Hawk: An Efficient NALM System for Accurate Low-Power Appliance Recognition

Zijian Wang[1,2], Xingzhou Zhang[1,2], Yifan Wang[1,2], Xiaohui Peng[1,2], Zhiwei Xu[1,2,3]
[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Great Bay University, Dongguan, China
{wangzijian19z,zhangxingzhou,wangyifan2014,pengxiaohui,zxu}@ict.ac.cn

Abstract

Non-intrusive Appliance Load Monitoring (NALM) aims to recognize individual appliance usage from the main meter without indoor sensors. However, existing systems struggle to balance dataset construction efficiency and event/state recognition accuracy, especially for low-power appliance recognition. This paper introduces Hawk, an efficient and accurate NALM system that operates in two stages: dataset construction and event recognition. In the data construction stage, we efficiently collect a balanced and diverse dataset, HawkDATA, based on balanced Gray code and enable automatic data annotations via a sampling synchronization strategy called shared perceptible time. During the event recognition stage, our algorithm pipeline integrates steady-state differential pre-processing and voting-based post-processing for accurate event recognition from the aggregate current. Experimental results show that HawkDATA takes only 1/71.5 of the collection time to collect 6.34x more appliance state combinations than the baseline. In HawkDATA and a widely used dataset, Hawk achieves an average F1 score of 93.94% for state recognition and 97.07% for event recognition, which is a 47.98% and 11.57% increase over SOTA algorithms. Furthermore, selected appliance subsets and the model trained from HawkDATA are deployed in two real-world scenarios with many unknown background appliances. The average F1 scores of event recognition are 96.02% and 94.76%. Hawk's source code and HawkDATA are accessible at https://github.com/WZiJ/SenSys24-Hawk.

CCS Concepts

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Hardware** → *Sensor applications and deployments*; • **Computing methodologies** → *Supervised learning by classification*; *Feature selection.*

Keywords

NALM, human activity recognition, sampling synchronization, dataset construction, feature extraction

ACM Reference Format:
Zijian Wang, Xingzhou Zhang, Yifan Wang, Xiaohui Peng, Zhiwei Xu. 2024. Hawk: An Efficient NALM System for Accurate Low-Power Appliance Recognition. In *The 22nd ACM Conference on Embedded Networked Sensor Systems (SENSYS '24), November 4–7, 2024, Hangzhou, China.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3666025.3699359

SENSYS '24, November 4–7, 2024, Hangzhou, China

ACM ISBN 979-8-4007-0697-4/24/11
https://doi.org/10.1145/3666025.3699359

1 INTRODUCTION

Indoor human activity recognition (HAR) is an attractive issue. Appliance usage, as an essential part of indoor activities, reflects people's daily routines and impacts household energy consumption. Accurate detection of appliance usage enables various applications, such as detecting hazardous appliances (e.g. chargers for flammable lithium batteries), elderly care[39], building energy management[28, 48], and optimizing operating plans for residential[43] and industrial users[30]. Non-intrusive appliance load monitoring (NALM)[16] detects the usage of individual appliances by monitoring the main circuit without indoor sensors. Due to its ease of deployment, privacy protection, and low hardware cost, NALM is very attractive[32, 38] and is considered vital for smart meters[14, 42].

Despite its potential, existing NALM systems struggle to balance algorithm accuracy with the efficiency of dataset construction. NALM is a typical single-channel blind source separation (SCBSS) problem[52], which is highly underdetermined[19]. Current algorithms mainly focus on recognizing a few noticeable appliances[4, 19, 52] and are hard to identify low-power ones from the total current. Moreover, long-term data collection is necessary to capture diverse real-world samples for model training[24, 45], and manual inspection is essential for accurate data annotation[12], which becomes a burden for supervised NALM algorithms[55]. Therefore, building a NALM system that accurately identifies all household appliances, especially low-power ones, faces three challenges:

Inefficiency of dataset collection. Like most HAR scenarios, NALM data collection suffers from *data scarcity*[45] and *imbalance*[29, 53]. Most appliances, such as 86% in the BLUED dataset[12], are switched less than ten or even zero times a day. In addition, the distribution of the usage of different appliances is highly imbalanced. The distribution of usage varies over time and in different households, making it difficult for the training sets collected to represent various real-world conditions[45]. Therefore, traditional NALM data collection involves long-term data acquisition in different households[24], which is time consuming and error prone[8]. Imbalanced appliance usage leads to redundant samples[40, 47, 53] and insufficient usage data for some appliances[12, 50], which burdens feature extraction and model training[40].

Inefficiency of dataset annotation. Traditional NALM systems collect the aggregated current and individual appliance status for data labeling during the dataset construction phase. Sampling synchronization in such typical distributed acquisition scenarios

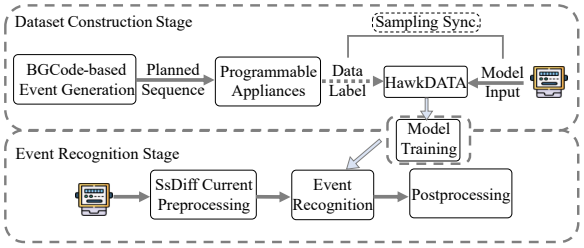

Figure 1: The methodological pipeline of Hawk.

affects the accuracy of the labeling[12, 35]. Traditional synchronization methods[27, 35] are two-step processes: synchronizing the sensor nodes' clocks to a standard clock, followed by responding and timestamping the sensor data. This *two-step sync* introduces high upper-bound cumulative errors. Thus, manual inspection[12] is essential for accurate event timestamps, which is time-consuming and error-prone, especially for low-power appliances.

Inaccuracy of the low-power appliances recognition: NALM is a typical SCBSS task that separates individual source signals from a single aggregated signal (typically total current or total power). The identification of target appliances is affected by other appliance currents and electrical fluctuations. Inspired by the Signal-to-Interference-plus-Noise Ratio (SINR) for assessing the separability of SCBSS[54], lower-power appliances are generally more difficult to recognize with the same level of interference. Thus, many SOTA algorithms[19, 52] only identify a few apparent appliances and have a high computational overhead[46], which become key obstacles for *using one sensor for all household appliances*.

To summarize the above challenges, we pose the following question: *can we **efficiently** train a model capable of **accurately** recognizing (F1 score > 90%) **low-power** (< 50w) appliances from aggregate current in **real world** with low computational overhead?* To this end, we propose Hawk, an efficient NALM system for accurate low-power appliance recognition. As illustrated in Figure 1, Hawk's methodological pipeline is divided into two stages: dataset construction and event recognition. First, Hawk generates diverse and balanced event sequences using *grouped randomized balanced Gray code (BGCode)*. Programmable appliances execute schedules while their current is recorded to label the aggregated current. Then, high-frequency sampled data from distributed sensors are synchronized using a *shared perceptible time* strategy, which eliminates the intermediate clock in the two-stage sync process. During the event recognition stage, a *steady-state differential (SsDiff) current* pre-processing method enhances the SINR of appliances events, especially low-power ones, from the aggregated current.The post-processing stage reports events after a popularity-based voting method to improve the algorithm's robustness to false predictions.

The application scenario of the Hawk system is also illustrated in Figure 1. Model developers efficiently construct datasets with a set of appliances and train the corresponding models to recognize these appliances of the same models. Users may be interested in any subset of the collected appliances and download the corresponding models to recognize the usage of the appliances against unknown background appliances. Identifying appliances by appliance model imposes stronger constraints than recognition by type, but it is more reasonable. This is because the electric signature of an appliance is primarily determined by its front-end circuitry[18] and power consumption, rather than by a strong mapping to the type of appliance. Thus, two appliances of the same type may have different electrical signatures, and two appliances with identical electrical signatures may be of entirely different types. Consequently, previous studies that validate the recognition accuracy in the same appliance type in different households[24] always result in a low recognition accuracy[4, 52], which complicates the evaluation of the effectiveness of the algorithm[19].

To our knowledge, Hawk is the first NALM system to accurately recognize low-power appliances from aggregated current using the model trained in the laboratory and inference in the wild. Full-stack optimizations of the Hawk system contribute to the final recognition accuracy. The Hawk sampling synchronization strategy (like the coordinated claws) reduces the maximum error bound to a sampling interval, 1/17.2 of 802.11 TSF. It enables automated data annotation to reduce labor costs and potential deviation. HawkDATA (like huge and balanced wings) achieves 6.34 times more appliance state combinations in just 1/71.5 of collection time compared to baseline[33]. It also improves the category balance ratio of the event and state combination by 151.4 and 7188.5 times. Hawk's algorithm (like the sharp eyes) achieves an average F1 score of 92.71% for event recognition and 93.93% for state identification in HawkDATA. The F1 score of state identification increases by 47.98% compared to [19, 46]. With the same model trained on HawkDATA, Hawk achieves an average F1 score of 96.03% with five appliances in a residential area and 94.76% with six appliances, including five low-power ones, in an official area, both with many unknown backgrounds appliances. In the BLUED dataset[12], Hawk's average F1 score for event classification reaches 97.07%, improved by 11.57% to SOTA[50]. In addition, Hawk's average cost of single-point, high-frequency (16Khz, 24-bit) stabilized sampling is reduced to $4.12, about 1/3 of Hz-level sampling board[9].

The rest of the paper is organized as follows: Section 2 introduces related work. Section 3 and Section 4 detail dataset construction scheme and algorithm pipeline. Section 5 describes hardware prototype, evaluation settings and in-the-wild deployment. Section 6 presents full-stack evaluation results. Section 7 discuss limitation and failure scenario of Hawk. Section 8 concludes the paper.

2 RELATED WORK

NALM Sampling Synchronization. The NALM datasets capture the current or power consumption in the main circuit as input to the model and collect the current or power consumption of the individual appliances for data annotation. In such a typical distributed data sampling scenario, accurate clock synchronization is the foundation of accurate automated annotation. [5] and [44] rely on hardware voltage zero crossing detectors to synchronize sampling and clocks. However, such hardware-based methods are vulnerable to voltage fluctuation due to hardware error[24] or some appliances events (e.g., some low-end fans). Most works adopt network-based synchronization methods. BLOND [27] employs NTP with one deviation tested as 6.8 ms, while LIT2020 [35] uses a wireless clock distributor based on a 433MHz RF transmitter with a precision of less than 5 ms. BLUED[12] uses a collision-free wireless

protocol but still needs visual inspections to realign the annotation of events. Other fields include [6] achieving 1us-accuracy PTP-like protocol in WiFi networks via TSF. However, the performance of such wireless methods varies with the network jitters[41].

NALM Dataset Construction. Like many HAR scenarios, NALM suffers from data scarcity[8]. Thus, traditional NALM dataset construction extend data collection periods and scenarios to capture diverse, authentic electricity usage behaviors. REDD[26] and UK-DALE[24] collect data from numerous residential houses over long periods. BLOND[27] collects a high-frequency dataset in office settings. However, such long-period data collection is time-consuming and error-prone[8]. What is worse, the scarce and imbalanced usage of appliances always results in an imbalanced dataset: redundant samples for certain state combinations[40, 47, 53], or absence of several appliances' activity[12, 55]. Some datasets[21, 21] choose controlled laboratory conditions to automate the data collection process, which consists of many appliances. However, they lack modeling for balanced state combinations and only capture current from individual appliances, missing aggregate current in NALM settings. Moreover, it is challenging to generate reliable synthetic data[35, 45] since the appliance current is an analog variable with multiple influences and complex interference between appliances.

NALM State/Event Identification. Existing NALM systems try to identify the state or event of appliances from aggregated current or power. State identification algorithms recognize appliance statuses, basic operational states[19], and power consumption[52]. Event recognition algorithms detect and classify state transitions[17, 32, 48]. Previous studies have used features at various frequencies for NALM systems. [17] pioneers the use of **low-frequency** features through an unsupervised learning pipeline. HMMs and their variants are prevalent [25, 37] and are used in industrial cases [30]. Deep learning based on low-frequency features also demonstrates promise[4, 11, 19, 23, 51, 52]. To address the diversity in dataset formats and algorithm design, NILMTK[2] provides a unified dataset parsing interface and integrates several classic low-frequency NALM algorithms for algorithm comparison. [22] validates NALM different algorithms in industrial settings. While low-frequency features require less computation and economic infrastructure, these approaches may struggle to detect low-power appliances, distinguish similar-power ones, and perform poorly in energy consumption trace disaggregation[3]. Some works[15, 32] adopt **ultra-high-frequency** noise sampling (over 1 MHz) to identify electrical events and recognize devices equipped with SMPS. However, noise is attenuated along the cable[32, 48], limiting the distance between the sensor and appliances. Additionally, the hardware infrastructure required for ultra-high-frequency sampling is much more expensive, and a considerable computational overhead exists. Recent work [46, 48, 50] utilize **high-frequency** (1 kHz to 1 MHz) currents with DNNs and achieve promising results. [14] utilizes EMI to recognize server-level power consumption, even distinguishing between different servers of the same model.

3 Dataset construction stage

The NALM dataset consists of two stages: dataset collection and data annotation. The collection process can be simplified as collecting aggregated and individual signals while a set of appliances A executes an events sequence S. Hawk makes the process efficient by automating the execution of balanced and diverse sequences using programmable appliances. Additionally, Hawk synchronizes the distributed sampling by shared perceptible time, enabling automated data annotation and significantly reducing labor costs.

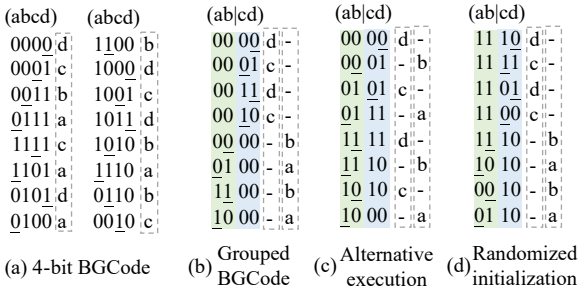

Figure 2: Four-bit balanced Gray code and our proposed variant for four appliances (a/b/c/d).

3.1 Grouped Randomized Balanced Gray Code

Hawk aims to efficiently collect a NALM dataset that satisfies the balance and diversity demands of two primary tasks: event and state recognition. This goal comes from Hawk's application scenarios, where users are interested in recognizing any subset of A with arbitrary background appliances. Therefore, there should be no assumed background currents and priority among appliances.

However, in traditional NALM dataset construction[24, 26], appliance sets A_{real} are chosen from a limited number of volunteer households, and event sequences S_{real} are executed naturally by the household residents. The S_{real} is naturally temporally sparse and biased according to the number of residents, their appliance usage habits, and exogenous conditions. This leads to long-term data collection periods for diverse but imbalanced samples, requiring additional software preprocessing over TB-level volumn of raw data, which burdens algorithm users. Thus, some works[21, 34, 35] collect datasets in a controlled laboratory to bridge appliance event execution sequence with demands of model training. However, they lack an abstraction of appliance event execution to satisfy the diverse and balanced demands of event/state-based algorithm model training, and most only collect individual currents[21, 34].

Therefore, we model appliance OFF-ON states as 0-1 and propose an event generation strategy based on a variant of balance Gray code (BGCode). As shown on the (a) of Figure 2, the BGCode has the following properties that make it ideal for generating event sequence that guides the execution of appliance events.

- Balance: BGCode ensures a nearly balanced flip count for each bit, balancing switch counts among appliances.
- One-bit clipping: BGCode differs only one bit among adjacent states, aligning with independent appliance switches.
- No duplicate travel: BGCode travels the entire state space with no duplicate state, minimizing switching operation.
- Scalability: BGCode can be extended to present any even number of appliances' OFF-ON states[36].

Traversing the entire state space of 2^n combinations becomes extremely time-consuming when the number of appliances becomes large. Additionally, each state combination must stay for sufficient time intervals to collect adequate samples. Thus, we introduce grouping and randomization strategies in event generation to reduce the size of event sequences while maintaining a diverse and balanced property. **Grouping** aims to reduce the scale of execution of balanced Gray code sequences. Given that we have n (4 in (a) of Figure 2) appliances divided evenly into m groups (2 in (b) of Figure 2), each group is selected to execute complete balanced Gray code sequences internally. Then, the number of events required is reduced from 2^n to $m * 2^{n/m}$. Such a sequence still meets the requirement of balance, but limits the range of traversal. Therefore, the grouped strategy will be executed in multiple rounds to achieve a more diverse combination of states. In each round, one or more groups of appliances will be activated and events within the groups will be alternated as (c) of Figure 2. To reduce the probability of state combination collisions in multiple generation rounds, we introduce **Randomization** in terms of grouping, execution sequences and initial state within groups, as shown in (d) of Figure 2.

In addition, this strategy comes with certain constraints. For example, the number of groups activated per round is limited to ensure safe automated execution of electricity usage without supervision. Additionally, to maintain the balance of the overall data set, the activation frequency of each group must be maintained throughout the process. Finally, in the training set generation phase, Algorithm 1 is executed for 30 rounds, yielding a sequence of 2880 events and 2584 unique states. As for the validation set, we performed the strategy for 18 rounds, resulting in 2112 events and 1962 unique states. This data generation strategy ensures a low overlap between the training and test sets, less than 17%, requiring the model to learn the underlying data patterns. The generated sequences are stored in a file and executed by the programmable appliances described in the following subsection.

3.2 Programmable Appliance

In the actual dataset collection process, it is necessary to implement programmable appliances that execute generated event sequences automatically without ambiguity and maintain the consistency of the appliance's state in cyberspace and physical space. Hence, the following two requirements must be satisfied: accurate event execution and timely feedback of the appliance state.

The programmable appliance is a logical entity implemented with an event executor, an individual current sensor, and a controller. The event executors consist of esp32-controlled servos and relays to simulate human operations, and ESP32s provide interfaces as Restful servers. Each executor is assigned a static address and pre-calibrated to ensure **accurate event execution**. The controller detects the state from the cycle-level root mean squared (RMS) current collected by the individual current sensor. However, some appliances exhibit continual variations in the RMS current. Hence, we denoised the RMS current waveform by averaging values in a window of length l, specific for each appliance, excluding the n largest and the m smallest values. And we switch the appliances multiple times in advance to determine these parameters corresponding to specific appliances. We keep l as short as possible (averaged 60 ms) to enable **timely feedback of the appliance status**. When the appliance is turned on or off due to a malfunction or timed switch, the abnormal event will be reported and automatically corrected.

3.3 Shared Perceptible Time

Clock synchronization accuracy is crucial for NALM data annotation, especially for datasets with larger granularity annotations, such as the 20-ms cycle level for HawkDATA and the 6s level for UK-DALE[24]. Given a synchronization strategy with a maximum deviation of E, events occurring within E around the label bounds may cause a label-level bias, which misguides model training and data analysis. Traditional NALM data acquisition suffer from the cumulative error of the *two-step sync*. Clock synchronization protocols in cyberspace are always influenced by network jitters. Additionally, the delay in CPU/MCU data response can be unstable. For example, using FPGA for high-frequency ADC data buffering may cause fluctuations in event timestamping[27].

NALM dataset construction focuses more on internal synchronization than on external synchronization. Hence, we propose a sampling synchronization strategy called shared perceptible time (SPT) to synchronize sampling directly. The core of the strategy is that time can be represented not only as counters driven by crystal oscillators in cyberspace but also by continuously varying physical signals. In the context of NALM dataset construction, most appliances are connected in parallel and share the same voltage within a household. We can utilize multichannel simultaneous ADCs to capture current alongside the voltage and timestamp each sample with (cycle number, phase value) from the voltage value. Cycle ID is confirmed by the TSF value of the first sample in the voltage cycle. The phase value is assigned a continuous distance to the first zero-crossing point, which is quantified by the sampling interval. We assume that the sampling interval remains stable over a short period, such as a voltage cycle of 20 ms, so the interval can be used to quantify the phase position. Ultimately, we construct the global current sequence by constructing the global voltage sequence.

4 Event Recognition Stage

NALM algorithms suffer from low signal-to-interference-plus-noise ratios (SINR), where signals and fluctuations from other appliances influence targeted appliance recognition. Hawk's enhances the target events' SINR by steady-state differential current pre-process and improves the algorithm's robustness by a popularity-based voting post-processing method. Thus, some basic classifiers can achieve high accuracy and robustness in recognition while maintaining low computational overhead.

4.1 Steady-state Differential Current

Unlike previous work that directly applies traditional signal processing methods[50] or relies on model learning capabilities[48], we aim to enhance the SINR of target appliances through pre-processing by leveraging the stability and periodicity of appliance currents. We propose a feature extraction called steady-state differential (SsDiff) current, which has two key advantages.

4.1.1 Enhanced SINR from Differential Current. To calculate SINR, we express the signal power as the power consumption of the appliance. Given the power of the ith appliance at time t is $P_{i,t}$. The

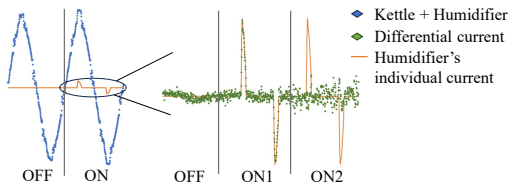

Figure 3: Comparison between aggregated current and differential current. The right subplots show the humidifier's current and the differential current calculated from three differential pairs: both before the event, at the two ends of the event (ON1), and both after the event(ON2).

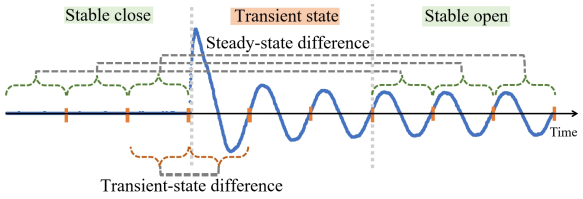

Figure 4: The incandescent lamp's individual current changes around turning on. When background currents are stable, the steady-state differential operation on total current is supposed to get changes in the incandescent lamp's current.

background noise is $N_{B,t}$. The SINR for the kth targeted appliance from raw data at time t is expressed as:

$$SINR_{k,t} = \frac{P_{k,t}}{\sum_{i!=k}^{n} P_{i,t} + N_{B,t}} \tag{1}$$

Therefore, smaller power appliances are generally more difficult to recognize directly from the aggregated current when running concurrently with larger ones. However, the differential current offers a different perspective. Differential current's effectiveness relies on Kirchhoff's current law, simplified as $I_{sum} = I_{target} + I_{background}$. When background appliance currents remain relatively stable over short durations, turning on or off the target appliances should result in a proportionate change at the main meter. Given that a switch event of kth appliance occurs within the range of (t-d, t), one of $P_{k,t-d}$ and $P_{k,t}$ corresponds to the power of the ON stage, $P_{k,ON}$, while the other corresponds to $P_{k,OFF}$. And $P_{k,OFF}$ is approximately zero for most appliances. Therefore, the SINR of the kth appliance event from the differential current waveform with differential pair (t-d, t) can be expressed as follows:

$$\begin{aligned} \mathrm{SINR}_{k,t} &= \frac{|P_{k,t} - P_{k,t-d}|}{\sum_{i\neq k}^{n} N'_{i,t} + N'_{B,t}} = \frac{P_{k,ON} - P_{k,OFF}}{\sum_{i\neq k}^{n} N'_{i,t} + N'_{B,t}} \\ &\approx \frac{P_{k,ON}}{\sum_{i\neq k}^{n} N'_{i,t} + N'_{B,t}} \end{aligned} \tag{2}$$

$N'_{i,t}$ represents the power of differential noise from the ith appliance. Since background appliances always remain relatively stable for a short period, $N'_{i,t}$ is significantly smaller compared to the original power. For example, assume an indoor environment with a 40±1W humidifier and a 1500±20W electric kettle working together, with a background noise of 10±10W. The humidifier's maximum SINR is 41/1480 if we identify directly from the total current. However, after applying differential current, the minimum SINR for the humidifier's switching event is 39/(40+20) = 39/60.

To demonstrate differential current's benefits more vividly, we present an example in Figure 3. The left subplot shows the total current for the humidifier's on and off state when the background appliance is a kettle. Due to the high-power background kettle, the total current waveform indistinguishable changes before and after the humidifier switch. On the other hand, when background appliances are relatively stable (often the case), and the differential current pairs are distributed around the switching events, the differential current will capture the waveform of the individual current change from the aggregated current. As seen in Figure 3, the waveform corresponding to ON1 matches the current values of the target appliance and makes it more distinguishable. Conversely, only electrical noise remains if the differential pair is in the same appliance state combination (such as OFF and ON2 in Figure 3).

4.1.2 Enhanced Robustness from Windowed Voting. As shown in Figure 3, differential total current can extract the individual current of switching appliances while background appliances are running steadily. Moreover, most appliances undergo a transient state between stable on and off states[34], as shown in Figure 4. Such transient current waveforms are more random and easily confused with waveforms from other background appliances. Therefore, we insert a fixed interval between differential current pairs, allowing us to bypass the transient state and directly capture the differential current between stable on and off states.

Unlike previous work that expects models to learn differential information from raw inputs[48] or extract steady-state differential currents for classification after detecting events[10], we employ an one-step sliding window approach for streaming steady-state differentials, as shown in Figure 4. Additionally, we leverage another observed characteristic of the differential currents to enhance our algorithm's robustness further. Given a differential interval of D and an appliance transient length of t, the steady-state differential current can ideally obtain $(D - t)$ cycles of differential current signature corresponding to the event (as illustrated by the three pairs of differential currents in Figure 4). In other words, the classifier will output the event $(D - t)$ times. Thus, we propose a voting-based post-processing method that reports events when the report number of the classifier greater or equal to a threshold T. Such a process can tolerate $(D-t-T)$ false negatives and $T-1$ false positives caused by background interference, which will effectively enhance the robustness of the model predictions.

4.2 Considerations of Model Design

Problem Definition. Different from appliance state recognition, we define appliance event recognition in as a time-series multi-class classification problem instead of a multi-label classification problem. In SustDataED2, out of 12,252 events involving 18 appliances, only two human-related events occurring within a specific second. Out of the 799 events for the eight appliances in the BLUED dataset, also only two events occur in the same 1-second interval. The statistics indicate a low collision probability for human-related events and set reference for maximum differential gap length.

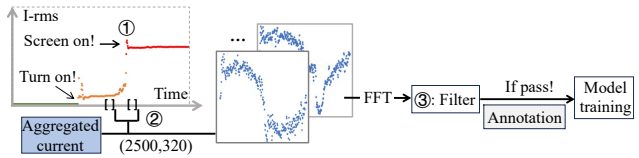

Figure 5: Training data preparation pipeline. The smart screen is a multi-component appliance and undergoes two events after switch on: *system boot* and *screen on*, the second one is more noticeable since the screen costs more energy.

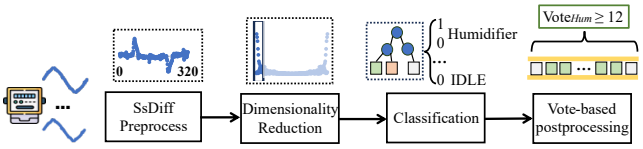

Figure 6: The algorithm pipeline of the event recognition. The pipeline's input is read from main meter. When vote number is over threshold, the event is reported.

Determination of crucial parameters. Differential interval and voting threshold are critical parameters in the algorithm pipeline. The differential interval for calculating SsDiff current should be long enough to jump over transient features but reasonable to avoid simultaneous events and rising risk of background noise. Moreover, specific thresholds for each appliances directly affect recognition accuracy, which is determined during model training by maximizing the F1-score of event predictions on the training set.

Classifier Selection. We conducted preliminary experiments with various deep neural networks for time series classification[20] and found that CNN and CNN-LSTM achieve higher accuracy and better computational efficiency. Different appliances show varying power ranges and harmonic content in their current waveforms. Hence, we combined FFT with XGBoost[7], which empirically performs better on unsmooth target functions[13]. We ultimately selected these three classifiers for comparison in our evaluation.

4.3 Algorithm Pipelines

The differential currents of events of low-power appliances are vulnerable to fluctuations caused by background appliances, so the pipeline design for model training and inference must account for noise interference with recognition.

4.3.1 Training Data Preparation. To reduce the effect of contaminated data on training and achieve data augmentation. We propose a three-step pipeline (depicted in Figure 5) to automatically extract and filter a sufficient number of SsDiff currents with corresponding labels for model training.

(1) **Obvious Event Localization**: Unlike strictly accurate dataset annotation, we choose more noticeable current change points during inevitable stages as their on/off events to predict (such as Smart Screen's *screen on* event in Figure 5). Therefore, we raise the RMS-current thresholds to automatically locate more apparent event positions to prepare training data.
(2) **Data Augmentation**: Steady-state differential currents are obtained by performing pairwise differential operations on the current from 50 stable ON and 50 stable OFF states on either side of appliance switches, generating 2500 pairs of differential current differences for each appliance event.
(3) **Sample Filtering**: The enhanced samples are filtered by comparing the fundamental frequency harmonics with the individual appliances', reducing contaminated samples.

4.3.2 Inference pipeline. Hawk's inference pipeline follows a basic preprocess-classifier-postprocess structure. During inference, Hawk's algorithm pipeline reads the aggregated current, organized by voltage cycles, from the smart meter to detect and classify appliance events. Our algorithm pipeline based on FFT-XGBoost is shown in Figure 6, while the pipelines based on CNN and CNN-LSTM use the raw current waveform as input.

(1) **Differential Current Calculation**: We compute Steady-state Differential currents by direct cycle-level current subtraction from a 30-cycle circular buffer.
(2) **Dimensional Reduction**: We take the first ten harmonics, including the real part, imaginary part, and magnitude, of differential current as model inputs after FFT transformation.
(3) **Classification**: Reduced features are fed into a big XGBoost model to identify events. The model outputs 37 states, including OFF-ON states of 18 appliances and an IDLE state.
(4) **Post-processing**: A post-processing strategy involves voting within an 30-cycle window. The most likely event, surpassing the voting threshold (e.g., 12 for the humidifier turn on event), is reported eventually.

We can use the *second step* to reduce the dimensionality of features while preserving sufficient information of the raw waveform because most of the current waveform energy of appliances is more concentrated in the low-frequency region. Such an operation can also eliminate the effect of high-frequency noise. Because Hawk performs continuous inference, the *classifier* includes an IDLE state, indicating no events detected. The results of the streaming classification are buffered in a queue of length the same as the differential gap. Each time, we count the occurrences of each category in the queue, and when the count of a specific category exceeds its corresponding threshold, the classifier outputs that category. This postprocessing efficiently reduces false predictions caused by noise.

5 Prototype And Validation Settings

In this section, we first present the design of our sampling hardware prototype and then introduce different settings: laboratory, in-the-wild study in the residual and office area with a brief description of the collected dataset.

5.1 Hardware Prototype

Hawk hardware architecture, depicted in Figure 7, integrates three primary elements: the data acquisition board for high-frequency data capture, AC-(to)-AC sensors for current and voltage sensing, and a data recording infrastructure. This fundamental hardware architecture design is shared between the development and deployment stages. We developed two versions of the prototype corresponding to two stages, as shown in Figure 8.

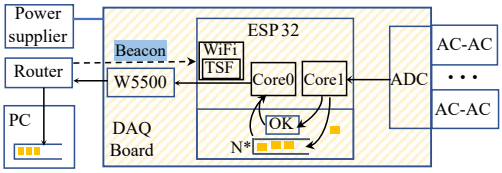

Figure 7: Hardware architecture of Hawk data acquisition. Voltage and current are collected by data acquisition boards through AC-AC sensors and transmitted via Ethernet to a data logging PC.

The data acquisition board consists of three components: a 24-bit ADS131M08 ADC for 16KHz simultaneous 8-channel sampling, an ESP32 MCU offering dual-core processing with integrated WiFi for TSF counter queries, and a W5500 Ethernet controller to transmit data reliably during dataset construction stage. To ensure stable and continuous sampling at 16KHz, as depicted in Figure 7, we assign the task of responding to ADC interrupts to Core1, which is free of default interrupts. Core1 reads and packages data by voltage cycles (the yellow squares in Figure 7). Conversely, Core0 is responsible for querying packet status and transmitting the cycle's data from the shared memory to the logging PC.

The AC-AC sensor unit comprises voltage and current sensors. Current sampling has three typical types of sensors[56]: shunt resistors, Hall effect sensors, and current transformers. We opted for contactless current transformers for their circuit non-intrusiveness, ease of deployment, and electrical safety.

The collected data are streamed through a wireless router. The wireless router facilitates the Ethernet connection between the data logging PC and the sampling boards via the MQTT protocol. It also synchronizes the TSF in each MCU for the sampling synchronization strategy. We utilize a desktop as the data logger in the laboratory environment. In real-household conditions, we employ an embedded computer as the collector to reduce disruption to the household's current and reduce costs. Additionally, the collected data is stored on an external hard drive.

The current transformer selected for the prototype at the main meter is rated at 4000:1, with a shunt resistor value of 10 ohms. The ADC reference voltage is 1.25V, featuring a resolution of 24 bits and a sampling frequency of 16 kHz. Ignoring sampling deviation, the theoretical resolution of the sampling is 59.6 μA. However, the sampling accuracy is constrained by the effective number of bits of high-frequency sampling and is also affected by the nonlinear conversion of the current transformer. Therefore, in the evaluation section, we assess the linearity of the current conversion.

The hardware design is designed to maximize component utilization, with multiple AC-AC sensors sharing a single data acquisition board and taking full advantage of dual-core performance. As a result, the average sampling cost per current channel based on the data acquisition board is reduced to $4.12, about one-third of the cost of the Hz-level sampling board[9]. This is remarkable, as the high cost of large-scale training data collection is a key obstacle for supervised method of NALM[55]. Even in the deployment phase, this optimized design can be taken in environments with multiple smart meters, such as green communities.

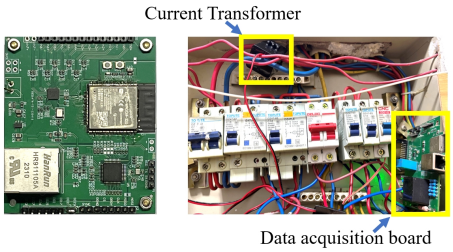

Figure 8: Two version of hardware prototypes. The left subplot shows the laboratory version powered by an external shared power supply. The deployed version in real environments will extend an AC-DC power module to avoid the need for battery maintenance.

5.2 Laboratory Setting and HawkDATA

We build a data collection laboratory to deploy 22 common appliances for data collection. Among these, 18 are programmable for future algorithms to recognize, as listed in Table 1. The other 4 appliances, such as a wireless router and refrigerator, are constantly on, free of human activity, and used as background appliances. We select diverse appliance representations, including those with similar functions but with different electronic circuits (e.g., LED, incandescent, and fluorescent bulbs for lighting). The electrical characteristics of these appliances span a wide range, from a 5W camera to a 2160W air heater and coverage of most front-end circuits[18]. Besides diversity, recognition challenges are also considered, and we also choose appliances with the same working principle but varying power ratings and appliances with close power rates. This is also an advantage of HawkDATA over others from real households with biased collections of appliances.

HawkDATA is collected in the laboratory environment, with programmable appliances executing the event schedules mentioned in Subsection 3.1. Each appliance state combination is planned to last for 20 seconds. Ultimately, the effective collection duration of our dataset is 32.2 hours, comprising a total of 5,796,650 voltage cycles for a 50Hz AC circuit. HawkDATA includes both raw and annotated versions stored in NPZ file format. The raw data version records the total current, voltage, and individual current of 18 appliances, all sampled at 16kHz. The raw-version file size is 127.8GB. The annotated version organizes and labels the data according to voltage cycles. For a 50Hz AC circuit, each cycle contains 320 points.

5.3 In-the-wild Settings

The field experiment aims to validate the Hawk system and ensure that it aligns with the application scenarios. The Hawk system, with a single sensor for inference (shown on the left of Figure 9), was tested in residential and office spaces, and target appliances subsets from HawkDATA were repurchased and deployed without affecting the original appliances. Including data collection laboratory, three places were selected from three different regions, with a total distance exceeding 2000 km between the points. All three regions operate under a 220V, 50Hz AC standard, but there are slight differences due to different electrical infrastructures and varying electricity consumption trends.

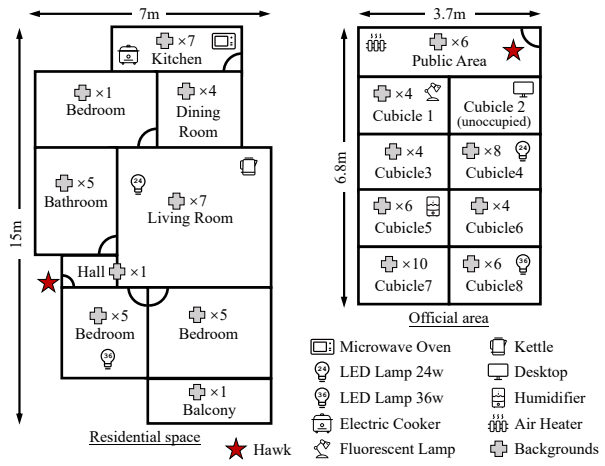

Figure 9: Deploy details of two in-the-wild settings.

For the residential environment, we selected a 103-m^2 three-bedroom apartment with three inhabitants and 36 unknown background appliances, of which more than 40% were lighting devices, totaling more than 42 operating states. We deployed five representative appliances from the HawkDATA set: two LED lights (24W and 36W), an electric cooker, an electric kettle, and a microwave oven, for seven days. Observations revealed the use of 35 of 36 background appliances during this period. For the office environment, we chose a 25-m^2 graduate student office with seven occupants and 48 appliances, totaling more than 51 operating states. Since most office appliances are less than 150W, we selected low-power appliances from HawkDATA, including a desktop, two 24W and 36W LED lights, a humidifier, a fluorescent lamp, and a high-power air heater as a background appliance when few appliances were used. Observations showed that 44 of 48 background appliances were used during this period.

The deployment of the Hawk model is in a manner of application scenarios mentioned in the Introduction section, which are deployed without modification. The inference program is implemented in C and is attached to a physical core of embedded computers (Jetson Orin NX 16GB) to enable real-time inference. It uses the MQTT protocol to subscribe to continuous data generated by a single sensor node at the main meter, as shown in Figure 9. Targeted appliances are manually switched on and off, with both switch actions and model recognition results recorded, including false positive samples when no switch action occurs. The switching frequency is adjusted according to the number of occupants, set to 15, 30, or 60-minute intervals. In addition, the deployed appliances do not interfere with the operation of original appliances, ensuring realistic and diverse background current.

6 EVALUATION

6.1 Data Acquisition Quality Evaluation

As a typical distributed data collection, the quality of NALM data is evaluated from two dimensions: sample synchronization accuracy between sensor nodes and single-point current conversion linearity.

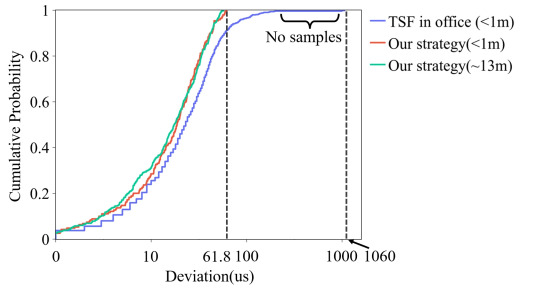

Figure 10: CDF of synchronization error of our strategy and TSF. Our CDF is smoother due to continuous timestamp.

6.1.1 *Sampling synchronization evaluation.*

Baseline. The baseline selected for comparison is the Time Synchronization Function (TSF) of the 802.11 protocol, which can achieve microsecond synchronization accuracy[6, 31]. Additionally, ESP32 provides a TSF interface, making it a suitable baseline for our low-cost distributed solution.

Evaluation method. We assess synchronization error by comparing the timestamps of the same current surge point recorded by two nodes. Incandescent lamps experience a current surge when switched on. Such transient current surge is simultaneously detectable by the main meter and the submeter attached to the incandescent lamp. We switch on and off the incandescent bulb using a relay to collect the synchronization deviation. Two meters for testing are placed less than 1 meter from the accessed wireless router to test the best performance of TSF synchronization. In addition, a distance of around 13m is set to evaluate the impact of electrical circuit length on SPT. The wireless router used for the experiment is an Asus AC66U-B1. The deviation of the SPT synchronization strategy is a continuous phase value, while the TSF-based synchronization strategy has a resolution of 1 microsecond.

Result Analysis. The result in Figure 10 shows that Hawk's synchronization strategy based on SPT can achieve an average synchronization accuracy of 20.60us, improved by 59.2% compared to TSF, averaging 32.8us. Our maximum synchronization error is 61.80us, about 1/17.2 of TSF. We can see a gap in the error distribution of TSF; 99.65% of the synchronization accuracy is within 220us, while the other 0.35% are distributed from 980us to 1060us. This gap is due to the 1-ms WiFi packet jitter[41]. Our experiments show that the jitter ratio of beacon packets is affected by wireless channel interference and the distance between the AP and the STA. In our testing scenario, several wireless APs in the office area are the sources of interference. On the other hand, our proposed SPT strategy is more robust. When the two sensor nodes are 13 meters apart, the overall accuracy improves. This is due to introduced random quantification errors, by equating the switch events of a continuous position of the events to a sampling point. The more comprehensive analysis of our synchronization error will be presented in future work.

6.1.2 *Current conversion linearity evaluation.*

Evaluation method. One of the challenges for sampling accuracy evaluation experiments is that we lack an ideal measurement for high-frequency current sampling instruments as a standard. Such standards invariably have their deviation ranges and come with

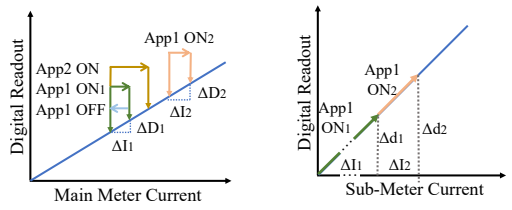

Figure 11: Current conversion differential linearity of different events with different background current. The ratios are calculated from diverse situations.

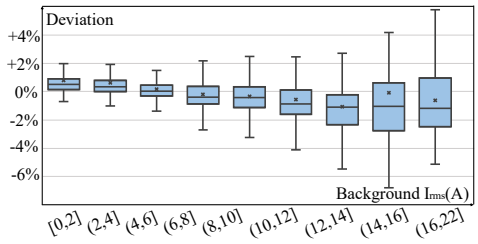

Figure 12: Differential linearity of current conversion in HawkDATA.

prohibitive costs. Inspired by a commonly used evaluation metric of ADC/DAC, differential linearity, we designed an assessment method. Differential linearity of ADC refers to the stable relationship between the digital output changes and the analog input variations at different starting points. As shown in Figure 11, we evaluated the differential linearity of HawkDATA by comparing the stability of the ratio between the total meter reading differences and current differences before and after switching events for different appliances under varying background currents. The current difference is presented as sub-meter reading differences. Since most appliances operate within a more stable and lower current range when compared with aggregated current, we assume that the current differences maintain a consistent ratio with the reading differences. Finally, we organize different appliance switches' average normalized linearity rates within the same range of background currents into a box and show their changes along the change of background currents in Figure 12.

Ensuring the differential linearity of our sampled values also enables our proposed steady-state differential current-based feature extraction. Though not guaranteeing that the acquired data represents exact values, differential linearity ensures a stable proportional relationship. Such deviations are often deemed inconsequential after typical pre-processing steps like normalization.

Result. According to Figure 12, both mean and median values deviate by no more than 2% from the ideal unity, substantiating the stability of our data conversion process. Since this metric is statistics of raw HawkDATA, the final results ensure that within the HawkDATA, the average reading deviation of the same current variation, influenced by background current, does not exceed 2%. The data visualized in Figure 12 reveals two phenomena with increased background current. The first observation is a slight reduction in the average normalized linearity rate which may related to property of current transformer. The second observation is the widening spread of the linearity rate distribution as the background current intensifies. The broadening is likely a consequence of the increased electrical noise and interference associated with a more significant number of active background appliances, affecting the calculation of differential linearity rates.

6.2 Dataset Construction Evaluation

Although the sequence generated from group randomized balanced Gray code guarantees balance, the practical execution of programmable appliances may deviate from this due to factors like the timed switching or operational failures; for instance, the microwave oven in Table 1 exhibits more frequent switching.

Table 1: The number of events and on-state cycles for the appliance in the HawkDATA.

Appliances	Event	On-state	Appliances	Event	On-state
Monitor	302	1,774,054	SmartScreen	294	1,780,251
Humidifier	278	1,694,105	FluorescentLamp	292	1,798,803
LEDLamp24w	278	1,666,026	Television	290	1,707,688
IncandescentBulb	296	1,856,095	Washer	288	1,693,044
ElectricCooker	296	1,656,548	MicrowaveOven	1310	1,628,551
InductionCooker	288	1,705,172	AirHeater	282	1,712,366
LEDLamp36w	280	1,718,433	ElectricKettle	286	1,687,541
Stirrer	290	1,651,833	PhoneCharger	294	1,793,826
Desktop	286	1,827,268	SweepingRobot	292	1,695,928

Baseline. Our baseline is SustDataED2[33], a high-frequency NALM dataset collected from a real household. SustDataED2 comprises the same number of appliances as the programmable appliances in Hawk, making it a reasonable baseline as representative of the traditional NALM dataset construction schema.

Metrics. The metric to measure dataset balance is balance ratio (BR), the inverse of a commonly used imbalance ratio (IR)[49], to avoid the minimal value of zero[12]. The category BR is defined as,

$$BR = \frac{N_{min}}{N_{max}} \tag{3}$$

where N_{min} is the sample size of the minoriest category and N_{max} is the sample size of the majoriest category. What's more, there are different balance requirements for state and event recognition algorithms. The balance requirement of the event recognition algorithm entails a uniform distribution of switch events across different appliances, maximizing the category BR. In contrast, state recognition algorithms require balance in two aspects[40]. First, the number of ON and OFF state samples for each type of appliance should be as close as possible. Second, the number of ON states between different appliances should be balanced.

As for the metrics of dataset diversity, we use the number of state combinations to represent the diversity of states and qualitatively represent event diversity by a heatmap of the appliance on/off distributions with different levels of background RMS current.

Result Analysis. Table 1 records the distribution of our event and state combinations. Most appliances have a consistent number of collected events and ON states, except that the microwave oven's timed shutdown feature caused many unplanned switch events during data collection. Ultimately, HawkDATA's category BR of

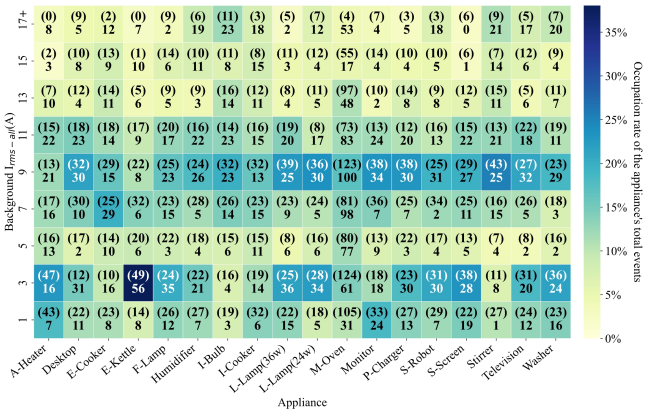

Figure 13: Each cell contains two numbers: the top number denotes the count of events occurring in the training set, and the bottom number corresponds to the testing set.

the event is 0.212, significantly higher than that of SustDataED2, which is 0.0014. And the category BR of ON states across appliances is 0.877, exceeding SustDataED2's 1.22*10^{-4}. We use the average ON-OFF BR as the evaluation metric of the overall balance between ON and OFF states. HawkDATA's average ON-OFF BR is 0.424, surpassing SustDataED2's 0.338.

As for event diversity evaluation, all appliances exhibit switch events across all current ranges in Figure 13. As for state diversity, HawkDATA records 4,558 unique state combinations across data collection of 32.2h. As a comparison, SustDataED2, a dataset collected from real households, acquired only about 718 unique states in 2,304 hours. HawkDATA records a unique state every 25.43 seconds and a switch event every 18.91 seconds during data collection. The number of unique state combinations collected per unit of time, which we refer to as the diversity density or information density of the dataset, is much higher than others, reflecting the efficiency of Hawk dataset construction.

6.3 Algorithm Pipeline Evaluation

6.3.1 Overall Setup. The evaluation of our algorithm pipeline consists of two parts: first, a set of ablation experiments targeting the three key designs of the Hawk system; second, a comparison of event classification accuracy on the BLUED dataset, which includes multiple multi-state appliances under different AC settings.

We design three ablation experiments focusing on the data balance, classifier performance, and steady-state differential current of the Hawk system. The first experiment compares event recognition accuracy across different classifiers trained on balanced and imbalanced HawkDATA. The second experiment adjusts the differential interval to evaluate its impact on recognition accuracy. The last experiment evaluates the effect of differential current processing on state recognition accuracy and compares it with open-source state recognition algorithms[19, 46].

Baselines. MSDC[19] uses CNN to recognize appliance OFF-ON states from low-frequency power signals, followed by state corrections with conditional random fields (CRF). *CALM*[46] is based on high-frequency signals and takes BiLSTM to infer appliance

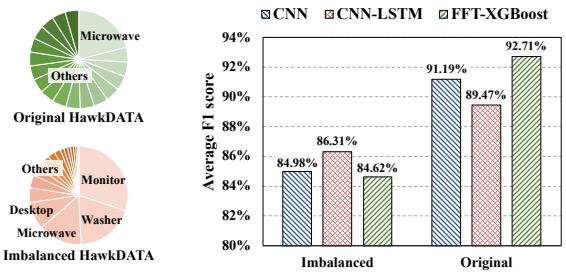

Figure 14: Comparison of different classifier trained on different dataset.

states from the raw high-frequency current. *MTS-Shapelet*[50] employs a multi-time-scale shapelet to process high-frequency current waveforms. It provides detailed accuracy results for appliance classification events, facilitating the comparison of different appliances.

Datasets. HawkDATA, as previously discussed in Subsection 5.2, is collected in a laboratory environment with 220V, 50Hz AC power. The average power is calculated from the total current and voltage to generate a labeled dataset for MSDC based on low-frequency signals. The *imbalanced HawkDATA* is resampled based on the relative proportions of appliance usage in actual households ([12, 33], as shown in the left part of Figure 14), keeping the total event count constant. We add differential current during periods without appliance events to increase the sample size for underrepresented types, based on Kirchhoff's current law. *BLUED*[12] is a high-frequency (12KHz) NALM dataset collected in a residential environment with 240/120V, 60Hz AC power over seven days. Some of collected appliances (such as fridge) are multi-state appliances.

Metrics. We use the F1 score to evaluate recognition accuracy and average F1 score to evaluate overall performance on balanced HawkDATA. For the imbalanced BLUED dataset, we also employed the weighted average F1 score[50]. However, we still prefer the average F1 score. This is because the weighted F1 score disproportionately emphasizes appliances like fridge, which have frequent automatic switches but are less related to human behavior.

6.3.2 Impact of classifier and dataset balance. Taking advantage of the enhanced SINR and robustness of the steady-state differential current algorithm pipeline, these basic network structures achieve high-accuracy event recognition, yielding similar final results. As shown in Figure 14, we choose FFT-XGBoost as our classifier, as it exhibited the highest recognition accuracy on the balanced dataset while maintaining minimal algorithmic overhead, enabling real-time recognition on embedded CPU platforms.

Regarding the impact of dataset balance, all classifiers demonstrated decreased recognition accuracy on the imbalanced dataset, even though most HawkDATA appliances are commonly used and the sample sizes are not significantly reduced. Additionally, different classifiers exhibited varying tolerances to dataset imbalance, with CNN-LSTM showing the least impact.

6.3.3 Impact of differential distance. As illustrated in Figure 4, the distinction between steady and transient state differential current lies in the differential distance, which is consistent with the length of the voting window. A smaller differential distance results in a

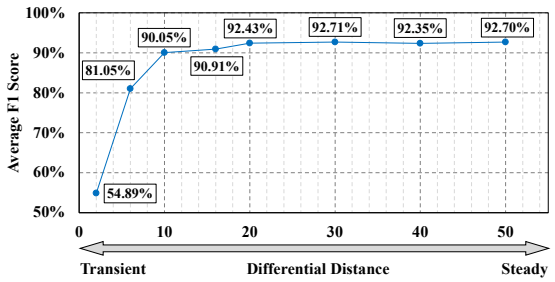

Figure 15: Impact of differential distance.

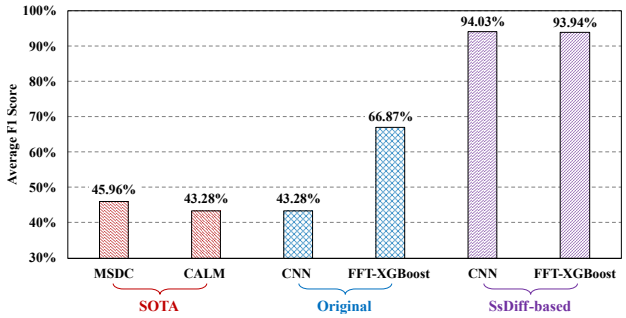

Figure 16: Accuracy comparison of state identification on HawkDATA. Original classifiers take raw data as input.

higher proportion of transient features within the voting window that cover events, while a larger differential distance increases the proportion of steady-state features. Moreover, the voting length affects the model's tolerance to false predictions; a larger differential interval increases the model's tolerance for errors. However, larger intervals also raise the likelihood of random noise interference and the occurrence of multiple events. Consequently, average accuracy exhibits fluctuations in Figure 15 when the interval is between 30 and 50. Ultimately, we select 30 to minimize the probability of multiple events occurring in the difference interval.

6.3.4 State Identification Accuracy on HawkDATA. Figure 16 evaluates the impact of integrating steady-state differential current into the algorithm pipeline on state recognition accuracy and compares it with SOTA performance.

The results in Figure 16 show that the same XGBoost classifier, and CNNs with identical parameter sizes, significantly improve state recognition accuracy when integrating differential current processing. Additionally, compared to the best SOTA performance, our simple classifier based on steady-state differential processing improved state recognition accuracy by 47.98% to 48.07%.

We observe that the previous two SOTA works performed poorly on HawkDATA. The evaluation results of the low-frequency signal-based MSDC align with earlier observations[21, 34], behaving poorly in power trace decomposition. In contrast, the high-frequency signal-based CALM was evaluated on a single-appliance dataset.

6.3.5 Event Classification Accuracy on BLUED Dataset. To validate the effectiveness of the Hawk algorithm pipeline for multi-state appliances and different AC standards, we trained the Hawk recognition model on the BLUED dataset for event classification. The event recognition of Hawk for multi-state appliances is similar to OFF-ON appliances, requiring pre-labeling all possible events for target appliances. The final recognition results are summarized in Table 2, compared with SOTA published results.

According to Table 2, Hawk event classification average F1 score outperforms MTS-Shapelet by 11.57%, and the weighted average F1 score improved by 4.13%. Moreover, higher recognition accuracy is achieved across all appliances, with three appliances achieving F1 scores of 100%. MTS-Shapelet and Hawk achieve 100% classification accuracy on the Washroom light appliance, which only has six samples in total, indicating a potential issue of insufficient validation and highlighting the importance of data balance.

Table 2: Comparison of event classification accuracy on BLUED. Hawk is based on FFT-XGBoost.

Appliances	Power (w)	Number	MTShapelet[50]	Hawk
Hair Dryer	1600	8	80.00%	**100.00%**
Kitchen Aid Chopper	1500	16	80.00%	**93.75%**
Air Compressor	1130	20	67.00%	**100.00%**
Bedroom Lights	190	19	86.00%	**90.91%**
Fridge	120	616	97.00%	**99.67%**
Washroom light	110	6	**100.00%**	**100.00%**
Bathroom upstairs lights	65	98	85.00%	**98.46%**
Backyard lights	60	16	89.00%	**93.75%**
Average	-	-	85.50%	**97.07%**
Weighted average	-	-	95.00%	**99.13%**

6.4 In-The-Wild Evaluation

6.4.1 Overall Setup. In the field experiments, our objective is to validate the effectiveness of the Hawk system in real world scenarios, which includes computational overhead, real-time performance, and recognition accuracy in various unknown background appliance settings. The Hawk model was deployed without modification in two real scenarios that align with the application scenario.

Computational Platforms. To validate Hawk run-time performance, we performed real-time streaming inference performance assessments on three CPU-based platforms, as listed in Table 3.

Metrics. We use the F1 score to evaluate the accuracy of Hawk system recognition, with the average F1 score used to assess overall performance. For runtime performance, we measure the Hawk's average streaming inference latency and the actual memory footprint (Residential Set Size, RSS) of the entire program.

6.4.2 Run-time Performance Evaluation. This section aims to evaluate the performance of streaming inference, which processes individual samples sequentially. The real-time requirement for streaming inference is that the average inference latency must be less than the average generation delay (20 ms in our AC settings). This is a stringent performance criterion and are proved valuable in real application scenarios[1]. Performance on the three platforms is summarized in Table 3.

Table 3 shows that our model satisfies the real-time requirement of immediate inference upon data generation on all three platforms. For future application developers, we recommend using buffered

Table 3: Average streaming inference latency and memory footprint of Hawk on three distinct platforms.

Platform	CPU	Latency (ms)	Mem (MB)
Desktop	Intel i7-10700@2.9GHz	3.31	42.50
Jetson Orin NX	Cortex-A78AE@2GHz	4.07	40.19
Raspberry PI 4B	Cortex-A72@1.8GHz	6.59	32.60
Real-time Requirement	Single core	< 20	-

data for batch processing, parallel programming, and filter samples based on differential power for performance optimization.

6.4.3 Event Recognition Accuracy Evaluation. The event recognition accuracy in two real environments is summarized in Table 4.

Table 4: Event recognition accuracy in two real settings.

Appliances	Power (w)	HawkDATA	Residence	Office
AirHeater	2160	**100.00%**	-	98.00%
ElectricKettle	1500	**100.00%**	96.29%	-
MicrowaveOven	1315	**100.00%**	98.82%	-
ElectricCooker	500	**99.60%**	98.71%	-
Desktop	40	**99.18%**	-	93.19%
Humidifier	40	**100.00%**	-	85.41%
LEDLamp36W	36	**99.58%**	94.88%	98.81%
LEDLamp24W	24	96.52%	91.42%	**96.83%**
FluorescentLamp	19	96.20%	-	**96.30%**
Average	-	**99.01%**	96.02%	94.76%

As shown in Table 4, all appliances except for the humidifier experienced less than a 10% decline in F1 score compared with accuracy on HawkDATA. The recognition accuracy of some low-power appliances is even improved in office environments. This experiment meets our initial goals and demonstrates the effectiveness of the Hawk system. Notably, even with many continuously changing background appliances, such as laptops and desktops, low-power appliances are still accurately recognized. However, the accuracy drop for the humidifier is significant. We analyze the reasons for this performance decline in the next section.

7 Discussion And Future work

In this section, we discuss some limitations of the Hawk system and analysis of recognition failures.

Limitations of Binary Modeling: The balanced Gray code-based strategy models appliance OFF-ON states as binary bit value, which does not accommodate multi-state appliances. OFF-ON appliances can be considered a particular case of multi-state appliances. The performance of Hawk's algorithm pipeline on the BLUED dataset, which includes multi-state appliances, proves that the processes of recognizing OFF-ON events and multi-state events are interconnected. However, this characteristic limits the diversity of the current HawkDATA appliance collection. A new balanced abstraction to model multi-state appliances is needed.

Limitations of Application Scenarios: Recognizing appliances based on their model rather than their type is more practical but more restrictive. The current Hawk model still requires data collection from specific appliance sets, which remains a significant overhead. Future work could reduce this overhead through multimodal unsupervised approaches [55] and few-shot learning, and our proposed algorithm pipeline design will benefit these efforts.

Limitations of Appliance Selection: Despite careful selection, the number of appliances in the HawkDATA collection is still limited compared to the variety of background appliances. For instance, the humidifier with the highest-power LC circuit suffers numerous false positive results in field experiments. Expanding the variety of appliance types and usage periods will be a future effort.

Failures Due to Voltage Variations: Voltage fluctuations in residential areas contribute to most false-negative results of electric kettle recognition. In real-world environments, the voltage of the appliances fluctuates over time and shows different trends in different areas. Moreover, different types of appliances exhibit different current responses to the changes: resistive appliances show a positive correlation between current and voltage, while capacitive appliances (e.g., LED light) with AC-DC front ends show a negative correlation to keep output power stable. Future work will further analyze the impact of voltage on recognition results and incorporate voltage waveforms for more precise appliance identification.

Falures Due to Complex Appliances: Hawk achieves its goals and performs well with low-power appliance events recognition in selected two real scenarios. However, the algorithm pipeline still struggles with recognizing ultra-low power (<10w), continuously varying power, and among identical appliances. Future work will focus on reducing background noise interference and distinguishing between different appliances of the same model[14].

8 CONCLUSION

Non-intrusive appliance load monitoring (NALM) faces the challenges of high dataset construction overhead and low appliance identification accuracy. This paper presents Hawk, an efficient NALM system for accurate low-power appliance recognition. To improve the efficiency of dataset construction, we propose an automatic dataset construction scheme based on grouped randomized balanced Gray codes and a sampling synchronization strategy based on shared perceptible time to enable automatic data annotation. Taking advantage of the enhanced SINR and robustness provided by our steady-state differential current-based algorithm pipeline, some basic classifiers show high accuracy and low computational overhead in real-time recognition across different tasks, even with low-power appliances. To our knowledge, Hawk is the first NALM system to accurately and in real time identify low-power appliance usages in real-world scenarios, with an average cost of stable high-frequency current sampling of only $4.12. A balanced and diverse NALM dataset, HawkDATA, has been published.

9 Acknowledgement

We are grateful to anonymous reviewers for their constructive suggestions. We sincerely thank Xingwu Liu, Shun Lu, Zheming Yang, and Xiaoyu Wang for their valuable suggestions on this paper. We also appreciate the Nanjing Institute of InfoSuperBahn for providing the data collection environment. This research was partly supported by the National Natural Science Foundation of China under Grant Nos. 62402475 and 62072434, and the Innovation Funding of ICT, CAS under Grant No. E361050.

References

[1] 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), 213–234.

[2] Nipun Batra, Rithwik Kukunuri, Ayush Pandey, Raktim Malakar, Rajat Kumar, Odysseas Krystalakos, Mingjun Zhong, Paulo Meira, and Oliver Parson. 2019. Towards reproducible state-of-the-art energy disaggregation. In *Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation*. 193–202.

[3] Nipun Batra, Amarjeet Singh, and Kamin Whitehouse. 2015. If you measure it, can you improve it? exploring the value of energy disaggregation. In *Proceedings of the 2nd ACM international conference on embedded systems for energy-efficient built environments*. 191–200.

[4] Gissella Bejarano, David DeFazio, and Arti Ramesh. 2019. Deep latent generative models for energy disaggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 850–857.

[5] Bradford Campbell, Ye-sheng Kuo, and Prabal Dutta. 2018. From Energy Audits to Monitoring Megawatt Loads: A Flexible and Deployable Power Metering System. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 189–200.

[6] Paizhuo Chen and Zhice Yang. 2021. Understanding Precision Time Protocol in Today's Wi-Fi Networks: A Measurement Study. In *USENIX Annual Technical Conference*. 597–610.

[7] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SigKDD International Conference on Knowledge Discovery and Data Mining*. 785–794.

[8] Wenqiang Chen, Shupei Lin, Elizabeth Thompson, and John Stankovic. 2021. Sensecollect: We need efficient ways to collect on-body sensor-based human activity data! *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (2021), 1–27.

[9] Samuel DeBruin, Branden Ghena, Ye-Sheng Kuo, and Prabal Dutta. 2015. Powerblade: A low-profile, true-power, plug-through energy meter. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 17–29.

[10] Shaokang Ding and Hongwei Zhu. 2020. Non-intrusive load monitoring based on deep neural network and differential current. In *IOP Conference Series: Earth and Environmental Science*, Vol. 474. IOP Publishing, 052103.

[11] Anthony Faustine, Lucas Pereira, Hafsa Bousbiat, and Shridhar Kulkarni. 2020. UNet-NILM: A deep neural network for multi-tasks appliances state detection and power estimation in NILM. In *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring*. 84–88.

[12] Adrian Filip et al. 2011. Blued: A fully labeled public dataset for event-based non-intrusive load monitoring research. In *2nd workshop on data mining applications in sustainability (SustKDD)*, Vol. 2012.

[13] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. 2022. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems* 35 (2022), 507–520.

[14] Pranjol Sen Gupta, Zahidur Talukder, Tasnim Azad Abir, Phuc Nguyen, and Mohammad A Islam. 2023. Enabling Low-Cost Server-Level Power Monitoring in Data Centers Using Conducted EMI. In *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 237–250.

[15] Sidhant Gupta, Matthew S Reynolds, and Shwetak N Patel. 2010. ElectriSense: single-point sensing using EMI for electrical event detection and classification in the home. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*. 139–148.

[16] George W Hart. 1985. Prototype nonintrusive appliance load monitor. In *MIT Energy Laboratory Technical Report, and Electric Power Research Institute Technical Report*.

[17] George William Hart. 1992. Nonintrusive appliance load monitoring. *Proc. IEEE* 80, 12 (1992), 1870–1891.

[18] Dawei He, Liang Du, Yi Yang, Ronald Harley, and Thomas Habetler. 2012. Front-end electronic circuit topology analysis for model-driven classification and monitoring of appliance loads in smart buildings. *IEEE Transactions on Smart Grid* 3, 4 (2012), 2286–2293.

[19] Jialing He, Jiamou Liu, Zijian Zhang, Yang Chen, Yiwei Liu, Bakh Khoussainov, and Liehuang Zhu. 2023. MSDC: exploiting multi-state power consumption in non-intrusive load monitoring based on a dual-CNN model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 5078–5086.

[20] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. *Data mining and knowledge discovery* 33, 4 (2019), 917–963.

[21] Matthias Kahl, Anwar Ul Haq, Thomas Kriechbaumer, and Hans-Arno Jacobsen. 2016. Whited-a worldwide household and industry transient energy data set. In *3rd International Workshop on Non-Intrusive Load Monitoring*. 1–4.

[22] Florian Kalinke, Pawel Bielski, Snigdha Singh, Edouard Fouché, and Klemens Böhm. 2021. An evaluation of nilm approaches on industrial energy-consumption data. In *Proceedings of the twelfth ACM international conference on future energy systems*. 239–243.

[23] Jack Kelly and William Knottenbelt. 2015. Neural nilm: Deep neural networks applied to energy disaggregation. In *Proceedings of the 2nd ACM international conference on embedded systems for energy-efficient built environments*. 55–64.

[24] 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), 1–14.

[25] J Zico Kolter and Tommi Jaakkola. 2012. Approximate inference in additive factorial hmms with application to energy disaggregation. In *Artificial intelligence and statistics*. PMLR, 1472–1482.

[26] 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.

[27] Thomas Kriechbaumer and Hans-Arno Jacobsen. 2018. BLOND, a building-level office environment dataset of typical electrical appliances. *Scientific Data* 5, 1 (2018), 1–14.

[28] Wenjin Jason Li, Xiaoqi Tan, and Danny HK Tsang. 2015. Smart home energy management systems based on non-intrusive load monitoring. In *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 885–890.

[29] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. 2024. Open Long-Tailed Recognition in a Dynamic World. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 3 (2024), 1836–1851.

[30] Wenpeng Luan, Fan Yang, Bochao Zhao, and Bo Liu. 2022. Industrial load disaggregation based on hidden Markov models. *Electric Power Systems Research* 210 (2022), 108086.

[31] Aneeq Mahmood, Reinhard Exel, and Thilo Sauter. 2014. Impact of hard-and software timestamping on clock synchronization performance over IEEE 802.11. In *2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014)*. IEEE, 1–8.

[32] Shwetak N. Patel, Thomas Robertson, Julie A. Kientz, Matthew S. Reynolds, and Gregory D. Abowd. 2007. At the flick of a switch: detecting and classifying unique electrical events on the residential power line. In *Proceedings of the 9th International Conference on Ubiquitous Computing*. Springer, 271–288.

[33] Lucas Pereira, Donovan Costa, and Miguel Ribeiro. 2022. A residential labeled dataset for smart meter data analytics. *Scientific Data* 9, 1 (2022), 134.

[34] Thomas Picon, Mohamed Nait Meziane, Philippe Ravier, Guy Lamarque, Clarisse Novello, Jean-Charles Le Bunetel, and Yves Raingeaud. 2016. COOLL: Controlled on/off loads library, a public dataset of high-sampled electrical signals for appliance identification. *arXiv preprint arXiv:1611.05803* (2016).

[35] Douglas Paulo Bertrand Renaux, Fabiana Pottker, Hellen Cristina Ancelmo, André Eugenio Lazzaretti, Carlos Raiumundo Erig Lima, Robson Ribeiro Linhares, Elder Oroski, Lucas da Silva Nolasco, Lucas Tokarski Lima, Bruna Machado Mulinari, et al. 2020. A dataset for non-intrusive load monitoring: Design and implementation. *Energies* 13, 20 (2020), 5371.

[36] John P Robinson and Martin Cohn. 1981. Counting sequences. *IEEE Trans. Comput.* 100, 1 (1981), 17–23.

[37] Kiarash Shaloudegi, András György, Csaba Szepesvári, and Wilsun Xu. 2016. SDP relaxation with randomized rounding for energy disaggregation. *Advances in Neural Information Processing Systems* 29 (2016).

[38] Wei Sun, Tuochao Chen, Jiayi Zheng, Zhenyu Lei, Lucy Wang, Benjamin Steeper, Peng He, Matthew Dressa, Feng Tian, and Cheng Zhang. 2020. Vibrosense: Recognizing home activities by deep learning subtle vibrations on an interior surface of a house from a single point using laser doppler vibrometry. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 3 (2020), 1–28.

[39] Nagender Kumar Suryadevara, Subhas C Mukhopadhyay, Ruili Wang, and RK Rayudu. 2013. Forecasting the behavior of an elderly using wireless sensors data in a smart home. *Engineering Applications of Artificial Intelligence* 26, 10 (2013), 2641–2652.

[40] Adane Nega Tarekegn, Mario Giacobini, and Krzysztof Michalak. 2021. A review of methods for imbalanced multi-label classification. *Pattern Recognition* 118 (2021), 107965.

[41] Minh-Thuyen Thi, Sosthène Guédon, Siwar Ben Hadj Said, Michael Boc, David Miras, Jean-Baptiste Dore, Marc Laugeois, Xavier Popon, and Benoit Miscopein. 2022. IEEE 802.1 TSN time synchronization over Wi-Fi and 5G mobile networks. In *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. IEEE, 1–7.

[42] Naveen Kumar Thokala, Spoorthy Paresh, and M Girish Chandra. 2022. An effective electrical load disaggregation approach for low-sampled smart meter data. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 149–158.

[43] Mark Valovage, Akshay Shekhawat, and Maria Gini. 2018. Model-free iterative temporal appliance discovery for unsupervised electricity disaggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[44] Sreejaya Viswanathan, Rui Tan, and David KY Yau. 2016. Exploiting power grid for accurate and secure clock synchronization in industrial IoT. In *2016 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 146–156.

[45] Tianshi Wang, Jinyang Li, Ruijie Wang, Denizhan Kara, Shengzhong Liu, Davis Wertheimer, Antoni Viros i Martin, Raghu Ganti, Mudhakar Srivatsa, and Tarek

Abdelzaher. 2024. SudokuSens: Enhancing Deep Learning Robustness for IoT Sensing Applications using a Generative Approach. In *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems (SenSys)*. Association for Computing Machinery, New York, NY, USA, 15–27.

[46] XiaoYu Wang, Hao Zhou, Nikolaos M Freris, Wangqiu Zhou, Xing Guo, and Xiang-Yang Li. 2021. CALM: Contactless Accurate Load Monitoring via Modality Distillation. In *2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 1–9.

[47] XiaoYu Wang, Hao Zhou, Nikolaos M Freris, Wangqiu Zhou, Xing Guo, Zhi Liu, Yusheng Ji, and Xiang-Yang Li. 2021. LCL: Light Contactless Low-delay Load Monitoring via Compressive Attentional Multi-label Learning. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*. IEEE, 1–6.

[48] Qian Wu and Fei Wang. 2019. Concatenate convolutional neural networks for non-intrusive load monitoring across complex background. *Energies* 12, 8 (2019), 1572.

[49] Han-Jia Ye, De-Chuan Zhan, and Wei-Lun Chao. 2021. Procrustean Training for Imbalanced Deep Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 92–102.

[50] Heyang Yu, Chongjun Xu, Guangchao Geng, and Quanyuan Jiang. 2024. Multi-Time-Scale Shapelet-Based Feature Extraction for Non-Intrusive Load Monitoring. *IEEE Transactions on Smart Grid* 15, 1 (2024), 1116–1128. https://doi.org/10.1109/TSG.2023.3285117

[51] Zhenrui Yue, Camilo Requena Witzig, Daniel Jorde, and Hans-Arno Jacobsen. 2020. Bert4nilm: A bidirectional transformer model for non-intrusive load monitoring. In *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring*. 89–93.

[52] Chaoyun Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard, and Charles Sutton. 2018. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[53] Jianjun Zhang, Xuanqun Chen, Wing WY Ng, Chun Sing Lai, and Loi Lei Lai. 2019. New appliance detection for nonintrusive load monitoring. *IEEE Transactions on Industrial Informatics* 15, 8 (2019), 4819–4829.

[54] Shujie Zhang, Tianyue Zheng, Hongbo Wang, Zhe Chen, and Jun Luo. 2022. Quantifying the Physical Separability of RF-Based Multi-Person Respiration Monitoring via SINR. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 47–60.

[55] Guanzhou Zhu, Dong Zhao, Kuo Tian, Zhengyuan Zhang, Rui Yuan, and Huadong Ma. 2023. Combining Smart Speaker and Smart Meter to Infer Your Residential Power Usage by Self-supervised Cross-modal Learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 3 (2023), 1–26.

[56] Silvio Ziegler, Robert C Woodward, Herbert Ho-Ching Iu, and Lawrence J Borle. 2009. Current sensing techniques: A review. *IEEE Sensors Journal* 9, 4 (2009), 354–376.