	0.49	0.56

As baseline, features were calculated using the provided timestamps for each event, using $0.5s$ before the timestamp as reference value for pre-event behavior, and $0.5s$ starting at the timestamp as transient. We performed forward feature selection, as is commonly used [11], using our adapted feature calculation (see Appendix B) based on the feature overview of Kahl et al. [11]. The baseline was compared to the exact same set of events but using our timestamp adjustment and transient extraction method as post-processing of the event timestamps.

The results of the best performing classifier, which was the Random Forest, are summarized in Table 2. As clearly visible, both standard metrics of classification (accuracy and macro F1-score) increase by an absolute of 20% in almost all cases. Although the transient extraction might still be improved, as discussed at the end of Section 4, the impact of applying our optimization strategies is already non-negligible. More refined models might already provide better baseline results, however, these results already hint at there being improvement potential regardless of the model type and dataset, albeit not always with a similar margin.

B Feature Calculation on Aggregated Data

To exploit the additive property of more than simply the power features (**Active Power P**, **Reactive Power Q**, **Apparent Power S**) in aggregated data, we developed formulae based on the feature overview of Kahl et al. [11].

The **Admittance Y** holds the same additive property as power, i.e., $Y_{event} = Y_{post} - Y_{pre}$. Since the **Resistance R** does not hold the additive property, as it is the multiplicative inverse of admittance, we can instead leverage the calculation of Y to arrive at R:

$$R_{event} = \frac{1}{Y_{event}} = \frac{1}{Y_{post} - Y_{pre}} = \frac{1}{\frac{RMS(I_{post})}{RMS(U_{post})} - \frac{RMS(I_{pre})}{RMS(U_{pre})}}$$

Many transient features make use of the harmonic characteristics of the current. For this purpose, the fast Fourier transform (FFT) is applied to the post-event window and the transient phase, using a Hamming window to minimize spectral leakage. Since the current intrinsically holds the additive property, we can leverage this and the basic idea of FFT (that the signal is composed of a superposition of periodic components) to extract the influence of the transient itself: For all multiples f^i of the fundamental frequency f^{base}, we subtract the amplitudes of the pre-event ROI from the amplitudes during the transient:

$$f^i_{event} = f^i_{transient} - f^i_{post}$$

Afterwards, features such as **Harmonics Energy Distribution**, **Tristimulus**, **Total Harmonic Distortion (THD)** or **Odd Even Harmonics Ratio** can be calculated using the adjusted frequency amplitudes according to established procedures [11].

Some features are based on waveform analysis, i.e., analyzing the shape of the transient. To remove components of the transient data that have been caused by background noise, we approximate the signal in the pre-event ROI to later on subtract it from the transient signal. For this purpose, firstly the pre-event ROI is divided into segments of length of one period. The segments are averaged and the result is smoothed using a low-pass filter. This representative is then concatenated to itself to fit the duration of the transient and shifted in phase to align with the transient, to finally be subtracted from the transient, leaving just the impact the switching of device states has caused on the overall signal. Waveform analysis features that are based on this artificially created signal of the transient include **Peak Mean Ratio**, **Mean Variance Ratio** and **Form Factor** [11].

The last set of features captures the behavior of the RMS current during the transient. **Current Over Time (COT)** [11] is adjusted by subtracting the mean RMS of the pre-event window from each of the first 25 RMS values of the transient:

$$COT = [RMS(I_1) - RMS(I_{pre}), ..., RMS(I_{25}) - RMS(I_{pre})]$$

where I_i denotes the current during the i^{th} period starting at the transient. **Periods to Steady State** simply is the length of the transient, converted to the number of periods, and needs no adjustment regardless of isolated or aggregated appliance data.