energies

Article

A Framework to Generate and Label Datasets for Non-Intrusive Load Monitoring

Benjamin Völker *, Marc Pfeifer, Philipp M. Scholl and Bernd Becker

Chair for Computer Architecture, University of Freiburg, 79110 Freiburg, Germany;
pfeiferm@informatik.uni-freiburg.de (M.P.); pscholl@informatik.uni-freiburg.de (P.M.S.);
becker@informatik.uni-freiburg.de (B.B.)
* Correspondence: voelkerb@informatik.uni-freiburg.de

Abstract: In order to reduce the electricity consumption in our homes, a first step is to make the user aware of it. Raising such awareness, however, demands to pinpoint users of specific appliances that unnecessarily consume electricity. A retrofittable and scalable way to provide appliance-specific consumption is provided by Non-Intrusive Load Monitoring methods. These methods use a single electricity meter to record the aggregated consumption of all appliances and disaggregate it into the consumption of each individual appliance using advanced algorithms usually utilizing machine-learning approaches. Since these approaches are often supervised, labelled ground-truth data need to be collected in advance. Labeling on-phases of devices is already a tedious process, but, if further information about internal device states is required (e.g., intensity of an HVAC), manual post-processing quickly becomes infeasible. We propose a novel data collection and labeling framework for Non-Intrusive Load Monitoring. The framework is comprised of the hardware and software required to record and (semi-automatically) label the data. The hardware setup includes a smart-meter device to record aggregated consumption data and multiple socket meters to record appliance level data. Labeling is performed in a semi-automatic post-processing step guided by a graphical user interface, which reduced the labeling effort by 72% compared to a manual approach. We evaluated our framework and present the FIRED dataset. The dataset features uninterrupted, time synced aggregated, and individual device voltage and current waveforms with distinct state transition labels for a total of 101 days.

Keywords: data annotation; non-intrusive load monitoring; semi-automatic labeling; smart meter

Citation: Völker, B.; Pfeifer, M.; Scholl, P.M.; Becker, B. A Framework to Generate and Label Datasets for Non-Intrusive Load Monitoring. *Energies* **2021**, *14*, 75. https://dx.doi.org/10.3390/en14010075

Received: 27 November 2020
Accepted: 18 December 2020
Published: 25 December 2020

1. Introduction

The United Nations has outlined 17 Sustainable Development Goals [1] for 2030. Related to the production and consumption of electric energy are three of them: *stop global warming* by *clean energy* in *sustainable cities*.

One important step to achieve these goals is to reduce the electricity consumption in our homes. In the residential domain, energy monitoring and 'eco-feedback' techniques have proven to help by raising the awareness of an unnecessary electricity consumption of a particular device. In addition, these techniques can be combined with demand-side flexibility to schedule their usage, so that mostly renewable energy is used. Ehrhardt-Martinez et al. [2] found that per device consumption feedback can achieve high energy savings when provided frequently. More specifically, according to this meta-study, real-time aggregated electricity consumption feedback can preserve around 8.6 % of electricity on average. If the feedback is provided appliance-wise, they spotted that the savings are up to an average of 13.7 %. These savings are achieved by simply raising the user awareness. The actual savings might even be increased, if the feedback system is combined with a smart home agent. Smart home agents can learn user behavior, adapt knowledge of other agents, and can either directly control smart appliances to save electricity or recommend specific energy saving strategies to the user.

There are mainly two possibilities to obtain the device specific electricity consumption of certain devices in a home. (1) Each device is equipped with a dedicated electricity meter—known as Intrusive Load Monitoring (ILM). (2) A single electricity meter is installed that measures the composite load of all appliances. Specially designed and often individually trained algorithms disaggregate this aggregated load into the load of each individual consumer. This approach is known as Non-Intrusive Load Monitoring (NILM) and is said to be more feasible (compared to ILM) as it only requires a single smart electricity meter.

In many countries, the standard electricity meter (Ferrari meter) has already been exchanged by a smart electricity meter. For instance, the roll-out of smart meters in Germany began with heavy consumers (>6000 kW h per year) by the beginning of 2020 [3]. Smart meters are promoted to bring features like device level electricity feedback to our homes by using NILM.

NILM research already started in 1992 when a first NILM prototype was introduced by G.W. Hart [4,5]. Recent promotion of smart meters, associated research fundings (e.g., SINTEG [6]), and emerging machine learning algorithms accelerated research in this field. Even if the concept is already known for 35 years, Armel et al. [7] stated that "disaggregation may be the lynch-pin to realizing large-scale, cost-effective energy savings in residential and commercial buildings." Over the last three decades, various NILM algorithms have been developed by researchers. These can be roughly categorized into (1) event-based algorithms which relate signal state changes to appliance state changes (such as [4] or [8]) and (2) event-less algorithms which estimate an overall system state using techniques such as Factorial Hidden Markov Models [9,10].

To train, evaluate, and compare these algorithms, public available datasets are used. Even though a lot of datasets have been published such as REDD [9], UK-Dale [11], BLOND [12], and many more (see [10] for a comprehensive overview), they can only hardly be used to compare different disaggregation techniques because of a low sampling frequency which does not allow to test event-based approaches or because of missing or incorrect ground truth information. Besides these, no datasets—except BLUED [13] to some extent—includes labels of internal appliance state changes (e.g., changing the channel of a television). Unfortunately, such information is of particular interest for event-based NILM approaches and electricity-based human activity recognition systems such as [14,15].

Retrospectively generating fine grain labels is not possible for datasets that have already been recorded years ago and hardly possible while generating new datasets. It would require the residents to manually log every action in the home (e.g., every key-press of the television remote) with precise timestamps. In other domains such as activity recognition, the problem of generating ground truth data is typically addressed by recording a video contemporaneous to e.g., accelerometer signals. The labeling step is then performed manually afterwards by going through the video on a frame by frame basis. This technique could hardly be applied to electricity datasets as: (1) electricity datasets typically cover a long time period of several weeks or months, (2) privacy concerns if all rooms or residents are equipped with a camera, and (3) internal or automatic state changes of appliances (like the cooling cycle of the fridge) can not be identified via video.

Therefore, we propose a hardware and software framework to generate and label data for NILM that feature fine-grained labels based on intrusive meters, additional sensors, and a smart labeling tool. The system allows to record time synced data of a home's electrical input (aggregated data) and nearly all individual consumers. Furthermore, smart appliances are incorporated to log their states. For devices that do not expose their states (e.g., old TVs), custom logging devices are used such as infrared sniffers. A post-conducted, semi-automatic algorithm identifies appliance steady states and state changes in the individual appliance data and applies preliminary labels to the data. Therewith, the overall labeling effort is reduced significantly to a human supervision step.

This report summarizes two publications [16,17] and extends them by (1) including more related work, (2) an in depth explanation of the used hardware and software components bundled into the proposed expandable framework, and (3) a deeper evaluation of the FIRED dataset that has also been extended to include more recording days and high quality event labels.

The remainder of the paper is structured as follows: Section 1.1 describes how others have recorded and labeled NILM datasets. In Section 2, we identify remaining challenges to record NILM datasets and describe the hardware and software of our proposed framework. We successfully utilized the framework to generate and label the FIRED [17] dataset which we present in Section 3. Finally, a discussion of the FIRED dataset and our observations with our framework concludes the paper in Sections 4 and 5.

1.1. Related Work

As interest in evaluating and comparing electricity related algorithms has increased over the recent decade, several datasets and the hardware used to record them have been published. Some of these datasets which have been recorded either in residential or industrial environments are briefly discussed in the following.

The Reference Energy Disaggregation Dataset (REDD) was introduced by Kolter et al. in 2011 [9]. The authors used a custom-built meter to record the whole house electricity consumption of six different homes in the US. *NI-9239* (National Instruments) analog to digital converters were used to measure the mains' voltage and *SCT-013* (YHDC) split core current transformers to measure current in a secure and non-intrusive way. The readings were acquired by a recording laptop at 16.5 kHz with an ADC resolution of 24 bit. High frequency mains' data of the complete recording duration is, however, only available as compressed files generated with a custom lossy compression. Socket and sub-circuit-level data are only available as unevenly sampled low frequency data of approximately 1/3 Hz. Furthermore, these data show gaps of several days.

The UK Domestic Appliance-Level Electricity dataset (UK-DALE) introduced by Kelly et al. in 2015 [11] covers the whole house electricity demand of five homes in the UK. In particular, three of the houses (1, 2, and 5) have been recorded at a sampling rate of 16 kHz. The aggregated power consumption was recorded with off-the-shelf USB sound cards with stereo line input. AC-AC transformers were used to scale down the voltage, while split core current transformers were used to measure current. The recording duration of house 1 was up to 1629 days resulting in the longest whole house recording known to us. Appliance level data was sampled using off-the-shelf 433 MHz electricity meter plugs (Eco Manager Transmitter Plugs developed by Current Cost) paired with a custom self-developed base station. Devices directly connected to the mains are metered using current clamp meters (Current Cost transmitter) that are sampled using the same custom base station. The data of these devices were sampled with a low sampling rate of around 1/6 Hz and contains several gaps too.

The Electricity Consumption and Occupancy (ECO) dataset was introduced by Beckel et al. in 2014 [18]. The authors leverage the communication interface of an off-the-shelf smart electricity meter (*E750 from Landis + Gyr*) to read out the aggregated consumption of six homes in Switzerland. The consumption data include different electricity related metrics such as active power, RMS voltage, and current as well as the phase shifts of all three supply legs. Furthermore, 6–10 *Plugwise* smart plugs have been deployed per house to record individual appliance active power measurements of selected appliances at around 1 Hz (the actual sampling rate varied due to a sequential readout, but the data have been resampled to 1 Hz). Home occupancy information is also available recorded by tablet computers and passive infrared sensors. The low sampling rate, dropouts, and low individual appliance coverage makes it difficult to use the dataset to evaluate event-based NILM and activity recognition approaches, as multiple events may happen between two samples.

The Almanac of Minutely Power dataset (AMPds) was introduced by Makonin et al. [19] in 2013. It features electricity, water, and gas readings at one minute resolution of a residential building in Canada. They used an off-the-shelf *Powerscout18* meter (DENT Instruments) to record the whole house consumption and the consumption of individual circuits over a time period of two years. Data from the same house is also available at 1 Hz resolution in the Rainforest Automation Energy (RAE) Dataset [20] introduced by the same authors in 2018. The RAE dataset covers 72 days of electricity data without any power events marked.

In the non-residential domain, Kriechbaumer et al. proposed the Building-Level Office eNvironment Dataset (BLOND) in 2018 [12]. They recorded aggregated and device level data of an office building in Germany over a time period of around 260 days. The authors used custom-built hardware for both aggregated and individual appliance readings. At an aggregated level, they used Hall effect current transformers and AC-AC transformers to record the 3-phase power grid with up to 250 kHz. Individual appliances have been recorded using the same principle (AC-AC transformer + Hall effect current transformers) embedded into off the shelf power strips with up to 50 kHz. Their dataset is split into two measurement series. BLOND-50 features 50 kHz aggregated and 6.4 kHz device level data over 213 days and BLOND-250 features 250 kHz aggregated and 50 kHz device level data over 50 days. For both sets, the 1 Hz apparent power has been derived from the voltage and current waveforms. However, downloading the dataset requires storing ≈40 TB of data. Moreover, the authors have not used their recording system to generate a residential dataset yet.

These datasets have successfully been used to evaluate different event-less NILM algorithms (e.g., in [10,18]). Event-based NILM methods, however, require information about all appliance events in order to evaluate the detection and classification of these events. Such information is not available in the presented datasets. The lack of datasets for event-based NILM algorithms has already been explored by Pareira et al. in [21]. They proposed a post-conducted labelling approach which can be applied to the individual device data of a dataset. Their method uses an automatic event detector based on the log likelihood ratio to recognize events in the power signal. They evaluated the detector using the REDD [9] and AMPds [19] datasets. It achieved F_1 scores of 84.52 % for REDD and 94.87 % for AMPds. The detector results highly depend on the quality of the data and set parameters. Therefore, a supervision is still required.

The Building-Level fUlly-labeled dataset for Electricity Disaggregation (BLUED) introduced by Anderson et al. [13] in 2012 was specifically recorded with event-detection in mind. The authors recorded voltage and current measurements with a resolution of 16 bit and a sampling rate of 12 kHz. Significant appliance transients were labeled manually and using additional sensors and switchable sockets. However, no individual appliance electricity measurements are available in this dataset.

We identified different shortcomings of existing datasets: (1) Larger time periods in which no data or only a part of the data are available (REDD, ECO, UK-DALE). (2) Relative low sampling rate for appliance level data (REDD, UK-DALE, ECO, AMPds) or no appliance level data at all (BLUED). (3) Missing information about the time and type of appliance events (REDD, ECO, UK-DALE, BLOND, AMPds). (4) Unknown number and type of devices which are not monitored individually (ECO, UK-DALE, BLUED, AMPds). (5) No standard procedure to load the data or explore the dataset.

2. Materials and Methods

The shortcomings of existing datasets have been expressed in Section 1.1. In order to overcome these shortcomings, we define a set of challenges that need to be addressed when recording datasets to evaluate load monitoring or other electricity related algorithms in the residential domain. In particular, event-based NILM algorithms and event detection algorithms cannot be evaluated using the existing datasets, as they lack ground truth information (time and type) of events. The challenges have been summarized in Table 1.

Table 1. Challenges that need to be addressed when recording datasets for Non-Intrusive Load Monitoring.

ID	Challenge
C1	**Simultaneous recordings** of a home's aggregated electricity consumption and the consumption of the individual appliances. The individual data can be used to validate the appliance estimates of NILM algorithms. Furthermore, it can be explored how semi-supervised hybrid NILM algorithms such as [22] can benefit from individual appliance data.
C2	**High sampling rates** of the aggregated and individual appliance data. This allows for the extraction of high frequency features from the individual waveforms which might further improve traditional NILM algorithms. Kriechbaumer et al. [12] focused on recording a dataset with a very high sampling rate but, therefore, require to download ≈75 GB of data per day. To not sacrifice usability, a trade-off between high sampling rates and file size needs to be examined.
C3	**Continuous data recording** for multiple days is crucial to understand and explore different consumption behavior based on the *time-of-day* or *day-of-week*.
C4	**High quality dataset labels** to evaluate event-based NILM and event detection algorithms. These labels should consist of a timestamp describing when the event occurred, the device responsible for the event and a textual description of the event.
C5	**High temporal accuracy** of the data and its labels is required. Labels should always reflect the associated change in the signal. This requires that the data streams are in sync and do not drift apart.
C6	**Usability** is one of the most underrated factors of a dataset. However, researchers should be able to explore and utilize a dataset in a quick and easy way.

Based on the stated challenges, we have developed a framework to record and label NILM datasets. The overall flow of this framework is shown in Figure 1. It consists of an aggregated electricity meter (Smart Meter) which records high frequency voltage and current waveforms at the aggregated level, and multiple distributed meters which record voltage and current waveforms of individual appliances. Further sensors can be added to measure other quantities (e.g., temperature or movement). The current and voltage waveforms as well as the sensor data are collected by a recording PC and stored in multimedia containers. Other electricity related metrics such as active and reactive power are derived from the raw current and voltage waveforms. These power data is stored with different sampling rates and is used to generate data labels semi-automatically. A post-processing step extracts events and assigns labels to these events. Both events and labels are refined by a human using a graphic user interface (GUI) resulting in a final set of label files. Each part of the framework is explained in more detail in the remainder of this section.

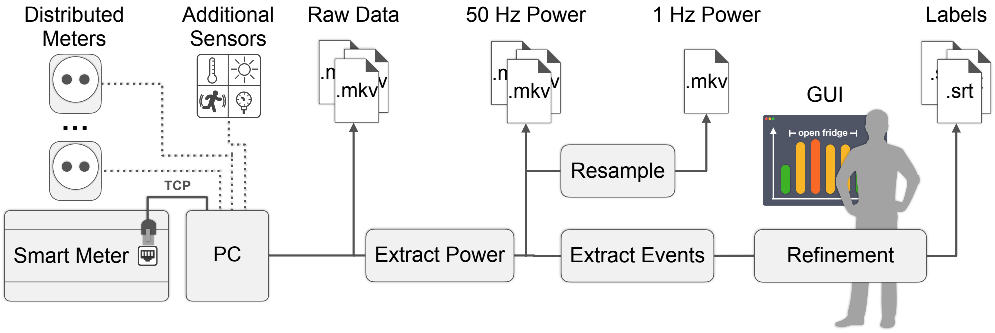


Figure 1. Overall flow of the presented framework to record advanced NILM datasets.

2.1. Smart Meter

Aggregated data are recorded using a custom-built measurement system referenced as the *SmartMeter* from now on. The system was introduced by Völker et al. in [23,24]. A schematic wiring diagram of the smart meter can be seen in Figure 2. It shows the required connections to the power grid. As the analog to digital converter (ADC) requires input voltage levels of 2 V maximum, we use a voltage dividers with a ratio of 1:1000 to scale down the mains voltage levels. Likewise, we use current transformers (*YHDC SCT-013* with a ratio of 1:2000) to convert the home's current consumption into a voltage signal that can be measured by the ADC. Using the split-core variant of the current transformer allows us to measure current in the most non-intrusive way. The home's scaled voltage levels and current consumptions are sampled at up to 32 kHz using the *ADE9000* ADC from *Analog Devices* [25]. The ADC can handle seven input signals at a resolution of 24 bit and a signal-to-noise ratio of 96 dB. It further has an internal Digital Signal Processor (DSP) to calculate attributes like active or apparent power as well as electrical energy. The sampled data is retrieved by an *ESP32* microcontroller (Espressif Systems) over an isolated SPI interface. The microcontroller converts the raw fixed point data to 32 bit float values representing the actual voltage and current measurements (in *Volt* and *Milliampere*, respectively). The data can be sent to a sink via either a *USB Serial*, a *TCP* or a *UDP* connection. An external flash memory allows for buffering the data on short network disconnections. We used 8 MB in the installed system which can hold up to ≈ 41 s of data at a sampling rate of 8 kHz. Furthermore, a Real Time Clock (RTC) is used to sync the sampling rate of the installed ADC (see Section 3.3.4). An Ethernet connection adds a reliable cable connection to the measurement system using the *LAN8720* chip (Microchip Technology) with an ordinary *RJ-45* connector. It is also possible to use WiFi communication, as the ESP32 comes with WiFi on-board. However, in our findings, Ethernet is more reliable in a fuse box environment and should be preferred.

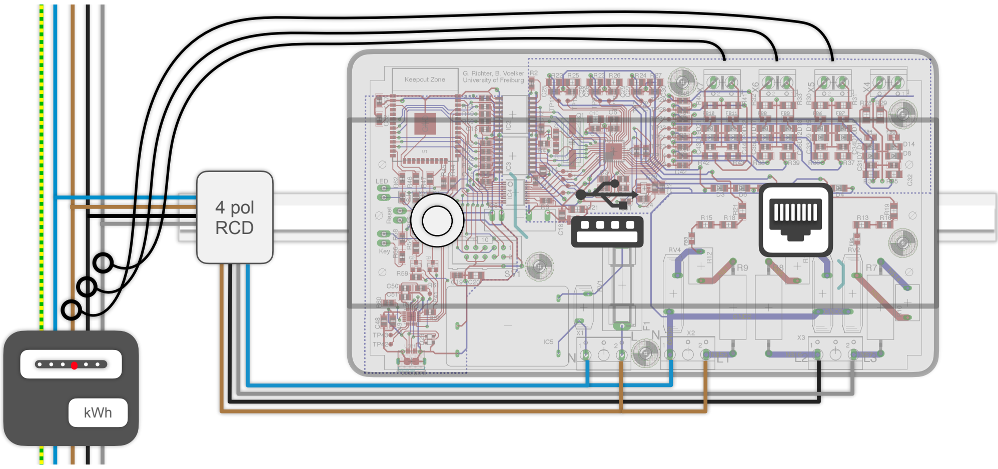


Figure 2. Schematic of the SmartMeter wiring for a three-phase power supply inside the fuse box.

The measurement system is encapsulated in a fire proof DIN housing. This allows the system to be installed at a DIN rail inside the fuse box (as shown on the left side of Figure 6).

2.2. Distributed Meters

Individual appliances can be recorded using a set of *PowerMeters* (see right side of Figure 6). PowerMeters are custom-built smart plugs designed to measure current and voltage waveforms of individual appliances. The plugs were introduced by Völker et al. in [22,23]. Their general system architecture is nearly identical to the architecture of the SmartMeter. A PowerMeter scales down the power outlet's voltage by a factor of 1:580 using a voltage divider and the current drawn by the connected appliance using a 3 mΩ shunt resistor. The analog signals are sampled using a *STPM32* ADC from *STMicroelectronics* [26]. The ADC can sample at up to 7.875 kHz with 24 bit resolution and has an internal DSP. The DSP again allows for calculating other electricity metrics directly inside the smart plug. Data from the ADC is collected by the same ESP32 microcontroller used inside the SmartMeter. 4 MB external flash storage allows for being resilient against ≈ 250 s network dropouts at a sampling rate of 2 kHz. Each distributed meter also includes an RTC for clock synchronization. Data is sent over WiFi, as a wireless solution should be preferred over a wired interface in such a distributed setup. The cost of a single PowerMeter is comparatively low with approximately €35.

The power consumption of each PowerMeter itself is quite low (0.56 W) and compares to the smart plugs used to record other NILM datasets e.g., *Plugwise* as used in [11] (0.5 W).

2.3. Additional Sensors

The framework allows for recording arbitrary sensor values or additional appliance information together with the electricity measurements using an MQTT-API. MQTT [27] provides a standardized publish–subscribe messaging system and has emerged to one of the standard protocols in the world of IoT. If an instance wants to share some information, it can send a message for a given *topic*. If other instances are interested in this information, they can *subscribe* to the specific topic. Each new message *published* under a certain topic is relayed by the MQTT broker to all instances subscribed to this topic. MQTT builds on top of the TCP network protocol which guarantees the successful transmission of data.

The recording PC (see Figure 1) hosts a central MQTT broker. A small *Python* script listens for incoming messages under a general topic *recording* and will handle storing the incoming data into *CSV* files. If a sensor should be added to the recording infrastructure, it simply needs to connect to the broker and send its data on a unique sub-topic (e.g., *recording/livingroom_temp*). Data must follow the *JSON* format. Each JSON key corresponds to a header entry in the resulting CSV file. A timestamp is added by the Python script for each entry if the key ”ts” is not present in the data. An example for a valid message of a temperature sensor is *recording/livingroom_temp {”value”: 20.5}*.

We further highlight three examples of how additional appliance states or sensors can be added:

- **Smart lighting:**
  Many light bulbs are nowadays substituted with smart light bulbs. Most of these can be controlled via a *ZigBee* gateway. Such a gateway can be incorporated to pass information if a light bulb changes its state, dimm state, or color. We have implemented a Python script which interfaces with such a gateway to log the state changes of all light bulbs connected to the gateway using our MQTT-API. This allows for deriving power consumption estimates without intrusively metering each light individually and provides further room occupancy information.
- **Sensors:**
  We show an example flow of how a custom sensor can be developed using the provided MQTT-API in Figure 3. The ESP32 has WiFi built-in and provides certain inter-system interfaces such as *SPI*, *I2C*, or *UART*. This allows for rapid prototyping different sensors like temperature or occupancy.
- **Bridges:**
  The same system overview as shown in Figure 3 can be used to develop different gateways. As an example, we developed a 433 MHz gateway that logs state changes of switchable sockets, wall switches, or remote button presses of devices that are equipped with 433 MHz. We further implemented an infrared sniffer that logs all commands received from off-the-shelf remotes to MQTT. This helps to capture interactions with televisions, HiFi systems, and air conditioners.

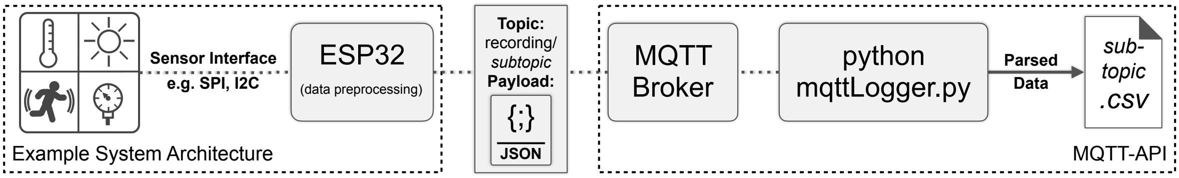


Figure 3. Example of extending the recording system by logging additional sensor data using the MQTT-API.

The Python script further publishes recording information each 600 s under the topic *recordingInfo*. The information includes the current recording state, the number of meters active, and the average power of each meter since the last message.

2.4. Extracting Events

Evaluating event-based NILM algorithms requires having ground truth data for all events in the dataset. The authors of the UK-Dale [11] dataset therefore recorded appliance turn on/off events for house 1 using switchable sockets. If the user pressed the button on such a switchable socket. The current timestamp, device, and state of the socket is logged. We particularly see three drawbacks of such an approach: (1) Devices that are hardwired to the mains like the stove or lighting cannot be equipped with such a socket. (2) Only on/off events can be logged. Most of our household appliances are multi state devices that have more than just a binary state *on* or *off*. (3) Devices that change their state without user interactions can not be labeled (e.g., a kettle turns off automatically if the water is boiling).

We build on the idea of a post-conducted labelling approach introduced by Pereira et al. in [21] and developed a semi-automatic labeling algorithm that consists of three steps: event detection, unique event identification, and high variance filtering.

2.4.1. Event Detection

The event detector utilizes the Log-Likelihood Ratio (LLR) test introduced by Pereira et al. in [21]. It has been enhanced by adaptive thresholding by Völker et al. in [23]. The detector calculates the likelihood ($L(i)$) that an event has happened at sample i by using a detection window over the power signal ($S(i)$). The detection window splits into two sub-windows, the *pre-event* window $[a, i[$, and the *post-event* window $[i, b]$. $L(i)$ calculates as

$$L(i) = \ln\left(\frac{\sigma_{[a,i[}}{\sigma_{[i,b]}}\right) + \frac{\left(S(i) - \mu_{[a,i[}\right)^2}{2 \cdot \sigma^2_{[a,i[}} - \frac{\left(S(i) - \mu_{[i,b]}\right)^2}{2 \cdot \sigma^2_{[i,b]}}, \tag{1}$$

where $\sigma_{[a,i[}$, $\sigma_{[i,b]}$, $\mu_{[a,i[}$ and $\mu_{[i,b]}$ are the standard deviations and means of the pre-event and post-event window, respectively. This signal is cleaned using an adaptive threshold ($thres_i$). If the change of the mean value between the pre- and post-event drops below this threshold, $L(i)$ is forced to zero using

$$L(i) = \begin{cases} L(i), & if \left|\mu_{[a,i[} - \mu_{[i,b]}\right| > thres_i \\ 0, & otherwise \end{cases}. \tag{2}$$

$thres_i$ is defined as

$$thres_i = thres_{min} + m \cdot \mu_{[a,i[}, \tag{3}$$

with $thres_{min}$ being the minimum power change of interest and m a linear coefficient.

This coefficient causes a linear increase of $thres_i$ with the current power drawn (power of the pre-event window). Typically, the variance in the power signal is proportional to the amount of power drawn. This effect is caused by increasing noise in the appliance or the analog frontend of the electricity meter. If a fixed small threshold $thres_i$ is set, a large number of false events may occur at regions where more power is drawn. If a fixed high threshold is set, low power events may be missed. Pereira et al. used a relative large threshold of 30 W [28] which does not allow for detecting state changes of low power devices such as battery chargers or lights. We use the linearly increasing threshold as it adapts to possible larger fluctuations, preventing false events and missed events.

If an event is detected at sample i, the likelihood will also be non-zero around that sample as a mean change is still observable in close proximity to the event depending on pre-event and post-event window sizes. The exact sample at which the event occurred is identified using a *voting window*. This window is applied to the signal L. Inside the window, only the maximum of the absolute value of L is kept. We further restrict the minimum distance between two events with an additional parameter l.

This algorithm has six adjustable parameters: the duration of pre-event, post-event, and voting window, the minimum detection threshold $thres_{min}$, the linear coefficient m, and the minimum distance between two events l. A user should specifically adjust the parameters $thres_{min}$ and l according to prior knowledge of the data: a low threshold $thres_{min}$ is required if events with small mean changes are expected, and a short l should be chosen if events can happen close in time. Values that seem to work quite well across different devices are: pre-event window = 1 s, post-event window = 1.5 s, voting window = 2 s, $thres_{min} = 3\,\text{W}$, $m = 0.005$ and $l = 1\,\text{s}$.

2.4.2. Unique Event Identification

To further simplify the labeling effort, we try to identify similar events of the appliance to label them accordingly. We therefore utilize the fact that most of our home appliances draw different but constant power before and after an event (e.g., the kettle after switched on) which represent constant states of the home appliance (e.g., *off* and *on* for the kettle). Depending on its complexity, an appliance can easily have more than ten unique states (e.g., a dishwasher).

The data is split at each event and the mean power demand between these splits is calculated. Unique mean values (representing unique appliance states) are then identified using hierarchical clustering with a distance threshold determined by $thres_{min}$. Each cluster is given a textual ID which is used to assign labels to each event ($S0$, $S1$, . . ., as shown in Figure 11). As some appliances show a higher rush-in power followed by a power settling due to moving parts in the appliance, we remove the 10 % of the highest and lowest values before calculating the mean value.

2.4.3. High Variance Filtering

Appliances such as PCs or televisions draw variable power depending on the current context (i.e., calculations of the PC, content of a television). This causes a large number of false events using the LLR test. To filter these false events, we first identify regions in the signal that show such high variance and afterwards remove all events found in those regions. We therefore calculate the mean ($\mu(i)$) and variance ($\sigma(i)$) of a sliding window. If $\sigma(i)$ is larger than $n \cdot \mu(i)$, the window is marked. If the length of consecutively marked windows exceeds a certain length (w), all events in these windows are removed. The parameters n and w can be adjusted. Values which show good results were found empirically as $w = 4\,\text{s}$ and $n = 0.005$.

By using the event extraction algorithm, an appliance power signal can be pre-labeled. Each found event is marked and a unique label is assigned. The extensive task of labeling can therewith be reduced to supervision and inspection: Remaining falsely classified events (FP) need to be removed, events not found (FN) need to be added, and each unique state label should be changed to a meaningful label representing the state of the appliance.

2.5. Human Supervision

To combine the presented automatic event labeling (Section 2.4) with a simple graphic based human supervision, we have developed the Annoticity inspection and labeling tool. The tool is introduced by Völker et al. in [16]. Annoticity is implemented as an interactive web application. Its workflow is depicted in Figure 4. The tool allows for uploading your own data in the *Matroska* multimedia (MKV) [29] or *CSV* format. A user can further select data form several existing datasets such as REDD [9], ECO [18], BLOND [12], UK-DALE [11], or FIRED [17].

Annoticity is split into a server backend and client frontend. The backend loads the data and prepares it for visualisation. Data is down-sampled to a reasonable sampling rate according to the current time-span selected by the user. Furthermore, the automatic labeling algorithm presented in Section 2.4 can be performed, and file downloads (labels or data) are provided.

The graphical user interface of the client frontend is shown in Figure 5. After either uploading a file or selecting a device of an available dataset, the user can visually inspect the data. All available measures (e.g., active and reactive power) can be selected. Zooming into the data reveals more information as it leads to a data download at a higher sampling rate. The user can further execute the automatic labeling algorithm resulting in an initial set of labels. Each label consists of a start time and a textual description representing the event or the state after the event. The initial set can be adjusted by the user. Labels can be added, removed, or its text can be modified. If the user is only interested in the events' timestamps, all text can be removed. Furthermore, it is possible to modify all labels with the same text in one step. The frontend also allows for adjusting the parameters of the automatic labeling algorithm explained in Section 2.4. The final set of labels can be stored either as plain *CSV*, *ASS*, or *SRT* files or embedded into a *MKV* container with the original data.

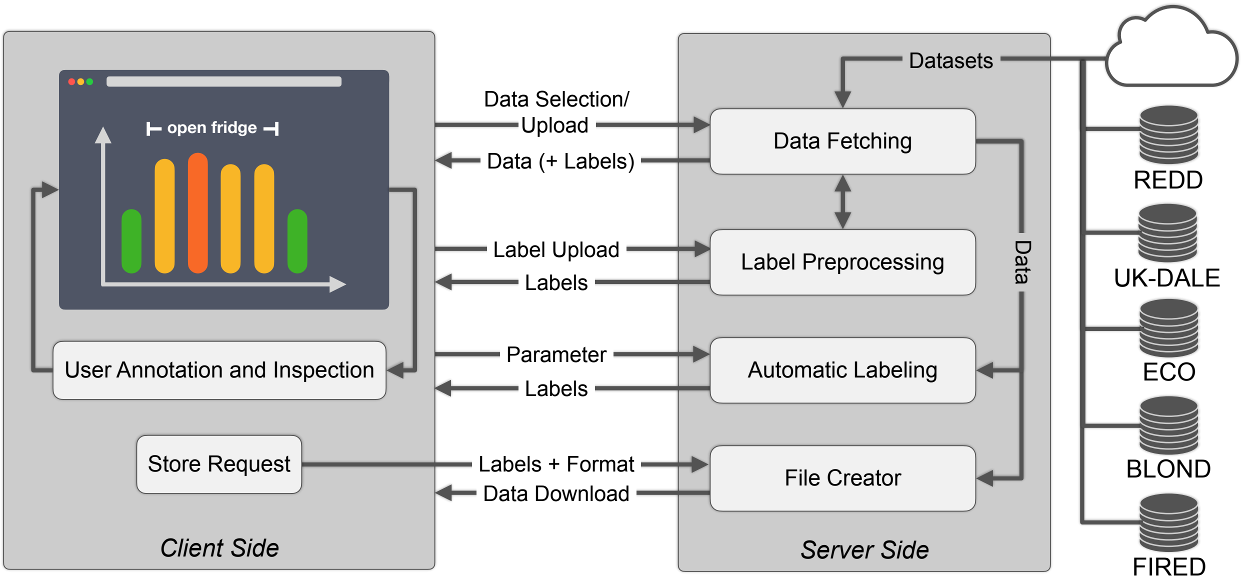
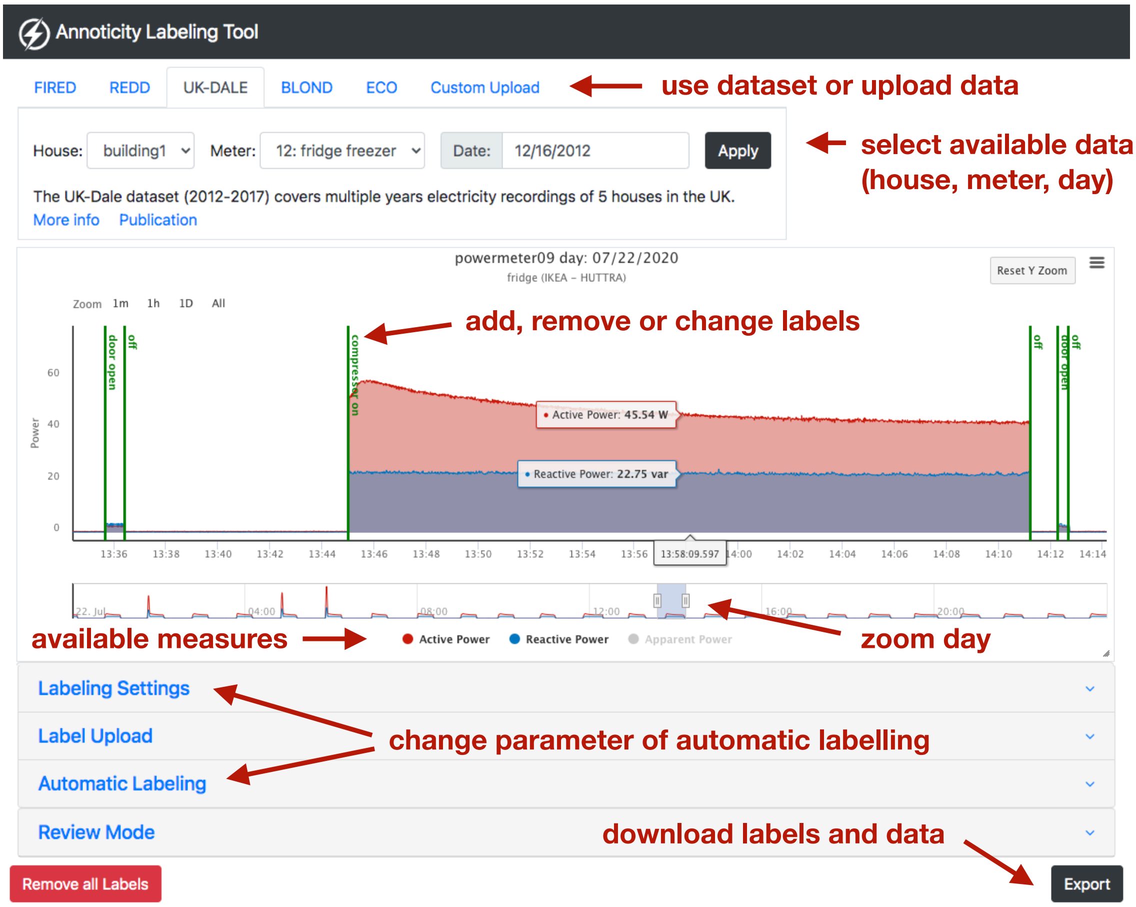


Figure 4. Flow of the Annoticity labeling tool. Data fetching, automatic labeling, and file creation are performed on the server side, while labeling and user interaction is performed on the client side (modified from [16]).



House: building1 Meter: 12: fridge freezer Date: 12/16/2012

The UK-Dale dataset (2012-2017) covers multiple years electricity recordings of 5 houses in the UK.

use dataset or upload data

select available data (house, meter, day)

add, remove or change labels

available measures

zoom day

change parameter of automatic labelling

download labels and data

Figure 5. The graphical user interface of the Annoticity inspection and labeling tool. The fridge events were generated and clustered automatically. Each text description (*off*, *compressor on*, and *door open*) was only set once by the user. All other occurrences were labeled accordingly (modified from [16]).

3. Results

We utilized the framework explained in Section 2 to record and label the Fully-labeled hIgh-fRequency Electricity Disaggregation (FIRED) dataset which was first introduced by Völker et al. in [17]. Version 2 of the FIRED dataset extends its time period to 101 days. The data includes aggregated three phase current and voltage measurements sampled at 8 kHz as well as 21 time synced individual appliance measurements sampled at 2 kHz from a residential apartment in Germany. Furthermore, it includes sensor readings such as room temperatures and additional state information of certain appliances and each light bulb in the apartment. Annoticity has been used to fully label all state changes of the sub-metered appliances over a time period of two weeks.

The FIRED dataset was collected at an apartment building constructed in 2017 with seven apartments on four floors. Data was collected from a three-room apartment with 79 m^2 of space (open combined kitchen and living room, bedroom, child's room partly used as office, hallway, bathroom, and storage room). The apartment is inhabited by two adults and one child. The building is heated via a district heating and most rooms are equipped with air filters with built-in recuperators. According to the building's energy certificate, it requires a primary energy consumption of 12 kWh/(m^{2}a). The apartment's power grid is a three-phase 50 Hz system consisting of L_1, L_2, L_3, and neutral (N) wires. L_1–L_3 has a phase shift of 120°. Access to the apartment's electrical system is given through a fuse box located in the hallway. All lights installed in the apartment are off-the-shelf smart light bulbs with a built-in *ZigBee* module. This allows the lights to be turned on or off via a smartphone application, voice assistant, or regular wall-light-switch. It further allowed to log all state changes during the recording of the dataset as explained in Section 2.3. The washing machine, dryer, and freezer are located in the basement of the building and are not part of the recording.

A SmartMeter (see Section 2.1) was installed in the apartment's fuse box. Split core current transformers were attached to the three incoming supply legs. For voltage measurements, L_1, L_2, L_3, and N were connected in parallel. The meter is supplied with power by an additional L_1 leg that is secured by a separate 16 A fuse. The final installation is shown in Figure 6 (left).

We further deployed 21 PowerMeters (see Section 2.2) in the apartment and connected them to WiFi. We further checked that the WiFi signal quality (*RSSI*) of each PowerMeter exceeded −60 dBm to be certain that data could be sent flawlessly. Some devices like the oven and the exhaust hood were directly connected to the mains. To measure those appliances, we connected a special version of our PowerMeters with screw terminals. Figure 6 (right) shows two PowerMeters connected to the espresso machine and coffee grinder.

Modern households can easily include more than 40 appliances (68 in the FIRED household). Many of these devices are only plugged in occasionally and sometimes at a different socket than before. Therefore, connecting a continuously sensing meter to each appliance is infeasible. Instead, we connected devices of the same category (e.g., routers) or devices which are only used simultaneously (e.g., monitor and PC) to the same PowerMeter. Devices which are only plugged in occasionally and typically not at the same time (e.g., mixer or vacuum cleaner) were connected to a dedicated PowerMeter (*pm11*). If an appliance was connected or disconnected, a corresponding entry was manually added to a log file. This means that, even if the socket was continuously sampling data, the appliance connected to it changed.

Moreover, temperature and humidity sensors were installed in most of the rooms, a ZigBee logger was set up, and both a 433 MHz and an infrared bridge were installed (as described in Section 2.3).

To properly connect all individual sensors to a central recording PC, the apartment was equipped with four WiFi access points. The power consumption of the recording PC and access points were recorded individually and also contribute to the apartment's aggregated consumption. The recording PC gathered the data of all electricity meters and sensors, stored these into files frequently, and pushed the files to a cloud server for persistent storage.

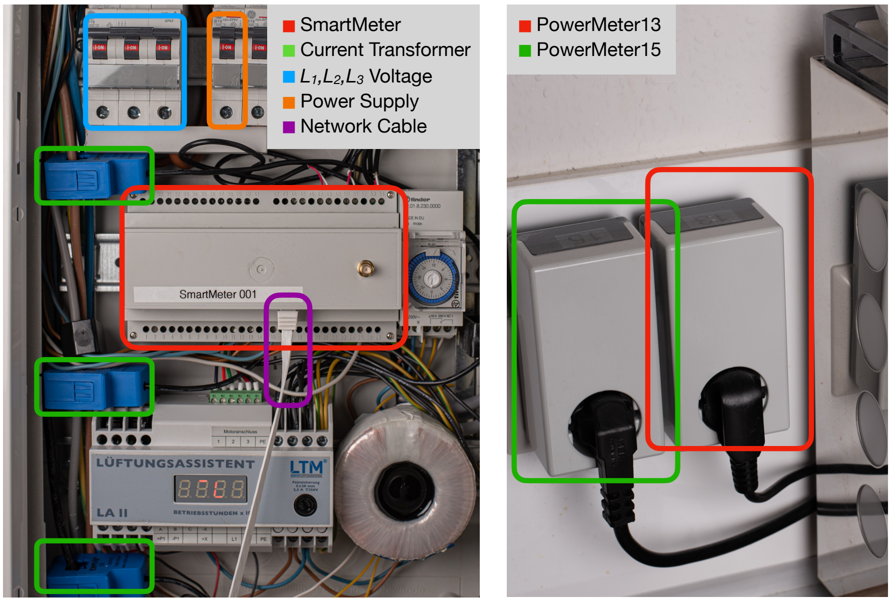


Figure 6. (**Left**) SmartMeter installed in the apartment's fuse box. (**Right**) PowerMeters with ID 13 and 15 connected to the coffee grinder and the espresso machine.

3.1. Data Records

The provided data include voltage and current measurements at high sampling rates taken from the aggregated mains' signal and 21 individual sockets. Furthermore, FIRED contains per day and device summary files with derived power measurements to get a quick insight into the data. The root directory of the dataset contains folders with the *raw* and *summary* data. The data are stored as multiple *MKV* container into sub-folders named *powermeter<ID>* and *smartmeter001*, respectively. We used multimedia containers to store the data, as we previously explored their benefits in [30]. While being optimized for audio or video streams, these containers allow for storing regularly sampled sensor data as time synced audio streams. Text based labels can further be stored as subtitle streams in the same file. The FIRED data of each metering device are stored as a single *WavPack* [31] encoded audio stream inside the multimedia container. Each stream has multiple channels for the voltage and current signals. As stated in [30], WavPack allows for a lossless compression while maintaining high compression rates for time series data. In particular, we achieved a compression ratio of 1.46 for the voltage and current measurements using WavPack (only 1.42 could be achieved with *hdf5* [32], which has been used e.g., for UK-DALE [11] and BLOND [12]). We also store different Metadata for each of the streams including the start timestamp (with microseconds resolution), the particular meter used, the sampling frequency, codec information, the name of the measured attributes, and the stream's duration. Therewith, each file is self descriptive and can be used without prior knowledge. File size and, therewith, file loading times are kept reasonable by splitting all files at regular time intervals. The local time of the first sample is appended to each filename as *<year>*_*<month>*_*<day>*__*<hour>*_*<min>*_*<sec>*.

Table 2 shows the mapping of the recorded appliance to the used PowerMeter (ID). For more information about the appliance, its brand and model are shown with its power rating (P) according to the device manufacturer as well as the average $\overline{P}$ and maximum power P_{max} observed during recording. Φ shows the live wire (L_1, L_2 or L_3) to which the PowerMeter was connected. A complete list of all appliances in the apartment is part of the dataset. It contains additional information and website links for each appliance.

3.1.1. Voltage and Current Data

All PowerMeters sampled the current and voltage waveforms at a rate of 2 kHz. While in theory data can be sampled and sent at up to ≈8 kHz using our PowerMeters, the available WiFi bandwidth limits the amount of data that can be sent simultaneously by all meters. Therefore, we chose a sampling rate of 2 kHz as a trade-off between reliability and temporal data resolution (see Section 3.3.3 for more information). The SmartMeter recorded voltage and current waveforms of L_1, L_2, and L_3 with a sampling rate of 8 kHz. The ADC installed in the SmartMeter allows for sampling these waveforms with a rate of up to 32 kHz, but, again, we preferred a higher reliability over a better time resolution. This is in line with Armel et al. [7] who stated that sampling rates between 15 kHz to 40 kHz will not improve the performance of NILM algorithms as higher frequency signal components are distracted by noise in real buildings.

Each file contains 600 s of data stored into a single audio stream inside the multimedia container. For the aggregated data, each audio stream has six channels (*v_l1*, *i_l1*, *v_l2*, *i_l2*, *v_l3*, *i_l3*) representing the current and voltage waveforms for the three supply legs. The audio streams for the individual appliance data contain two channels (v,i). The number of samples in each file should match the time distance to the next file. If this is not the case, no data is available for this particular meter during this time period. This occurred during a reliability reset each day at midnight and rarely for single meters due to occasional data loss as depicted in Section 3.3.3.

Data pre-processing is typically not required, as both the SmartMeter and all PowerMeters calculate the physical quantities (*Volt* for voltage and *Milliampere* for current measurements) from the raw ADC samples.

Table 2. Appliances recorded via PowerMeters. ID represents the identifier of the PowerMeter used for recording. For *PowerMeter11*, the connected appliance changed during recording. P is the power according to the device manufacturer, Φ is the *Live Wire* the device is connected to (L_1, L_2 or L_3), P_{max} is the maximum average power drawn for the duration of one second and $\overline{P}$ is the per day average power seen during the recording. The unit of all power measurements is Watts.

ID	Connected Appliance	Brand	Model	P	Φ	P_{max}	$\overline{P}$
08	Baby Heat Lamp	Reer	FeelWell	600	2	611.93	0.30
09	Fridge	IKEA	HUTTRA	1000	3	1138.79	18.02
10	Smartphone Charger #1	-	2 Port USB	10	3	12.74	1.68
11	Changing Device				3	1898.71	3.10
12	Smartphone Charger #2	-	4 Port USB	25	1	27.86	2.77
13	Coffee Grinder	Graef	Cm800	128	3	206.89	0.17
14	Smart Speaker	Apple	HomePod	15	3	3.60	0.23
15	Espresso Machine	Rocket	Appartamento	1200	3	1230.62	29.82
16	Kettle	Aigostar	Adam 30GOM	2200	3	1958.76	2.89
17	Hairdryer	Remington	D3190	2200	1	1934.85	1.00
18	Router #1	Apple	Airport Extreme A1521	10.3	1	27.97	19.34
	Router #2	Telekom	Speedport Smart 1	10			
	Telephone	Gigaset	A400	1			
19	Printer	EPSON	Stylus SX435W	15	1	21.45	0.16
20	Office PC	Apple	Mac Mini A1993	85	2	236.08	59.49
	27″ Display	Apple	Thunderbolt display	200			
	Speaker	Logitech	Z2300	240			
	Smartphone Charger #3	Apple	MD813ZM/A	5			
	Access Point #2	Apple	Airport Express A1264	8			
21	Media PC	Apple	Mac Mini A1347	85	3	45.38	12.98
22	HiFi System	Onkyo	TX-SR507	160	3	85.86	15.27
	Subwoofer	Onkyo	SKW-501E	105			
23	Television	Samsung	UE48JU6450	64	3	150.96	12.31
24	Light+Driver	IKEA	-	40	3	36.56	1.75
25	Oven	IKEA	MIRAKULÖS	3480	3	2491.03	9.69
26	Access Point #3	Apple	Airport Express A1392	2.2	3	2.67	2.21
27	Router #3	Netgear	R6250	30	1	56.91	15.74
	Recording PC	Intel	NUC8v5PNK	60			
28	Fume Extractor	IKEA	WINDIG	250	3	249.26	1.11

The provided voltage and current data without any pre-processing can be seen in Figure 7. Plots 1–3 show data of the SmartMeter while plots 4 and 5 show the simultaneous measurements of two additional PowerMeters. The figure does not only highlight the high temporal resolution of the data but also the achieved clock synchronization. The rush-in current shown in the PowerMeter data (Figure 7 plot 4) matches the rush-in current seen in L_3 of the SmartMeter (Figure 7 plot 3). A time shift between the measurement devices of around 10 ms can be observed. Even after 16 h of continuous recording, the offset between the SmartMeter and PowerMeters is below one mains cycle, highlighting the effectiveness of the realized clock synchronization (see Section 3.3.4).

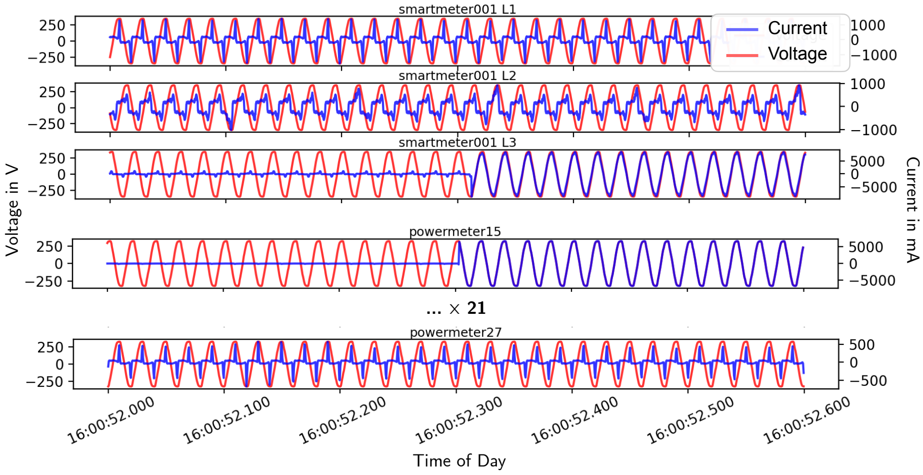


Figure 7. Voltage (red) and current (blue) waveforms of the SmartMeter, powermeter15 and powermeter27. The recording was taken on 9 June 2020 at around 4:00 p.m. The same appliance switch-on event of the espresso machine is visible in the recording of L3 of the smartmeter and of powermeter15.

3.1.2. Power Data

The power data was derived from the voltage (V) and current (I) waveforms. We calculated active, reactive, and apparent power from the raw voltage and current data of all recording devices. The data is stored as a single file for each day of the recording. The formulas used to calculate the individual powers based on the mains frequency $f_l = 50\,\text{Hz}$ are shown in (4), (5), and (6), respectively:

$$P(n) = \frac{1}{N} \cdot \sum_{i=0}^{N-1} V(i) \cdot I(i) \tag{4}$$

$$S(n) = I_{RMS}(n) \cdot V_{RMS}(n) \tag{5}$$

$$Q(n) = \sqrt{S(n)^2 - P(n)^2} \tag{6}$$

P, Q, and S are calculated for each non-overlapping window n. The length of the window is $N = \frac{f_s}{f_l}$. f_s is the sampling rate of the voltage and current measurements. For the 8 kHz, SmartMeter data N is $\frac{8000\,\text{Hz}}{50\,\text{Hz}} = 160$. I_{RMS} and V_{RMS} are calculated as follows:

$$I_{RMS}(n) = \sqrt{\frac{1}{N} \cdot \sum_{i=0}^{N-1} I(i)^2} \tag{7}$$

$$V_{RMS}(n) = \sqrt{\frac{1}{N} \cdot \sum_{i=0}^{N-1} V(i)^2} \tag{8}$$

Since data of commercial smart meters have a sampling frequency of 1 Hz to 0.01 Hz, an additional 1 Hz version of the power data is provided.

The 50 Hz and 1 Hz power are stored for each meter individually and contain one day of data. Times for which the power could not be calculated as no voltage and current data being available are marked with a power of constant zero to maintain an equidistant time period between samples.

Figure 8 shows the single day apparent power consumption of the apartment. The contribution of the six appliances which consumed the most power on this day is shown as individual colored blocks. All other appliances are summed and plotted as *Others*. The aggregated power consumption of the SmartMeter is shown as *mains*. Ideally, the superposition of the apparent power of all individual meters should match the aggregated apparent power. Nevertheless, a small margin can be observed in Figure 8. This gap is caused by hard-wired appliances such as lights and the ventilation system, which are not individually monitored (see Section 3.3.2 for more information).

3.1.3. Logs

The *annotation* folder contains 33 tab-separated *CSV* files. The first column of each file represents the timestamp associated with the event or sensor reading. We divide these files into three categories:

- **Smart lights:**
  State changes of each light in the apartment is logged. The filenames of these logs have the form:
  light__<room>__<deviceName>__<deviceModel>.csv
  room is the room the light is installed in, *deviceName* represents how this light is used (e.g., ceiling light) and *deviceModel* matches the light model name. The file's second column represents the state of the light (*on* or *off*), the third column is the light's intensity (0 % to 100 %) while the last column represents the light's *hex* color. If setting different colors is not supported by the light, the column only shows *None* values.
  As individual measurements of the apartment's lighting have shown, the installed smart lights consume constant apparent power linearly increasing with the light's intensity (dimm setting). As information of the lights' state and dimm setting is available for the complete recording duration, this information can be used to estimate the power consumption of each light.
  The smart light logs of two days are shown in Figure 9. The *hallway ceiling light* consists of three light bulbs and is triggered by a passive infrared sensor. Hence, all three light bulbs are turned on if a resident walks through the hallway, which can be seen in Figure 9 throughout the days.
- **Sensor readings:**
  The readings of temperature and humidity sensors are stored in files following the name scheme:
  sensor__<room>__<sensorType>.csv
  sensorType is either *hum* or *temp* for humidity or temperature readings. Each file's second column represents the sensor reading. Temperature readings are stored in degrees Celsius and humidity readings in %, respectively. All values have floating point precision. Samples are not acquired equidistantly, as the sensors only send new values if they have changed.
- **Device info:**
  Certain smart devices or bridges as explained in Section 2.3 allow for capturing events of devices in the apartment, e.g., pressing a certain key of the television remote. These events are logged in files with the following name format:
  device__<room>__<deviceName>__<deviceModel>.csv
  Each file's second column gives information about the current device state or happened event. The file of the HiFi system, for instance, includes key-presses such as *power* or *vol_up*. Figure 10 shows the logs for the television, the HiFi system, and the espresso machine.

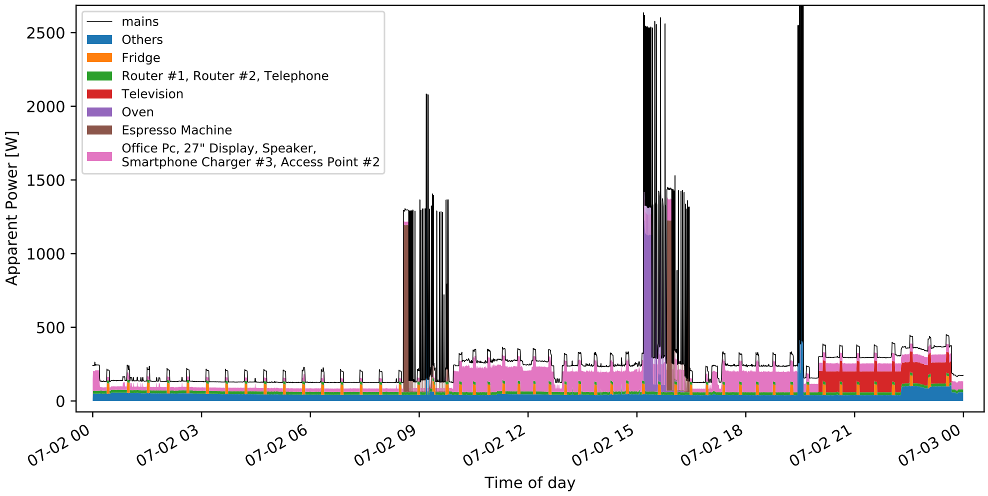


Figure 8. The power consumption of the apartment over one full day (2nd July 2020). The power is down-sampled to one sample every 3 s. The black line indicates the power consumption recorded by the SmartMeter. The contribution of the six top-most consumers is shown as stacked colored blocks. The consumption of the remaining individually metered appliances are aggregated and shown as the blue block *Others*. A slight offset between the SmartMeter and the accumulated power of all PowerMeters can be seen.

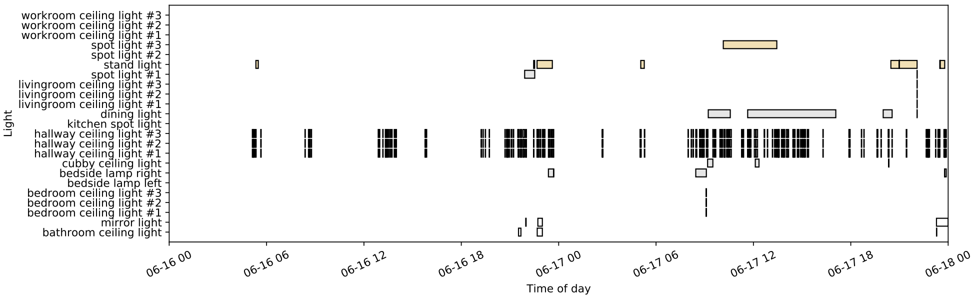


Figure 9. Two days of light usage information (16th–17th June 2020). The time of day is shown on the *x*-axis while the particular light is shown on the *y*-axis. If a light was used, a black box is shown during this period. The box is filled with the set color value and dimm setting.

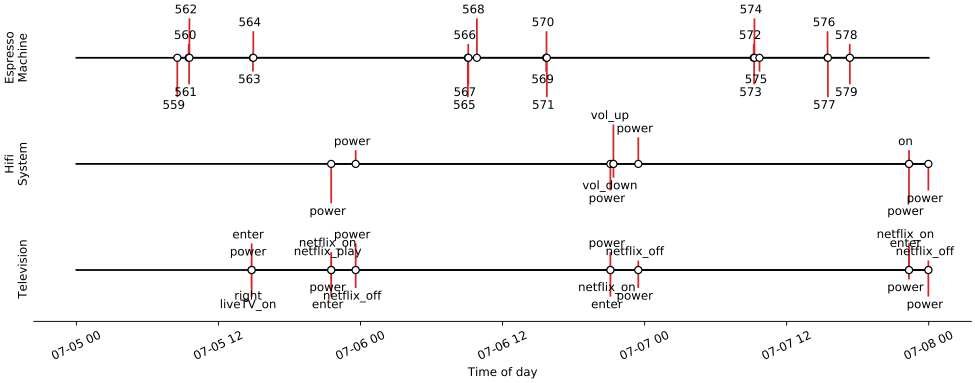


Figure 10. Three days of device logs (19th–22nd June 2020). The data of the espresso machine shows the numbers of espressos made, while the data of the HiFi system and television show key-presses on the remote. The television content is provided by a connected media PC.

3.1.4. Data Labeling

We used the Annoticity labeling tool to fully label all events that happened within two weeks of the FIRED data (22nd July 2020–4th August 2020). The automatic labeling tool was used to generate an initial set of labels. This set was modified by visually inspecting the data. We removed false events, added missing events, and assigned a distinct and descriptive label to each appliance state. The labels were stored as *CSV* files and are part of the dataset. During labeling, we also stored the initial set of labels obtained by the automatic labeling algorithm to evaluate the algorithm's performance. Figure 11 shows both the initial set of labels and the final labeled data of the 'espresso machine'.

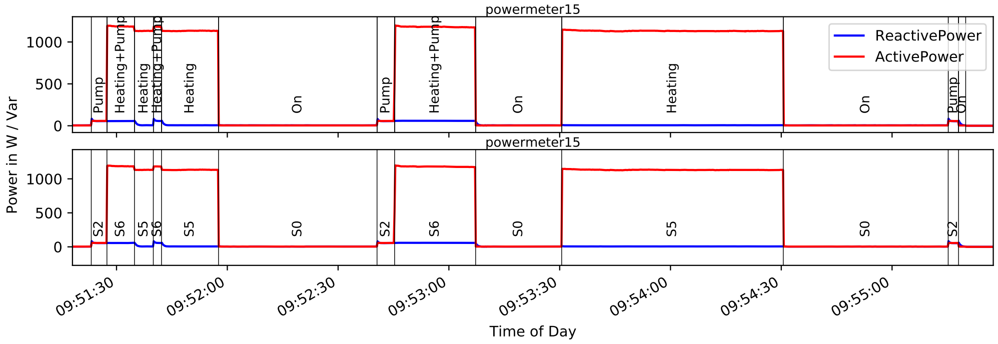


Figure 11. The fully labeled data of the espresso machine (30th of July at 9:50 a.m.). The bottom plot shows the initial labeling of the automatic labeling algorithm, while the top plot shows the final labeling after human supervision. (The rightmost event has been missed by the algorithm.)

To evaluate the performance of the labeling tool (see Section 2.4), we compared its result with the final set of manually labeled events. Evaluation was done in terms of True Positives (TP) i.e., true events, False Positives (FP), i.e., falsely classified events and False Negatives (FN), i.e., events not found by the tool. A TP is defined as a classified event which is reflected within 2 s in the manually labeled data. According to that, a FN is an event in the manually labeled data not found within 2 s during detection. An FP is a classified event which is not present in the manually labeled data. The F_1 score was used to summarize these metrics into a single score. We fixed the algorithm's parameters for all appliances to simplify the evaluation: pre-event window length = 1 s, post-event window length = 1.5 s, voting window length $= 2$ s, $thres_{min} = 3$ W, $m = 0.005$, and $l = 2$ s.

We experienced that the labeling algorithm is performing fairly well for devices which show distinct states in the power signal (such as the oven, kettle, or the espresso machine shown in Figure 11). For devices that draw variable power in between states (such as a PC or a coffee grinder), a large number of false events were triggered.

To put this into perspective, Section 3.1.4 shows the evaluation results split into two groups: #1 represents appliances that show distinct states and #2 represents appliances which draw variable power. Appliances for which no distinct events were labeled manually (e.g., network equipment) are omitted.

Section 3.1.4 indicates that most of the appliances present in residential homes (group #1) can be labeled in a semi-automatic way. The *Coffee Grinder* and the *HiFi System* show a comparable low performance with a high number of FP. This is due to higher variance if the motor is active or music is playing and could have been avoided by using a higher linear factor m or a higher threshold $thres_{min}$.

To get an overall estimate of how much the labeling effort can be reduced by using the automatic labeling, we compared the raw number of clicks required to label the data from scratch with the number of clicks required to supervise and modify the pre-labeled set generated by the automatic labeling algorithm for group #1. In total, 4379 events were labeled manually. If we omit the task of applying textual labels, labeling events would still have required at least 4379 clicks. As shown in Table 3, for devices in group #1, the labeling algorithm automatically placed 3232 events at the correct position. With 159 falsely classified events, 377 missing events, and the 770 missing labels of group #2 (which would require manual labeling), 1306 clicks were required to remove false events and add missing events. Therewith, the sheer amount of clicks was already reduced by 70.18% not accounting for the support provided by the Annoticity tool while applying textual labels:

$$Reduction = 1 - \frac{t_{add} \cdot MissedEvents + t_{del} \cdot FalseEvents}{t_{Add} \cdot AllEvents} \tag{9}$$

If we also accommodate the fact that it typically takes less time to remove a falsely placed label (t_{del}) compared to manually adding a label from scratch (t_{add}), we can apply Equation (9). Using $t_{add} = 10\,\mathrm{s}$ and $t_{del} = 5\,\mathrm{s}$ as a reasonable guess of this difference, the reduction in labeling effort is actually 71.99% compared to a fully manual approach. Considering that the parameters could have been manually adjusted and optimized for each appliance, the actual reduction might be even higher.

Table 3. Results for the automatic labeling algorithm split into two appliance groups. In #1, appliances are grouped which show distinct states in the power signal in which nearly constant power is drawn. #2 groups appliances that draw variable power. *Events* marks the number of ground truth events labeled manually.

Group	Appliance	Events	TP	FP	FN	F1
#1	Baby Heat Lamp	6	6	0	0	100.00
	Fridge	1006	863	2	143	92.25
	Coffee Grinder	348	250	114	98	70.22
	Espresso Machine	1880	1760	0	120	96.70
	Kettle	30	30	0	0	100.00
	Hairdryer	18	17	0	1	97.14
	Hifi System, Subwoofer	45	44	37	1	69.84
	Television	79	65	4	14	87.84
	Kitchen Spot Light	12	12	0	0	100.00
	Oven	138	138	1	0	99.64
	Fume Extractor	47	47	1	0	98.95
	Sum	3609	3232	159	377	92.34
#2	Smartphone Charger #1	96	83	1491	13	9.94
	Smartphone Charger #2	63	45	7999	18	1.11
	Office Pc	583	410	85,367	173	0.95
	Media Pc	28	10	26,632	18	0.07
	Sum	770	548	121,489	222	0.89

3.2. Data Statistics

Overall, we collected 53,328 h of raw current and voltage waveforms using the proposed framework. Figure 7 highlights the richness of the captured data for the SmartMeter and the PowerMeter units. Figure 8 shows the apparent power extracted for each individual appliance. It further emphasizes the contribution of each appliance to the total power consumption on this day.

According to [33], the average consumption of a comparable three person household in Germany is 7.12 kW h per day. Evaluating the SmartMeter data of the FIRED dataset results in an average electricity consumption of 6.06 kW h per day, which is slightly lower compared to the average, but this is expected as the data does not contain the electricity consumed by the washing machine, dryer, and freezer.

Figure 12a shows the consumption of six appliances at the hour of day averaged over the whole recording duration. This delivers a good indication for usage patterns. For example, the ’espresso machine’ shows two distinct peaks, one in the morning at around 9:00 a.m. (morning coffee) and one in the afternoon at 3:00 p.m. (coffee break). In comparison, the ’router’ does not show any significant peak. It can also be seen that the ’office PC’ has a high standby consumption of around 35 W and is used mainly between 9:00 a.m. and 6:00 p.m. Figure 12b shows the distribution of power demand for the same appliances. Some state information can already be derived from these plots. The ’hairdryer’ shows two distinct states representing two different temperature settings. The ’office PC’ shows three peaks. The peak around 35 W represents the already mentioned standby consumption, the peak around 50 W represents the PC in its *On* state, and the 140 W peak includes the *On* state of the 27-inch monitor connected to the same meter. The ’espresso machine’ consumes a huge amount of power (1200 W) during its heating cycles but is mostly idle (5 W) in between.

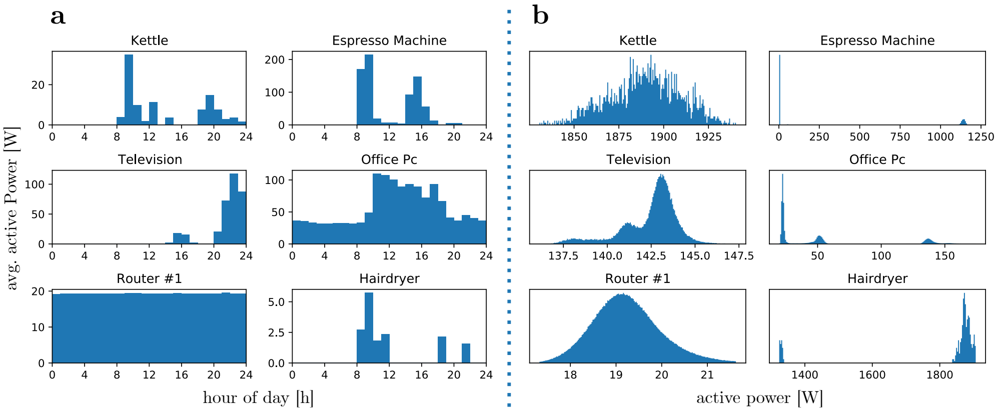


Figure 12. Appliance usage over the complete recording duration. (**a**) shows the daily usage patterns of the devices as the average power for each hour-of-day; (**b**) shows the histogram of the power demands. A 2 W threshold was set to omit data in which no power is drawn.

3.3. Technical Validation

Measurements of the FIRED dataset are provided without applying any pre-processing or filtering rather than the conversion to physical quantities. The recording framework is equipped with different mechanisms to cope with real world effects such as network dropouts or clock drifts, and the data integrity was analyzed in detail.

3.3.1. Calibration

We calibrated each meter in advance using an *ENERGY-LOGGER 4000* [34] (with a stated accuracy of 1%) as a reference. We used ten loads with different power consumption ranging from 5 W to 2000 W and applied a linear calibration. The calibration parameters of each meter were stored permanently in their non-volatile memory. We repeated the calibration after the recording duration to see if aging affects have already occurred. Such effects could not be identified.

3.3.2. Residual Power

The sum of all PowerMeters’ apparent power matches the apparent power recorded by the SmartMeter with a slight offset (residual power). The residual power is the portion of the total consumed power which is not metered by an individual meter, i.e., the portion for which no ground truth data is available. Our goal was to minimize this portion in order to provide reliable ground truth data that can be used by supervised machine learning algorithms.

The residual power observed in the FIRED dataset (see Figure 8) is mainly due to non-monitored, hard-wired devices in the apartment such as the lighting and the ventilation system but also due to the power consumption of the distributed meters. The individual consumption of each light-bulb can be estimated with the log files which we provide with our dataset. To show that this is feasible, we generated apparent power estimates using these log files and the provided individual light recordings. The consumption of the remaining unmonitored appliances (including the consumption of 21 PowerMeters) is the base power consumption of the apartment. It can be estimated at times when lights are turned off and the majority of appliances do not consume any power which is typically during the night or in the case of owner absence. We calculated the base power $\mathbf{P}_{baseLx}$ for each individual supply leg $x \in [1, 2, 3]$ as

$$\begin{aligned} PM_{Lx} &:= \{\, pm \in PM \mid \text{phase of } pm \text{ is } x \,\}, && (10)\\ L_{Lx} &:= \{\, l \in Lights \mid \text{phase of } l \text{ is } x \,\}, && (11)\\ \mathbf{P}_{baseLx} &= \mathbf{P}(SM_{Lx}) - \sum_{pm \in PM_{Lx}} \mathbf{P}(pm) - \sum_{l \in L_{Lx}} \mathbf{P}(l). && (12) \end{aligned}$$

SM_{Lx} is the SmartMeter data of live wire Lx, PM is the set of all PowerMeters, PM_{Lx} is the set of PowerMeters that are connected to live wire Lx, $Lights$ is the set of all lights, L_{Lx} is the set of lights connected to Lx and $\mathbf{P}(X)$ is the extracted power trace of a meter or light X. We assume that the base power is normally distributed and therewith remove all points in $\mathbf{P}_{baseLx}$ that are further away than σ from the mean value and calculate $\mathbf{P}_{baseLx}$ as the mean from the cleaned signal.

Figure 13 shows the apparent power consumption including the lighting and the estimated base power with a remaining Root-Mean-Square Error (RMSE) of 17 V A.

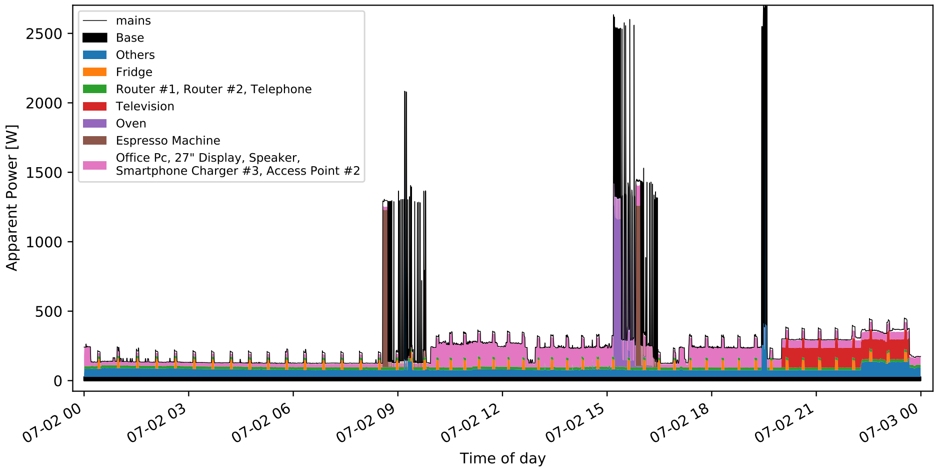


Figure 13. The power consumption of the apartment over one full day (2nd July 2020). The power is down-sampled to one sample every 3 s. The black line indicates the power consumption recorded by the SmartMeter. The contribution of the six top-most consumers is shown as stacked colored blocks. The consumption of all remaining appliances and the reconstructed consumption of the apartment's lighting are aggregated and shown as the blue block *others*. The black base block represents the apartment's base power estimated as 26.66 V A on average for this day.

3.3.3. Availability

The data availability over the complete recording duration was 99.96 %. The missing data amounts to 1405 min and is mainly due to a reliability reset which we perform each day at 12:00 a.m. accounting for approximately 20 s of missing data per meter and day. Figure 14 shows the availability of each metering device over the complete recording period. The reset each day at midnight can be clearly seen in the plot. Occasionally, due to WiFi connection outages and an erroneous implementation of the TCP/IP stack on the ESP micro-controller, some data packets can be lost. However, a packet only accounts for less than 20 ms of data. Once detected, we replace missing samples with zeros to maintain the correct timestamps for all remaining samples. It is still possible to identify these time periods as voltage, and current zero plateaus cannot occur in other situations. *PowerMeter14* and *PowerMeter22* show this behavior more frequently compared to all other meters. The reason might be an unstable WiFi condition as the RSSI values reported by both meters were the lowest of all.

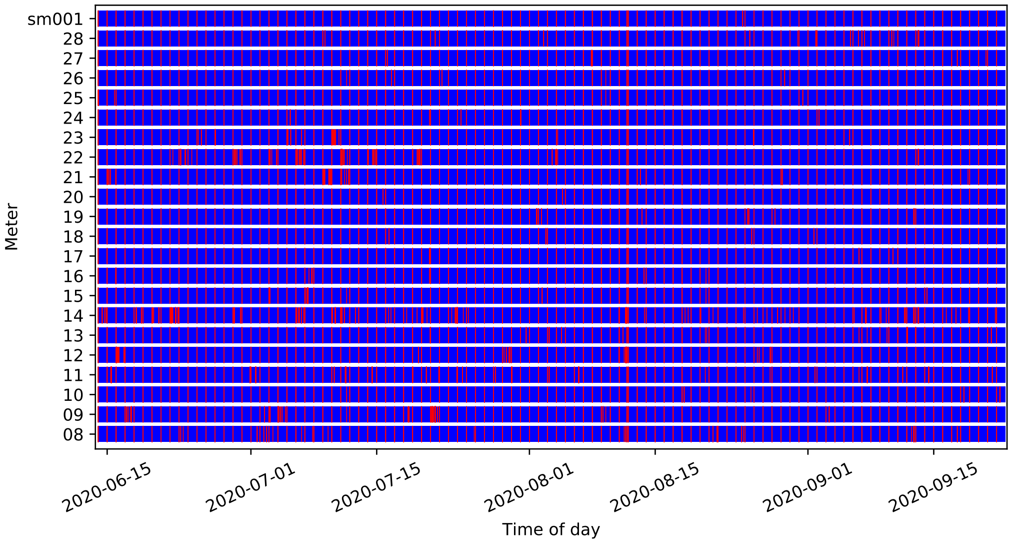


Figure 14. Time periods when measurement data are available. Each individual meter is shown on the y-axis (ID 08-28) while the time period of the complete recording is shown on the x-axis. Red areas indicate gaps in the dataset. Please note that gaps are plotted with at least one pixel width for better visibility and therewith appear significantly larger.

3.3.4. Clock Synchronization

In Section 2, we defined one challenge as "simultaneous recordings with high temporal accuracy". Our framework is comprised of numerous meters and sensors distributed across the home. To maintain precise timestamps for the measured data and the event labels, precise clock synchronization techniques are required.

We also observed non-negligible clock drifts in initial experiments originating from clock inaccuracies of the used ADC and microcontroller which vary depending on the meter and also over time for the same meter due to temperature and aging effects. To remove these drifts, we equipped each metering device (aggregated meter and distributed meters) with an RTC to synchronize the internal ADC clock. A NTP server in the recording network is utilized to sync each microcontroller's system time every 120 s. We calculate the system time as the NTP response plus half the time it takes for the NTP request to be answered by the server. Therewith, we can obtain the time from a NTP request which took e.g., 10 ms with an accuracy of ± 5 ms. Hence, the accuracy of each device's system time depends on the network latency, which is typically below 10 ms. As we can measure the NTP response time, which is a good indication for the network latency, we only utilize NTP requests with response times better than ± 10 ms to sync the system time. If we detect a time drift after an NTP synchronization, it is slowly reduced by removing or adding samples. The used technique adds only a minor jitter to the data ($1/\textit{samplingrate}$ seconds or 125 µs for 8 kHz data).

Figure 7 highlights the achieved clock synchronization. The figure shows the voltage and current traces of PowerMeter15 and the SmartMeter. PowerMeter15 shows a rapid rise in the current consumption due to a heating element in the connected espresso machine. The corresponding increase can be also observed in the data of L_3 of the SmartMeter. Both signals are shifted by around 10 ms highlighting the achieved clock synchronization. Such a time shift allows for syncing the voltage and current waveforms with sub-cycle precision.

3.4. Data and Code Availability

The FIRED dataset is available under the creative common license from our servers. The code which has been used to generate all plots in this work is provided as open source. The code and further information on how to download the data can be found at: https://github.com/voelkerb/FIRED_dataset_helper.

4. Discussion

In this work, we proposed a set of challenges which need to be addressed to record datasets which can be used to evaluate a wide variety of electricity related algorithms (especially event-based NILM). These challenges are summarized in Table 1. We further proposed a framework to record and label datasets which meet the defined challenges. It is comprised of the required hardware and software components to record the data, an algorithm to automatically find and label events in the recorded data and a tool to visually inspect the data and adjust the labels.

Using the framework, we recorded and labeled the FIRED dataset which features 101 days of electricity measurements (C3) of a residential apartment in Germany. This is significantly longer than most existing high frequency datasets such as REDD or BLUED. Aggregated level data are available as 8 kHz voltage and current waveforms while individual appliance data are available at 2 kHz for 21 appliances (C1, C2). While the aggregated sampling rate is matched or even exceeded by other datasets, we are currently unaware of any other residential dataset which features high frequency individual appliance recordings. The data is further time synced with an accuracy of around 10 ms (C5) and shows a coverage of 99.96 % over the complete recording time period (C3). Other datasets such as REDD or UK-DALE show a significant amount of missing samples due to bad wireless communication. Our framework also provides a 1 Hz and 50 Hz summary with derived active, reactive, and apparent power measurements. All data is stored in Matroska multimedia containers (C6) with included metadata information such as timestamps and measurands. Additional *CSV* files are included in the dataset which provide information about the apartments lighting states, room temperature, and device operation states (C4). Event positions and state labels have been added for two weeks of the data in a semi-automatic way using the presented Annoticity labeling tool (C4). No other dataset known to us includes such information. The dataset itself and the tools to process it are provided as open source (C6).

5. Conclusions

In summary, this work offers three main contributions to the community of smart meter data analytics:

1. We defined a set of challenges that an electricity dataset needs to address so that it can be used to evaluate a large set of electricity and smart meter related algorithms such as event-based and event-less Non-Intrusive Load Monitoring.
2. We proposed an expandable framework comprised of the hardware and software components required to record datasets that meet these challenges.
3. We introduced and evaluated a novel dataset to the community, which, compared to other residential electricity datasets such as BLUED, REDD or UK-DALE, features simultaneous high frequency recordings of the aggregated mains' signal and of individual household appliances as well as two weeks of fully labeled appliance events.

The high sampling rates, the achieved clock synchronization, and the marked events allow for using datasets that have been recorded using our framework (like FIRED) to evaluate event-based and event-less NILM algorithms. Additional data like detailed textual labels and additional sensor readings allow for developing disaggregation algorithms that utilize multi-modal information. Offering simultaneous, high frequency aggregated and individual appliance recordings will allow researchers to develop hybrid load monitoring systems which use individual appliance recordings in a semi-supervised fashion to aid the laborious training process of supervised NILM systems.

Besides the mentioned advantages, the framework is currently optimized for use in the residential domain as, e.g., plug level meters and WiFi are being used. While it should be possible to move the overall concepts to the commercial or industrial domain, such specialized environments may require additional adaption. Besides temperature and humidity readings, other environmental information such as occupancy or light sensors may be of interest and are theoretically supported by the framework but have not been installed while generating the FIRED dataset. Further increasing the sampling rate of the meters or the sheer number of meters is theoretically possible; however, in our findings, it reduced the reliability of our framework mainly due to bandwidth problems. One suggestion to overcome such a limitation in the future is to compress the data before it is sent over the bandwidth limited WiFi channel.

We provide electricity datasets and the software and hardware to record these so that researchers can set their focus on improving load monitoring and eco-feedback techniques. These have shown tremendous potential in saving our earth's energy resources.

Author Contributions: Conceptualization, B.V.; Writing—review and editing, M.P.; Writing—review and editing, P.M.S.; Supervision, B.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository that does not issue DOIs.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ILM	Intrusive Load Monitoring
NILM	Non-Intrusive Load Monitoring
RTC	Real Time Clock
FIRED	Fully-labeled hIgh-fRequency Electricity Disaggregation
CSV	Comma-Separated Values
SRT	SubRip Text
RMSE	Root-Mean-Square Error
TP	True Positives
FP	False Positives
FN	False Negatives
NTP	Network Time Protocol

References

1. Nations, U. About the Sustainable Development Goals—United Nations Sustainable Development. Available online: https://www.un.org/sustainabledevelopment/sustainable-development-goals (accessed on 29 August 2020).
2. Ehrhardt-Martinez, K.; Donnelly, K.A.; Laitner, S. *Advanced Metering Initiatives and Residential Feedback Programs: A Meta-Review for Household Electricity-Saving Opportunities*; American Council for an Energy-Efficient Economy: Washington, DC, USA, 2010.
3. Bundesministerium für Wirtschaft und Energie (BMWi). Gesetz zur Digitalisierung der Energiewende. 2016. Available online: https://www.bmwi.de/Redaktion/DE/Downloads/Gesetz/gesetz-zur-digitalisierung-der-energiewende.pdf (accessed on 29 August 2020).
4. Hart, G.W. Nonintrusive appliance load monitoring. *Proc. IEEE* **1992**, *80*, 1870–1891. [CrossRef]
5. Norford, L.K.; Leeb, S.B. Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms. *Energy Build.* **1996**, *24*, 51–64. [CrossRef]

6. Bundesministerium für Wirtschaft und Energie (BMWi). Information on the Funding Program Entitled 'Smart Energy Showcases—Digital Agenda for the Energy Transition' (SINTEG). 2016. Available online: https://www.bmwi.de/Redaktion/EN/Downloads/bmwi-papier-sinteg-kernbotschaften.pdf?__blob=publicationFile&v=3 (accessed on 29 August 2020).
7. Armel, K.C.; Gupta, A.; Shrimali, G.; Albert, A. Is disaggregation the holy grail of energy efficiency? The case of electricity. *Energy Policy* **2013**, *52*, 213–234. [CrossRef]
8. Hsu, C.Y.; Zeitoun, A.; Lee, G.H.; Katabi, D.; Jaakkola, T. Self-supervised learning of appliance usage. In Proceedings of the 8th International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.
9. Kolter, J.Z.; Johnson, M.J. REDD: A public data set for energy disaggregation research. In Proceedings of the Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA, USA, 21–24 August 2011; Volume 25, pp. 59–62.
10. Batra, N.; Kelly, J.; Parson, O.; Dutta, H.; Knottenbelt, W.; Rogers, A.; Singh, A.; Srivastava, M. NILMTK: An open source toolkit for non-intrusive load monitoring. In Proceedings of the 5th International Conference on Future Energy Systems, Cambridge, UK, 11–13 June 2014; pp. 265–276.
11. Kelly, J.; Knottenbelt, W. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Sci. Data* **2015**, *2*, 150007. [CrossRef]
12. Kriechbaumer, T.; Jacobsen, H.A. BLOND, a building-level office environment dataset of typical electrical appliances. *Sci. Data* **2018**, *5*, 180048. [CrossRef] [PubMed]
13. Anderson, K.; Ocneanu, A.; Benitez, D.; Carlson, D.; Rowe, A.; Berges, M. BLUED: A fully labeled public dataset for event-based non-intrusive load monitoring research. In Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD), Beijing, China, 8 December 2012; Volume 7.
14. Belley, C.; Gaboury, S.; Bouchard, B.; Bouzouane, A. An efficient and inexpensive method for activity recognition within a smart home based on load signatures of appliances. *Pervasive Mob. Comput.* **2014**, *12*, 58–78. [CrossRef]
15. Alcalá, J.M.; Ureña, J.; Hernández, Á.; Gualda, D. Assessing human activity in elderly people using non-intrusive load monitoring. *Sensors* **2017**, *17*, 351. [CrossRef] [PubMed]
16. Völker, B.; Pfeifer, M.; Scholl, P.M.; Becker, B. Annoticity: A Smart Annotation Tool and Data Browser for Electricity Datasets. In Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring, Yokohama, Japan, 18 November 2020; pp. 1–5.
17. Völker, B.; Pfeifer, M.; Scholl, P.M.; Becker, B. FIRED: A Fully-labeled hIgh-fRequency Electricity Disaggregation Dataset. In Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, Yokohama, Japan, 19–20 November 2020; pp. 294–297.
18. Beckel, C.; Kleiminger, W.; Cicchetti, R.; Staake, T.; Santini, S. The ECO data set and the performance of non-intrusive load monitoring algorithms. In Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings, Memphis, TN, USA, 4–6 November 2014; pp. 80–89.
19. Makonin, S.; Popowich, F.; Bartram, L.; Gill, B.; Bajić, I.V. AMPds: A public dataset for load disaggregation and eco-feedback research. In Proceedings of the 2013 IEEE Electrical Power & Energy Conference, Halifax, NS, Canada, 21–23 August 2013; pp. 1–6.
20. Makonin, S.; Wang, Z.J.; Tumpach, C. RAE: The rainforest automation energy dataset for smart grid meter data analysis. *Data* **2018**, *3*, 8. [CrossRef]
21. Pereira, L. Developing and evaluating a probabilistic event detector for non-intrusive load monitoring. In Proceedings of the 2017 Sustainable Internet and ICT for Sustainability (SustainIT), Funchal, Portugal, 6–7 December 2017; pp. 1–10.
22. Völker, B.; Scholls, P.M.; Schubert, T.; Becker, B. Towards the Fusion of Intrusive and Non-intrusive Load Monitoring: A Hybrid Approach. In Proceedings of the Ninth International Conference on Future Energy Systems, Karlsruhe, Germany, 12–15 June 2018; pp. 436–438.
23. Völker, B.; Scholl, P.M.; Becker, B. Semi-Automatic Generation and Labeling of Training Data for Non-Intrusive Load Monitoring. In Proceedings of the Tenth International Conference on Future Energy Systems, Phoenix, AZ, USA, 25–28 June 2019.
24. Völker, B.; Pfeifer, M.; Scholl, P.M.; Becker, B. A Versatile High Frequency Electricity Monitoring Framework for Our Future Connected Home. In Proceedings of the International Conference on Sustainable Energy for Smart Cities, Braga, Portugal, 4–6 December 2019; pp. 221–231.
25. Analog Devices. *ADE9000—High Performance, Multiphase Energy, and Power Quality Monitoring IC*; ; Analog Devices: Norwood, MA, USA, 2017; Rev. A.
26. STMicroelectronics. *STPM32, STPM33, STPM3—ASSP for Metering Applications with up to Four Independent 24-bit 2nd Order Sigma-Delta ADCs, 4 MHz OSF and 2 Embedded PGLNA*; STMicroelectronics: Geneva, Switzerland, 2016; Rev. 5.
27. OASIS. The MQTT 5.0 standard—A Machine-to-Machine (M2M) "Internet of Things" Connectivity Protocol. 2020. Available online: https://www.mqtt.org/ (accessed on 29 August 2020).
28. Pereira, L.; Nunes, N.J. Semi-automatic labeling for public non-intrusive load monitoring datasets. In Proceedings of the Sustainable Internet and ICT for Sustainability (SustainIT), Madrid, Spain, 14–15 April 2015; pp. 1–4.
29. Matroska, N.P.O. The Matroska File Format. 2020. Available online: https://www.matroska.org/ (accessed on 29 August 2020).
30. Scholl, P.M.; Völker, B.; Becker, B.; Van Laerhoven, K. A multi-media exchange format for time-series dataset curation. In *Human Activity Sensing*; Springer: Berlin, Germany, 2019; pp. 111–119.
31. WavPack. Hybrid Lossless Audio Compression. 2020. Available online: http://www.wavpack.com (accessed on 29 August 2020).

32. Group, T.H. THE HDF5 LIBRARY & FILE FORMAT. 2020. Available online: https://www.hdfgroup.org/solutions/hdf5/ (accessed on 11 December 2020).
33. co2online. Der Stromspiegel für Deutschland 2019. 2020. Available online: https://www.stromspiegel.de/stromverbrauch-verstehen/stromverbrauch-3-personen-haushalt/ (accessed on 29 August 2020).
34. Ag, C.E. Voltcraft Energy Logger 4000. 2020. Available online: http://www.voltcraft.ch/index.html (accessed on 29 August 2020).