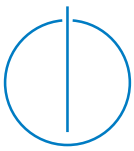
Daniel Rene Jorde

Learning from Power: Machine Learning on Electrical Signals

Technische
Universität
München



Technische Universität München

Fakultät für Informatik

Lehrstuhl für Anwendungs- und Middlewaresysteme

Learning from Power: Machine Learning on Electrical Signals

Daniel Rene Jorde

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: apl. Prof. Dr. Georg Groh

Prüfer der Dissertation:

1. Prof. Dr. Hans-Arno Jacobsen
2. Prof. Dr. Alexander Horsch

Die Dissertation wurde am 29.10.2020 bei der Technische Universität München eingereicht und durch die Fakultät für Informatik am 25.02.2021 angenommen.

Ad Astra

Abstract

Electricity is everywhere. It powers devices ranging from small appliances to large industrial machines. Seizing electrical signals to monitor the behaviour of electrical consumers delivers insights that various applications, such as, for example, disaggregation of energy measurements and industrial condition monitoring, rely on. Non-intrusive load monitoring (NILM) techniques are one means for extracting device-level information from electrical signals, without intrusively attaching sensors to each individual device. In this dissertation, we contribute new advancements to the steps of the NILM analysis pipeline: **data acquisition**, **event detection**, and **appliance identification**.

Data acquisition: We introduce the first extensive publicly-available dataset for NILM-based condition monitoring and analysis of industrial components in the field, namely, the coffeemaker electrical activity measurements (CREAM) dataset. More particularly, this dataset is the first one to include comprehensive ground-truth information and high-sampling-rate electrical signals of industrial electrical components that are actively working together to create products following a dedicated manufacturing process.

Event detection: Furthermore, we introduce a new multi-environment event detector (MEED) for high-sampling-rate electrical signals. MEED improves the current state of the art while being trained fully unsupervised. Thus, the algorithm does not require manual adaption when being used in new environments. In addition, we provide an extensive categorisation of the existing state of the art in event detection for NILM. We identify research gaps based on this review and conduct a benchmark of our MEED approach and four re-implemented state-of-the-art algorithms, showing that MEED achieves the highest precision and recall on the BLUED and the BLOND dataset.

Appliance identification: The subsequent appliance identification step relies on previously detected events. Appliance identification algorithms use hand-crafted features so far. Different appliances have different features that represent them best, making the feature engineering highly dependent on the respective appliance composition. We overcome this manual effort by introducing a new approach that applies a deep convolutional neural network to extract features from the raw signals. By evaluating the algorithm on the WHITED and PLAID dataset, we show that it achieves F1-Scores of 1 and 0.69 respectively.

Zusammenfassung

Elektrizität ist überall. Sie treibt Geräte von kleinen Haushaltsgeräten bis hin zu großen industriellen Maschinen an. Die Nutzung von elektrischen Signalen zur Überwachung des Verhaltens von elektrischen Verbrauchern liefert Einblicke auf denen verschiedenste Anwendungen basieren, wie zum Beispiel, die Disaggregation von Energiemessungen und industrielle Zustandsüberwachung. Non-intrusive load monitoring (NILM) Techniken sind ein Mittel, um gerätespezifische Informationen aus elektrischen Signalen zu extrahieren, ohne dabei invasiv Sensoren an den einzelnen Verbrauchern anzubringen. In dieser Dissertation tragen wir neue Entwicklungen zu den Schritten der NILM-Analysepipeline bei: **Datanaquise**, **Eventerkennung** und **Geräteidentifikation**.

Datanaquise: Wir stellen den ersten ausführlichen, öffentlich verfügbaren Datensatz für NILM basierte Zustandsüberwachung und Auswertungen auf dem Gebiet vor, den coffeemaker electrical activity measurements (CREAM) Datensatz. Insbesondere ist dieser Datensatz der Erste, der umfassende Referenzdaten und hochaufgelöste elektrische Signale von industriellen, elektrischen Komponenten enthält, die aktiv in einem dedizierten Fertigungsprozess zusammenarbeiten, um mehrere Produkte zu erstellen.

Eventerkennung: Darüber hinaus stellen wir einen neuen, multi-umgebungs Eventerkennungsalgorithmus (MEED) für hochaufgelöste elektrische Signale vor. MEED verbessert den aktuellen Stand der Technik, obwohl er vollständig unüberwacht trainiert wird. Daher benötigt der Algorithmus keine manuellen Anpassungen, wenn er in neuen Umgebungen verwendet wird. Außerdem stellen wir eine umfangreiche Kategorisierung des existierenden Standes der Technik in der Eventerkennung für NILM zur Verfügung. Wir identifizieren dabei, basierend auf diesem Überblick, Forschungslücken und führen einen Benchmark unseres MEED Ansatzes mit vier neu implementierten Algorithmen des aktuellen Standes der Technik durch und zeigen, dass MEED die höchsten Precision und Recall Werte auf den Datensätzen, BLUED und BLOND, erreicht.

Geräteidentifikation: Der nachfolgende Schritt der Geräteidentifikation stützt sich auf die zuvor erkannten Events. Algorithmen zur Geräteidentifikation verwenden traditionell händisch erarbeitete Charakteristiken. Verschiedene Geräte haben verschiedene Eigenschaften durch die sie am besten repräsentiert werden, was die Entwicklung der Eigenschaften stark abhängig von der entsprechenden Gerätezusammenstellung macht.

Wir überwinden diesen manuellen Aufwand durch die Einführung eines neuen Ansatzes, der ein tiefes, faltendes (convolutional) neuronales Netz verwendet, um Eigenschaften aus den Rohsignalen zu extrahieren. Anhand der Auswertung des Algorithmus auf dem WHITED und dem PLAID Datensatz zeigen wir, dass er entsprechende F1-Werte von 1 und $0,69$ erreicht.

Acknowledgments

All work contributing to this dissertation took place at the Department of Informatics of the Technische Universität München under the supervision of Prof. Dr. Hans-Arno Jacobsen.

First, I would like to thank Prof. Dr. Hans-Arno Jacobsen for supporting me throughout my journey to this dissertation with his encouragement, his valuable feedback, his vast experience, and for providing everything that allowed me to pursue my research.

Furthermore, I would like to thank Prof. Dr. Alexander Horsch for all the interesting talks we had, his encouragement, and his agreement to be the second examiner of my dissertation. Also, I would like to thank Prof. Dr. Georg Groh for acting as chair of the committee and for inspiring me with his lecture on social computing.

I also want to thank all my colleagues at the chair. I am grateful for not only your feedback and support on research matters but also for the great time we had, especially when a table-football match with you pushed me back on track. In particular, a huge thank you to Thomas Kriechbaumer, who acquired the funding for my position and who, together with Christoph Doblander, continuously helped me with matters of all kinds. In addition, I want to especially thank Elias, Jan, Matthias, and Alexander for the great time we had.

I am grateful for all the support and the true friendship from by best friends, Benedikt and Tetsuya. They cheered me up whenever we met and talked, despite the distance.

I am especially indebted to my parents Andrea and Norman, as well as my brothers Marcel and Sammy, for always listening to me and backing me up throughout my entire life. You never stopped believing in me. Also, I want to thank my grandparents and my uncle Manfred for their continuous support.

Last but not least, I want to thank my wife, Ela. Thank you for all your love and understanding for my quirks and the long working hours. For your empathy and optimism, especially when I felt desperate. You are taking the burden from my shoulders by supporting me whenever needed. Without you, all this would never have been possible. I also want to thank my yet unborn daughter Mila. Thank you for showing me what is really important in life with your slight kicks in your mum's belly. I love the two of you.

Contents

1

Introduction

Electrical signals can be found everywhere, from consumer products to large industrial machinery. The specific inner workings and behaviour of appliances influence the electrical signals that power them [1, 2]. Thus, measuring these signals exhibits information about the components and appliances they are acquired from. We differentiate two paradigms for measuring the respective voltage and current signals, namely, intrusive load monitoring (ILM) and non-intrusive load monitoring (NILM). They represent the trade-off of either attaching a measurement device to every individual appliance and component or of measuring multiple ones at the same time in a non-intrusive way [1]. For obtaining the behaviour of individual appliances from the aggregate signal in the latter approach, intelligent algorithms are necessary to retrieve information from the signal.

The non-intrusive paradigm bears several advantages over ILM, such as, for example, a reduction in costs and maintenance, and the possibility to monitor appliances and components for which an individual sensor placement is unfeasible. Monitoring the electrical load of appliances and other electrical components enables various applications, such as energy breakdowns and condition monitoring of machinery [1, 3].

In this work, we focus on the NILM paradigm to monitor the electrical load of both consumer appliances and industrial components. NILM relies on the algorithms that are

used to disaggregate the aggregate sensor readings. This thesis identifies research gaps along the analysis pipeline of NILM and overcomes them subsequently by proposing new algorithms and data.

1.1 Motivation

Two trends and challenges raise the demand for the analysis of electrical consumer appliances in households and office environments on the one hand, and on industrial electrical equipment on the other hand.

First, the depletion of natural resources with the simultaneous rise in demand for (electrical) energy is one of today's main challenges [4]. Consequently, innovations to reduce the amount of energy consumed and to improve the use of limited resources are necessary. One way is to advise end-users, both residential and industrial ones, on how they can save electrical energy [5, 6]. This can be done by providing detailed energy breakdowns to identify anomalies and potential sources of energy waste [6]. These energy breakdowns and appliance anomaly detection applications can be realised by implementing respective measurement hardware and NILM algorithms. By non-intrusively measuring and disaggregating the power consumption of the entities of interest, such as, for example, residential households, one can provide per-appliance usage and energy demand information to raise end-user awareness [5, 6]. NILM was initially developed to provide per-appliance power usage information in residential environments by applying disaggregation algorithms but is increasingly adapted in other domains [7, 8, 9, 10, 11].

The second major trend that lays the basis for an application area beside the residential and office sector is the increasing adaption of cyber-physical systems as the backbone of new industrial developments [12]. Detailed information about the behaviour of electrical components lays the basis for implementing condition monitoring (CM) processes in the industry [13]. NILM techniques can provide such per-component information, without the need for deploying sensors to machinery in an intrusive way, as shown by the work of Suzuki et al. [11]. Besides, analysing industrial equipment with NILM algorithms enables

energy breakdowns that offer the potential to achieve energy savings in the industrial sector.

In conclusion, both developments, the depletion of natural resources with the rising energy demand, as well as the adoption of condition monitoring techniques in the industry motivate the further development of NILM algorithms. In particular, these algorithms contribute to reduce energy waste and to optimise the usage of industrial equipment without intrusively attaching sensors to individual appliances and components.

1.2 Problem Statement

Recent advances in machine learning, such as the advancements in several fields through the adoption of neural network-based algorithms, offer opportunities to further improve the state of the art in NILM along all steps of the analysis pipeline. We divided the NILM process, in particular, when being applied to high-sampling-rate data, into three steps that finally result in the disaggregated load, namely, data acquisition, event detection, and appliance identification, as shown in Figure 2.1.3.

Thus, the key problems and challenges of NILM that are covered in this thesis are described with respect to the step they belong to in the following. As NILM can be applied in various settings, the entities that are analysed can differ substantially from whole appliances to individual electrical components in industrial machinery. To facilitate readability and to follow the traditional naming of the field, we use the terms *appliance* and *component* interchangeably in the following to refer to the entity that is the target of the NILM analysis process.

Data acquisition NILM was initially developed to disaggregate load profiles from residential environments [7]. As a result, most of the published datasets are from this domain [8]. To further transfer NILM techniques into the domain of condition monitoring, the need for a dataset to benchmark such algorithms arises. Currently, there is no high-sampling-rate dataset containing industrial components, such as motors and pumps, that are triggered following a dedicated production process pattern.

Event detection In the field of NILM, various algorithms have been proposed to detect relevant signal segments and state-changes in the high-sampling-rate voltage and current waveform [14, 15, 16, 17, 18, 19]. The existing algorithms are often tailored to a specific setting with a fixed appliance composition. Consequently, the algorithms need to be manually adapted by human experts when being used in new environments, with a potentially dynamic appliance composition. Furthermore, there is no comprehensive overview of the existing approaches and no standard for evaluating the algorithmic performance, making the algorithms hard to compare.

Appliance identification Different appliances have unique fingerprints that are best suited to identify them [2, 20]. Traditional NILM approaches rely on manually derived features, that need to be fined tuned for specific appliance compositions and settings. In contrast, neural network-based representation approaches promise to automatically extract features from the raw data, allowing the algorithms to generalise better without human interference and manual fine-tuning [21]. Thus, the adaption of such algorithms can overcome the need for appliance specific feature engineering.

1.3 Approach

This dissertation presents multiple advancements along the complete analytical pipeline of NILM that have the potential to overcome the existing problems and challenges in the field.

Experimental results show, that NILM can be used for applications besides energy disaggregation, in particular, for monitoring industrial equipment [3, 11]. Most of the NILM algorithms have been developed with the purpose of disaggregating residential load profiles. In order to transfer these algorithms to an industrial setting for applications, such as condition monitoring, new datasets are necessary that contain industrial electrical components. The usage of these components has to follow the patterns of a dedicated production process. Thus, we introduce the coffeemaker electrical activity measurements for condition monitoring (CREAM) dataset. The dataset contains the fully-labelled ground-truth electrical signals of two industrial-grade coffeemakers with typical industrial

electrical components. The purpose of this publicly available dataset is to provide a baseline for benchmarking condition monitoring algorithms on electrical signals, as it includes over 370000 expert labelled electrical events and corresponding maintenance and production process labels.

The event detection step is fundamental to all subsequent actions in the NILM pipeline on high-sampling-rate data, as it determines which signal segments are further processed for appliance identification and other applications. As outlined before, the existing event detection algorithms for NILM are developed and evaluated for specific, often not publicly available environments [22]. In addition, the algorithms need to be tediously fine-tuned when being transferred to another setting [22]. Furthermore, it is hard to compare the wide variety of existing algorithms as they are often evaluated on non-public data and use unknown parameters in the evaluation. We propose an unsupervised multi-environment event detector (MEED) for NILM to overcome the first issue [23]. This event detector can identify relevant signal segments without human supervision in a fully unsupervised way. At its heart, MEED relies on a denoising autoencoder model with bidirectional long short-term memory (LSTM) layers for encoding and decoding the input. The only hyperparameter of MEED is a threshold on the mean-square-error (MSE) reconstruction error to determine events in the signal. This parameter is automatically set with respect to the error produced at the end of the training procedure. We have released all source code of MEED to facilitate reproducibility. In addition to introducing MEED, we have conducted an extensive literature review of the high-sampling-rate event detection algorithms for NILM to overcome the issues of reproducibility and comparability in the field [22]. Based on this review, we have re-implemented four of the state-of-the-art event detection algorithms and evaluated them against MEED on two publicly available datasets, showing the superiority of MEED in event detection for NILM. To make our approach reproducible, we published all source code, including the four re-implemented approaches, as open source.

Based on the detected events, one can determine the appliance that was responsible for the respective event by classifying the signal segments. Handcrafting features is a tedious task, that is dependent on the specific appliances that are used. The datasets in the field have a comparatively small number of samples per appliance type, making the classification task on high-sampling-rate data challenging, due to the curse-of-dimensionality. By

introducing a convolutional neural network (CNN) architecture, we show how the raw voltage and current waveform can be used to automatically perform the classification based on the representation the network extracts from the raw input [24].

1.4 Contribution

With the goals of improving central aspects of the NILM analytical pipeline on the one hand and of facilitating the usage of NILM algorithms in industrial environments for condition monitoring, on the other hand, this work presents new algorithms and a new dataset. In pursuit of achieving this, this dissertation includes the following main contributions to the three steps of the analysis pipeline:

1. We introduce the coffeemaker electrical activity measurements for condition monitoring (CREAM) dataset to overcome the lack of datasets to benchmark NILM algorithms on industrial components for advanced applications other than energy disaggregation. In particular, the dataset contains 370600 hand-labelled electrical events and fine-granular labels of manufactured products and maintenance actions taken for two industrial-grade coffeemakers. The coffeemakers were selected as they fulfil the requirements for a closed system in which industrial electrical components, such as, for example, pumps and motors, are triggered according to a pre-defined manufacturing process to produce various products. The dataset can be used to evaluate various algorithms, such as non-invasive condition monitoring algorithms, component classification techniques, and event detectors, on a comprehensive set of labels.

2. We present an unsupervised multi-environment event detector (MEED) that outperforms the existing state of the art in event detection on high-sampling-rate electrical signals. The algorithm is fully unsupervised and requires no human intervention when being used in multiple settings, such as residential and office environments. Our approach lowers the amount of missed and falsely identified events compared to the existing state of the art while generalising well between different environments.

3. We conduct an extensive review of the existing state of the art in event detection for high-sampling-rate NILM. By categorising the publications and distilling the evaluation criteria, we present an overview of algorithms that can be compared to each other. Based on this, we identify shortcomings in the evaluation of the existing algorithms and re-implement four state-of-the-art algorithms to evaluate them against our MEED algorithm. In addition, we make all source code publicly available, including the functions for evaluation, to overcome the issues of reproducibility and comparability in the field.

4. We consider a new approach for performing appliance classification on raw, high-dimensional voltage and current signals. The approach does not require hand-crafted, appliance specific feature engineering, as the CNN automatically extracts a suitable representation from the input. With the CNN based approach, we are able to achieve state-of-the-art results without dedicated feature engineering. In addition, we discuss possible data augmentation techniques to overcome the problems resulting from the small datasets in the NILM field that arise when training deep neural networks.

Parts of the content and contributions of this work have been accepted and published in:

- D. Jorde, T. Kriechbaumer, T. Berger, S. Zitzlsperger, and H.-A. Jacobsen. "CREAM, a component level coffeemaker electrical activity measurement dataset." In: *Scientific Data* (2020), accepted for publication on 15.10.2020

- D. Jorde, M. Kahl, and H.-A. Jacobsen. "MEED: An Unsupervised Multi-Environment Event Detector for Non-Intrusive Load Monitoring." In: *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm).* 2019, pp. 1–6. DOI: 10.1109/SmartGridComm.2019.8909729

- D. Jorde and H.-A. Jacobsen. "Event Detection for Energy Consumption Monitoring." In: *IEEE Transactions on Sustainable Computing* (2020), pp. 1–1. DOI: 10.1109/TSUSC.2020.3012066

- D. Jorde, T. Kriechbaumer, and H.-A. Jacobsen. “Electrical Appliance Classification using Deep Convolutional Neural Networks on High Frequency Current Measurements.” In: *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. 2018, pp. 1–6. DOI: 10.1109/SmartGridComm.2018.8587452

The latter paper on appliance identification [24] is based on the master thesis of the author of this dissertation. The master thesis was supervised by Prof. Dr. Hans-Arno Jacobsen and advised by Dr. Thomas Kriechbaumer and is entitled "Identification of Individual Electronic Appliances in High Frequency Energy Data using an Artificial Neural Network Approach". The thesis was submitted on the 12th of March 2018.

1.5 Organization

This dissertation is organised as follows. Chapter 2 presents the background to relevant topics and our methodology for improving the existing state of the art along the NILM analytical pipeline. Chapter 3 presents a short summary of the publications that this thesis comprises. In particular, we describe the main achievements of each paper and highlight the author’s contributions. We have attached the respective publications to this thesis in the Appendices A, B, C, and D. In Chapter 4, we discuss the results and our contributions to NILM in the larger context of the field. Chapter 5 concludes this thesis.

2

Methodology

This chapter gives an overview of the Non-Intrusive Load Monitoring (NILM) methodology, that is applied to disaggregate electrical signals in this thesis. In the first Section 2.1, we provide an overview of the analysis pipeline and the general setup of Non-Intrusive Load Monitoring techniques. In Section 2.2, we give an overview of data acquisition systems that are used to collect the data that is processed by the NILM algorithms. In particular, we describe the hardware used to collect the CREAM dataset in detail. In the subsequent Section 2.3, we introduce the datasets that are used in this thesis to evaluate the developed algorithms. The commonly used metrics to evaluate NILM algorithms are then described in Section 2.4. Most of the NILM algorithms rely on the computation of features from the raw voltage and current signals. Hence, we describe commonly used features in Section 2.5. We then concluded this chapter by giving an overview of machine learning methodologies applied to the electrical signals and features in Section 2.6.

2.1 Non-Intrusive Load Monitoring

When monitoring the electrical load of individual electrical components, or appliances, one can take two approaches. One can either intrusively attach sensors to each component to measure the power consumption, or one can use a single sensor to measure the

consumption of multiple components at the same time and apply intelligent algorithms to separate the aggregate signal into the individual component signals afterwards [7]. The amount of sensors used and the ease of monitoring the individual component's power consumption represents a trade-off. The intrusive placement of sensors introduces costs, such as for the acquisition and maintenance of the sensors, and is not feasible in some cases, due to the character and locations of individual components [1, 7]. The Non-Intrusive Load Monitoring methodology was introduced to reduce the number of sensors used by disaggregating the aggregate consumption of multiple components measured by a single sensor [1, 7].

The general setup for measuring the consumption of individual components with NILM is depicted in Figure 2.1.1. The metering hardware acquires the electrical signals from the aggregate entity, that consist of multiple, individual electrical components. Each of these components has unique characteristics that are reflected in their power consumption. NILM algorithms rely on these individual characteristics to disaggregate the aggregate signals into the individual component ones [1, 7]. By doing that, one can monitor multiple components with a single sensor.

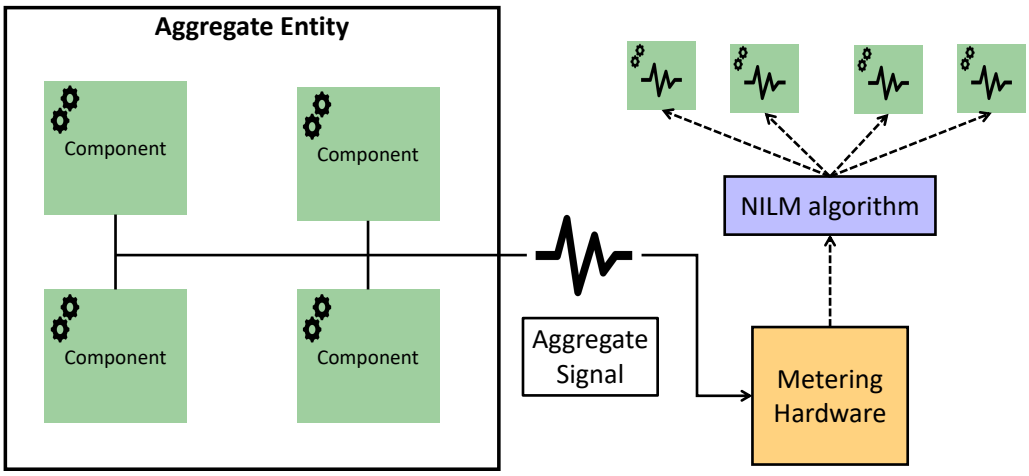


Figure 2.1.1: NILM metering setup

NILM was originally introduced by Hart [7] to monitor residential appliances based on the aggregate consumption of a residential home. Referring to the terminology introduced before in Figure 2.1.1, each component is an electrical appliance, and the aggregate entity is a residential home. The metering hardware is then installed at the electrical mains of the house to monitor the aggregate signal of the house. After applying the NILM algorithms, one obtains the detailed consumption information for each of the appliances, such as typical household appliances like fridges. By analysing the aggregate voltage and current recorded by a load monitor, Hart determines the turn-on and switch off moments of each appliance and their power consumption [7].

Other settings than the residential one, are, for example, industrial machinery or vessels. In the case of industrial machinery, the aggregate entity can be a complete machine consisting of electrical components, such as heaters, pumps, and motors.

The input to the NILM algorithms, an aggregate electrical signal, and the respective output, the signals of each component, is shown in Figure 2.1.2. In the case of the example in Figure 2.1.2, the overall apparent power consumption of three office devices, namely, a personal computer (PC), and two screens, are shown.

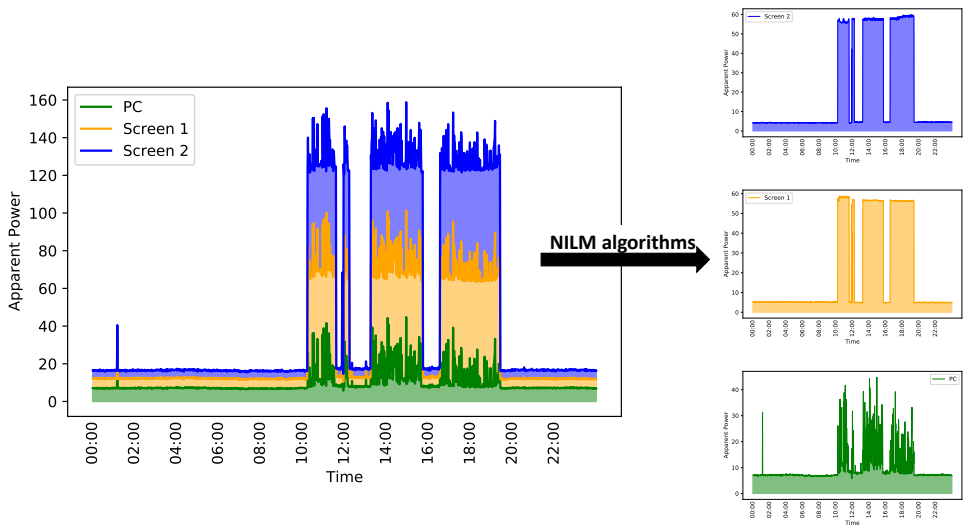


Figure 2.1.2: NILM disaggregation procedure

Hart also derived a load model that describes the underlying problem of NILM. The central idea behind this model is the parallel wiring of the components to be monitored. Consequently, the power the components consume is additive (to a first order approximation) [7], resulting in the following equation for the aggregate power consumption at time t [7]:

$$P(t) = \sum_{i=1}^{n} a_i(t)P_i + e(t) \tag{2.1.1}$$

The load of each component i when it is operating is modelled as a vector P_i. The error term $e(t)$ represents the existing background noise and other errors. The boolean vector $a(t)$ describes the state of the aggregate system at time t. When a component is activated at time t, the vector has an 1 entry at index i [7].

Finding the correct combination of components that results in the aggregate power consumption measured, is an NP-complete "weighted set problem" [7]. Thus, various heuristics and machine learning based algorithms have been designed to solve the problem and to determine the correct vector $a(t)$ of activated and inactivated components for a certain time t [1, 7].

The wide variety of NILM algorithms can be divided into two sets: state-based (i.e., non-event based) NILM algorithms and event-based ones [26]. State-based algorithms aim to disaggregate the electrical signal into the individual components directly. In contrast, event-based algorithms rely on separating relevant and irrelevant signal segments before determining the appliance status. Event-based algorithms detect state-changes of the components in the signal. Thus, the event-based NILM algorithms consist of multiple-steps: the detection of events, the identification of the component that is responsible for the event, and then the disaggregation of the aggregate signal [17]. The complete analysis pipeline is depicted in Figure 2.1.3.

The precondition of NILM algorithms is the data acquisition, that is usually done using smart meter devices or other dedicated hardware solutions for measuring the electrical signals. The two types of NILM algorithms are applied to data with different sampling rates,

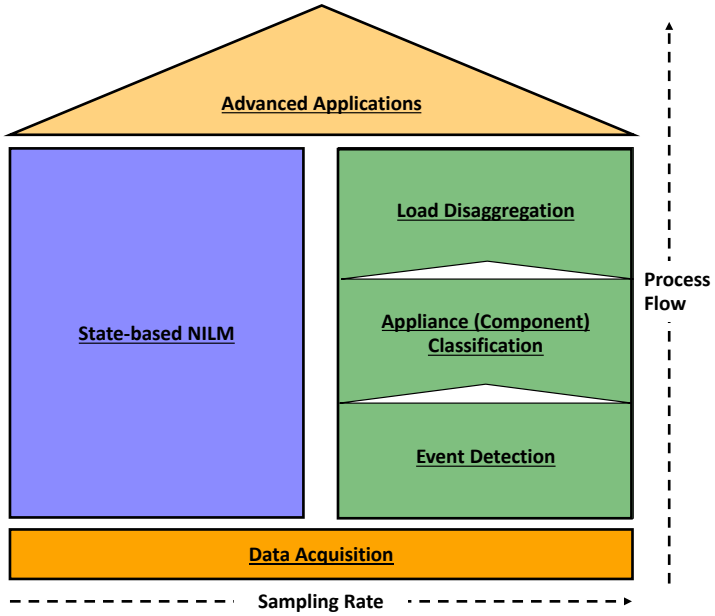


Figure 2.1.3: House of NILM

respectively, as most of the state-based algorithms become computationally intractable on higher sampling rates. Consequently, event-based algorithms are predominantly applied to high-sampling-rate data [26]. The sampling rate gives information on the number of samples-per-second (sps) the metering hardware collects.

A commonly used definition for low- and high-sampling-rates ρ is introduced by Liang et al. [27]:

$$\rho = \begin{cases} \text{low}, & \text{if} \quad \rho \leq 1 \quad \text{sps} \\ \text{high}, & \text{if} \quad \rho > 1 \quad \text{sps} \end{cases} \tag{2.1.2}$$

Based on the results of either the state-based or the event-based approaches, one can build other advances applications that rely on the disaggregated energy profiles of the components. Examples of such advanced applications are, for instance, activity monitoring of elderly people and condition monitoring for naval vessels [3, 28]. In the first example, Alcala et al. [28] use NILM algorithms to build a low-cost and scalable

activity monitoring system. A NILM algorithm determines the activity of people based on the appliances that are active at certain moments in time. By doing so, the authors are able to find deviations from normal in the monitored people's behaviour that can be used to improve health-care services [28]. In the second example, Lindahl et al. [3] show the usage of disaggregated electrical load profiles to perform fault detection in naval vessels by analysing the health condition of the respective machinery in a non-intrusive way [3].

In contrast to small sampling rates ($\rho \leq 1$), higher ones posses several advantages, such as the possibility to disaggregate more and even smaller components from the aggregate signal [6]. Furthermore, more features can be extracted from the high-sampling-rate data. In particular, frequency-based features, such as, for example, harmonics, need a minimum sampling rate to compute them. Due to these advantages and the gaps in the research on high-sampling-rate NILM, this thesis focuses on this domain. Consequently, we discuss event-based algorithms in more detail than state-based ones.

2.2 Data Acquisition

The data the NILM algorithms process can be acquired using a wide variety of electrical meters. Depending on their sampling rate capabilities, the meters can be classified as low-frequency energy meters or high-frequency meters. The features and signal characteristics that can be extracted from the electrical signals depend on the sampling rate. As an example, higher-order harmonics can only be extracted from high-sampling-rate signals full-filling the Nyquist-Shannon sampling criteria [1]. In a comprehensive study, ul Haq and Jacobsen [29] analysed the capabilities of off-the-shelf electrical meters. They show that more than 80% of the investigated hardware has a maximum sampling rate of 1 sps. Only a minority of the measurement devices provides sampling rates in the range of multiple thousands of samples-per-second [29]. High-sampling-rate data possess advantages over lower sampling rates, as it increases the capabilities for successful NILM in settings with multiple devices [6]. On the other hand, the acquisition of such high-sampling-rate poses various challenges, such as the storage of the data [8]. To solve the challenge of acquiring high-sampling-rate signals for appliances, Kriechbaumer et al. [30] designed the low-cost Mobile Energy Data Acquisition Laboratory (MEDAL). In

the following, we further detail the MEDAL hardware, as it measured the data for the CREAM dataset that is described in Appendix A. MEDAL is a cyber-physical system that allows for high-sampling-rate signal acquisition and on-device algorithm execution for preprocessing these signals [30]. The hardware consists of a six-socket power strip that is equipped with one voltage and six current sensors to monitor the individual sockets. In addition to the sensor hardware, MEDAL consists of an embedded-PC, a Raspberry pi-3, to process the collected data [30]. The overall setup of the MEDAL system is depicted in Figure 2.2.1.

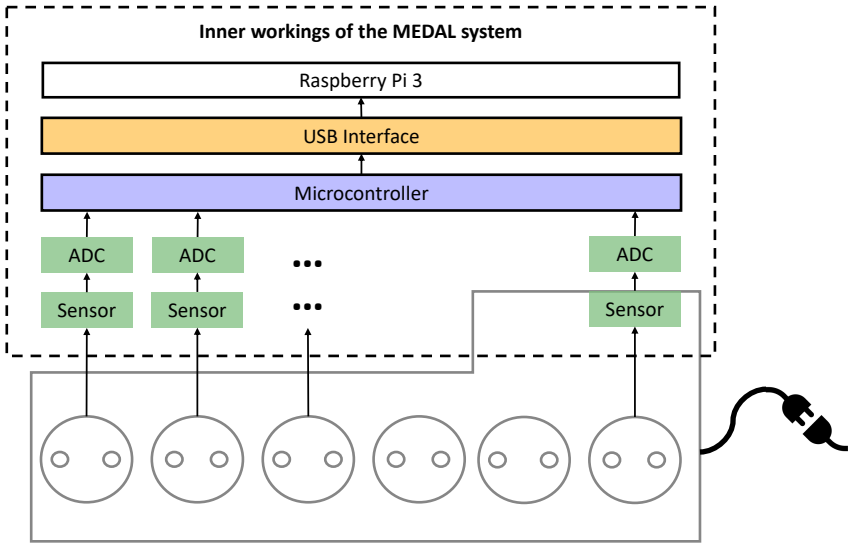


Figure 2.2.1: MEDAL case and inner system architecture

Each of the sockets of the MEDAL's power strip is connected to one current sensor and one analog-digital-converter (ADC), as shown in Figure 2.2.1. Furthermore, it contains an AC-AC transformer for sensing the voltage. This transformer also acts as a galvanic isolator and a step-down converter [30]. The sensors deployed for measuring the six current signals are Hall-effect-based sensors. The sensors and ADCs are controlled by a microcontroller that collects the data and forwards it to the single-board PC (raspberry pi 3) over a USB connection. As a result, the MEDAL system is capable of collecting the signals at a sampling rate of up to 50 kilo-samples-per-second (ksps). The single-board PC can then be used to perform various tasks, such as preprocessing the data, sending the data to a permanent cloud storage, and locally executing NILM algorithms [30].

2.3 Datasets

There are various publicly available datasets from different research domains that are commonly used to evaluate NILM-based algorithms. Most of the datasets have been collected with a specific purpose, such as energy disaggregation, occupancy detection, demand prediction, and anomaly detection [31]. In the following, we describe the datasets that are used in this thesis. As the focus is on event-based NILM algorithms, the following datasets are all high-sampling-rate ones.

WHITED The Worldwide Household and Industry Transient Energy Dataset (WHITED) [32] contains isolated measurements of start-up moments of various appliance types. The appliances were mostly recorded in multiple residential homes spread around the world. For each of the 110 appliances, multiple 5 second samples were recorded using a custom-designed low-cost sound card meter [32]. The sound card meter sampled WHITED at a rate of 44 ksps with a 16 bit resolution ADC.

PLAID The Plug-Level Appliance Identification Dataset (PLAID) [33] is available in multiple-versions. The original version of the datasets contains the start-up moments of 11 appliance classes, resulting in 1049 measurements. The dataset is sampled at 30 ksps. The appliances are residential ones, measured in 56 households in the USA [33].

BLUED In contrast to WHITED and PLAID that contain isolated appliance measurements, the Building Level fUlly-labeled dataset for Electricity Disaggregation (BLUED) [34] contains the aggregate power consumption of a US American home for one week. The signals of the two phases of the house are sampled at 12 ksps [17]. The aggregate measurements are complemented with turn-on and switch-off events for all appliances. These events were recorded with plug-level power meters and light-intensity sensors near overhead lights. The two phases, phase A and B, contain 872 and 1548 events respectively [17]. Looking at the appliances connected to each phase, one can see that the ones that are more complicated to disaggregate are connected to phase B. As the ground-truth for this high-sampling-rate dataset is the most comprehensive one that is publicly available, the BLUED dataset became the de-facto standard for evaluating event detection algorithms for NILM.

BLOND The previously mentioned datasets are all recorded in residential environments. The creators of the Building-Level Office eNvironment Dataset (BLOND) intended to overcome this by making a high-sampling-rate, long-term dataset from an office environment publicly available [8]. The dataset contains aggregate measurements of all three phases of a German office space. Furthermore, the authors used the previously described MEDAL [30] system to acquire the plug-level ground truth signals for every appliance in the office space [8]. The overall BLOND dataset is a composite of two parts: one dataset with the aggregate data being sampled at 50 ksps and another part with it being sampled at 250 ksps [8].

Looking at the existing datasets, one can see that there is no high-sampling-rate dataset with industrial components to benchmark NILM-based algorithms. Hence, we have created the CREAM dataset that is described in Appendix A to overcome this shortcoming.

2.4 Performance Metrics

When evaluating event-based NILM algorithms, several metrics are commonly used. Most of them are based on the values of the confusion matrix between ground-truth and predicted values [17, 35]. The confusion matrix contains the counts of samples that are True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). Based on these scores, one can compute the following metrics, that are also commonly used in other machine learning application areas:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (2.4.1)$$
$$\text{FPP} = \frac{\text{FP}}{\text{TP} + \text{FN}} \quad \text{F1-Score} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

For the evaluation of event detection algorithms, it is necessary to define the determination of the confusion matrix scores precisely. In the related literature, researchers applied different procedures to compute the respective scores [17, 35, 36]. As events can spread

over a longer time frame than a single point in time, a certain tolerance in time is required to determine correctly classified events. Furthermore, the tolerance limit τ is required as the event labels in commonly used datasets can be imprecise, due to their often manual generation. Thus, for event detection algorithms, a true-positive event is a detected event d that lies in temporal proximity of a ground truth event g, such that:

$$\exists g \; : \; d - \tau \leq g \leq d + \tau \tag{2.4.2}$$

When evaluating appliance (component) classification algorithms with multiple classes, the metrics for the overall performance for the classes in the dataset is computed by taking the unweighted average of all per-class scores [37]. In particular, each metric is calculated for every class individually first, before averaging them. In case there is a large class imbalance or if the misclassification of one class is more severe then the one of others, one can weight the individual scores with a factor before averaging them.

2.5 Electrical Features

Based on the electrical signals measured, various characteristics, i.e., features, can be computed [2]. Different appliances produce different fingerprints when they are activated. The various features that are proposed in the literature capture these fingerprints and allow algorithms to distinguish between the appliances or to detect events [2]. Careful feature selection eases NILM related tasks, such as, for instance, appliance classification. In Figure 2.5.1, two sets of features for multiple instances of three appliances from the BLUED [34] dataset are shown. As it can be seen visually, the appliances are clearly separated from one another based on the feature set in the right plot, whereas the characteristics are overlapping when using a different feature set in the left plot.

As NILM research was and still is mainly focused on consumer appliances, researchers divided these appliances into four categories according to common characteristics [1]

The appliances of the first type have two states of operation. They can either be turned-on or switched-off and do not possess any intermediary statuses [1].

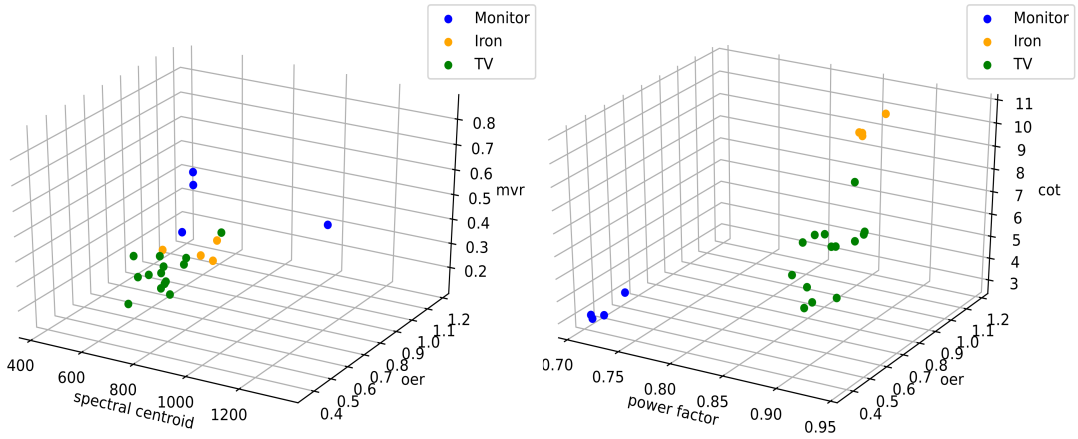


Figure 2.5.1: Different feature sets for appliance (component) classification

The second category consists of appliances that do not only have two states but a finite number of states they can be operated in. The state-transitions of these appliances follow regular patterns that can be seized by NILM algorithms. [1].

In contrast to the second category, appliances of the third category have an infinite number of states. Thus, they are named continuously variable devices. Their power characteristics do not exhibit repeated patterns [1].

Appliances that are continuously operating, i.e., that are active over long periods, are assigned to the fourth category [1].

The different appliance categories can be seized to design NILM approaches, such as hierarchical classification algorithms. Similar to the (consumer) appliances, it is also common in the NILM research field to categorise features into three sets, namely, transient features, steady-state features, and non-traditional features [1, 2]. Transient features are extracted from signal areas that belong to transitions between states of the appliances, in particular when they are turned on. In contrast, steady-state features describe the stable state of operation of an appliance. In addition, other features, besides steady-state and transient ones, are used to improve NILM algorithms. These features belong to the non-traditional feature category [1].

When extracting features, especially transient ones, this is done based on a fundamental assumption on appliance behaviour that was introduced by Hart in his seminal paper on NILM, namely, the Switching-Continuity-Principle (SCP) [7, 38]. This principle states that, in a small time interval, only up to one component (appliance) changes its state. Depending on the setting and the appliance composition, the SCP should only be rarely violated [38]. Based on the SCP, one can compute an appliance fingerprint, in particular, based on the state-transitions, that belongs to only one individual appliance.

The three-dimensional fingerprints in Figure 2.5.1 are computed based on the first two seconds after the respective appliances were turned on. In order to give an overview of potential features for NILM, the ones in Figure 2.5.1 are further described in the following. Experiments by Kahl et al. [2] have shown, that a combination of features from the time-domain and ones from the frequency-domain perform best. All features are derived from the raw voltage (U) and current (I) signal. The power factor feature is the ratio between the real (P) and apparent ($|S|$) power of the signal segment of interest, with the rms being the root-mean-square value of the respective physical quantity and ϕ being the phase angle between I and U [1, 2].The feature is computed for a certain region-of-interest (roi) with n measurement samples.

$$\text{rms}(I_{roi}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} I_i^2} \tag{2.5.1}$$

$$\text{rms}(U_{roi}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} U_i^2} \tag{2.5.2}$$

$$P = \text{rms}(I_{roi}) \times \text{rms}(U_{roi}) \times cos(\phi) \tag{2.5.3}$$

$$S = \text{rms}(I_{roi}) \times \text{rms}(U_{roi}) \tag{2.5.4}$$

$$Q = \text{rms}(I_{roi}) \times \text{rms}(U_{roi}) \times sin(\phi) \tag{2.5.5}$$

$$\text{power factor} = \frac{P}{|S|} \tag{2.5.6}$$

Various scalar quantities can be used to capture the current waveform, such as, for example, the mean-variance-ratio (mvr) feature. This feature is computed based on the absolute of the current signal in the region-of-interest.

$$\text{mean}(|I_{roi}|) = \frac{1}{n}\sum_{i=1}^{n}|I_i| \tag{2.5.7}$$

$$\text{var}(|I_{roi}|) = \frac{1}{n}\sum_{i=1}^{n}(|I_i| - \text{mean}(|I_{roi}|))^2 \tag{2.5.8}$$

$$\text{mvr} = \frac{\text{mean}(|I_{roi}|)}{\text{var}(|I_{roi}|)} \tag{2.5.9}$$

Another feature that is multi-dimensional in contrast to the previous ones is the current-over-time (cot) vector, with every element being the rms of the i'th period of the signal [2].

In addition to the features that capture the signal waveform in the time domain, frequency-based features, such as the spectral centroid and the odd-even-harmonics ratio (oer), are commonly used [2, 20].

$$\text{spectral centroid} = \frac{\sum_{f\in f_{bins}} x_f \times f}{\sum_{f\in f_{bins}} x_f} \tag{2.5.10}$$

$$\text{oer} = \frac{mean(x_{f_1}, x_{f_3}, ..., x_{f_{19}})}{mean(x_{f_2}, x_{f_4}, ..., x_{f_{20}})} \tag{2.5.11}$$

Both features are based on the results returned by a discrete Fourier Transformation of the current signal, with x_f being the magnitude and f being the frequency of the respective bin of the discrete analysis [2].

2.6 Machine Learning

Machine learning algorithms are Artificial Intelligence (AI) algorithms that extract knowledge from patterns in data [21].

Modern machine learning approaches have levelled up the performance of NILM algorithms [39]. Dependent on the kind of experience machine learning algorithms use

during learning, one can classify them into three categories: supervised- , unsupervised-, and semi-supervised learning algorithms [21]. Supervised machine learning techniques rely on datasets that contain labels for each sample in the dataset. The learning algorithm uses these labels to differentiate between the individual classes in the datasets [21]. In contrast to supervised learning algorithms, unsupervised techniques do not rely on labelled data. Semi-supervised learning algorithms are a hybrid form of both, supervised and unsupervised techniques, and rely on partially labelled data. In the NILM research field, a different definition of supervised- and unsupervised-learning is frequently used [38, 40]. Unsupervised NILM algorithms can be trained in a supervised-way, in the sense of the machine learning definition with labelled training data. The unsupervised nature of these algorithms does not refer to the use of labelled training data but refers to the non-availability of prior knowledge of the appliances or components in the setting of interest [38, 40]. In particular, general models of the existing appliances and components are transferred to an unknown setting. Using the three learning categories from machine learning, unsupervised NILM algorithms are semi-supervised machine learning algorithms [38]. Hence, one has to carefully look at the definition of the term unsupervised that is used in the respective publications. In this thesis, we use the classic machine learning definition.

The representation of the data, i.e., the electrical signals, influences the performance of machine learning algorithms. Good representations of the information in the data facilitate the learning task [21]. The same dataset can be represented using various feature combinations, as it is shown in Figure 2.5.1. Instead of manually handcrafting features to represent the data, one can also use machine learning algorithms to extract a good representation from the data automatically. In particular, deep learning techniques are capable of building complex representations from simpler concepts that are distilled from the data [21]. Several challenges in machine learning have motivated the usage of deep neural networks (NN), such as, for example, the curse-of-dimensionality. This problem refers to the circumstance that many problems become more difficult when the data is high-dimensional [21]. The high-frequency electrical signals used in this thesis to perform NILM related analysis tasks are high-dimensional and pose several challenges, such as an increase in computational complexity, to the machine learning algorithms. The application of NN to NILM problems is promising and increasingly adapted [39, 41, 42]. Commonly used NN architectures and building blocks are fully-connected neural

networks (FCNN), convolutional neural networks (CNN), and recurrent neural networks (RNN) [21]. These particular building blocks are used in different variations in this thesis; thus, they are briefly described in the following.

FCNN The fully-connected feedforward type of neural network aims to approximate a function f^*. In the case of a classifier, for example for appliance classification, the FCNN learns the parameters Θ of the mapping function $y = f(x; \Theta)$. In particular, the model selects the parameters during learning that match the true, underlying function best [21]. In the feedforward FCNN, the information passes without feedback connections from the input to the output of the network [21].

CNN Convolutional neural networks are designed to process data with a grid-like topology [21]. Time series data, such as electrical signals, can be interpreted as a one-dimensional grid with samples at regular (time) intervals [21]. The network relies on the mathematical convolution operation. Convolutional neural networks often apply pooling layers to learn hierarchical representations of the input. In contrast to traditional, plain FCNNs, CNNs rely on parameter sharing to reduce the number of parameters [21].

RNN Recurrent neural networks are designed to process sequential data [21]. Hence, they are well suited to process the raw electrical signal time-series data. Through parameter sharing over the time indices, RNNs are able to process variable-length sequences. For every (time) step in the sequence, RNNs produce an output. This output is then combined with the input of the next time step. A special type of RNNs is the long short-term memory (LSTM) cell network that relies on multiple gate mechanisms to capture long-term dependencies from the sequential input [21].

In the next chapter, we summarise our NILM-related publications, showing how we contribute new advancements to the field of NILM by using the previously described neural network architectures.

3

Summary of Publications

In this chapter, we summarise the individual contributions of this publication-based dissertation. In particular, we provide the key ideas and achievements and the author's contribution to each of the four accepted peer-reviewed publications.
The following sections are ordered bottom-up with respect to the analysis pipeline of NILM. First, we describe the CREAM dataset for condition monitoring in Section 3.1. Subsequently, we introduce our event detector MEED in Section 3.2. This is followed by our extensive literature review and algorithmic benchmark of the state of the art in event detection for NILM in Section 3.3. Finally, we conclude this chapter by providing the details on our appliance identification approach in Section 3.4.

3.1 CREAM, a component level coffeemaker electrical activity measurement dataset

Reference: D. Jorde, T. Kriechbaumer, T. Berger, S. Zitzlsperger, and H.-A. Jacobsen. "CREAM, a component level coffeemaker electrical activity measurement dataset." In: *Scientific Data* (2020), accepted for publication on 15.10.2020

Full-text version enclosed: Appendix A

Summary: Non-intrusive condition monitoring delivers insights into the internal states of industrial machinery. By analysing electrical signals, non-intrusive load monitoring techniques can be used to derive the conditions of electrical components.

We introduce the first publicly available dataset for analysing the electrical signals of industrial electrical components that follow a variety of pre-defined processes to output various products. Based on these requirements, we have selected two industrial-grade coffeemakers, as they resemble a closed system that mimics an industrial manufacturing process. The dataset contains the continuous voltage and current readings of the coffeemakers, sampled at 6400 samples-per-second with the MEDAL measurement device and additional ground-truth information. In particular, we provide 370600 expert-labelled electrical events, 1734 product events, and 3646 maintenance-related events.

We have implemented labelling tools to annotate the raw electrical signals and to refine the ones that are automatically generated by the respective coffeemaker at a one-minute granularity, i.e., the product- and maintenance events. All tools and related source code are released to the public to enable researchers to further extend the dataset.

The dataset can be used to benchmark various analysis tasks, e.g. to monitor the condition of the pumps, heaters, and motors of the coffeemakers. In addition, the dataset provides the most extensive amount of labelled electrical events in the field at the time of publication that can be used to develop new event detection algorithms.

Author's contributions: Conceived and recorded the dataset. Adjusted and implemented tools. Executed parts of the labelling. Wrote the paper.

3.2 MEED: An Unsupervised Multi-Environment Event Detector for Non-Intrusive Load Monitoring

Reference: D. Jorde, M. Kahl, and H.-A. Jacobsen. "MEED: An Unsupervised Multi-Environment Event Detector for Non-Intrusive Load Monitoring." In: *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm).* 2019, pp. 1–6. DOI: 10.1109/SmartGridComm.2019.8909729

Full-text version enclosed: Appendix B

Summary: The fundamental step of high-sampling-rate NILM is the detection of state-transitions of appliances and components. Enabled by accurate detection, various applications, such as appliance classification and energy disaggregation, can be implemented. Existing algorithms for detecting such events rely on expert made pre-defined rules and patterns to detect relevant signal segments. These algorithms are customised to specific environments, preventing them from generalising well to other environments without manually adapting them. We overcome this limitation by introducing a new unsupervised, multi-environment event detector (MEED). At its heart, the algorithm applies a two-step procedure to detect the events with high-precision in time. In the first step, the cumulative sum of a window of the current signal is fed to a denoising autoencoder. As the events are rare by nature compared to non-event segments, the model parameters are fine-tuned to reconstruct non-event windows. Based on this, we use an automatically determined threshold on the reconstruction error to detect event windows. In the second step of MEED, we apply a peak-detection procedure to precisely locate the events. We compare our approach to two state-of-the-art algorithms on the office-environment BLOND and the residential BLUED dataset. We outperform the existing algorithms on both environments with respect to the recall and precision metric while training MEED fully unsupervised. In particular, no manual adaption of the algorithm is necessary. We release all models and code to facilitate reproducibility.

Author's contributions: Conceived, developed, and implemented the approach. Devised optimisations. Conducted analysis and experimental evaluation. Wrote the paper.

3.3 Event Detection for Energy Consumption Monitoring

Reference: D. Jorde and H.-A. Jacobsen. "Event Detection for Energy Consumption Monitoring." In: *IEEE Transactions on Sustainable Computing* (2020), pp. 1–1. DOI: 10.1109/TSUSC.2020.3012066

Full-text version enclosed: Appendix C

Summary: In the field of NILM, various approaches for detecting relevant signal segments and events have been proposed. As there is no unified standard for evaluating these algorithms, many are evaluated on non-public datasets and according to unclear criteria, making them hard to compare. We conduct an extensive literature review on the existing state of the art in event detection for high-sampling-rate NILM approaches. In particular, we categorise the relevant publications with respect to the approaches proposed and the evaluation methods applied. Consequently, we are able to identify several publications that can be compared to each other when carefully investigating the datasets and evaluation methods used. Besides, we also list all approaches that are hard to compare to each other.

Based on this literature review, we select four state-of-the-art algorithms to perform an algorithmic benchmark. We re-implement these approaches as no publicly available source code exists. We evaluate the approaches on two publicly available, heterogeneous datasets from different environments, namely the BLUED and the BLOND dataset. Furthermore, we compare the four algorithms with our recently proposed fully unsupervised, multi-environment event detector (MEED), showing that MEED improves the existing state of the art with respect to both precision and recall.

Author's contributions: Conceived, developed, and implemented the algorithms and the benchmark. Performed literature review. Conducted analysis and experimental evaluation. Wrote the paper.

3.4 Electrical Appliance Classification using Deep Convolutional Neural Networks on High Frequency Current Measurements

Reference: D. Jorde, T. Kriechbaumer, and H.-A. Jacobsen. "Electrical Appliance Classification using Deep Convolutional Neural Networks on High Frequency Current Measurements." In: *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm).* 2018, pp. 1–6. DOI: 10.1109/SmartGridComm.2018.8587452

Full-text version enclosed: Appendix D

Summary: Appliance Identification is the central step in the NILM analysis pipeline that extracts the appliance level signal information that various applications, such as energy disaggregation and occupancy detection, directly rely on. We propose a new approach that can directly be used on raw high-dimensional electrical signals to perform the classification task. Traditional approaches for appliance classification on NILM are built based on hand-crafted features to identify the devices. Studies have shown that the ideal feature sets for identifying electrical appliances are specific to the type of appliance that is investigated. Consequently, experts need to manually develop and select features, tailored to the appliance composition of the respective setting. We overcome this by proposing a NN architecture that automatically extracts a suitable representation from the raw signals. The NN consists of one-dimensional convolutional and corresponding pooling layers. In addition to the algorithm, we propose two data augmentation methods to increase the training dataset size, to overcome the issue of small datasets in the field. We evaluate our approach on two publicly-available datasets, namely WHITED and PLAID, and achieve F1-scores of 1 and 0.69 respectively without manual feature engineering.

Author's contributions: Conceived, developed, and implemented the approach. Devised optimisations. Conducted analysis and experimental evaluation. Wrote the paper. The approach is based on the author's master thesis, entitled "Identification of Individual Electronic Appliances in High Frequency Energy Data using an Artificial Neural Network Approach" (submitted on the 12.03.2018 at the Technical University of Munich).

4

Discussion

In this chapter, we discuss our contributions and findings in the larger context of research on NILM techniques for analysing electrical signals.

On industrial NILM NILM was originally developed to provide detailed energy breakdowns based on the disaggregation of the electrical mains power consumption of private households [1, 6, 7]. Over the years, researchers transferred the methods to other domains, such as individual industrial components [43], monitoring office buildings [8], monitoring elderly people by detecting occupancy in houses [28], and monitoring the condition of naval vassals [3]. Fundamental for developing algorithms for these domains is the availability of adequate datasets. Without publicly available datasets, in particular, when the source code of the algorithm is also not available, comparing algorithms becomes challenging and prone to errors. Already 25 years ago, researchers applied NILM techniques to industrial components [43], but the main focus remained with residential households. Recently, the interest in industrial applications increased, and Suzuki et al. [11] showed the applicability of NILM algorithms to monitor industrial equipment. The dataset the authors used to evaluate their approach is not publicly available. There are two publicly available datasets containing electrical signals of industrial components to the best of our knowledge, the laboratory-measured industrial load of appliance characteristic dataset (LILACD) [10] dataset and the one by Martins et al. [9]. The first dataset comprises several industrial components, such as, for example, motors, that

are measured isolated and when being activate simultaneously. The LILAC dataset is recorded under laboratory conditions, and the switching-patterns of the components does not follow a dedicated process pattern. Instead, the components are activated systematically to cover different combinations of them [10, 44]. The dataset is measured at high-sampling-rates and contains 1302 samples. The second industrial dataset was published by Martins et al. [9]. The authors used smart metering hardware to record the electrical energy consumption of a poultry feed factory in Brazil. In this factory, pellets of ration for poultry are produced based on corn and soybeans [9]. The dataset includes electrical components of heavy-machinery from the factory, namely pelletisers, exhaust fans and double pole contactors [9]. The smart meters output the measured signal parameters once per second, which is low-frequency according to the previous definition. Using this dataset, the authors show the disaggregation of the consumption of the heavy-machinery with a neural network-based approach [9]. In the larger context of monitoring the conditions of machinery as non-invasive as possible, Suzuki et al. [11] and Lindahl et al. [3] have shown the usefulness of NILM techniques. Despite this, there is no extensive dataset with industrial electrical components that are activated to manufacture products according to dedicated patterns publicly available. In particular, there is no dataset that is sampled at high rates and that provides an extensive ground-truth of electrical events, manufactured products, and implemented maintenance actions of a closed system that is suited for benchmarking algorithms. With the CREAM dataset, we have released such a dataset to overcome the aforementioned issues and to further facilitate research on industrial applications of NILM algorithms. In addition to the use case of condition monitoring, the CREAM dataset provides 370600 labelled electrical events, making it the largest dataset in the field to be used for evaluating event detection algorithms.

On reproducibility in NILM Event detection is fundamental for all high-sampling-rate NILM applications to separate relevant signal segments from irrelevant ones. In our literature review [22], we categorise the existing approaches for high-sampling-rate electrical signals. Hence, we provide the first extensive overview of such algorithms in the field. We find several critical problems that need to be overcome to further improve the existing state of the art. First, the evaluation procedures for event detection need to be unified. Besides the use of different metrics, one can observe that commonly used metrics, such as confusion matrix based ones, are not computed in a uniform way.

Researchers need to communicate evaluation methods and potential tolerance levels for computing true positive events more clearly. To make our approach reproducible and comparable with future work, we have publicly released all source code, in particular, also the evaluation functions. In addition to the evaluation procedures themselves, many approaches are evaluated on small and non-public datasets, making a comparison between the approaches difficult [22]. Furthermore, there is no publicly-available code base for state-of-the-art event detection algorithms in the field. Hence, we have publicly released all re-implemented algorithms and our MEED event detector to facilitate the reproducible in the field. The problem of evaluating event detection algorithms is also discussed in other publications, such as, in the paper by Pereira et al. [35]. The authors empirically explore 23 performance metrics for event detection algorithms. The authors conclude that domain-specific metrics are dataset dependent, making it hard to use them in cross-dataset evaluations. Furthermore, Pereira et al. state that it is important to clearly highlight the trade-off between the classical machine learning-based metrics that are based on the confusion matrix, such as recall, precision and F-measures [35]. This particularly concerns recall and precision, as detectors that are optimised with respect to either of them have a different focus while potentially achieving similar F-scores [35]. In both publications on event detection, we discuss this and the drawbacks of focusing on either precision or recall. In general, we agree with Pereira et al. [35] that new metrics are necessary to benchmark event detection algorithms. If such metrics become available, public source code and datasets are fundamental for evaluating new approaches against the existing state of the art with respect to the new metrics.

On the generalisability of NILM With our event detection approach MEED, we present an event detector that can be used in multiple environments for NILM without the need for manual adaptions of the algorithms with respect to the setting [23]. Recently, researcher such as, for example, Kahl et al. [45] evaluated NILM algorithms across multiple-datasets to show their performance independent of specific datasets. The researchers aim to overcome the common practice of the field, namely the evaluation of algorithms on isolated datasets, and, thus, on specific appliance compositions. By introducing an event detector that can be used in multiple environments, such as residential and office ones, we similarly aim to overcome the issue of algorithms that are only developed for specific settings and appliance compositions. Besides the lack of generalisability between environments, NILM algorithms, such as, for appliance identification, rely on feature sets

that are dependent on the appliances used, as shown in the feature study by Kahl et al. [2]. We introduce an approach based on convolutional neural networks to circumvent manual feature engineering by automatically extracting a good representation from the raw, high-dimensional waveforms [24]. Neural networks are well suited for representation learning [21]. Thus, they also gained popularity in the field of NILM, as a number of publications relying on neural networks and the state-of-the-art results they achieve are indicating [39, 41, 42, 46, 47, 48, 49]. The increasing adoption of neural network approaches, similar to the ones in this dissertation, promises new advances in NILM, in particular, regarding the ability of the algorithms to generalise between datasets and environments.

5

Conclusions

NILM algorithms harvest insights from electrical signals without intrusively attaching sensors to each consumer. The NILM analysis pipeline for high-sampling-rate data comprises multiple steps, with event detection and appliance (component) identification being the fundamental ones. This thesis presented multiple advancements along the analysis pipeline, in particular, a new industrial component dataset, a new event detection algorithm, a survey and extensive benchmark of the existing state of the art in event detection for NILM, and a new appliance identification algorithm. The algorithms developed in this dissertation reduce manual interference by a domain expert and alleviate the effort of adapting them when being used in multiple, heterogeneous environments.

With the new dataset for condition monitoring on electrical signals, namely the CREAM dataset, we provide an extensive ground-truth for high-sampling-rate, industrial electrical signals. CREAM is the first publicly available dataset that enables the evaluation of NILM based condition monitoring algorithms. For doing so, the dataset provides 370600 expert labelled electrical events, information on the components responsible for these events, and relevant manufacturing-related labels, such as products manufactured and maintenance actions taken. In order to provide such a dataset, we have selected two different industrial-grade coffeemakers that closely resemble a manufacturing process with electrical components, while being a closed and fully controllable system. By creating and refining labels for this dataset, we provide full transparency on the components used,

the products created, and the maintenance actions taken for both coffeemakers. We have added a second coffeemaker to the dataset, as it further enables the implementation of comparative benchmarks. With the intention to facilitate reproducibility and to enable researchers to extend the dataset if needed, we have publicly released all related source code and tools.

The fundamental step of the NILM analysis pipeline is event detection. We improve the current state of the art and further contribute an extensive literature review with algorithm implementations to the field in this thesis. First, we introduce MEED, a new multi-environment event detection algorithm that can be used fully unsupervised in contrast to the existing approaches. The algorithm can be used in different environments, such as offices spaces and residential houses, without the need for manually adapting the algorithm to the respective setting. We compare MEED to two re-implemented state-of-the-art algorithms on two publicly available datasets, outperforming them with respect to the recall and precision metrics. A two-step procedure lies at the heart of the algorithm, with the first step being a window-based denoising autoencoder to detect event windows, and the second step being a peak-detection algorithm for precisely allocating the events in time.

Furthermore, we contribute to the field of event detection by providing a comprehensive overview and categorisation of the existing state of the art in NILM. Based on this overview, we identify research gaps and select four state-of-the-art algorithms to re-implement them. Subsequently, we evaluate these algorithms against our MEED algorithm on two publicly available datasets. By releasing the source code of the algorithms and the evaluation to the public, we aim to provide a reusable library for evaluating new algorithms.

Appliance identification is the step following the detection of events in the electrical signal. The output of the appliance identification algorithms is the identification of the electrical consumer that is responsible for a certain event. Traditionally, manually engineered features are computed for a window around the event that captures the state of the appliance responsible for the event. These hand-crafted appliance signatures are custom-tailored to specific appliance types and settings. Thus, they require interference by domain experts. We introduce a new approach that uses the raw voltage and current waveform

to extract a suitable representation and perform the classification task automatically. We overcome the challenge of the high-dimensional input and the comparatively small dataset sizes by introducing two data-augmentation techniques. By evaluating the approach on two publicly available datasets, namely the WHITED and the PLAID dataset, we achieve state-of-the-art results while avoiding manual feature engineering.

In the course of our work, we have identified several aspects that may be targeted in future work. We think that pursuing the following research directions has the potential to further promote the field of NILM and the adaption of the algorithms in the industry:

More focus on applications other than energy disaggregation Energy disaggregation, in particular, on low-sampling-rate data has matured to a certain extend. The past research has mainly focused on energy disaggregation, while other application areas, such as occupancy detection and monitoring the conditions of industrial electrical equipment, were not pursued as much. Recent publications have shown the potential of NILM, in particular, for analysing industrial equipment, in a non-invasive way [11]. Thus, NILM research on other applications areas is promising.

More datasets from different domains Most of the existing datasets for NILM are acquired at residential households. Only recently, a few datasets from other domains, such as, for example, office ones have been released. To further pursue the adaption of NILM techniques in other application areas, more publicly-available datasets from other domains are necessary to develop and evaluate the corresponding algorithms.

Unification of the methods for evaluating event detection methods While there are efforts to unify the evaluation metrics and methods for event detection algorithms in NILM, the publications in the field still use heterogeneous evaluation approaches. Using common metrics, or even a common code-based for evaluating high-sampling-rate event detection methods for NILM bears the chance to further advance the field.

More usage of publicly available datasets and source code Similar to the unification of the methods for evaluating event detection algorithms, the increasing usage of publicly-available datasets for evaluating the algorithms enables new ways to reproduce the research and to compare new algorithms with existing ones. Furthermore, only a few source code repositories for corresponding publications on NILM exist currently. Most of

the code released is on low-sampling-rate NILM, such as the NILM toolkit (NILMTK) [50]. More publicly-available source code would enable more transparent benchmarks with new algorithms, analogous to the use of public data.

Moving away from the standalone claim of NILM Most of the existing work on NILM focuses on the standalone usage of NILM algorithms for energy disaggregation and other applications and does not seize side-channel information, despite such features being mentioned in various publications [1, 51]. We are convinced that, in particular in an industrial setting, other sensor information than electrical signals can be used to improve the insights into machine conditions. Electrical signals deliver a straightforward, non-invasive mean for collecting information on machinery, but under certain conditions, intrusive sensors can be applicable and improve the results of NILM.

Glossary

ADC analog-digital-converter

AI Artificial Intelligence

BLOND Building-Level Office eNvironment Dataset

BLUED Building Level fUlly-labeled dataset for Electricity Disaggregation

CNN Convolutional Neural Network

cot current-over-time

CREAM CoffeemakeR Electrical Activity Measurements

FCNN Fully-Connected Neural Network

FN False Negative

FP False Positive

LSTM Long Short-Term Memory

MEDAL Mobile Energy Data Acquisition Laboratory

MEED Multi-Environment Event Detector

NILM Non-Intrusive Load Monitoring

NILMTK Non-Intrusive Load Monitoring Toolkit

NN Neural Network

oer odd-even-harmonics ratio

PC personal computer

PLAID Plug-Level Appliance Identification Dataset

rms root-mean-square

RNN Recurrent Neural Network

roi region-of-interest

SCP Switching-Continuity-Principle

TN True Negative

TP True Positive

WHITED Worldwide Household and Industry Transient Energy Dataset

List of Figures

Bibliography

[1] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar. "Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey." In: *Sensors* 12.12 (2012), pp. 16838–16866. DOI: 10.3390/s121216838.

[2] M. Kahl, A. Ul Haq, T. Kriechbaumer, and H.-A. Jacobsen. "A Comprehensive Feature Study for Appliance Recognition on High Frequency Energy Data." In: *Proceedings of the Eighth International Conference on Future Energy Systems*. (Shatin, Hong Kong). New York, NY, USA: ACM, 2017, pp. 121–131. DOI: 10.1145/3077839.3077845.

[3] P. A. Lindahl, D. H. Green, G. Bredariol, et al. "Shipboard Fault Detection Through Nonintrusive Load Monitoring: A Case Study." In: *IEEE Sensors Journal* 18.21 (2018), pp. 8986–8995.

[4] European Commission. *EU Energy in Figures*. 2018. ISBN: 978-92-76-08818-9.

[5] J. Kelly and W. J. Knottenbelt. "Does Disaggregated Electricity Feedback reduce Domestic Electricity Consumption? A Systematic Review of the Literature." In: *3rd International Workshop on NILM*. 2016, pp. 1–5. URL: http://arxiv.org/abs/1605.00962 (visited on 06/06/2020).

[6] C. Armel, A. Gupta, G. Shrimali, and A. Albert. "Is Disaggregation the Holy Grail of Energy Efficiency? The Case of Electricity." In: *Energy Policy* 52 (Jan. 2013), pp. 213–234.

[7] G. W. Hart. "Nonintrusive Appliance Load Monitoring." In: *Proceedings of the IEEE* 80.12 (Dec. 1992), pp. 1870–1891.

[8] T. Kriechbaumer and H.-A. Jacobsen. "BLOND, a Building-Level Office Environment Dataset of Typical Electrical Appliances." In: *Scientific Data* 5.180048 (2018).

[9] P. B. M. Martins, J. G. R. C. Gomes, V. B. Nascimento, and A. R. de Freitas. "Application of a Deep Learning Generative Model to Load Disaggregation for Industrial Machinery Power Consumption Monitoring." In: *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. Aalborg: IEEE, Oct. 2018, pp. 1–6. DOI: 10.1109/SmartGridComm.2018.8587415.

[10] M. Kahl, V. Krause, R. Hackenberg, et al. *Measurement System and Dataset for in-depth Analysis of Appliance Energy Consumption in Industrial Environment*. https://www.in.tum.de/i13/resources/lilacd/. 2019.

[11] R. Suzuki, S. Kohmoto, and T. Ogatsu. "Non-Intrusive Condition Monitoring for Manufacturing Systems." In: *2017 25th European Signal Processing Conference (EUSIPCO)*. 2017, pp. 1390–1394.

[12] A. W. Colombo, S. Karnouskos, O. Kaynak, Y. Shi, and S. Yin. "Industrial Cyberphysical Systems: A Backbone of the Fourth Industrial Revolution." In: *IEEE Industrial Electronics Magazine* 11.1 (2017), pp. 6–16.

[13] *Condition Monitoring and Diagnostics of Machines — General Guidelines*. Standard. Geneva, CH: International Organization, May 2018.

[14] T. Lu, Z. Xu, and B. Huang. "An Event-Based Nonintrusive Load Monitoring Approach: Using the Simplified Viterbi Algorithm." In: *IEEE Pervasive Computing* 16.4 (Oct. 2017), pp. 54–61.

[15] L. d. Baets, J. Ruyssinck, D. Deschrijver, and T. Dhaene. "Event Detection in NILM using Cepstrum Smoothing." In: *3rd International Workshop on NILM*. May 2016, pp. 1–4. URL: https://users.ugent.be/~didschri/papers/2016_05__NILM_Conf.pdf (visited on 09/02/2020).

[16] J. M. Alcala, J. Urena, and A. Hernandez. "Event-based Detector for Non-Intrusive Load Monitoring based on the Hilbert Transform." In: *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation*. Sept. 2014, pp. 1–4.

[17] K. D. Anderson, M. E. Berges, A. Ocneanu, D. Benitez, and J. M. F. Moura. "Event Detection for Non Intrusive Load Monitoring." In: *38th Annual Conference on IEEE Industrial Electronics Society*. Oct. 2012, pp. 3312–3317.

[18] Y. Jin, E. Tebekaemi, M. Berges, and L. Soibelman. "Robust Adaptive Event Detection in Non-Intrusive Load Monitoring for Energy Aware Smart Facilities." In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. May 2011, pp. 4340–4343.

[19] K. S. Barsim and B. Yang. "Sequential Clustering-Based Event Detection for Non-Intrusive Load Monitoring." In: *Proceedings of the 6th International Conference on Computer Science and Information Technology*. Jan. 2016, pp. 77–85.

[20] N. Sadeghianpourhamami, J. Ruyssinck, D. Deschrijver, T. Dhaene, and C. Develder. "Comprehensive feature selection for appliance classification in NILM." In: *Energy and Buildings* 151 (2017), pp. 98–106. DOI: 10.1016/j.enbuild.2017.06.042.

[21] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. Cambridge, Massachusetts and London, England: MIT Press, 2016. URL: http://www.deeplearningbook.org/.

[22] D. Jorde and H.-A. Jacobsen. "Event Detection for Energy Consumption Monitoring." In: *IEEE Transactions on Sustainable Computing* (2020), pp. 1–1. DOI: 10.1109/TSUSC.2020.3012066.

[23] D. Jorde, M. Kahl, and H.-A. Jacobsen. "MEED: An Unsupervised Multi-Environment Event Detector for Non-Intrusive Load Monitoring." In: *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. 2019, pp. 1–6. DOI: 10.1109/SmartGridComm.2019.8909729.

[24] D. Jorde, T. Kriechbaumer, and H.-A. Jacobsen. "Electrical Appliance Classification using Deep Convolutional Neural Networks on High Frequency Current Measurements." In: *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. 2018, pp. 1–6. DOI: 10.1109/SmartGridComm.2018.8587452.

[25] D. Jorde, T. Kriechbaumer, T. Berger, S. Zitzlsperger, and H.-A. Jacobsen. "CREAM, a component level coffeemaker electrical activity measurement dataset." In: *Scientific Data* (2020), accepted for publication on 15.10.2020.

[26] Y. F. Wong, Y. Ahmet Sekercioglu, T. Drummond, and V. S. Wong. "Recent approaches to non-intrusive load monitoring techniques in residential settings." In: *IEEE Symposium on Computational Intelligence Applications in Smart Grid (CIASG 2013)*. IEEE, 2013, pp. 73–79.

[27] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng. "Load Signature Study—Part II. Disaggregation Framework, Simulation, and Applications." In: *IEEE Transactions on Power Delivery* 25 (2 2010), pp. 561–569. DOI: 10.1109/TPWRD.2009.2033800.

[28] J. Alcalá, J. Ureña, and Á. Hernández. "Activity supervision tool using Non-Intrusive Load Monitoring Systems." In: *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*. 2015, pp. 1–4.

[29] A. Haq and H.-A. Jacobsen. "Prospects of Appliance-Level Load Monitoring in Off-the-Shelf Energy Monitors: A Technical Review." In: *Energies* 11.1 (2018), p. 189. DOI: 10.3390/en11010189.

[30] T. Kriechbaumer, A. Ul Haq, M. Kahl, and H.-A. Jacobsen. "MEDAL." In: *Proceedings of the Eighth International Conference on Future Energy Systems*. (Shatin, Hong Kong). New York, NY, USA: ACM, 2017, pp. 216–221. DOI: 10.1145/3077839.3077844.

[31] Y. Himeur, A. Alsalemi, F. Bensaali, and A. Amira. "Building power consumption datasets: Survey, taxonomy and future directions." In: *Energy and Buildings* 227 (2020). DOI: https://doi.org/10.1016/j.enbuild.2020.110404.

[32] M. Kahl, A. U. Haq, T. Kriechbaumer, and H.-A. Jacobsen. "Whited-a worldwide household and industry transient energy data set." In: *3rd International Workshop on Non-Intrusive Load Monitoring*. (Vancouver, Canada). 2016.

[33] J. Gao, S. Giri, E. C. Kara, and M. Bergés. "PLAID. A Public Dataset of High-resolution Electrical Appliance Measurements for Load Identification Research: Demo Abstract." In: *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. BuildSys'14. (Memphis, TN, USA). ACM. New York, NY, USA: ACM, 2014, pp. 198–199. DOI: 10.1145/2674061.2675032.

[34] K. D. Anderson, A. Ocneanu, D. Benitez, et al. "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*. Jan. 2012, pp. 1–5.

[35] L. Pereira and N. Nunes. "An Experimental Comparison of Performance Metrics for Event Detection a Algorithms in NILM." In: *4th International Workshop on NILM*. 2018. URL: http://nilmworkshop.org/2018/proceedings/Paper_ID07.pdf (visited on 06/06/2020).

[36] M. Valovage and M. Gini. "Label Correction and Event Detection for Electricity Disaggregation." In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. May 2017, pp. 990–998.

[37] M. Kahl, T. Kriechbaumer, A. U. Haq, and H. Jacobsen. "Appliance classification across multiple high frequency energy datasets." In: *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. 2017, pp. 147–152.

[38] S. Makonin. "Investigating the Switch Continuity Principle assumed in Non-Intrusive Load Monitoring (NILM)." In: *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. Piscataway, NJ, USA: IEEE, 2016, pp. 1–4. DOI: 10.1109/CCECE.2016.7726787.

[39] J. Kelly and W. Knottenbelt. "Neural NILM." In: *Proceedings of the 2nd ACM BuildSys*. New York NY: ACM, 2015, pp. 55–64. DOI: 10.1145/2821650.2821672.

[40] O. Parson, S. Ghosh, M. Weal, and A. Rogers. "An unsupervised training method for non-intrusive appliance load monitoring." In: *Artificial Intelligence* 217 (2014), pp. 1–19. DOI: https://doi.org/10.1016/j.artint.2014.07.010.

[41] H. Lange and M. Berges. "The Neural Energy Decoder: Energy Disaggregation by Combining Binary Subcomponents." In: *3rd International Workshop on NILM*. 2016.

[42] K. S. Barsim, L. Mauch, and B. Yang. "Neural Network Ensembles to Real-time Identification of Plug-level Appliance Measurements." In: *3rd International Workshop on NILM*. Vol. 2. 2016.

[43] S. B. Leeb, S. R. Shaw, and J. L. Kirtley. "Transient Event Detection in Spectral Envelope Estimates for Nonintrusive Load Monitoring." In: *IEEE Transactions on Power Delivery* 10.3 (July 1995), pp. 1200–1210.

[44] M. Kahl, V. Krause, R. Hackenberg, et al. "Measurement system and dataset for in-depth analysis of appliance energy consumption in industrial environment." In: *tm - Technisches Messen* 86.1 (2019). DOI: https://doi.org/10.1515/teme-2018-0038.

[45] M. Kahl, T. Kriechbaumer, A. U. Haq, and H. Jacobsen. "Appliance classification across multiple high frequency energy datasets." In: *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. 2017, pp. 147–152.

[46] P. Dash and K. Naik. "A Very Deep One Dimensional Convolutional Neural Network (VDOCNN) for Appliance Power Signature Classification." In: *2018 IEEE Electrical Power and Energy Conference (EPEC)*. Toronto, ON: IEEE, Oct. 2018, pp. 1–6. DOI: 10.1109/EPEC.2018.8598355. (Visited on 05/28/2019).

[47] P. Davies, J. Dennis, J. Hansom, et al. "Deep Neural Networks for Appliance Transient Classification." In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, United Kingdom: IEEE, May 2019, pp. 8320–8324. DOI: 10.1109/ICASSP.2019.8682658. (Visited on 05/29/2019).

[48] T.-T.-H. Le, J. Kim, and H. Kim. "Classification performance using gated recurrent unit recurrent neural network on energy disaggregation." In: *2016 International Conference on Machine Learning and Cybernetics (ICMLC)*. Jeju Island, South Korea: IEEE, July 2016, pp. 105–110. DOI: 10.1109/ICMLC.2016.7860885. (Visited on 05/29/2019).

[49] M. Kaselimi, N. Doulamis, A. Doulamis, A. Voulodimos, and E. Protopapadakis. "Bayesian-optimized Bidirectional LSTM Regression Model for Non-intrusive Load Monitoring." In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, United Kingdom: IEEE, May 2019, pp. 2747–2751. DOI: 10.1109/ICASSP.2019.8683110. (Visited on 05/29/2019).

[50] N. Batra, J. Kelly, O. Parson, et al. "NILMTK." In: *Proceedings of the 5th international conference on Future energy systems - e-Energy '14* (2014). DOI: 10.1145/2602044.2602051.

[51] N. Roux, B. Vrigneau, and O. Sentieys. "Improving NILM by Combining Sensor Data and Linear Programming." In: *2019 IEEE Sensors Applications Symposium (SAS)* (2019), pp. 1–6. DOI: 10.1109/SAS.2019.8706021.

Appendices

A CREAM, a component level coffeemaker electrical activity measurement dataset

SCIENTIFIC DATA

CREAM, a component level **c**offeemake**r** **e**lectrical **a**ctivity **m**easurement dataset

Daniel Jorde [1], Thomas Kriechbaumer [1], Tim Berger [1],
Stefan Zitzlsperger [1], Hans-Arno Jacobsen [1]

October 26, 2020

1. Department of Computer Science, Chair for Application and Middleware Systems, Technical University of Munich, 85748 Garching, Germany. Correspondence and requests for materials should be addressed to D.J. (email: daniel.jorde@tum.de).

Abstract

Monitoring the internal conditions of a machine is essential to increase its production efficiency and to reduce energy waste. Non-intrusive condition monitoring techniques, such as analysing electrical signals, provide insights by disaggregating a composite signal of a machine as a whole into the individual components to determine their states. Developing and evaluating new algorithms for condition monitoring and maintenance-related analysis tasks require a fully-labelled dataset for a machine, which comprises standard industrial components that are triggered following a typical manufacturing process to produce goods. For this purpose, we introduce CREAM, a component level electrical measurement dataset for two industrial-grade coffeemakers, simulating industrial processes. The dataset contains continuous voltage and current measurements provided at 6,400 samples per second, as well as the product and maintenance-related event labels, such as 370,600 expert-labelled component-level electrical events, 1,734 product ones and 3,646 maintenance ones. CREAM provides fully-labelled ground-truth to establish a benchmark and comparative studies of manufacturing-related analysis in a controlled and transparent environment.

Background & Summary

Recent advances in artificial intelligence and the increasing implementation of modern cyber-physical systems in the manufacturing industry constitute the backbone of a new industrial revolution [1]. The monitoring of current conditions and internal states of industrial machines is fundamental to increase the production and energy efficiency [3]. The placement of sensors, to obtain

SCIENTIFIC DATA

detailed information about the behaviour of the machine's individual components, is fundamental in the condition monitoring (CM) process [3]. Instead of intrusively measuring each component of a machine individually, an aggregated signal for multiple components can be considered. In a subsequent step, algorithms to extract the per-component information from an aggregate signal can be applied. Such an approach can allow for avoiding invasive interference that causes various problems, such as high costs associated with sensor implementation and warranty issues. Initially developed to provide feedback on energy consumption in residential environments, non-intrusive load monitoring (NILM) is widely used for other purposes, such as CM [4, 5]. NILM algorithms can be used to disaggregate power signals measured at the electrical mains of a building into the individual appliances [6, 7]. By implementing sensors, such as Hall Effect current ones, electrical signals of a machine or appliance can be measured in a non-intrusive manner [5]. Sampling the voltage and current signals at high rates is necessary to identify individual components when many other components are concurrently activated and to enable differentiation between smaller ones [9]. To the best of our knowledge, there are two public datasets containing the data on electrical measurements of industrial-like machinery. Both of them have drawbacks, such as being either sampled at low rates [10] or comprising only individual appliances from a laboratory environment [11]. The first dataset contains electrical parameters of a poultry feed factory, recorded for a duration of 111 days. The smart meters at the factory sample the data internally at 8000 sps, but send out the down-sampled electrical features once per second. This dataset provides insights into the energy consumption of a factory using NILM techniques for energy disaggregation. The machine components in this factory produce pellets of ration for poultry by processing corn or soybeans. The dataset comprises two pelletisers, two double-pole contactors, two exhaust fans and two milling machines. All appliances measured are horizontal motors [10]. In the second dataset, electrical signals for fifteen residential and industrial electrical components were sampled at 50,000 sps in a laboratory environment. However, the utilised devices were not activated according to a dedicated pattern, for example, such as a production process, and no complementary information about conditions of components is provided [11]. In addition to these two datasets, several other datasets containing sensor measurements for CM concerning individual components were established [12, 13, 14]. These datasets contain information about the isolated components using a dedicated sensor infrastructure to obtain various parameters. The milling dataset by Agogino and Goebel [12], for example, provides records on the wear of the milling insert of a milling machine, recorded at different speeds, feeds and depth of cut [12]. Some of the datasets provide additional information about detected faults of components, such as, for example, a hydraulic test rig [14]. In particular, in this dataset, measurements on the condition of hydraulic components in a primary working and a secondary cooling-filtration circuit are presented [14]. The sensor data includes features such as, for example, pressure, motor power, temperature, and vibration, measured at least once per second. The dataset includes component-specific failure information. The failure information for each

SCIENTIFIC DATA

component is structured hierarchically, from full functionality to failure of the component [14].

To construct a dataset that would enable the evaluation of algorithms for non-invasive CM, event detection, and other manufacturing-related analysis tasks, we formulated the following requirements. First, a considered machine had to execute an industrial process, including typical electrical components that are used in manufacturing, triggered following dedicated process patterns. Second, the environment and the machine had to be fully-controllable to avoid any unknown external interference. Third, the machine had to be equipped with sensors to record reliable ground-truth for events caused by components. We focused on the events related to the fabricated products and performed maintenance actions. Following these requirements, we selected two distinct fully-automated, industrial-level coffeemakers to construct the proposed coffeemaker electrical activity measurement (CREAM) dataset, to enable individual machine analysis and comparative studies between the coffeemakers. We provide high-resolution continuous measurements of the voltage and current signals of the coffeemakers acquired at 6,400 sps. During signal acquisition, the machines produced eight different product types, each following a unique internal process. Furthermore, we provide 370,600 expert-labelled electrical events, triggered by the machine components. In addition, CREAM contains the labels for the three main components of the coffeemakers, namely the respective heaters, pumps, and motors of the milling plants. The data are marked with the product and maintenance labels, containing the information about the fabricated products and performed maintenance actions. Therefore, the resulting dataset can be considered as a source for a wide variety of tasks, such as CM, product analysis, and maintenance prediction.

Methods

We constructed the CREAM dataset based on the previously defined requirements. For the *JURA GIGA X8*, the information accumulated in the dataset was recorded for a period from 23 August 2018 to 8 October 2018. We recorded the *JURA GIGA X8* dataset for sixteen hours every day, except for the last one, 8 October 2018, that was measured for eight hours. The data for the *JURA GIGA X9* was recorded for 20 days, starting from 22 December 2018, for 15 hours per day. The daily data acquisition time frames were chosen to cover the main periods the coffeemakers were activated. For both coffeemakers, the data acquisition process was divided into three sub-steps, that apply equally to both machines. The data acquisition setup is shown in Figure 1. First, we sampled the voltage and current signals of each coffeemaker at 6,400 sps. To execute this step, we utilised a custom measurement device for high-sampling rate plug-level appliance recordings to achieve this, namely, the mobile energy data acquisition laboratory (MEDAL) measurement unit [15]. Simultaneously, we extracted the product and maintenance-related event logs from a serial port of the coffeemaker using a single-board personal computer (PC). As the methodology described in

the following is meant to apply generically to other scenarios, we refer to the types of coffee the coffeemakers produce as products. Lastly, three experts labelled the electrical component events and refined the automatically generated product and maintenance logs. In the next section, we provide general information about the considered coffeemakers, such as their individual components, the production processes, and other relevant characteristics.

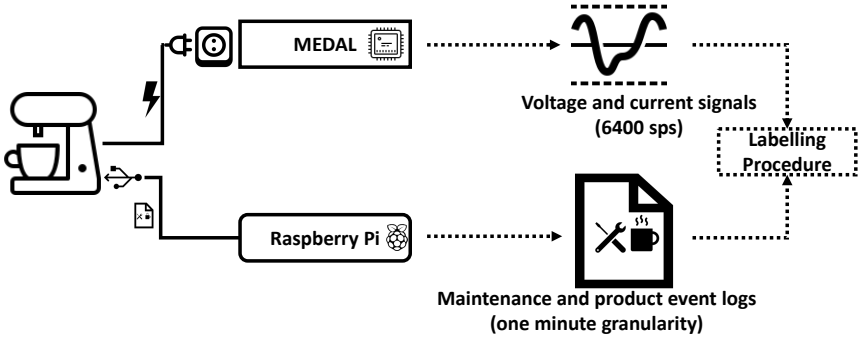


Figure 1: **Data collection architecture**. The setup consists of the MEDAL unit for measuring the voltage and current signals and the Raspberry Pi for pulling the maintenance and product event logs with a one-minute resolution from the serial ports of the coffeemakers. Afterwards, we applied the three-step labelling procedure, as shown in Figure 2.

Domain knowledge

We measured the voltage and current consumed by two different professional coffeemakers, the *Jura Giga X8 Professional* [16] and the *Jura Giga X9 Professional*, and combined them with the hand-labelled components and machine-generated event-logs. Below, we describe the domain knowledge and architecture useful to interpret the generated data, based on the official technical description and the components of the machines. Concerning the general production approach, pre-defined processes trigger the individual components of the machine for brewing a requested coffee product. First, a grinder is launched to grind pre-roasted beans and feeds the ground coffee to the brewing unit. Second, if not already pre-heated, heating units are utilised to heat the water or to produce steam for the requested coffee product. The water or steam is then pressed through the brewing unit, which controls the water flow through the ground coffee. A dedicated steam or a water pump moves steam or water through the machine. Then, the brewing unit presses water through the coffee to extract the ingredients, such as caffeine and oils, from coffee. After the brewing unit, the brewed coffee flows into a drinking container and is output from the coffeemaker. The brewing unit then pushes the residual coffee into

the coffee tray. Depending on a particular coffee product, intermediary steps such as heating milk and producing milk foam, are executed. The *Jura Giga X8* has two thermal heating blocks. Therefore, steam required to produce milk foam and hot water is generated simultaneously. In contrast, the *Jura Giga X9* has one additional thermal heating block and an additional pump, to speed up the production process, especially for hot water [17]. In addition to the described brewing process, the coffeemakers have other maintenance programmes to ensure the long-term functionality of the machines and to speed-up the brewing process. These maintenance processes involve various actions, such as, for example, regular cleaning and descaling the coffee and milk systems.

Each coffeemaker is comprised of several major components and a variety of small ones. The main components involved in the production process are pumps, thermal heating blocks, and ceramic grinding modules, as listed in Table 1. The selection of the main components was performed according to the feasibility of detecting them visually in the electrical signal by the human experts. Therefore, the other components included in the coffeemakers, such as, for example, lights, valves, a touchscreen, and a drainage motor, were excluded from consideration due to their small power consumption or more complex power usage patterns.

Component	X8	X9	Characteristics	Power
water pump	1x	2x	15 Bar pressure	65 W
steam pump	1x	1x	15 Bar	28 W
thermal heater	2x	3x	-	1,080 W
grinding motor	2x	2x	DC motor	26 W - 236 W

Table 1: **Main components of the coffeemaker.** The list outlines the main components, key characteristics, and their energy consumption.

The *Jura GIGA X8* is composed of two grinders, one for espresso beans and one for coffee ones, launched depending on a requested product. Each of the grinders is powered by a directed current (DC) motor. The motor energy usage depends on the speed it is running at. Therefore, its power consumption is within a specific range, as outlined in Table 1. Furthermore, thermal heating blocks are employed to produce hot water and steam when the machine generates a product or when the built-in pre-heating controller launches the heating process. In this way, the coffee-making process is sped up, as heating water to the required temperature is time-consuming. Hot water and steam are transported through the machine using the corresponding pumps. At the end of the process, water is pressed through the brewing unit. The timing and energy consumption for these components varied according to particular products and settings of the machine. As previously mentioned, the *Jura GIGA X9* has three thermal heating blocks and three pumps to speed up the production process.

The entire *Jura Giga X8* coffeemaker has a nominal capacity of 2,700 W and a standby power consumption of approximately 0.5 W when operating it at the base-frequency of 50 Hz [16]. The *Jura Giga X9* differs, as is has a nominal capacity of 2,300 W, while having the same standby power consumption [17].

SCIENTIFIC DATA

Voltage and current monitoring

A single MEDAL measurement unit was used to collect the voltage and current signals [15]. MEDAL comprises an off-the-shelf power strip, a voltage, and a current sensor, as well as an embedded single-board PC for processing recorded measurements. The MEDAL system was initially developed to record a long-term office environment dataset for energy disaggregation [8]. Therefore, it complies with the high-requirements concerning data quality and long-term continuous recording. Each MEDAL unit has six sockets available, enabling it to measure six devices simultaneously. The data for each coffeemaker was collected independently and sequentially. Hence, we describe the setup exemplary for one of the coffeemakers in the following. We used two sockets to monitor the coffeemaker. The coffeemaker was plugged-into one socket (socket 1), and the other socket (socket 6) was used to record the background-noise generated by the measurement device. In this way, we facilitate noise filtering for users. Socket 1 was explicitly designed for measuring high-power devices (up to 3,600 W). In the case of exceeding this limit, the recorded signal is limited to the maximum value, while keeping the operation electrically safe. The measurement unit itself consumes 5 W.

A hall effect-based sensor from the *Allegro ACS712* family recorded an independent current signal for each of the sockets. Furthermore, one voltage signal was recorded for each coffeemaker. The MEDAL's sampler board is used to digitise the analogue signal and to transmit the data via USB connection to the single-board PC, a Raspberry Pi 3. Here, seven independent single-channel analogue digital converters (ADC) *MCP3201* with a 12-bit resolution are used [18]. Despite utilising independent ADCs, MEDAL samples the signals simultaneously, coordinated by an *ATmega324PA* microcontroller. The recorded data were stored on a SSD hard-drive connected to the MEDAL via USB.

MEDAL is capable of recording the signals with a high temporal resolution without introducing data losses and gaps, which allows capturing the voltage and current signals at 6,400 sps. These high sampling rates enable extracting the frequency-domain related features for various analytical purposes [9].

Product and maintenance events

In addition to recording electrical signals, we collected the product and maintenance event logs that were automatically generated by the coffeemakers and read out over the serial maintenance ports of the machines. We used the setup and the information described in the coffeemaker reengineering project repository and documentation provided by the company Q42 [19]. For each coffeemaker, a Raspberry Pi microcontroller was connected to the serial maintenance port, using its receiver and transmitter pins to establish an 8-N-1 serial connection. Then, the events were extracted from each coffeemaker's internal EEPROM using the reverse-engineered codes provided in the repository and stored on the SSD hard-drive. The raw events generated by the machines were marked by timestamps with a one-minute time resolution and were created after or close

SCIENTIFIC DATA

to the completion of an event.

Name	Milling Plant	Milk	Two X8 | X9
cappuccino	espresso	Yes	Yes | Yes
coffee	coffee	No	Yes | Yes
espresso	espresso	No	Yes | Yes
hot_water	-	-	-
latte_macchiato	espresso	Yes	Yes | Yes
white_coffee	espresso	Yes	Yes | -
ristretto	espresso	No	Yes | Yes
espresso_macchiato	espresso	Yes	Yes | Yes

Table 2: **The list of fabricated products of both coffeemakers.** The products have different production processes depending on the involvement of a type of a milling plant and the usage of milk. Some products can be produced simultaneously, as indicated by the column *Two* for both coffeemakers respectively.

While measuring electrical signals, eight different products were produced by the two coffeemakers. In addition to producing these products, the coffeemakers were capable of providing a wide variety of other hot water and milk-based products. The products mentioned in the dataset are listed in Table 2. We omitted the products that were not produced when data collection was enabled.

The product considered indicates which components were utilised during the preparations process. When attempting to separate the behaviour of components that are built-in into the coffeemakers multiple times, such as grinding modules, the product information was analysed to identify the particular component involved. In addition to the product events listed in Table 2, we also recorded maintenance-related events, as listed in Table 3. Certain events were triggered to request a user to perform maintenance activities, such as, for example, to rinse the milk system. Other events refereed to the executed action, such as, for example, the machines rinsing the milk system. The *Type* column in Table 3 indicates whether an event is an alert for action (type$P$) or an action executed by the machines (type A). Both event types could be considered to extract and predict the maintenance-related information from the electrical data, as they described the current state of the system. To illustrate this, we consider the following example. When rinsing milk or the coffee system, water is pumped through the respective pipes to remove the remains of the coffee making process. The *RinseMilkSystem* and *RinseCoffeeSystem* activities can be launched either automatically by the coffee maker or manually by a user after the *CleanMilkSystem* or *Time2Clean* alerts appeared on the screen. In contrast to using water for rinsing the system, the *CleanMilkSystem* alert requests a user to insert a cleaning agent into the machine. The standard procedure is to perform this task daily. The *Clean* alert requires the following actions from the user: the drip tray and the ground coffee container have to be removed and emptied. Then, a cleaning agent has to be used to clean the whole system. The coffeemakers can

SCIENTIFIC DATA

Name	Description	Type
MillingPlantEspresso	Grinding espresso beans	A
MillingPlantCoffee	Grinding coffee beans	A
CleanMilkSystem	Cleaning the milk system	P
RinseMilkSystem	Rinsing the milk system	A
Time2Clean	Alert: Clean the coffee system	P
RinseCoffeeSystem	Rinsing the coffee system	A
Clean	Clean the whole system	A/P
Time2Descale	Descale the whole system	A/P

Table 3: **Maintenance-related events of both coffeemakers.** The list represents all maintenance-related events in the dataset and their purpose.

not be used before the completion of the cleaning process, which takes approximately 20 minutes. Similarly, the *Time2Descale* alert requests a user to add a descaling tablet into the water and to run the descaling programme that takes approximately 50 minutes [16, 17].

Labelling procedure

The behaviour of electrical components was captured in the voltage and current signals recorded by the MEDAL unit. The electrical signals were marked according to three sets of labels aiming to facilitate a wide variety of supervised and unsupervised analysis techniques. We have defined an electrical event for both coffeemakers individually, based on the key characteristics the acquired signals exhibit. For the *Jura GIGA X8*, an electrical event was defined as an increase in the current signal equal to approximately one ampere that lasted over a time frame of at least 1 s. In order to capture all significant events, there can be slight deviations from the event definition, as the data exhibits some variation that we also captured in the labelling process. In contrast, the *Jura GIGA X9* generated a vast amount of patterns with a shorter duration. Hence, to capture this behaviour, we have created two sets of electrical event labels for the *Jura GIGA X9*. The first set contains electrical events lasting over a time frame of at least 1 s, similar to the component events of the *Jura GIGA X8*, to enable comparative studies with the other coffeemaker. The second set of component events of the *Jura GIGA X9* extends the first one with events lasting at least 0.1 s. Thus, we have labelled 92,449 electrical events for the *Jura GIGA X8*. For the second coffeemaker, we have created 278,151 electrical events, including the 44,219 events labelled with a minimum duration of 1 s.

Among all registered electrical events, we created a subset that contained the expert-labelled information about the individual main components that had triggered these events. Furthermore, the two sets of the maintenance and product-related events that were automatically generated by the coffeemakers were specified for each of the coffeemakers individually. Due to the aforementioned granularity of one minute, these events were manually refined to match

the associated electrical signals as precisely as possible. The three sets of labels were constructed by applying a three-step labelling procedure conducted by three human experts, as outlined in Figure 2. The experts involved in the labelling procedure own a university degree in computer science and have vast experience in signal processing and machine learning, making them suitable for the task. We have ensured a consistent labelling of events by definition the key characteristics, as explained above, and by using example events from the data to guide the experts. The labelling tools allow for high precision labelling of the time series data, as shown in Figure 3. To reduce human labelling bias and to reduce errors, all events were peer-reviewed.

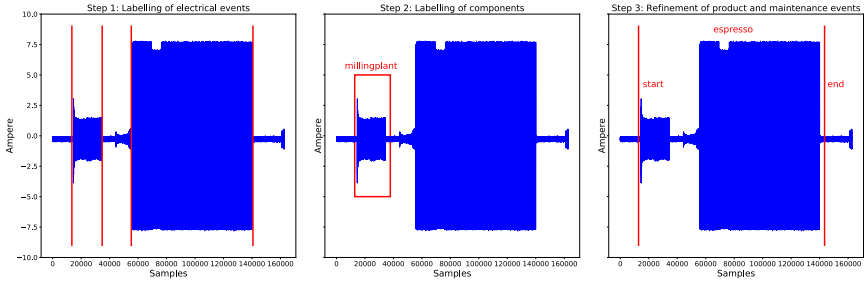


Figure 2: **Three-step labelling procedure.** The red, vertical lines denote the labels added at the respective step. In Step 1, the electrical events are labelled. In Step 2, events are assigned to the respective initiating component. In Step 3, the product and maintenance events registered with the one-minute granularity are precisely allocated in time.

In the first step, we hand-labelled the electrical events that were triggered by the main electricity consumers in the coffeemaker. For this purpose, we developed a labelling tool, that enabled the experts to inspect signal segments and mark potential events visually. The labelling procedure was established according to the previously stated event definition. Furthermore, an event had to exhibit a significant and re-emerging pattern. After completion of the labelling procedure by two of the experts, all generated labels were revised and corrected by the third expert. The vast amount of events could be used to develop and benchmark event detection algorithms on a high number of samples, in contrast to the existing datasets.

In a second step, we assigned a subset of the labelled events to the corresponding main component, as listed in Table 1, that had caused these events to occur. In this way, we aimed to facilitate the development of supervised machine learning algorithms requiring prior labelling to identify main components in the coffeemakers. Two of the main components, grinding motors and thermal heating blocks, are installed multiple times in the coffeemakers. In the labelling process, we were unable to distinguish the components of the same type visually certainly. Therefore, we summarised the main components from Table 1 according to the three following classes: *heater*, *millingplant*, and *pump*. The precision

SCIENTIFIC DATA

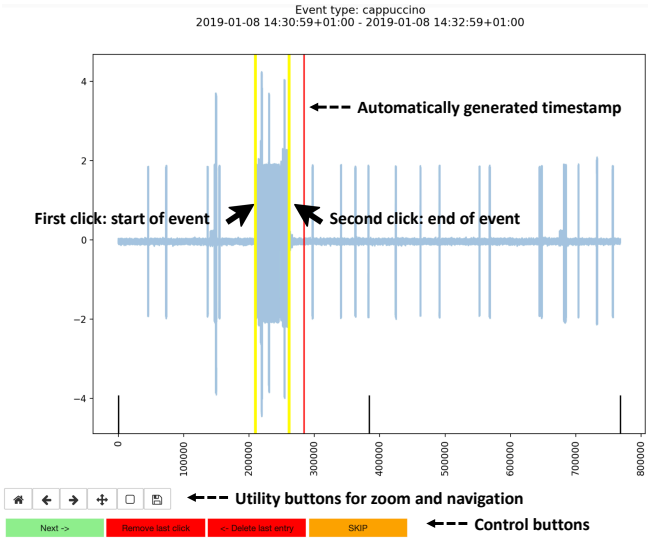


Figure 3: **Labelling tool for step 3 of the labelling procedure.**

ceramic disc grinding modules of the milling plants are powered by one motor each. The signals corresponding to these demonstrate a characteristically sharp spike after being switched on when a motor is initially accelerated. Afterwards, the amplitude slowly decreases when the motors settle into their steady-state. The grinder motors are the most prominent components in the coffeemakers, being clearly visible in the current signal. We considered the *MillingPlantEspresso* and *MillingPlantCoffee* events from the maintenance events list to obtain isolated milling events for the labelling process. After selecting a random subset from these events, the human experts manually labelled it using the labelling tool. For the *heater* events, we used the signals recorded on Saturdays. On these days, no products were generated, and no maintenance tasks were executed by the machines, as the locations where the coffeemakers were placed at were not occupied during weekends. Despite that, the machines were not switched off completely, and the installed pre-heating system periodically initiated the heating procedure to remain prepared for future provisioning. Therefore, we could observe the isolated heating events on these days, which facilitated labelling a larger sample of heater signals for the ground-truth. The labelling of pumps was performed using the *hot_ water* product events, as they involved no usage of grinders that infer with the pumping process. The *heater* components involved in the *hot_ water* process could be visually separated by the experts, as they steadily consumed the same amount of energy.

In the last step of the labelling procedure, depicted in Figure 3, we manually refined the automatically generated product and maintenance event timestamps. The machine-generated timestamps had a one-minute resolution in time and marked the completion of a given procedure. Therefore, we plotted the signals

SCIENTIFIC DATA

enhanced with the labels from Step 2. The human experts then manually specified the start and end timestamps for a considered event by investigating the signal in the window of interest around the automatically generated timestamp. All labelling tools are available in the provided repository [21].

Code availability

The source files for the data collection using the MEDAL measurement units is available in the BLOND data repository [20]. For completion, we have also added these files to the CREAM repository [21] in the *data_ collection* folder. This repository contains all the scripts used for the technical validation of the measurement hardware capabilities. The code to reproduce the extraction of the product and maintenance events through the serial maintenance ports of the coffeemakers is available in the coffeemaker project repository provided by Q42 [19]. We implemented the data processing, labelling tools, and utility functions in Python 3. The labelling tools were implemented in three Jupyter Notebooks, one corresponding to each step of the labelling pipeline. The individual source files are available in the CREAM repository [21]. All labelling steps can be fully reproduced and extended if necessary, using the supplied tools. Furthermore, we provide the utility class containing all necessary functions for loading and pre-processing the signals.

Known issues

The signals measured using the MEDAL system may introduce a slight direct current bias, occurring due to changes in the DC reference voltage and the use of a unipolar ADC. Appropriate signal calibration and filtering, as shown in the CREAM repository [21], can be applied to correct this issue during pre-processing [8]. Furthermore, the events represented in the CREAM dataset are imbalanced, as shown in Table 4 for the *Jura GIGA X8* and in Table 5 for the *Jura GIGA X9*. When evaluating the performance of algorithms on the dataset, it is necessary to adjust for this bias by applying appropriate techniques for the imbalanced data, such as oversampling.

In addition, it should be noted that due to customisation possibilities and due to unexpected user behaviour, such as aborting the coffee-making process, the event durations may vary, as shown in Table 4 and Table 5. This heterogeneity needs to be considered in the analysis, as the intra-class variance is high; namely, the signals for samples corresponding to the same type of event can deviate between each other.

The obtained voltage and current signals acquired are the aggregate ones corresponding to the individual component activities. Therefore, in the analysis, it is necessary to consider overlapping activities, such as heating and activating a milling plant.

SCIENTIFIC DATA

Event type	Samples	Mean	Standard deviation
cappuccino	521	50.26	3.51
coffee	361	29.43	3.24
espresso	313	23.43	3.08
hot_water	157	24.16	6.33
latte_macchiato	109	46.10	8.40
white_coffee	10	29.86	0.94
ristretto	3	21.29	1.99
espresso_macchiato	2	37.79	1.06
MillingPlantEspresso	1316	4.53	1.65
MillingPlantCoffee	1008	4.99	1.93
RinseMilkSystem	418	17.04	2.01
CleanMilkSystem	47	49.44	6.86
Time2Clean	47	2.56	7.67
RinseCoffeeSystem	22	76.50	28.54
Clean	9	106.75	10.51
Time2Descale	1	0.53	0

Table 4: **Statistics of the product and maintenance event durations of the *Jura GIGA X8***. The list shows the mean and standard deviation of event durations of the product and maintenance events produced by the *Jura GIGA X8*.

Event type	Samples	Mean	Standard deviation
cappuccino	95	36.20	20.10
espresso	58	18.26	8.54
coffee	47	24.91	12.12
hot_water	43	47.19	57.52
latte_macchiato	14	53.68	41.13
espresso_macchiato	1	36.19	0
MillingPlantEspresso	392	2.40	1.12
MillingPlantCoffee	209	2.34	1.03
RinseMilkSystem	117	8.73	11.14
CleanMilkSystem	20	24.08	23.34
Time2Descale	15	1.28	1.02
Time2Clean	14	1.03	0.82
RinseCoffeeSystem	8	15.70	16.394
Clean	3	154.00	55.31

Table 5: **Statistics of the product and maintenance event durations of the *Jura GIGA X9***. The list shows the mean and standard deviation of event durations of the product and maintenance events produced by the *Jura GIGA X9*.

SCIENTIFIC DATA

Data Records

The CREAM [21] dataset contains the three measured signals generated by each of the coffeemakers: the voltage, current and background-noise signal registered by a socket in the MEDAL measurement unit. Furthermore, it comprises the labels of electrical components, as well as the information about the product and maintenance events. The dataset is divided into two subfolders, one for the *Jura GIGA X8* and one for the *Jura GIGA X9* respectively.

Data files

All signals were sampled with 6,400 samples per second at the mains frequency of 50 Hz. The signals obtained from the sensor input were stored as-is: in particular, no dedicated pre-processing of the raw signals was performed to ensure unbiased analysis of the data. In the CREAM repository, we provide examples of possible pre-processing steps [21]. The dataset was structured with respect to the individual days of recording so that one subfolder contains the data files for each day in the data acquisition process. The raw data and the metadata were stored in HDF5 files. The utilised data formats and the metadata are similar to the ones used in the BLOND office environment dataset, as the MEDAL hardware was used in the latter as well.

Functionality to process this type of file is available in a variety of open-source and commercially available tools, making them easily accessible [8]. Into each of the HDF5 files, we embedded the corresponding file metadata in the form of HDF5 attributes that could be accessed either directly in the file root or in a specific HDF5-dataset, as described in Table 6. The value types of the data are either short integer, floating point or ASCII-encoded byte strings. Parts of the metadata information is also encoded in the file names, for example, *coffeemaker-2018-08-23T07-00-03.783395T+0200-0000001.hdf5*: The first sample of this file was recorded approximately at 07:00 23 October 2018, with a time zone offset of 2 hours. Furthermore, each file within a day has a sequence number, such as the sequence number 1, as represented in the example file name. The sequence number uniquely identifies the file order within a particular day. All timestamps in CREAM, in particular, the ones from the labels and from the data recordings, are synchronised. In the CREAM repository, we provide the examples for handling timestamps and time zone information [21]. Each HDF5 file contains one hour of data, and each day of CREAM, except the last one, contains sixteen HDF5 files, with the first file starting at approximately 06:00, and the last file ending at 22:00, covering the usual working hours. No daylight saving time transitions or leap seconds have occurred during the process of recording. Therefore, one can fully rely on the timestamps provided in the data. The MEDAL units automatically create the one hour file chunks, while measuring the electrical signals without interruptions at 6,400 sps.

SCIENTIFIC DATA

Path	Attribute	Description
/	name	Name of the measurement unit
/	first_trigger_id	Internal trigger number to detect gaps
/	last_trigger_id	Internal trigger number to detect gaps
/	sequence	day-internal sequence number
/	frequency	nominal samples per second
/	year	Year of this file
/	month	Month of this file
/	day	Day of this file
/	hours	Hours of first sample
/	minutes	Minutes of first sample
/	seconds	Seconds of first sample
/	microseconds	Microseconds of first sample
/	timezone	Timezone offset
/<dataset>	calibration_factor	Factor for signal calibration
/<dataset>	removed_offset	Removed DC-offset

Table 6: **HDF5 file metadata.** The metadata attributes are accessible via a HDF5-attribute-path. All physical values are provided in base units (Volt, Ampere, Hertz), and the timestamp information refers to the first sample in the respective data file. The *<dataset>* placeholder can be either *voltage*, *current1* for the coffeemaker's current from socket 1, or *current6* for the socket 6 background-noise current.

Labels

The labels resulting from the labelling procedure represented in Figure 2 are stored as comma-separated value (csv) files in the sub folder of the respective coffeemaker. All label timestamps have the following format: *year-month-day hours:minutes:seconds.microseconds+timezone*. The electrical component events are stored in the *component_events.csv* file for the *Jura GIGA X8*, as described in Table 7. In contrast, there are two component event files in the the *Jura GIGA X9* subfolder, one for the previously defined minimum duration of the electrical events. The 1 s events are stored in the *component_events_coarse* CSV file and the 0.1 s events in the *component_events_fine* CSV file. The fine-grained events of the *Jura GIGA X9* can be matched with the corresponding coarse events, using the *ID* column of the label files. The events are either turn-on (*On*) or switch-off (*Off*) events. The *On* / *Off* information was determined automatically, by comparing the mean power in a 0.1 s window before the event and 0.1 s after the event occurs. If the mean power before the event is lower than afterwards, we labelled the event to be an *On* event. On the other hand, *Off* events exhibit a drop in power in between the pre-event and the post-event window. As stated before, we assigned one of the three components (heater, millingplant, or pump) to a subset of the events. The events without a component label are declared as *unlabeled* in the respective column.

SCIENTIFIC DATA

Column	Description
Filename	File name containing the event
Timestamp	Event timestamp
Amplitude	Current value (ampere) of event
Event_Type	*On* or *Off* event
ID	Unique event identifier, sequentially numbered
Component	Name of event invoking component

Table 7: **Description of the component events files.** Columns of the files containing the electrical component timestamps and supplementary information, such as the amplitude of the current drawn.

Column	Description
Start_Timestamp	Start time of event
End_Timestamp	End time of event
Automatic_Timestamp	Original automatically generated timestamp
Event_Type	Product name or maintenance activity
Event_Duration_Seconds	Seconds between start and end timestamp
Date	Format: *year-month-day*

Table 8: **Description of the product and maintenance event files.** Columns of the *product_events.csv* and *maintenance_events.csv* files. The files contain the timestamps of the start and end of each event, resulting from the refinement in Step 3 of the labelling process.

The refined product and maintenance events are stored in the respective *.csv* files. These files have the same column structure, as shown in Table 8.

The *Event_Type* column represents the product events from Table 2 or the maintenance events from Table 3, respectively. The timestamps in the *Automatic_Timestamp* column correspond to the one-minute granularity timestamps that were automatically generated by the machines.

The refined automatic timestamps from Step 3 of the labelling procedure, as stated in the corresponding description before, are stored in the *Start_Timestamp* and *End_Timestamp* columns. As a result of using the coffeemakers in an office building, its energy patterns differ considerably between working and non-working days. Therefore, we include an additional CSV-file for each coffeemaker, namely, the *day_information.csv*, to provide this information, as shown in Table 9.

In addition to the labels that were generated as a result of the labelling procedure, we also include the raw label files of the product and maintenance events automatically generated by the coffeemakers in the *raw_coffee_maker_logs* subfolders. These files contain the one-minute granularity timestamp, and the columns named *Activity* corresponding to the maintenance events or *Product* for the product events, accordingly.

SCIENTIFIC DATA

Column	Description
Date	Format: *year-month-day*
WorkingDay	*True* if working day, *False* if not
Weekday	Day of the week

Table 9: **Description of the day and date information files.** Columns of the *day_ information.csv* files that contain the information about working days in Germany and the weekday information for all days in the dataset, for each of the coffeemakers.

Technical Validation

Signal acquisition

The data collection capabilities of the MEDAL system were thoroughly evaluated concerning the long term measurements presented in the BLOND dataset. In the following subsection, we describe the major characteristics of the hardware technical validation. Additional details can be found in the corresponding data descriptor of the BLOND dataset [8]. We applied the same data sanity checks as the ones implemented for the BLOND dataset collection. The data acquisition unit was used to perform the checks aiming to detect continuity and transmission errors [8]. Furthermore, each file created during a day has a unique sequence number to detect gaps in recordings. To perform offline verification, each HDF5 file included two trigger IDs in its metadata, as presented in Table 6, aiming to ensure a continuous and uninterrupted signal. No discontinuities were presented in CREAM, according to the utilised sequence numbers. MEDAL recorded the signals with the fixed nominal sampling rate of 6,400 sps. The actual sampling rate could differ from the nominal one due to minor deviations corresponding to the MEDAL's internal oscillator that was used to control the ADC conversion [8]. Based on the analysis conducted for the BLOND dataset [20], we analysed the average sampling rate in the data obtained per day. The results of the analysis were in-line with the ones published for the BLOND dataset, indicating that concerning CREAM, the actual sampling rate did not differ from the nominal one. Furthermore, we performed additional sanity checks per file, implemented on the basis of the ones outlined for BLOND. We checked the dataset for completeness, considering the expected number of samples per file, the number of files, and the number of days in the dataset. The analysis results confirmed that no gaps were presented in the dataset, and the data for all days in the considered period were recorded appropriately. The nominal mains frequency of the electrical network was 50 Hz. We estimated the actual mains frequency based on the voltage signal by selecting the strongest bin in a fast Fourier transform. Deviations from the nominal frequency could indicate malfunctions of the ADC [8]. However, no difference in the frequency was observed. In addition, we implemented the checks to control multiple parameters of the voltage, and current signals, such as the root mean squared (RMS) values. The parameters and their corresponding thresholds are

SCIENTIFIC DATA

Parameter ϕ	**Value range of ϕ**
Voltage RMS	$210 <= \phi <= 240$
Voltage mean	$0 <= \|\phi\| <= 5$
Voltage crest factor	$1.2 <= \phi <= 1.6$
Voltage value range	$\phi >= 2000$
Voltage bandwidth	$\phi >= 50$
Voltage minimum	$-300 <= \phi <= -355$
Voltage maximum	$300 <= \phi <= 355$
Current RMS	$0 <= \phi <= 16$
Current mean	$0 <= \|\phi\| <= 1$
Current crest factor	$\phi >= 1.2$

Table 10: **Validated voltage and current parameters.** Based on the technical validation performed for the MEDAL units in the BLOND dataset [8], we validated the signal with respect to the parameters listed in this table. The parameter ϕ needs to be within the specified value range to pass the validation.

provided in Table 10. The parameters were checked for both current and voltage signal, and as a result, we observed that the tests were passed successfully for all files. In addition to these checks, we validated whether the signals contained flat regions with individual periods consisting only of constant values. The scripts to reproduce the data sanity checks are provided in the CREAM repository [21], in the *technical_validation* subfolder.

Label validation

We applied several measures to ensure the appropriate labelling quality throughout all steps of the labelling procedure, as outlined in Figure 2. The main validation component was a double-review of all labelled events. Therefore, all event labels were at least checked by two experts independently. In the case of errors or inaccuracies, the labels were corrected by the reviewer. During the initial labelling, examples of existing event types were provided to guide the experts through the process. We established the labelling notebooks to prevent labelling errors by introducing pre-labelled event examples to the experts. The electrical event labels from Step 1 are uniformly distributed over the day, and no gaps exist, as shown in Figure 4 for both coffeemakers.

In Step 2 of the labelling process, we assigned each component a subset of the corresponding electrical events. Figure 5 represents the mean instantaneous power consumed by every component, grouped by the corresponding coffeemaker. Due to imbalance in the number of labelled components, we subsampled 100 of them to obtain comparable values. The characteristics of the components differ between the two coffeemakers. The components of the *Jura GIGA X9* are often triggered simultaneously, as the uniform distribution of the instantaneous power consumed shows. This raises the demand for energy disaggregation algorithms to filter out the individual components from overlapping

SCIENTIFIC DATA

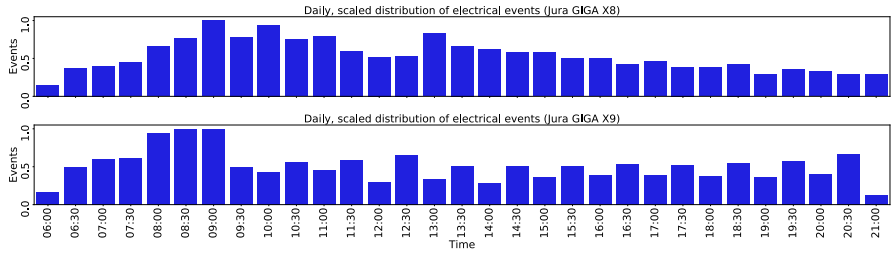


Figure 4: **Daily distribution of electrical events.** The upper figure shows the scaled event distribution of the *Jura GIGA X8*, whereas the bottom one visualises the event distribution of the *Jura GIGA X9*. Accumulated over all days in the dataset, the distribution of events is balanced with peaks in the morning and after lunch, as expected in the office environment.

signal segments. In contrast, most of the labelled the component patterns in the *Jura GIGA X8* exhibit a uniform power consumption pattern, except for a few outliers. Similar to the *Jura GIGA X9*, deviations from the mean occur when the components are activated simultaneously with the other active ones. Consequently, the signals from individual components superimpose each other. When analysing the duration of the refined product and maintenance events

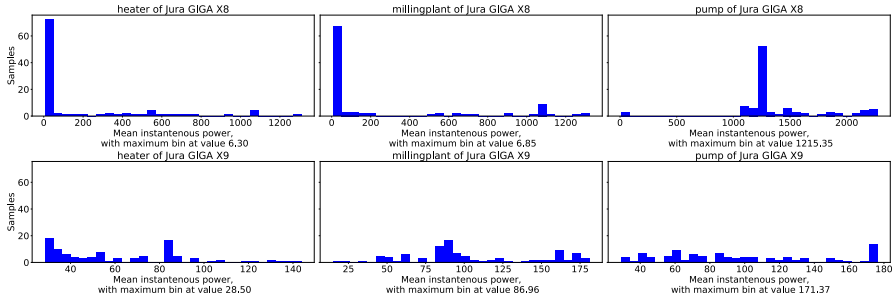


Figure 5: **Mean instantaneous power consumed by each main component**. We computed the power for a subset of 100 samples per component to address the class imbalance. The first row of the figure contains the components of the *Jura GIGA X8* and the second one the ones of the *Jura GIGA X9*. For the *Jura GIGA X8*, each component has a clear peak power corresponding to the consumption of the majority of events, whereas the *Jura GIGA X9* exhibits more uniform characteristics.

of the *Jura GIGA X8* obtained at Step 3 of the labelling process, one can see in Table 4 that most of them have a small variation and are labelled with a uniform length in time. In comparison, the duration of the events created by the *Jura GIGA X9* exhibit a higher variation. The deviations can be observed due to various reasons, such as changes in the coffeemaker settings, processes

SCIENTIFIC DATA

```python
import h5py

file_path = "filepath_of_interest" # data location

# Open the file
with h5py.File(file_path, 'r', driver='core') as f:

    """
    Extract the signal.
    The file contains the "current1" coffeemaker channel, the "current6" noise channel, and the "voltage" data.
    """
    signal = f["current1"][:]

    """
    Extract the metadata attributes.
    The "calibration_factor" attribute can be replaced by any another one.
    """
    calibration_factor = f[name].attrs["calibration_factor"]
```

Listing 1: **Exemplary usage of the h5py python library for extracting the electrical signals of the coffeemakers and the metadata from the HDF5 data files.** In the CREAM repository [21], we provide pre-built functions for reading and processing the full dataset.

that differ between the coffeemakers or diverging user behaviour. The labelling procedure itself was precise, as confirmed by visual inspection.

Usage Notes

In the CREAM repository [21], we provide the code to reproduce the dataset establishment and the examples that can be used to facilitate the usage of the dataset. The code is provided in the *source_code* folder. All source code is implemented in Python 3. The recorded electrical signals are stored in files in the format of HDF5 files. The HDF5 format is supported by most of the scientific computing libraries, such as Python (h5py/numpy/scipy), MATLAB (h5read), and R (rhdf5). The code snippet in Listing 1 shows the usage of the h5py python library for extracting the data.

We provide the relevant metadata in HDF5 attributes within the individual files and the respective filenames, as documented in Table 6. While creating the HDF5 files, we have used the following widely supported filters: gzip compression, shuffle to improve the compression ratio, and Fletcher to add checksums to prevent the data from being corrupted. The repository contains examples of loading and pre-processing the CREAM data. Furthermore, we provide the labelling tools utilised to produce the labels, as outlined in Figure 2 and as shown in 3. Therefore, the created labels can be reproduced independently. Moreover, the set of existing labels can be extended if necessary. In CREAM, we provide

the raw measurements to avoid any bias caused by data pre-processing. Despite that, we recommend applying two pre-processing steps for most of the potential analysis techniques. First, we recommend to calibrate the signals according to the calibration factors provided in the file metadata (see Table 6). Second, we suggest removing any DC-bias by subtracting the mean offset from each mains-cycle in the signal. We provide the implementations for both pre-processing steps in the repository, according to the instructions outlined in the BLOND repository [20].

Acknowledgements

This research was supported by the Federal Ministry for Economic Affairs and Energy based on a decision by the German Bundestag and the Technical University of Munich (TUM) within the funding programme Open Access Publishing. We would like to thank the company All for One Steeb AG for providing access to coffeemaker logs and data. Furthermore, we want to thank the company Q42 for publicly providing the code for accessing the serial port of the coffeemakers. The icons used in Figure 1 are provided by the platform icons8 (https://icons8.de/) for free usage.

Author Contributions

Daniel Jorde designed and performed the dataset collection, technical validation steps and developed the software code and the manuscript. Thomas Kriechbaumer developed the MEDAL measurement unit and set up the necessary data collection infrastructure. Tim Berger and Stefan Zitzlsperger, together with Daniel Jorde, were responsible for the labelling of the events. Hans-Arno Jacobsen incepted the research project, lead and oversaw it, and provided conceptual guides while supervising Daniel Jorde's Ph.D.

Competing Interests

The authors declare no competing interests.

References

[1] Colombo, A. W., Karnouskos, S., Kaynak, O., Shi, Y. & Yin, S. Industrial Cyberphysical Systems: A Backbone of the Fourth Industrial Revolution. *IEEE Industrial Electronics Magazine* **11**, 6–16 (2017).

[2] International Energy Agency. World Energy Outlook 2019. `https://www.iea.org/reports/world-energy-outlook-2019` (2019).

[3] Condition Monitoring and Diagnostics of Machines — General Guidelines. Standard, International Organization for Standardization, Geneva, CH (2018).

[4] DeNucci, T. *et al.* Diagnostic Indicators for Shipboard Systems using Non-Intrusive Load Monitoring. In *IEEE Electric Ship Technologies Symposium*, 413–420 (IEEE, Piscataway, NJ, 2005).

[5] Suzuki, R., Kohmoto, S. & Ogatsu, T. Non-Intrusive Condition Monitoring for Manufacturing Systems. In *2017 25th European Signal Processing Conference (EUSIPCO)*, 1390–1394 (2017).

[6] Zoha, A., Gluhak, A., Imran, M. A. & Rajasegarar, S. Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey. *Sensors* 16838–16866 (2012).

[7] Makonin, S. & Popowich, F. Nonintrusive Load Monitoring (NILM) Performance Evaluation. *Energy Efficiency* **8**, 809–814 (2014).

[8] Kriechbaumer, T. & Jacobsen, H.-A. BLOND, a Building-Level Office Environment Dataset of Typical Electrical Appliances. *Scientific Data* **5** (2018).

[9] Armel, C., Gupta, A., Shrimali, G. & Albert, A. Is Disaggregation the Holy Grail of Energy Efficiency? The Case of Electricity. *Energy Policy* **52**, 213–234 (2013).

[10] Martins, P. B. M., Gomes, J. G. R. C., Nascimento, V. B. & de Freitas, A. R. Application of a Deep Learning Generative Model to Load Disaggregation for Industrial Machinery Power Consumption Monitoring. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 1–6 (IEEE, Aalborg, 2018).

[11] Kahl, M. *et al.* Measurement System and Dataset for in-depth Analysis of Appliance Energy Consumption in Industrial Environment. `https://www.in.tum.de/i13/resources/lilacd/` (2019).

[12] Agogino, A. & Goebel, K. Milling Dataset. `https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/` (2007).

[13] Saxena, A. & Goebel, K. Turbofan Engine Degradation Simulation Data Set. `https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/` (2008).

[14] Condition Monitoring of Hydraulic Systems Data Set. `https://archive.ics.uci.edu/ml/datasets/Condition+monitoring+of+hydraulic+systems` (2018).

SCIENTIFIC DATA

[15] Kriechbaumer, T., Ul Haq, A., Kahl, M. & Jacobsen, H.-A. Medal: A cost-effective high-frequency energy data acquisition system for electrical appliances. In *Proceedings of the Eighth International Conference on Future Energy Systems*, e-Energy '17, 216–221 (Association for Computing Machinery, New York, NY, USA, 2017).

[16] JURA Elektroapparate AG. Giga X8 professional datasheet. `https://www.jura.com/-/media/global/pdf/manuals-global/professional/GIGA-X8-G2/download_manual_jura_giga_x8.pdf` (2020).

[17] JURA Elektroapparate AG. Giga X9 professional datasheet. `https://us.jura.com/-/media/global/pdf/manuals-global/professional/GIGA-X9/download_manual_jura_giga_x9.pdf` (2020).

[18] Microchip Technology Inc. Mcp3201 datasheet. `http://ww1.microchip.com/downloads/en/DeviceDoc/21290D.pdf` (2007).

[19] Q42. Coffee hack. `https://github.com/Q42/coffeehack` (2017).

[20] Kriechbaumer, T. Kriechi/building-level-office-environment-dataset: v1.0. *Zenodo* `https://doi.org/10.5281/zenodo.838974` (2017).

[21] Jorde, D. & Jacobsen, H. CREAM: CoffeemakeR Electrical Activity Measurements for Condition Monitoring. *Technical University of Munich* https://doi.org/10.14459/2020mp1554766 (2020).

B MEED: An Unsupervised Multi-Environment Event Detector for Non-Intrusive Load Monitoring

This is the author's accepted version of an article that has been published at the 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). Changes were made to this version by the publisher prior to publication. The final version of record is available at https://doi.org/10.1109/SmartGridComm.2019.8909729

MEED: An Unsupervised Multi-Environment Event Detector for Non-Intrusive Load Monitoring

Daniel Jorde, Matthias Kahl, and Hans-Arno Jacobsen
Chair for Application and Middleware Systems
Technische Universität München, Germany
Email: daniel.jorde@tum.de

***Abstract*—The accurate detection of transitions between appliance states in electrical signals is the fundamental step that numerous energy conserving applications, such as Non-Intrusive Load Monitoring, rely on. So far, domain experts define rules and patterns to detect changes of appliance states and to extract detailed consumption information of individual appliances subsequently. Such event detectors are specifically designed for certain environments and need to be tediously adapted for new ones, as they require in-depth expert knowledge of the environment. To overcome this limitation, we propose a new unsupervised, multi-environment event detector, called MEED, that is based on a bidirectional recurrent denoising autoencoder. The performance of MEED is evaluated by comparing it to two state-of-the-art algorithms on two publicly available datasets from different environments. The results show that MEED improves the current state of the art and outperforms the reference algorithms on a residential (BLUED) and an office environment (BLOND) dataset while being trained and used fully unsupervised in the heterogeneous environments.**

I. Introduction

One of the challenges humanity is facing nowadays is the depletion of natural energy resources, while the overall energy demand keeps increasing, especially the demand for electrical energy [1]. For this reason, researchers are striving to find solutions to improve the way limited energy resources are used. By making both industrial and residential consumers aware of their detailed electricity consumption, one aims to reduce the waste of energy [2]. Several surveys indicate that appliance-level information can reduce energy consumption by raising consumer awareness [3]. The consumption of individual appliances can be acquired using Non-Intrusive Load Monitoring (NILM) methods with a low-cost single-sensor approach to record an aggregated signal, measured only at the mains of a building or industrial plant [2]. After extracting relevant signal segments using an event detection algorithm, the appliances that had caused the events can be identified and the signal can be decomposed into the individual appliances. Besides using appliance-level information to save energy, it enables other applications, such as detecting malfunctions in appliances to reduce maintenance costs [2].

Most of the electrical data used in NILM is collected by smart meters, which usually sample the signals at a low rate (< 1 kHz). As a result, only some of the major devices can be detected [2]. Data that are sampled using higher rates increase the probability for successful NILM [4] and allow to detect more devices, especially low consumption ones [2]. Although high-sampling-rate data contains a high amount of information, most machine learning algorithms can not be used on it as they suffer from the curse-of-dimensionality caused by the sampling rate [5]. Hence, it is necessary to reliably extract relevant segments from the overall signal, i.e., appliance-state transitions, that can be used to identify appliance-level behavior.

In the past, researchers focused on methods for low-sampling-rate data, driven by the high costs (for metering, storage, and processing) associated with the acquisition of high-sampling-rate data. As there is an evident lack of methods and because of the advantages of high-sampling-rate data, we focus on this domain. Detecting events and distinguishing them from signal noise is particularly challenging and prone to errors. So far, researchers focused mainly on residential buildings and their appliances. Thus, there are multiple residential datasets publicly available [6]. Recent work also investigates industrial and office settings and new datasets are published [7].

Existing event detection algorithms exhibit one important disadvantage: Most rely on customized, expert-made event definitions. This prohibits such approaches to generalize well to a setting they were not designed for. Subsequently, the hyperparameters and the algorithms themselves need to be tediously fine-tuned for being used in a new setting.

The main contribution of this paper is to present a new multi-environment event detector (MEED) that does not rely on a dedicated event definition while being trained and used fully unsupervised. Hence, MEED can be used in different environments without the need to preset additional hyperparameters, enabling new possibilities for NILM and energy-related applications in general. Furthermore, MEED detects events more reliable than the existing state of the art, inducing fewer errors into subsequent analysis steps. We compare MEED with two state-of-the-art algorithms on two publicly available datasets from different environments. By doing so, we improve the current state of the art on the residential BLUED [6] and the office environment BLOND [7] datasets.

The rest of the paper is organized as follows: In Section II we give an overview on event detection in NILM and relevant metrics. Section III summarizes related work, followed by the description of MEED in Section IV. Subsequently, we detail our experimental setup in Section V and discuss the corresponding results in Section VI. Section VII then concludes this paper.

II. Background

Based on the foundational work on energy disaggregation by George Hart [8], multiple new algorithms that use events to extract relevant signal segments have been proposed.

A. Event Detection

As the majority of researchers use different, specific definitions of events that reduce their capability to generalize to new settings, we use a general event definition. In particular, we define events to be transitions between states of individual appliances. Event detection algorithms can be divided into ones using supervised- and unsupervised learning, depending on the use of labeled data during training (supervised) or not (unsupervised). Supervised algorithms are less flexible in tasks like event detection than unsupervised algorithms, as they are highly dependent on the event definition used. Thus, we propose an unsupervised event detector.

Another possibility to classify event detection methods are the three categories introduced by Anderson et al. [9], namely Expert Heuristics (EH), Probabilistic Models (PM), and Matched-Filters (MF). Rule-based approaches, including simple threshold-based ones, and methods using machine learning are considered to be EH, whereas approaches using statistical metrics to determine events belong to the PM category. MF approaches match learned event masks with the signal to detect events [9].

B. Metrics

The following commonly used metrics are based on the scores of the confusion matrix, namely the amount of records that are True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

$$\text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \quad \text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP}+\text{TN}} \quad \text{FPP} = \frac{\text{FP}}{\text{TP}+\text{FN}}$$

$$\text{F1-Score} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

Another common metric is the True Positive Percentage (TPP), which is defined as $\frac{\text{TP}}{\text{E}} = \frac{\text{TP}}{\text{TP}+\text{FN}}$ [9]. As this is equal to the recall metric, we omitted the TPP in our evaluation. It is of particular importance to define how the single scores of the confusion matrix are computed to ensure the comparability of the results [10]. Ground truth events are often generated by human experts, and thus can be imprecise with respect to their exact location in time. Hence, it is common to define a tolerance limit τ for the matching of detected e_{det} and ground truth events e_{gt} [10]–[12]. A detected event e_{det} is a TP if there exists a ground truth event within an interval of $\pm\tau$, i.e., if $\exists e_{gt} : e_{det} - \tau \leq e_{gt} \leq e_{det} + \tau$.

III. Related Work

In the following, we summarize the current state of the art in event detection in NILM on high-sampling-rate data and identify gaps. The essential characteristics of the related algorithms are listed in Table I, with the scores being rounded to the second decimal digit. In case the exact value for a particular metric is unclear from the publication, we declared the result to be approximate (≈). We further categorize the related work according to the following criteria. The "Setting" column indicates the environment the algorithms are designed for and evaluated in, namely Residential (R) or Industrial (I) environments. The cross-validation criterion (CV) reports whether cross-validation was used in the evaluation, as suggested by Makonin and Popowich [25] to improve the reliability of the results. Most of the publications exhibit no information about the tolerance limit τ used to calculate the metrics. Hence, the scores can not be directly, but only approximately, compared.

Valovage and Gini introduce a PM [11] that relies on a Bayesian detection method at its core. The algorithm tries to partition the signal into run sequences, followed by declaring the transitions between such sequences as events.

Pereira [12] introduces another PM. The algorithm makes use of a log likelihood ratio detector to estimate a detection statistic. The algorithm then searches the signal for extreme values to determine events.

In contrast to the other PM, Wild et al. [17] use a kernel Fisher discriminate analysis to detect start and end times of event and non-event segments in a supervised way.

Alcala et al. [13] use the signal's envelop of the normalized current and voltage RMS values and a threshold to detect events. Another approach using the signal's envelope is based on a Hilbert Transform [14]. It further applies an average and a derivative filter to obtain a set of spikes to detect transitions.

The algorithm proposed by de Baets et al. [15] transforms the signal into the frequency domain and applies a threshold on the computed Cepstrum components to detect events.

As part of an unsupervised NILM system, Barsim et al. [16] developed a three-step unsupervised event detection algorithm. First, they separate steady and transient states by applying the mean-shift clustering algorithm to logarithmized real and reactive power values. Afterward, they use expectation-maximization clustering and Gaussian mixture models to detect the time limits of the events. In the last step, the algorithm verifies the detected events according to expert-defined constraints for the specific setting.

The approach by Trung et al. [20] makes use of the cumulative sum statistics (CUMSUM) and an adaptive threshold to detect the start and the end of the transient segments in the signal. Furthermore, Zhu et al. [22] use CUMSUM to detect events in a residential setting.

In contrast to methods designed for residential settings, Cox et al. [24] use the voltage distortion to detect events in industrial data.

Yang et al. [23] compare an EH and a goodness-of-fit (GOF) event detector, concluding that the latter performs better.

Barsim and Yang [18] use the unsupervised DBSCAN clustering algorithm. Zheng et al. [19] also show the applicability of DBSCAN to detect events. The clustering algorithm detects various appliance states and their transitions before they are post-processed with dedicated thresholds on power and time to

TABLE I
RELATED WORK (ACRONYMS ARE EXPLAINED IN THE TEXT)

Ref.	Learning	Setting	Type	Dataset	CV	Recall	Precision	FPR	FPP	F1-Score	Limit τ
[11]	Unsup.	R	PM	BLUED A | B	No	-	-	-	-	$\approx$ 0.98 | $\approx$ 0.80	2 s
[12]	Unsup.	R	PM	BLUED A | B	No	-	-	-	-	$\approx$ 0.96 | $\approx$ 0.72	1 s
				REDD	No	-	-	-	-	$\approx$ 0.80	-
[13]	Unsup.	R	EH	BLUED A | B	No	0.94 | 0.88	-	0.88e-3 | 0.12	-	-	-
[14]	Unsup.	R	EH	REDD subset	No	0.93	-	0	0	-	-
[15]	Sup.	R	EH	BLUED A | B	Var.	-	-	-	-	0.98 | 0.80	-
[16]	Unsup.	R	EH	BLUED A | B	No	0.99 | $\approx$ 0.70	-	-	$\approx$ 0.55 | $\approx$ 0.09	-	-
[17]	Sup.	R	PM	BLUED A | B	No	0.99 | 0.86	0.99 | 0.92	-	-	0.99 | 0.89	-
[18]	Unsup.	R	EH	BLUED A | B	No	0.97 | 0.68	0.99 | 0.93	-	0.78e-2 | 0.49e-1	0.98 | 0.79	-
[19]	Unsup.	R	EH	BLUED A | B	No	0.99 | 0.88	-	0.99 | 0.69	0.71e-2 | 0.4	0.99 | 0.77	3 s
[20]	Unsup.	R	EH	REDD subset	No	0.94	-	-	-	-	-
[21]	Unsup.	R	PM	Non Public	-	-	-	-	-	-	-
[22]	Unsup.	R	EH	Non Public	-	-	-	-	-	-	-
[23]	Unsup.	R	PM, EH	Non Public	-	-	-	-	-	-	-
[24]	Unsup.	I	EH	Non Public	-	-	-	-	-	-	-

filter out duplicates and FPs. For its transparent evaluation and its state-of-the-art performance on BLUED, we have selected the latter approach as one of the two algorithms to benchmark the performance of MEED.

In addition to the first reference algorithm, we have selected the GOF based approach by Jin et al. [21] as it has multiple advantages over other probabilistic event detectors, despite not being evaluated on a public dataset. In contrast to other algorithms, this GOF approach provides a guideline for the selection of the event window hyperparameter and a closed form for the decision threshold. Furthermore, the authors show the superiority of their algorithm over the widely used generalized log likelihood detector [21].

Based on the related work, the base requirements for MEED are as follows: MEED has to be designed independent of any specific event definition, allowing it to generalize better, even to different settings. Furthermore, this multi-environment event detector has to identify events unsupervised to avoid the need for costly human labeled events.

IV. EVENT DETECTION APPROACH

Based on the introduced requirements, we design a new window-based EH event detector. MEED applies a two-step process to the pre-processed input data to detect events, as shown in Figure 1. The first step is based on the mean-squared reconstruction error (MSE) of the signal window, which is computed by an autoencoder. Subsequently, we apply a peak detection algorithm to find the exact timestamps within the windows that exceed an absolute MSE value.

A. Data Pre-Processing

One of the features that is used in multiple event detection methods is CUMSUM [20], [22] as it has shown to reveal trends and changes in time-series data successfully. Small fluctuations are suppressed in CUMSUM, while substantial changes in the data are revealed. The input to the first step of MEED is the CUMSUM of the five-period root-mean-square (RMS) of the current signal. The CUMSUM at time t_i is computed by accumulating the deviations of the RMS values from the window's mean value for all $t \leq t_i$. Therefore, CUMSUM reveals trends in the signal, while suppressing small fluctuations. During training, we scale the CUMSUM input signal to a value range between -1 and 1.

B. Coarse Event Detection Autoencoder

The first step of MEED is to apply a denoising autoencoder to the input windows. In doing so, we seize the rarity of events to build a model that can detect signal windows that contain transitions. A similar idea is used to solve classical anomaly detection tasks, as presented in the summary on this topic by Chandola et al. [26]. The authors present several methods to identify unexpected events. One of these methods applies a neural network, in particular, an autoencoder, to learn a hidden representation h of the input x to reveal events [26].

Autoencoders are neural networks that are typically used in representation learning tasks. They consist of an encoder $h = f(x)$ and a decoder function $r = g(h)$ [5]. As events are rare compared to non-events, they constitute the minority class, making it hard for the model to learn a proper event representation h. Instead, the model learns to represent and reconstruct the non-event majority class. Hence, a large reconstruction error is produced when the model faces a deviation from normal behavior, i.e., an event. MEED's hyperparameter settings can generalize well between environments, as events rarely occur in all electrical environments. Denoising autoencoders, a special case of the general autoencoder, minimize a loss function $L(x, g(f(\widetilde{x})))$, while trying to reconstruct the input x from intentionally corrupted input data $\widetilde{x}$. The corrupted input $\widetilde{x}$ is obtained by adding white Gaussian noise ($\mu = 0$, $\sigma = 0.25$) to x.

Each of the three hidden layers of MEED are bidirectional long short-term memory (LSTM) cells that have shown to be able to learn long term dependencies and patterns from sequential data [5]. Briefly, a LSTM cell is a memory cell with non-linear gating units (i_t, o_t, f_t and c_t) that control its information flow, as summarized in the following equations,

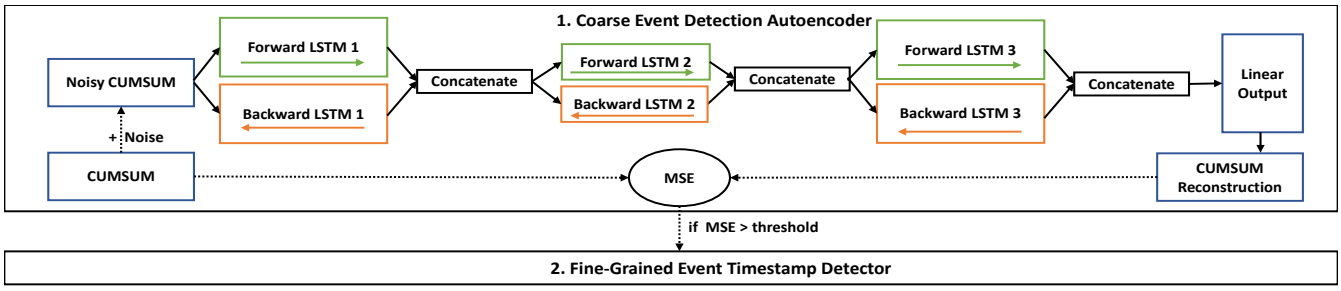


Fig. 1. MEED Architecture, with step 1 (the coarse detection autoencoder) and step 2 (the fine-grained event timestamp detector)

with sig being the logistic sigmoid, $tanh$ the hyperbolic tangent function, and $\circ$ the element-wise product [5]:

$$
\begin{aligned}
f_t &= sig(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= sig(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= sig(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ tanh(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ tanh(c_t)
\end{aligned}
$$

The input window to MEED has a fixed size, thus, we use bidirectional LSTMs to make use of future context information to improve the overall performance [27], as depicted in the autoencoder step in Figure 1. The bidirectional networks consist of one LSTM to process the data in the forward direction, i.e., from the beginning of the window to the end, and one separate LSTM to do so in the opposite direction [27]. The outputs of both directions are concatenated into a single output to compute the MSE to optimize the network. On top of the LSTM layers, the linear output layer ensures that the in- and output have the same dimensionality. We evaluated several hyperparameters and different architectural components using a grid search on the training data, resulting in the final optimal settings shown in Table II.

TABLE II
AUTOENCODER HYPERPARAMETERS, IMPLEMENTED WITH KERAS [28]

Parameter	Value	Parameter	Value
learning rate	0.001	LSTM 1 cell size	216
optimizer	Adam [29]	LSTM 2 cell size	108
initializer	Glorot uniform [30]	LSTM 3 cell size	216
merge mode	concatenate	window size	10 s

C. Fine-Grained Event Timestamp Detector

In case the coarse event detector determines an input window to be an event window, i.e., the reconstruction error threshold is exceeded, we apply the fine-grained detection procedure to determine the exact number and timestamps in the window. The algorithm aims to detect significant state-changes, while supressing ones that are caused by noise. We compute the RMS values over five periods of the raw input signal. Then, we convert the RMS signal into a binary one, setting values higher than the mean of the values to 1 and the smaller ones to 0. Afterward, all transitions between the binary values are determined. We use two hyperparameters to distinguish between relevant and noise related transitions, namely the min_time and the $fluctuation\_threshold$ parameter. The first one is used to ensure a minimum time between transitions. In particular, consecutive transitions are suppressed that belong to the same event, therefore it is set to a value of 2. The latter parameter filters out events that are caused by noise in the standardized signal. It suppresses small fluctuations that amount to an RMS value smaller than 1.

V. DATASETS AND EXPERIMENTAL SETUP

We evaluated our approach on two distinct publicly available datasets from a residential and an office environment. Furthermore, we compare our algorithm to the re-implemented approaches by Jin et al. [21] and Zheng et al. [19]. We use cross-validation to evaluate all algorithms. In the coarse detection step, we selected a threshold on the MSE of 2, guided by the reconstruction errors produced during training. In general, a wide range of thresholds is feasible, as non-event windows produce MSE errors close to zero, while events result in high errors.

For the evaluation, we set the tolerance limit for the calculation of the performance metrics to $\tau = 1$ s, as proposed by Pereira [12], ensuring a minimum precision in time that is necessary for the majority of the NILM algorithms. The first dataset we used, namely BLUED, contains the voltage and current, sampled at 12 kHz, of a house with a two-phase connection [6]. The first phase (Phase A), has only a few devices attached to it, producing over-optimistic detection results [12]. Phase B, on the other hand, resembles the actual consumption of typical households, as it contains more diverse devices and a higher, more realistic noise level than Phase A. Consequently, we use Phase B to obtain a performance benchmark in a realistic setting. The BLUED dataset contains a few minutes of corrupted data, reducing the number of events for Phase B by 4 to 1574. In particular, the events on 26.10 from 01:22:00 to 01:24:00 are removed. As BLUED is divided into 16 folders, we apply 16-fold cross-validation with one day allocated for training in each iteration.

Additionally, we use an office environment dataset, namely BLOND [7]. The aggregated electrical signals are sampled at 50 kHz, making BLOND the only publicly available office

dataset that is sampled with high frequency. The appliance composition in office environments is substantially different from the one in households, as they contain more devices like, for example, laptops that induce noise into the signals. The results are 5-fold cross-validated, with the first five days of November 2016 being used for training and the subsequent ten days used for testing. BLOND is the only office environment dataset that is publicly available, but as it does not provide ground truth labels, we sampled a subset of the detected events for each approach separately to calculate the TPs and FPs and thus the precision to compare the algorithms. The appropriate sample size n is determined by assuming a binomial probability distribution for the TPs and FPs, using a Wilson score interval to estimate n [31]. We use the scores of MEED on the BLUED dataset as a prior guideline for the success probability p of the distribution. We expect the number of the TPs to be slightly lower, due to the noise level of the BLOND dataset. Hence, we account for this by subtracting 0.1 from p. The sample size is then calculated using an α value of 0.1 and a confidence interval with a conservative width of 0.4 around the estimate for p. This results in a minimum sample size per day of $n = 10$ for MEED, of $n = 13$ for the EH by Zheng et al. [19], and of $n = 14$ for the PM by Jin et al. [21].

VI. Experimental Results

The following results are obtained by running the input windows through the coarse event detection autoencoder and the fine-grained timestamp detector that constitute MEED. The cross-validated results of the detection performance benchmark on BLUED are shown in Figure 2, with MEED clearly outperforming the other algorithms. Furthermore, Figure 2 shows that the cross-validation scores are within a narrow range for all algorithms, hence the influence of the individual training folds in BLUED is limited.

TABLE III
Mean Scores achieved on BLUED Phase B, with $\tau = 1$ s

Algorithm	F1-Score	Recall	Precision	FPR	FPP
MEED	**0.75**	0.69	**0.83**	0.03e-3	0.14
Jin et al. [21]	0.51	**0.87**	0.36	0.04e-2	1.55
Zheng et al. [19]	0.51	0.41	0.69	0.046e-3	0.18

Averaging the scores over the folds of the cross-validation, one obtains a stable result for the overall performance of the algorithms, as shown in Table III. The autoencoder clearly outperforms the two reference algorithms with regard to the F1-Score. The algorithm by Zheng et al. [19] performs worse compared to the results presented in the original paper. This can be explained by the use of a different tolerance limit τ. When using $\tau = 3$ s, like Zheng et al. [19] in the original paper, MEED achieves an F1-Score of 0.79, hence, also outperforming the F1-Score that is reported by Zheng et al. [19]. The scores achieved show that MEED is more resilient to noise than the two reference algorithms, resulting in higher overall precision. The PM by Jin et al. [21] achieves a higher recall, but at the cost of a high amount of FP events. When

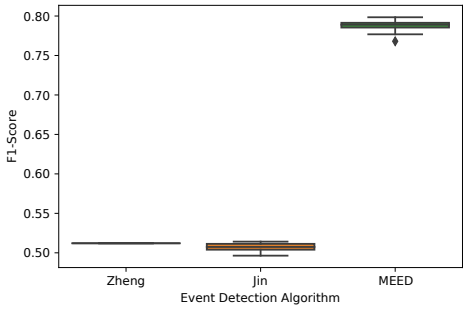


Fig. 2. Cross-validated F1-Scores for the algorithms of Zheng et al. [19], Jin et al. [21] and MEED on BLUED Phase B

trading off precision against recall, precision should be favored as FP events propagate through the entire analytical pipeline of NILM, leading to problems in the subsequent steps. In this context, missing out single events is not as severe as detecting multiple FPs. Regarding the FPs that MEED detected on BLUED, the majority of them corresponds in fact to true events that are not labeled accordingly in the data. Despite the use of BLUED as the de-facto standard dataset for the evaluation of event detection algorithms, one can see that the ground truth has some flaws, as also claimed by Zheng et al. [19]. As there are no labels for the BLOND dataset, our sampling procedure only allows us to compute the TP and FP events, thus, no F1-Score can be calculated. Looking at the results on the BLOND dataset in Table IV, one has to account for human labeling bias and the uncertainty caused by the sampling process. Despite this, MEED achieves a precision that significantly outperforms the other two algorithms on all three channels (1, 2, 3) of BLOND.

TABLE IV
Best Scores achieved on BLOND, with $\tau = 1$ s

	TP			FP			Precision		
Algorithm	1	2	3	1	2	3	1	2	3
MEED	86.8	69	89.6	64.4	41.6	46	**0.58**	**0.62**	**0.66**
Jin et al. [21]	53.6	39.2	80	138.4	113.4	116	0.28	0.26	0.40
Zheng et al. [19]	49	31	60	126	68	69	0.28	0.31	0.47

The amount of samples differs between the channels and the algorithms, as some of the evaluation days are no work days. Due to the low activity on such days, the number of detected events is smaller than the sample size n per day, resulting in an overall lower amount of samples. The variation of the scores between the MEED models as shown in Figure 3 indicates that an intelligent selection of typical work days for training can improve the model performance. In conclusion, one can see that MEED detected events with high precision in both settings. Consequently, MEED substantially reduces the amount of FPs that are induced into the analysis pipeline, while reliably detecting the majority of the events.

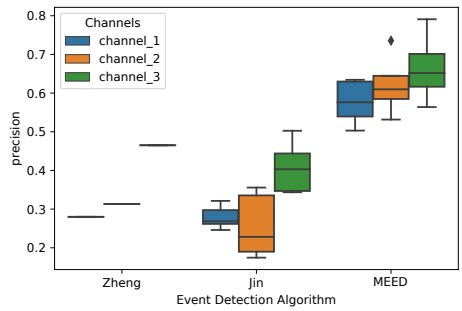


Fig. 3. Cross-validated precision scores for the algorithms of Zheng et al. [19], Jin et al. [21] and MEED on BLOND

VII. Conclusion

We propose MEED, a new multi-environment event detector, that does not rely on a dedicated event definition and that generalizes well to different environments without the need to adapt the model. In contrast to the present state of the art, MEED requires no dedicated expert knowledge about the environment it is used in. We compare MEED, a fully unsupervised bidirectional recurrent denoising autoencoder, to two reference event detectors on a residential (BLUED) and an office environment dataset (BLOND). In doing so, MEED achieves state of the art results with an F1-Score of 0.75 on BLUED and clearly outperforms the reference algorithms on BLOND, while using a tolerance limit of one second.

Acknowledgment

This research was supported by the Federal Ministry For Economic Affairs and Energy based on a decision by the German Bundestag.

References

[1] European Commission, "EU Energy in Figures," 2018.
[2] C. Armel, A. Gupta, G. Shrimali, and A. Albert, "Is Disaggregation the Holy Grail of Energy Efficiency? The Case of Electricity," *Energy Policy*, vol. 52, pp. 213–234, 2013.
[3] J. Kelly and W. J. Knottenbelt, "Does Disaggregated Electricity Feedback reduce Domestic Electricity Consumption? A Systematic Review of the Literature," *CoRR*, vol. abs/1605.00962, 2016. [Online]. Available: http://arxiv.org/abs/1605.00962
[4] R. Dong, L. Ratliff, H. Ohlsson, and S. Sastry, "Fundamental Limits of Nonintrusive Load Monitoring," *HiCoNS 2014 - Proceedings of the 3rd International Conference on High Confidence Networked Systems*, Oct. 2013.
[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, Massachusetts and London, England: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org/
[6] K. Anderson, A. Ocneanu, D. Benitez, D. Carlson, A. Rowe, and M. Berges, "BLUED: A Fully Labeled Public Dataset for Event-based Non-Intrusive Load Monitoring Research," *Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability*, pp. 1–5, Jan. 2012.
[7] T. Kriechbaumer and H.-A. Jacobsen, "BLOND, a Building-Level Office Environment Dataset of Typical Electrical Appliances," *Scientific Data*, vol. 5, 2018.
[8] G. W. Hart, "Nonintrusive Appliance Load Monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.
[9] K. D. Anderson, M. E. Berges, A. Ocneanu, D. Benitez, and J. M. F. Moura, "Event Detection for Non Intrusive load monitoring," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Oct. 2012, pp. 3312–3317.
[10] L. Pereira and N. Nunes, "An Experimental Comparison of Performance Metrics for Event Detection a Algorithms in NILM," in *4th International Workshop on NILM*, 2018.
[11] M. Valovage and M. Gini, "Label Correction and Event Detection for Electricity Disaggregation," in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '17. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 990–998.
[12] L. Pereira, "Developing and Evaluating a Probabilistic Event Detector for Non-Intrusive Load Monitoring," in *2017 Sustainable Internet and ICT for Sustainability (SustainIT)*, Dec. 2017, pp. 1–10.
[13] J. M. Alcalá, J. Ureña, and Á. Hernández, "Event-Based Energy Disaggregation Algorithm for Activity Monitoring From a Single-Point Sensor," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 10, pp. 2615–2626, Oct. 2017.
[14] ——, "Event-based Detector for Non-Intrusive Load Monitoring based on the Hilbert Transform," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, Sept. 2014, pp. 1–4.
[15] L. de Baets, J. Ruyssinck, D. Deschrijver, and T. Dhaene, "Event Detection in NILM using Cepstrum Smoothing," in *3rd International Workshop on NILM*, 2016, pp. 1–4.
[16] K. S. Barsim, R. Streubel, and B. Yang, "Unsupervised Adaptive Event Detection for Building-Level Energy Disaggregation," *Proceedings of power and energy student summit (PESS)*, 2014.
[17] B. Wild, K. S. Barsim, and B. Yang, "A new Unsupervised Event Detector for Non-Intrusive Load Monitoring," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2015, pp. 73–77.
[18] K. S. Barsim and B. Yang, "Sequential Clustering-Based Event Detection for Non-Intrusive Load Monitoring," in *Computer Science & Information Technology*, vol. 6, 2016, pp. 77–85.
[19] Z. Zheng, H. Chen, H. Xiaowei, and L. Xiaowei, "A Supervised Event-Based Non-Intrusive Load Monitoring for Non-Linear Appliances," *Sustainability*, vol. 10, p. 1001, March 2018.
[20] K. N. Trung, E. Dekneuvel, B. Nicolle, O. Zammit, C. N. Van, and G. Jacquemod, "Event Detection and Disaggregation Algorithms for NIALM System," in *2nd International Workshop on NILM*, 2014.
[21] Y. Jin, E. Tebekaemi, M. Berges, and L. Soibelman, "Robust Adaptive Event Detection in Non-Intrusive Load Monitoring for Energy Aware Smart Facilities," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 4340–4343.
[22] Z. Zhu, S. Zhang, Z. Wei, B. Yin, and X. Huang, "A Novel CUSUM-Based Approach for Event Detection in Smart Metering," *IOP Conference Series: Materials Science and Engineering*, vol. 322, no. 7, 2018.
[23] C. C. Yang, C. S. Soh, and V. V. Yap, "Comparative Study of Event Detection Methods for Non-intrusive Appliance Load Monitoring," *Energy Procedia*, vol. 61, pp. 1840 – 1843, 2014.
[24] R. Cox, S. B. Leeb, S. R. Shaw, and L. K. Norford, "Transient Event Detection for Nonintrusive Load Monitoring and Demand Side Management using Voltage Distortion," in *21st Annual IEEE Applied Power Electronics Conference and Exposition, 2006. APEC '06.*, March 2006.
[25] S. Makonin and F. Popowich, "Nonintrusive Load Monitoring (NILM) Performance Evaluation," *Energy Efficiency*, vol. 8, pp. 809–814, Dec. 2014.
[26] V. Chandola, A. Banerjee, and V. K. Vipin, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 15:1–15:58, July 2009.
[27] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
[28] F. Chollet *et al.*, "Keras," https://keras.io, 2015.
[29] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR*, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980
[30] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *Proceedings of the 13. International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.
[31] W. W. Piegorsch, "Sample Sizes for Improved Binomial Confidence Intervals," *Computational Statistics and Data Analysis*, vol. 46, no. 2, pp. 309–316, June 2004.

C Event Detection for Energy Consumption Monitoring

This is the author's accepted version of an article that has been published at the IEEE Transactions on Sustainable Computing. Changes were made to this version by the publisher prior to publication. The final version of record is available at https://doi.org/10.1109/TSUSC.2020.3012066

Event Detection for Energy Consumption Monitoring

Daniel Jorde, Hans-Arno Jacobsen, *Fellow, IEEE*

Abstract—The accurate detection of appliance state transitions in electrical signals is fundamental for numerous energy-conserving applications. We present an extensive overview and categorization of the current state in event detection on high-sampling-rate signals. Existing approaches are designed for specific environments and need to be tediously adapted for new ones. Thus, we propose an unsupervised, multi-environment event detector, outperforming four state-of-the-art algorithms on two heterogeneous public datasets.

Index Terms—Event detection, non-intrusive load monitoring, neural nets, machine learning, energy-aware systems

1 Introduction

One of the challenges humanity is facing nowadays is the depletion of natural energy resources, while the overall energy demand keeps increasing, especially the demand for electrical energy [1]. For this reason, researchers are striving to find solutions to improve the way limited energy resources are used. By making both industrial and residential consumers aware of their detailed electricity consumption, one aims to reduce the waste of energy and build more sustainable electrical applications [2]. Several surveys indicate that appliance-level information can reduce energy consumption by raising consumer awareness [3] and enable a variety of sustainable smart city applications for energy savings [4]. The consumption of individual appliances can be acquired using Non-Intrusive Load Monitoring (NILM) methods with a low-cost single-sensor approach to record an aggregated signal, measured only at the mains of an entity, such as a building or a composite device [2]. After extracting relevant signal segments using an event detection algorithm, the appliances that had caused the events can be identified, and the signal can be decomposed into the individual appliances. Most of the electrical data used in NILM is collected by smart meters, which usually sample the signals at a low rate ($<$ 1 kHz). As a result, only some of the major devices can be detected [2]. Data that are sampled using higher rates increase the probability for successful NILM [5] and allow to detect more devices [2]. Due to problems arising with high sampling rates, such as the curse-of-dimensionality for machine learning algorithms [6] and the amount of data to store and process, online applications must pre-process the data by reliably extracting relevant segments, i.e., appliance-state transitions [7]. By only sending segments over the network, one can enable smart city applications, while overcoming infrastructural problems due to the amount of communication in large-scale Internet of Things settings [8]. In the past, researchers focused on low-sampling-rate data, driven by the high costs (for metering, storage, and processing) associated with the acquisition of high-sampling-rate data. As there is an evident lack of methods, and because of the advantages of high-sampling-rate data, we focus on this domain. Detecting events and distinguishing them from signal noise is particularly challenging and prone to errors. So far, researchers focused mainly on residential buildings and their appliances. Thus, there are multiple residential datasets publicly available [9]. Recent work investigates industrial and office settings, and new datasets containing measurements of computational equipment are published [10]. Existing event detection algorithms exhibit one important disadvantage: Most rely on customized, expert-made event definitions. This prohibits such approaches from being able to generalize well to a setting they were not designed for. Subsequently, the algorithms need to be tediously fine-tuned for being used in a new setting.

The main contributions of this paper are as follows: First, we present the current state of the art in event detection for energy consumption monitoring with NILM, based on an extensive literature review. In a comprehensive overview, we facilitate algorithm comparison by reporting all relevant metrics and by categorizing existent approaches based on key-characteristics. Second, we present, based on our previous work [11], the state-of-the-art, multi-environment event detector (MEED) that does not rely on a dedicated event definition while being trained and used fully unsupervised. Hence, MEED can be used in different environments with an automatically determined decision hyperparameter, enabling new possibilities for NILM and energy-related applications in general. Third, we conduct an extensive benchmark test of the current state of the art by implementing four algorithms and by evaluating them compared to MEED on two publicly available datasets from different environments. We are publicly releasing all source code, all models, and the parametrization for all algorithms used in this paper in a publicly available repository [12]. By doing so, we provide the first and most comprehensive implementation of high-sampling-rate event detection algorithms in the field.

The rest of the paper is organized as follows: In Section 2 we give an overview on event detection in NILM and relevant metrics. Section 3 summarizes related work, followed by the description of MEED in Section 4. Subsequently, we

- *D. Jorde is with the Department of Computer Science, Technical University of Munich, Germany. E-mail: daniel.jorde@tum.de*
- *H.-A. Jacobsen is with the Department of Computer Science, Technical University of Munich, Germany. Email: arno.jacobsen@tum.de*

detail our experimental setup in Section 5 and discuss the corresponding results in Section 6. Section 7 then concludes this paper.

2 Background

Based on the edge detection algorithm for NILM introduced in 1992 by George Hart [13], a variety of new algorithms that use events to extract relevant signal segments have been proposed.

2.1 Event Detection

As most researchers use different, specific definitions of events that reduce their capability to generalize to new settings, we use a general event definition. In particular, we define events to be transitions between states of individual appliances. Event detection algorithms can be divided into ones using supervised- and unsupervised-learning, depending on the use of labeled data during training (supervised) or not (unsupervised). Algorithms relying on a small number of labeled samples and a large number of unlabeled samples are considered to be semi-supervised. Supervised and semi-supervised algorithms are less flexible in tasks like event detection than unsupervised algorithms, as they are dependent on the event definition used for labeling. Thus, we propose an unsupervised event detector. Another possibility to classify event detection methods is the three categories introduced by Anderson et al. [14], namely Expert Heuristics (EH), Probabilistic Models (PM), and Matched-Filters (MF). Rule-based approaches, including simple threshold-based ones, and methods using machine learning, are considered to be EH, whereas approaches using statistical metrics to determine events belong to the PM category. Algorithms that match previously learned event masks with the signal to detect events are considered to be MF approaches [14].

2.2 Metrics

The following commonly used metrics are based on the scores of the confusion matrix, namely the amount of records that are True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

$$\text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \quad \text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}$$

$$\text{FPP} = \frac{\text{FP}}{\text{TP}+\text{FN}} \quad \text{F1-Score} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

Metrics based on TN scores are omitted, because of the absence of a uniform definition of true non-events, as they can occur any time no event exists. Another metric that is commonly used in the NILM literature is the True Positive Percentage (TPP), which is defined as $\frac{\text{TP}}{\text{E}} = \frac{\text{TP}}{\text{TP}+\text{FN}}$ [14]. As this is equal to the recall metric, we omitted the TPP in our evaluation. It is of particular importance to define how the single scores of the confusion matrix are computed to ensure the comparability of the results [15]. Ground truth events are often generated by human experts, and thus can be imprecise with respect to their exact location in time. Hence, it is common to define a tolerance limit τ for the matching of detected events e_{det} and ground truth events e_{gt} [15], [16], [17]. A detected event e_{det} is a TP if there exists a ground truth event within an interval of $\pm\tau$, i.e., if $\exists e_{gt} : e_{det} - \tau \leq e_{gt} \leq e_{det} + \tau$.

3 Related Work

In the following, we summarize the current state of the art in event detection in NILM on high-sampling-rate data and identify gaps. To ease the comparison of different algorithms, we have categorized the related literature into approaches that use complete, or clearly defined subsets of public datasets and into works that use non-public, or undefined subsets of public data to evaluate their results. Based on this, publications falling into the first category are listed in Table 1, and publications from the latter category are gathered in Table 2. We only list approaches that have been evaluated on real-world data. The essential characteristics of the related algorithms are listed in these tables, with the scores being rounded to the second decimal digit. In case the exact value for a particular metric is unclear from the publication, we declared the result to be approximate ($\approx$). The algorithms make use of either supervised- (sup.) or unsupervised-learning (unsup.) techniques. To the best of our knowledge, there are no semi-supervised event detection approaches for high-sampling-rate NILM, in contrast to several publications on low-sampling-rate data, such as the one by Yang et al. [49] and Li et al. [50]. We further categorize the related work according to the following criteria. The "Setting" column indicates the environment the algorithms are designed for and evaluated in, namely Residential (R) or Industrial (I) environments. The cross-validation (CV) criterion reports whether CV was used in the evaluation, as suggested by Makonin and Popowich [51] to improve the reliability of the results. Most of the publications exhibit no information about the tolerance limit τ used to calculate the metrics. Hence, the scores can only be compared approximately and indirectly.

In the following, the publications listed in Table 1 are further detailed, as they share common characteristics in their evaluations, making them comparable. Valovage and Gini introduce a PM [16] that relies on a Bayesian detection method at its core. The algorithm tries to partition the signal into run sequences, followed by declaring the transitions between such sequences as events. Pereira [17] introduces another PM. The algorithm makes use of a log-likelihood ratio detector to estimate a detection statistic. The algorithm then searches the signal for extreme values to determine events. In contrast to other PM, Wild et al. [22] use a kernel Fisher discriminate analysis to detect start and end times of events in a supervised way. Alcala et al. [18] use the signal's envelope of the normalized current and voltage RMS values and a threshold to detect events. Another approach using the signal's envelope is based on a Hilbert Transform [19]. It further applies an average and a derivative filter to obtain a set of spikes to detect transitions. The algorithm proposed by de Baets et al. [20] transforms the signal into the frequency domain and applies a threshold on the computed Cepstrum components. A different frequency domain based feature is introduced by Held et al. [26]. The authors represent the

TABLE 1
Comparable Related Work (abbreviations are explained in the text)

Ref.	Learning	Setting	Type	Dataset	CV	Recall	Precision	FPP	F1-Score	Limit τ
[16]	unsup.	R	PM	BLUED A | B	No	-	-	-	$\approx$ 0.98 | $\approx$ 0.80	2 s
[17]	unsup.	R	PM	BLUED A | B	No	-	-	-	$\approx$ 0.96 | $\approx$ 0.72	1 s
				REDD	No	-	-	-	$\approx$ 0.80	-
[18]	unsup.	R	EH	BLUED A | B	No	0.94 | 0.88	-	-	-	-
[19]	unsup.	R	EH	REDD subset	No	0.93	-	0	-	-
[20]	sup.	R	EH	BLUED A | B	Var.	-	-	-	0.98 | 0.80	-
[21]	unsup.	R	PM	BLUED A | B	Var.	-	-	-	$\approx$ 0.98 | $\approx$ 0.80	-
[22]	sup.	R	PM	BLUED A | B	No	0.99 | 0.86	0.99 | 0.92	-	0.99 | 0.89	-
[23]	**unsup.**	**R**	**EH**	**BLUED A | B**	**No**	**0.97 | 0.68**	**0.99 | 0.93**	**0.78e-2 | 0.49e-1**	**0.98 | 0.79**	**-**
[24]	**unsup.**	**R**	**EH**	**BLUED A | B**	**No**	**0.99 | 0.88**	**-**	**0.71e-2 | 0.4**	**0.99 | 0.77**	3 s
[25]	unsup.	R	EH	REDD subset	No	0.94	-	-	-	-
[26]	unsup.	R	EH	BLUED A | B	No	1 | 0.94	0.99 | 0.95	-	0.99 | 0,94	-
[27]	unsup.	R	EH	BLUED A	No	0.94	-	0.79e-2	-	-

TABLE 2
Not directly comparable Related Work (abbreviations are explained in the text)

Ref.	Learning	Setting	Type	Dataset	CV	Recall	Precision	FPP	F1-Score	Limit τ
[28]	sup.	R	EH	BLUED A | B sampled 1:4	No	0.98 | 0.98	0.95 | 0.92	0.05e-1 | 0.09	0.96 | 0.95	-
[29]	unsup.	R	EH | PM	3 appliances	No	-	0.21	-	-	-
[30]	unsup.	R	PM	REDD subset	No	-	-	-	-	-
[31]	unsup.	R	EH	6 appliances	No	-	-	-	-	-
[32]	sup.	I	MF	4 appliances	No	-	-	-	-	-
[33]	sup.	I	MF	4 appliances	No	-	-	-	-	-
[34]	unsup.	R	EH	4 appliances	No	-	-	-	-	-
[35]	**unsup.**	**R**	**EH**	**BLUED subset | simulated**	**No**	**-**	**-**	**-**	**-**	**-**
[36]	**unsup.**	**R**	**PM**	**3 residential units**	**No**	**0.99**	**-**	**0.09e-1**	**-**	**-**
[37]	unsup.	R	PM	10 appliances	No	0.73	0.76	0.23	-	3 samples
[38]	unsup.	I	EH	4 appliances	No	-	-	-	-	-
[39]	unsup.	R	EH	6 homes	No	-	-	-	-	-
[40]	unsup.	R	EH	15 appliances	No	0.89	-	-	-	-
[41]	unsup.	R	EH	5 appliances	No	-	-	-	-	-
[42]	unsup.	R	EH	REDD subset	No	-	-	-	-	-
[43]	unsup.	R	PM, EH	REDD subset	No	-	-	-	-	-
[44]	unsup.	R	EH	8 appliances	No	0.96	1	0	0.98	-
[45]	unsup.	R	PM	29 appliances	No	-	-	-	-	-
[46]	unsup.	I	EH	14 appliances	No	-	-	-	-	-
[47]	unsup.	R	EH | PM	4 appliances	No	-	-	-	-	-
[48]	unsup.	I	EH	7 appliances	No	-	-	-	0.83	-

signal by applying a frequency invariant transformation to obtain a signal representation that emphasizes non-periodic, pulse-shaped components to detect events. The authors fine-tune the six hyper-parameters of the algorithm by running an optimization algorithm over the full evaluation dataset. As no separated test set is used for the evaluation, the algorithm's parameters overfit the data, and the evaluation scores do not resemble a real-world setting. The approach by Trung et al. [25] makes use of the cumulative sum statistics (CUMSUM) and an adaptive threshold to detect the start and the end of the transient segments in the signal. Furthermore, Zhu et al. use CUMSUM to detect events in a residential setting. In contrast to methods designed for residential settings, Cox et al. [38] use the voltage distortion of industrial data. Yang et al. [43] compare an EH and a goodness-of-fit (GOF) event detector, concluding that the latter performs better. An enhanced GOF approach, based on a chi-square test statistic, is introduced by Baets et al. [21]. In their work on event detection, Barsim and Yang [23] introduce three algorithm agnostic event models, that oppose different generic constraints on events. The authors evaluate their approach using the unsupervised DBSCAN clustering algorithm. We have implemented this approach because of its performance and well-defined event model to benchmark MEED. As many event detection algorithms suffer from false positives, Lu and Li introduce a hybrid approach, consisting of a threshold-based base algorithm to detect events and two additional ones to filter out false events [27]. Zheng et al. [24] also show the applicability of DBSCAN to detect events. The algorithm clusters appliance states and transitions before they are post-processed with dedicated thresholds on power and time to filter out duplicates and FPs. For its transparent evaluation and state-of-the-art performance on BLUED, we have selected this algorithm for the benchmark.

Despite the comparability issues, we have selected two distinguished algorithms from Table 2 to benchmark our approach. The first algorithm is the GOF based approach by Jin et al. [36] as it has multiple advantages over other probabilistic event detectors, despite not being evaluated on a public dataset. In contrast to other algorithms, this GOF approach provides a guideline for the event window hyperparameter and a closed form for the decision threshold. Furthermore, the authors show the superiority of their

algorithm over the widely used generalized log-likelihood detector [36]. Besides this PM event detector, we have selected a fast and straightforward, but well-performing technique by Liu et al. [35] that is based on a median filter and a ripple mitigation algorithm to separate meaningful events from other fluctuations in the signal [35].

The following base requirements for MEED are derived from the related work presented: MEED has to be designed independent of any specific event definition, allowing it to generalize better, even to different settings. Furthermore, this multi-environment event detector has to identify events unsupervised, without the need for user-defined decision hyperparameters. By doing so, one avoids costly human labelling of events and interventions. In contrast to existing unsupervised event detectors, MEED's decision hyperparameter is determined automatically.

4 Event Detection Approach

Based on the introduced requirements, we design a new window-based EH event detector. MEED applies a two-step process to the pre-processed input data to detect events, as shown in Figure 1. The first step is based on the mean-squared reconstruction error (MSE) of the window, which is computed by an autoencoder model. Subsequently, we apply a peak detection algorithm to find the exact timestamps within the windows that exceed an absolute MSE value.

4.1 Data Pre-Processing

One of the features that are used in multiple event detection methods is CUMSUM [25], [44], as it has shown to reveal trends and changes in time-series data successfully. Small fluctuations are suppressed in CUMSUM, while substantial changes in the data are revealed. The input to the first step of MEED is the CUMSUM of the five-period root-mean-square (RMS) of the current signal. The CUMSUM at time t_i is computed by accumulating the deviations of the RMS values from the window's mean value for all $t \leq t_i$. During training, we scale to values between -1 and 1.

4.2 Coarse Event Detection Autoencoder

The first step of MEED is to apply a denoising autoencoder to the input. In doing so, we seize the rarity of events to build a model that can detect windows that contain transitions. A similar idea is used to solve classical anomaly detection tasks, as presented in the summary on this topic by Chandola et al. [52]. The authors present several methods to identify events. One of these methods applies a neural network, in particular, an autoencoder, to learn a hidden representation h of the input x to reveal events [52]. Autoencoders are typically used in representation learning tasks. They consist of an encoder $h = f(x)$ and a decoder function $r = g(h)$ [6]. As events are rare compared to non-events, they constitute the minority class, making it hard for the model to learn a proper event representation h. Instead, the model learns to represent and reconstruct the non-event majority class. Hence, a large reconstruction error is produced when the model faces a deviation from normal behavior, i.e., an event. MEED's hyperparameter settings can generalize well between environments, as events rarely occur in electrical environments. Denoising autoencoders, a special case of the general autoencoder, minimize a loss function $L(x, g(f(\widetilde{x})))$, while trying to reconstruct the input x from intentionally corrupted input data $\widetilde{x}$. The corrupted input $\widetilde{x}$ is obtained by adding white Gaussian noise ($\mu = 0$, $\sigma = 0.25$) to x. Each of the three hidden layers of MEED is a bidirectional long short-term memory (LSTM) cell that has shown to be able to learn long term dependencies and patterns from sequential data [6]. Briefly, a LSTM cell consists of gating units (i_t, o_t, f_t and c_t) that control its information flow, as summarized in the following equations, with sig being the logistic sigmoid, $tanh$ the hyperbolic tangent function, and $\circ$ the element-wise product [6]:

$$
\begin{aligned}
f_t &= sig(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= sig(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= sig(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ tanh(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ tanh(c_t)
\end{aligned}
$$

The input window to MEED has a fixed size, thus, we use bidirectional LSTMs to make use of future context information to improve the overall performance [53], as depicted in the autoencoder architecture in step one in Figure 1. The bidirectional networks consist of one LSTM to process the data in the forward direction, i.e., from the beginning of the window to the end, and one separate LSTM to do so in the opposite direction [53]. The parameters of the forward and the backward LSTMs are not shared, but the outputs of both directions are concatenated into a single output to compute the MSE to optimize the network during training. Each of the three autoencoder components, namely, the encoding, the embedding and the decoding layer, are represented by a bidirectional LSTM. On top of the LSTM cells, the linear output layer ensures that the in- and output have the same dimensionality. We evaluated several training hyperparameters and different architectural components using a grid search on the training data, resulting in the final optimal settings shown in Table 3.

TABLE 3
Autoencoder Hyperparameters, implemented with Keras [54]

Parameter	Value	Parameter	Value
learning rate	0.001	LSTM 1 cell size	216
optimizer	Adam [55]	LSTM 2 cell size	108
initializer	Glorot uniform [56]	LSTM 3 cell size	216
merge mode	concatenate	window size	10 s

4.3 Fine-Grained Event Timestamp Detector

In case the coarse event detector determines an input window to be an event window, i.e., the reconstruction error threshold is exceeded, we apply the fine-grained detection procedure to determine the exact number and timestamps in the window. We compute the RMS values over five periods of the raw input signal. Then, we convert the RMS signal into a binary one, setting values higher than the mean of the values to 1 and the smaller ones to 0, revealing significant peaks in the window. Relevant and noise-related transitions

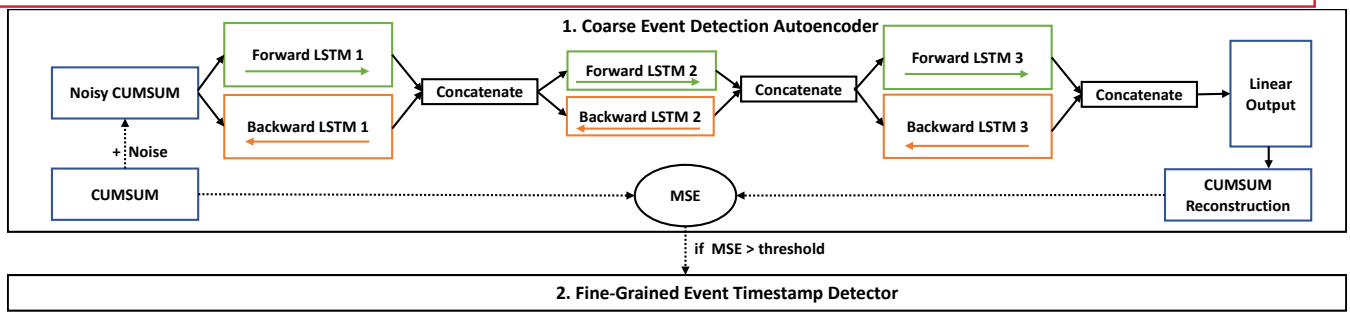


Fig. 1. MEED architecture, with step 1 (the coarse event detection autoencoder) and step 2 (the fine-grained event timestamp detector)

are distinguished by suppressing small fluctuations that do not amount to an RMS value greater than 1 and by ensuring a minimum event time of 2 samples, i.e., filtering out consecutive samples that belong to the same event.

5 Experimental Setup

We evaluated our approach on two distinct publicly available datasets from a residential and an office environment. Furthermore, we compare our algorithm to the four re-implemented approaches by Jin et al. [36], Zheng et al. [24], Liu et al. [35] and Barsim et al. [23]. We use CV to evaluate all algorithms. In the coarse detection step, the threshold on the reconstruction error is set to be the MSE produced at the end of the training. For the following experiments, we have rounded the average of the last 10 MSE values after training the model on one day of data (BLOND) or half a day (BLUED), leading to a threshold of 2 for both cases. In general, a wide range of thresholds is feasible, as non-event windows produce MSE errors close to zero, while events result in high errors. For the evaluation, we set the tolerance limit for the calculation of the performance metrics to $\tau = 1$ s, as proposed by Pereira [17], ensuring a minimum precision in time that is necessary for the majority of NILM algorithms.

5.1 Datasets

The first of the two datasets we used, namely BLUED, contains the voltage and current, sampled at 12 kHz, of a house with a two-phase connection [9]. The first phase (Phase A), has only a few devices attached to it, producing over-optimistic detection results, as also claimed by Pereira [17]. Phase B, on the other hand, resembles the actual consumption of typical households. It contains more diverse devices and a higher, more realistic noise level than Phase A. Consequently, we use Phase B to obtain a performance benchmark in a realistic setting. The majority of the events in Phase B occur when other appliances are running simultaneously, making the dataset challenging. The BLUED dataset contains corrupted data, reducing the number of events for Phase B by 4 to 1574. In particular, the events on 26.10 from 01:22:00 to 01:24:00 are removed. As BLUED is divided into 16 folders, we apply a 16-fold CV with one day allocated for training in each iteration. Additionally, we use an office and computational equipment dataset, namely BLOND [10]. The electrical signals are sampled at 50 kHz, making BLOND the only publicly available office dataset that is sampled with high frequency. The appliance composition in offices is substantially different from the one in households, as they contain more devices such as, for example, laptops that induce noise into the signal. The results are 5-fold cross-validated, with the first five days of November 2016 being used for training and the subsequent ten days for testing. As BLOND does not provide ground truth labels, we sampled a subset of the detected events for each approach to calculate the TPs and FPs and thus the precision to compare the algorithms. The sample size n is estimated assuming a binomial probability distribution for the TPs and FPs, using a Wilson score interval [57]. We use the scores of MEED on BLUED as a prior guideline for the success probability p of the distribution and calculate n using an α value of 0.1 and a confidence interval with a conservative width of 0.4. Thus, for each of the five algorithms n is as follows: MEED $n = 10$, Jin et al. [36] $n = 14$, Zheng et al. [24] $n = 13$, Liu et al. [35] $n = 12$, and Barsim et al. [23] $n = 9$. The sampled events are then expert labeled, using the ground truth provided in BLOND.

5.2 Reference Algorithms and Implementation

In the following, we will provide short descriptions of the algorithms used to benchmark MEED in our evaluation. The original papers and our source code of the algorithms can be consulted for additional details. The implemented algorithms and the settings for the hyperparameters used to conduct the experiments are provided in a publicly available repository [12]. The repository [12] also contains all trained models and the code for MEED. We encourage researchers to use the algorithms and the score function we provide to evaluate their approaches.

In their PM, Jin et al. [36] formulate the event detection problem as a binary hypothesis test. The authors compare the data distribution in a pre-event window to the one in a detection window. If the distributions are not equal, i.e., the null hypothesis can be rejected, an event is detected. This is done by using a chi-squared based threshold. The authors further introduce an estimate for the window size parameter [36]. Zheng et al. use the DBSCAN clustering algorithm to detect the transitions between steady states in the active power and RMS current signal. The transitions found are then post-processed by applying two preset thresholds to separate meaningful events from random fluctuations [24].

In contrast to the other algorithms, Liu et al. [35] solely rely on basic signal processing techniques, namely, a median filter and a ripple mitigation filter. The ripple mitigation filter computes the delta of consecutive power samples over multiple ranges around each data point. The authors then compute an absolute power delta curve by only keeping the highest, absolute power delta values around each sample. Therefore small ripples in the signal are filtered, and meaningful differences in the power consumption remain. Subsequently, a fixed threshold is applied to these absolute power delta values to detect events [35]. As the other algorithms oppose some constraints on consecutive detected events and apply additional post-processing, we do so accordingly for the one by Liu et al. [35], as we assume the authors have done the same. We do so, by filtering out consecutive events that are detected within a second of a positive detection

Besides applying the DBSCAN algorithm, Barsim et al. [23] introduce three formal event definitions with different constraints on what to consider an event. These constraints oppose criteria on the individual clusters from the DBSCAN and define a loss-function for the overall clustering structure. When a clustering fulfills the event model and the loss is below a certain threshold, an event is detected. Afterward, post-processing is applied to find more stable steady and transient-state regions in the signal.

6 Experimental Results

The following results are obtained by running the input windows through the coarse event detection autoencoder and the fine-grained timestamp detector that constitute MEED. When averaging the scores over the cross-validation, one obtains a stable result for the overall performance of the algorithms, as shown in Table 4 for the BLUED dataset. The scores of the CV on BLUED are within a narrow range for all algorithms. Hence, the influence of the individual training folds in BLUED is limited.

TABLE 4
Mean Scores achieved on BLUED Phase B, with $\tau = 1$ s

Algorithm	F1-Score	Recall	Precision	FPP
MEED	**0.75**	0.69	0.83	**0.14**
Jin et al. [36]	0.51	0.87	0.36	1.55
Zheng et al. [24]	0.51	0.41	0.69	0.18
Liu et al. [35]	0.53	**0.91**	0.38	1.5
Barsim et al. [23]	0.31	0.18	**0.95**	0.95

MEED clearly outperforms the four reference algorithms with regard to the F1-Score, as shown in Table 4. The two deviations in the performance of the re-implemented reference algorithms, compared to the original papers, can be explained by the nonavailability of necessary hyperparameter configurations and by the use of different tolerance limits τ. First, the recall of the algorithm by Barsim et al. [23] is higher (0.68) than in our experiments for the BLUED dataset. This can be explained by the strong influence of the publicly non-available decision threshold for the loss function of the algorithm. As we could not determine the exact configurations, we have conducted an extensive grid search ourselves to optimize the hyperparameters of the reference algorithms. We have included all detailed configurations into the code repository [12]. Second, the scores of the algorithm by Zheng et al. [24] deviate because of their use of a different tolerance limit of $\tau = 3$ s. When adopting this tolerance limit in our experiments, MEED achieves an F1-Score of 0.79, hence outperforming the original F1-Score reported by Zheng et al. [24] on the BLUED dataset. The algorithm by Liu et al. [35] achieves a higher recall, but at the cost of a high amount of FP events. On the other hand, the algorithm by Barsim et al. achieves a higher precision score than MEED, while missing most of the actual events. When trading off precision against recall, precision should be favored as FP events propagate through the entire analytical pipeline of NILM, leading to problems in the subsequent steps. Despite this, a minimum recall has to be ensured, as missing the majority of events prohibits applications relying on their accurate detection from working. While having a substantially high precision score, the results of MEED are the most balanced ones, resulting in the highest F1-Score. Regarding the FPs that MEED detected on BLUED, the majority of them correspond in fact to true events that are not labeled accordingly in the data. Despite the use of BLUED as the de-facto standard dataset for the evaluation of event detectors, one can see that the ground truth has some flaws, as also claimed by Zheng et al. [24]. Besides measuring the detection performance based on the F1-Score, we empirically determined the time-efficiency of the algorithms, as shown in Figure 2. The results show that by using the DBSCAN algorithm twice, in a forward- and a backward-pass, the algorithm by Barsim et al. [23], is significantly slower than the other algorithms. The fastest technique is proposed by Jin et al. [36]. It has to be noted, though, that all algorithms perform near real-time as they only require processing times of up to 1.306 s for every 10 s of the high-sampling-rate BLUED dataset.

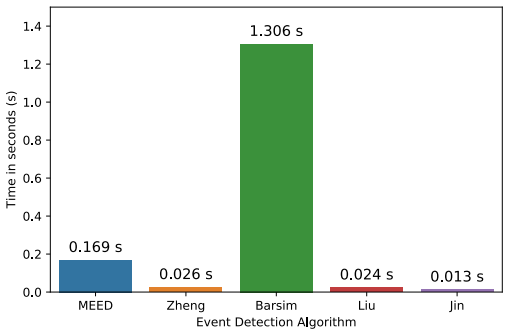


Fig. 2. Empirical time-efficiency of MEED and the algorithms of Jin et al. [36], Zheng et al. [24], Li et al. [35] and Barsim et al. [23] on BLUED Phase B. The measurements represent the time required for each of the algorithms to process 10 s of the BLUED dataset, measured on a machine with two Intel Xeon E5-2630 v3 8x@2.4GHz processors and 16GB RAM.

As there are no labels for the BLOND dataset, our sampling procedure only allows us to compute the TP and FP events, thus, no F1-Score can be calculated. Looking at the results on the BLOND dataset in Table 5, one has to account for human labeling bias and uncertainty caused by the sampling process. Despite this, MEED achieves a precision that significantly outperforms the other four algorithms on all three channels (1, 2, 3) of BLOND.

TABLE 5
Best Scores achieved on BLOND, with $\tau = 1$ s

Algorithm	TP 1	TP 2	TP 3	FP 1	FP 2	FP 3	Precision 1	Precision 2	Precision 3
MEED	86.8	69	89.6	64.4	41.6	46	**0.58**	**0.62**	**0.66**
Jin et al. [36]	53.6	39.2	80	138.4	113.4	116	0.28	0.26	0.40
Zheng et al. [24]	49	31	60	126	68	69	0.28	0.31	0.47
Liu et al. [35]	52	32	79	129	71	88	0.29	0.31	0.47
Barsim et al. [23]	32	17	35	68	59	65	0.32	0.22	0.35

The amount of samples differs between channels and algorithms, as some days are not workdays. Due to the low activity on such days, the number of detected events is smaller than the sample size n per day, resulting in an overall lower amount of samples. The variation in the cross-validated scores of MEED, as shown in Figure 3, indicates that an intelligent selection of typical workdays for training can further improve the model performance. In conclusion, one can see that MEED detected events with high precision in both settings. MEED substantially reduces the amount of FPs that are induced into the analysis pipeline, while reliably detecting events at fast speed.

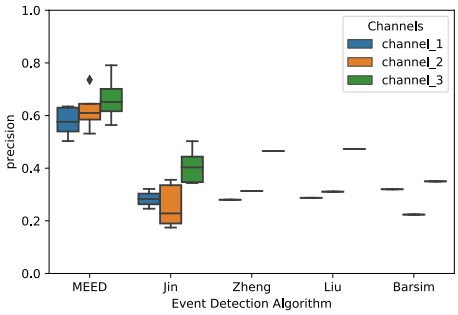


Fig. 3. Precision scores (CV) of MEED and the algorithms of Jin et al. [36], Zheng et al. [24], Li et al. [35], and Barsim et al. [23] on BLOND

7 Conclusion

We propose MEED, a new multi-environment event detector for energy consumption monitoring, that does not rely on a dedicated event definition and that generalizes well to different environments without the need to adapt the model. In contrast to the present state of the art, MEED requires no dedicated expert knowledge about the environment it is used in. We compare MEED, a fully unsupervised bidirectional recurrent denoising autoencoder, to four re-implemented reference algorithms on a residential dataset (BLUED) and an office environment dataset (BLOND). In doing so, MEED achieves state of the art results with a cross-validated F1-Score of 0.75 on BLUED and clearly outperforms the reference algorithms on BLOND, while using a tolerance limit of one second.

Acknowledgments

This research was supported by the Federal Ministry for Economic Affairs and Energy based on a decision by the German Bundestag.

References

[1] European Commission, *EU Energy in Figures*, 2018.
[2] C. Armel, A. Gupta, G. Shrimali, and A. Albert, "Is Disaggregation the Holy Grail of Energy Efficiency? The Case of Electricity," *Energy Policy*, vol. 52, pp. 213–234, Jan. 2013.
[3] J. Kelly and W. J. Knottenbelt, "Does Disaggregated Electricity Feedback reduce Domestic Electricity Consumption? A Systematic Review of the Literature," in *3rd International Workshop on NILM*, 2016, pp. 1–5. [Online]. Available: http://arxiv.org/abs/1605.00962
[4] G. Rostirolla, R. Righi, J. Barbosa, and C. d. Costa, "ElCity: An Elastic Multilevel Energy Saving Model for Smart Cities," *IEEE Transactions on Sustainable Computing*, vol. 3, no. 1, pp. 30–43, Sep. 2018.
[5] R. Dong, L. Ratliff, H. Ohlsson, and S. Sastry, "Fundamental Limits of Nonintrusive Load Monitoring," in *Proceedings of the 3rd International Conference on High Confidence Networked Systems*, Oct. 2013, pp. 11–18.
[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, Massachusetts and London, England: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org/
[7] R. Pal, C. Chelmis, M. Frincu, and V. Prasanna, "Towards Dynamic Demand Response on Efficient Consumer Grouping Algorithmics," *IEEE Transactions on Sustainable Computing*, vol. 1, no. 1, pp. 20–34, Nov. 2016.
[8] J. Huang, C. Xing, S. Y. Shin, F. Hou, and C. Hsu, "Optimizing M2M Communications and Quality of Services in the IoT for Sustainable Smart Cities," in *IEEE Transactions on Sustainable Computing*, vol. 3, Jan. 2018, pp. 4–15.
[9] K. D. Anderson, A. Ocneanu, D. Benitez, D. Carlson, A. Rowe, and M. Berges, "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*, Jan. 2012, pp. 1–5.
[10] T. Kriechbaumer and H.-A. Jacobsen, "BLOND, a Building-Level Office Environment Dataset of Typical Electrical Appliances," *Scientific Data*, vol. 5, no. 180048, 2018.
[11] D. Jorde, M. Kahl, and H. Jacobsen, "MEED: An Unsupervised Multi-Environment Event Detector for Non-Intrusive Load Monitoring," in *IEEE SmartGridComm*, Oct. 2019, pp. 1–6.
[12] D. Jorde, "Reference Algorithms, MEED Implementation, and Trained Models," Oct. 2019. [Online]. Available: http://doi.org/10.5281/zenodo.3490218
[13] G. W. Hart, "Nonintrusive Appliance Load Monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992.
[14] K. D. Anderson, M. E. Berges, A. Ocneanu, D. Benitez, and J. M. F. Moura, "Event Detection for Non Intrusive Load Monitoring," in *38th Annual Conference on IEEE Industrial Electronics Society*, Oct. 2012, pp. 3312–3317.
[15] L. Pereira and N. Nunes, "An Experimental Comparison of Performance Metrics for Event Detection a Algorithms in NILM," in *4th International Workshop on NILM*, 2018. [Online]. Available: http://nilmworkshop.org/2018/proceedings/Paper_ID07.pdf
[16] M. Valovage and M. Gini, "Label Correction and Event Detection for Electricity Disaggregation," in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, May 2017, pp. 990–998.
[17] L. Pereira, "Developing and Evaluating a Probabilistic Event Detector for Non-Intrusive Load Monitoring," in *Sustainable Internet and ICT for Sustainability*, Dec. 2017, pp. 1–10.
[18] J. M. Alcala, J. Urena, and A. Hernandez, "Event-Based Energy Disaggregation Algorithm for Activity Monitoring From a Single-Point Sensor," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 10, pp. 2615–2626, Oct. 2017.
[19] ——, "Event-based Detector for Non-Intrusive Load Monitoring based on the Hilbert Transform," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation*, Sep. 2014, pp. 1–4.
[20] L. d. Baets, J. Ruyssinck, D. Deschrijver, and T. Dhaene, "Event Detection in NILM using Cepstrum Smoothing," in *3rd International Workshop on NILM*, May 2016, pp. 1–4. [Online]. Available: https://users.ugent.be/~didschri/papers/2016_05__NILM_Cont.pdf
[21] L. d. Baets, J. Ruyssinck, C. Develder, T. Dhaene, and D. Deschrijver, "Optimized Statistical Test for Event Detection in Non-Intrusive Load Monitoring," in *IEEE International Conference on Environment and Electrical Engineering and IEEE Industrial and Commercial Power Systems Europe*, Jun. 2017, pp. 1–5.

[22] B. Wild, K. S. Barsim, and B. Yang, "A new Unsupervised Event Detector for Non-Intrusive Load Monitoring," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2015, pp. 73–77.

[23] K. S. Barsim and B. Yang, "Sequential Clustering-Based Event Detection for Non-Intrusive Load Monitoring," in *Proceedings of the 6th International Conference on Computer Science and Information Technology*, Jan. 2016, pp. 77–85.

[24] Z. Zheng, H. Chen, H. Xiaowei, and L. Xiaowei, "A Supervised Event-Based Non-Intrusive Load Monitoring for Non-Linear Appliances," *Sustainability*, vol. 10, no. 4, pp. 1001–1029, Mar. 2018.

[25] K. N. Trung, E. Dekneuvel, B. Nicolle, O. Zammit, C. N. Van, and G. Jacquemod, "Event Detection and Disaggregation Algorithms for NIALM System," in *2h International Workshop on NILM*, 2014. [Online]. Available: https://pdfs.semanticscholar.org/11fb/703fe0a1c7cddc8fd8cf06671d0f661aad38.pdf

[26] P. Held, D. Weißhaar, S. Mauch, D. O. Abdeslam, and D. Benyoucef, "Parameter Optimized Event Detection for NILM Using Frequency Invariant Transformation of Periodic Signals (FIT-PS)," in *IEEE 23rd International Conference on Emerging Technologies and Factory Automation*, Sep. 2018, pp. 832–837.

[27] M. Lu and Z. Li, "A Hybrid Event Detection Approach for Non-Intrusive Load Monitoring," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 528–540, Jan. 2020.

[28] O. P. Patri, A. V. Panangadan, C. Chelmis, and V. K. Prasanna, "Extracting Discriminative Features for Event-Based Electricity Disaggregation," in *IEEE Conference on Technologies for Sustainability*, Jul. 2014, pp. 232–238.

[29] S. Houidi, F. Auger, H. B. A. Sethom, L. Miègeville, D. Fourer, and X. Jiang, "Statistical Assessment of Abrupt Change Detectors for Non-Intrusive Load Monitoring," in *IEEE International Conference on Industrial Technology*, Feb. 2018, pp. 1314–1319.

[30] C. C. Yang, C. S. Soh, and V. V. Yap, "A Systematic Approach to ON-OFF Event Detection and Clustering Analysis of Non-Intrusive Appliance Load Monitoring," *Frontiers in Energy*, vol. 9, no. 2, pp. 231–237, May 2015.

[31] Z. Zhu, Z. Wei, B. Yin, S. Zhang, and X. Wang, "A Novel Approach for Event Detection in Non-Intrusive Load Monitoring," in *IEEE Conference on Energy Internet and Energy System Integration*, Nov. 2017, pp. 1–5.

[32] S. B. Leeb and J. L. Kirtley, "A Multiscale Transient Event Detector for Nonintrusive Load Monitoring," in *19th Annual Conference of IEEE Industrial Electronics*, Nov. 1993, pp. 354–359.

[33] S. B. Leeb, S. R. Shaw, and J. L. Kirtley, "Transient Event Detection in Spectral Envelope Estimates for Nonintrusive Load Monitoring," *IEEE Transactions on Power Delivery*, vol. 10, no. 3, pp. 1200–1210, Jul. 1995.

[34] L. Jiang, S. Luo, and J. Li, "Automatic Power Load Event Detection and Appliance Classification based on Power Harmonic Features in Nonintrusive Appliance Load Monitoring," in *8th Conference on Industrial Electronics and Applications*, Jun. 2013, pp. 1083–1088.

[35] M. Liu, J. Y. X. Wang, and J. Lu, "A new Event Detection Technique for Residential Load Monitoring," in *18th International Conference on Harmonics and Quality of Power*, May 2018.

[36] Y. Jin, E. Tebekaemi, M. Berges, and L. Soibelman, "Robust Adaptive Event Detection in Non-Intrusive Load Monitoring for Energy Aware Smart Facilities," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2011, pp. 4340–4343.

[37] F. Jazizadeh, A. Milad, and W. Jue, "Self-configuring Event Detection for Electricity Disaggregation," in *4th International Workshop on NILM*, Mar. 2018. [Online]. Available: http://nilmworkshop.org/2018/proceedings/Poster_ID14.pdf

[38] R. Cox, S. B. Leeb, S. R. Shaw, and L. K. Norford, "Transient Event Detection for Nonintrusive Load Monitoring and Demand Side Management using Voltage Distortion," in *21st IEEE Applied Power Electronics Conference and Exposition*, Mar. 2006, pp. 1751–1757.

[39] S. N. Patel, T. Robertson, J. A. Kientz, M. S. Reynolds, and G. D. Abowd, "At the Flick of a Switch: Detecting and Classifying unique Electrical Events on the Residential Power Line," in *International Conference on Ubiquitous Computing*, Sep. 2007, pp. 271–288.

[40] P. Meehan, C. McArdle, and S. Daniels, "An Efficient, Scalable Time-Frequency Method for Tracking Energy Usage of Domestic Appliances using Two-Step Classification Algorithm," *Energies*, vol. 7, no. 11, pp. 7041–7066, Oct. 2014.

[41] S. Wang and Y. Bo, "A Novel Nonintrusive Transient Event Detection Based on the Current," in *2nd International Conference on Mechanical Control and Automation*, 2017, pp. 298–304.

[42] T. Lu, Z. Xu, and B. Huang, "An Event-Based Nonintrusive Load Monitoring Approach: Using the Simplified Viterbi Algorithm," *IEEE Pervasive Computing*, vol. 16, no. 4, pp. 54–61, Oct. 2017.

[43] C. C. Yang, C. S. Soh, and V. V. Yap, "Comparative Study of Event Detection Methods for Non-intrusive Appliance Load Monitoring," *Energy Procedia*, vol. 61, pp. 1840 – 1843, Jan. 2014.

[44] Z. Zhu, S. Zhang, Z. Wei, B. Yin, and X. Huang, "A Novel CUSUM-Based Approach for Event Detection in Smart Metering," *Materials Science and Engineering*, vol. 322, no. 7, Mar. 2018.

[45] A. R. Rababaah and E. Tebekaemi, "Electric Load Monitoring of Residential Buildings using Goodness of Fit and Multi-Layer Perceptron Neural Networks," in *IEEE International Conference on Computer Science and Automation Engineering*, May 2012, pp. 733–737.

[46] H.-H. Chang, K.-L. Chen, Y.-P. Tsai, and W.-J. Lee, "A New Measurement Method for Power Signatures of Nonintrusive Demand Monitoring and Load Identification," *IEEE Transactions on Industry Applications*, vol. 48, no. 2, pp. 764–771, Mar. 2012.

[47] H. Gao, L. Zhang, L. Qiao, and Z. Tang, "An Improved Permutation Entropy Algorithm for Non-intrusive Load State Change Detection," in *IEEE Innovative Smart Grid Technologies - Asia*, May 2018, pp. 886–890.

[48] S. Yi, X. Yin, Y. Diao, B. Wang, and P. Wu, "A New Event-detection Method Based on Composite Windows in NILM for Industrial Settings," in *IEEE Sustainable Power and Energy Conference*, Nov. 2019, pp. 2768–2771.

[49] Y. Yang, J. Zhong, W. Li, T. A. Gulliver, and S. Li, "Semi-Supervised Multi-Label Deep Learning based Non-intrusive Load Monitoring in Smart Grids," *IEEE Transactions on Industrial Informatics*, pp. 1–11, Nov. 2019.

[50] D. Li and S. Dick, "Residential Household Non-Intrusive Load Monitoring via Graph-Based Multi-Label Semi-Supervised Learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 4615–4627, Aug. 2019.

[51] S. Makonin and F. Popowich, "Nonintrusive Load Monitoring (NILM) Performance Evaluation," *Energy Efficiency*, vol. 8, pp. 809–814, Dec. 2014.

[52] V. Chandola, A. Banerjee, and V. K. Vipin, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, Jul. 2009.

[53] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[54] F. Chollet and others, *Keras*, 2015. [Online]. Available: https://keras.io

[55] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations*, May 2015, pp. 1–15. [Online]. Available: http://arxiv.org/abs/1412.6980

[56] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *Proceedings of the 13. International Conference on AISTATS*, May 2010, pp. 249–256. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf

[57] W. W. Piegorsch, "Sample Sizes for Improved Binomial Confidence Intervals," *Computational Statistics and Data Analysis*, vol. 46, no. 2, pp. 309–316, Jun. 2004.

Daniel Jorde received his M.Sc. in Information Systems from the Technical University of Munich (TUM) in 2018. Currently, he is pursuing his Ph.D. at TUM, working on the MobiCM project. His research interests include various topics in the area of non-intrusive power monitoring, predictive maintenance, and machine learning in general.

Hans-Arno Jacobsen received his Ph.D. degree at Humboldt University in Berlin. He engaged in postdoctoral research at INRIA, before moving to the University of Toronto in 2001. He researches the intersection of distributed systems and data management, with a focus on middleware systems, event processing, and cyber-physical systems. He received the Alexander von Humboldt Professorship award to engage in research at TUM.

D Electrical Appliance Classification using Deep Convolutional Neural Networks on High Frequency Current Measurements

Electrical Appliance Classification using Deep Convolutional Neural Networks on High Frequency Current Measurements

Daniel Jorde, Thomas Kriechbaumer, and Hans-Arno Jacobsen
Chair for Application and Middleware Systems
Technische Universität München, Germany
Email: daniel.jorde@in.tum.de

***Abstract*—Monitoring the energy demand of appliances can raise consumer awareness and therefore reduce energy consumption. Using a single-point measurement of mains energy consumption can keep costs and hardware complexity to a minimum. This data stream of raw voltage and current measurements can be used in machine learning tasks to extract information. We apply Deep Convolutional Neural Networks on an electrical appliance classification task, using raw high frequency start up events from two datasets. We further present Data Augmentation techniques to improve the model performance and evaluate different data normalization techniques. We achieve a perfect classification on WHITED and a F1-Score of 0.69 on PLAID.**

I. Introduction

Researchers are striving to find solutions to reduce the total energy consumption, due to climatic change and the increasing energy demand of both industrial and residential consumers. By making consumers aware of their detailed energy consumption it is possible to achieve significant energy savings. Energy consumption data from electrical appliances can be recorded and analyzed to provide appropriate feedback to the consumers and the utilities [1], [2]. Other applications are motivated by the insights from performing such analysis: non-invasive monitoring of elderly people [3], anomaly detection of malfunctioning devices to reduce maintenance costs [4], or condition monitoring on naval vessels [5]. The required information to enable such applications can be derived by monitoring the electric appliances. One major approach is Non-Intrusive Load Monitoring (NILM). It aims to use aggregated electrical signals, for example, measured at the electrical mains of a building. NILM is a step-wise process and one of its major steps is to identify individual appliances from the voltage and current signals. The appliance identification problem is particularly challenging and therefore still not sufficiently solved [2], [6]. Past research focused mainly on residential areas and their appliances, which resulted in multiple public datasets of their measured electrical signals [7], [8].

Smart Meters are the most widespread devices to record such aggregated data, usually at a low sampling rate ($\sim$ 1 Hz) [4]. Low frequency data provides only enough information to identify some of the major appliances, making it challenging to identify multiple, smaller ones at the same time [4]. With higher sampling rates, it is possible to distinguish even such appliances. Most of the ongoing research is dealing with low frequency data from Smart Meters due to their growing deployment rate. There is a clear lack of methods for the high frequency domain [9]. Methods using datasets that are sampled with a frequency of 1 Hz or less are considered as low frequency methods according to [10]. We do not take low frequency data into account, due to the previously described limitations. Subsequently, approaches using datasets with a higher sampling frequency are considered as high frequency methods. In order to identify single appliances and to disaggregate loads, most methods use manually derived appliance signatures as features for various supervised machine learning classifiers [2], [9]. Such features require extensive domain knowledge and are dependent on different appliance types [9].

The main contribution of this paper is to investigate the usage of the raw, high resolution current signal in order to classify individual appliances with Deep Convolutional Neural Networks (DCNN). It is valid to assume the switching continuity principle because of the high sampling rate of the measurements [11] and therefore to use datasets that treat the appliance measurements isolated from one another. We compare our DCNN classification approach to the state of the art approaches using handcrafted (manually-derived) features. Furthermore, we investigate multiple Data Augmentation (DA) techniques to improve the performance of the appliance identification classifier. Besides this, we investigate different standardization and scaling methods to normalize the data. Our classifier is designed to distinguish between multiple appliance types. We use publicly available high frequency energy datasets: WHITED [7] and PLAID [8]. These datasets already provide a clean signal trace with annotated device labels suitable for machine learning. Datasets with long-term measurements, such as BLOND [12] or UK-DALE [13], do not provide this level of ground truth granularity. The evaluation considers multiple DCNN architectures and hyperparameter configurations.

The rest of the paper is organized as follows: In Section II, related work is presented. In Section III, we present the DA and the normalization techniques together with our model architecture. Section IV presents the experimental setup, followed by the evaluation of the experiments in Section V. Section VI then concludes this paper.

II. Related Work

Appliance identification, a subtask of energy disaggregation, can be modeled as a typical machine learning classification problem. Most of the machine learning techniques rely on using pre-computed sets of features, i.e., individual appliance signatures, to distinguish between appliances [9]. Frequently applied algorithms are Hidden Markov Models [14]–[16], Support Vector Machines [17], [18], and k-Nearest Neigbors [9], [17]. Besides these algorithms, Artificial Neural Networks (ANN) are increasingly applied on the problem, due to their success in other research fields. They are applied to both low [19], [20] and high frequency datasets [21], [22]. When looking at the high frequency domain, most of the works propose simple, fully-connected feedforward neural networks for residential settings [23]–[28]. Apart from fully-connected ANNs, Baets et al. [22] proposed a DCNN for identifying different appliances using VI-trajectory-based appliance signatures. On the other hand, several authors have applied Recurrent Neural Networks (RNNs) on low frequency data to both energy disaggregation [19], [29], [30] and appliance identification [20], [31] with great success, yet high frequency data have not been extensively tested with RNNs. The lack of methods using RNNs in the high frequency domain can be explained by the high computational requirements for high-dimensional (high frequency) input data. Even in the low frequency domain, some works in NILM rely on first using convolutional layers to down-sample the input signal and to detect features before applying recurrent layers to it [19], [29]. Besides the different algorithms that are applied to the task, no approach directly uses the raw signal to perform the appliance classification to the best of our knowledge. Studies have shown that different appliances and system settings require different appliance signatures [9]. Using a high frequency signal, without dropping information by pre-processing it extensively, promises to allow to distinguish between multiple, even smaller appliances [4]. In general, ANNs and in particular DCNNs, are able to efficiently process high dimensional data and to automatically extract meaningful features from it [32]. Problems in other research fields, especially in audio related tasks, exhibit a structure that is similar to the one of the appliance identification problem. In [33], the authors successfully used DCNNs to recognize environmental sounds, supporting the idea to also apply DCNNs to distinguish different appliance types. They propose a DCNN architecture which is able to handle long input sequences, containing up to 32000 data points. The input sequences we use for the appliance identification task exhibit a similar length to the data used in the audio related task [33]. Roos et al. [34] manually derived an appliance hierarchy, using the appliances' inherit electrical components and behaviour. Furthermore, the authors claim that a hierarchical classification approach is likely to enhance the identification capabilities of a classifier. DCNNs are designed to automatically extract hierarchical features while performing the classification task [32]. Therefore, it seems possible that DCNNs can make use of the hierarchy of the appliance types to efficiently perform the classification task. The manually designed appliance hierarchy supports the claim to deliver state of the art results on the identification problem.

III. Appliance Identification Approach

Our classification approach and the experiments we developed are designed based on the following main design principles: *Raw Input* and *Flexibility*. To fulfill the first requirement, as little pre-processing as possible should be applied on the data. Therefore, only minimal normalization and scaling of the raw input signal is to be used. For the *Flexibility* requirement, the chosen model needs to be flexible with respect to new appliances. In order to being able to compare the results to the ones obtained in [9], we used the same datasets and input sizes. Only the current signal is used as an input to the DCNNs in the following, because it exhibits most of the information necessary to perform the identification task [9]. Both datasets are comparatively small with high dimensional samples, resulting in a particular hard classification problem [32]. To overcome this issue, the number of samples is increased by applying two Data Augmentation techniques to both datasets and the architecture of the DCNN is designed to learn hierarchical features while gradually down-sampling the input signal. By doing this, the network learns a lower dimensional feature representation that is used as an appliance signature in the identification task.

A. Data Pre-Processing

We only apply a minimum amount of pre-processing to the raw current signal and we evaluate three pre-processing techniques and their influence on classification performance. Furthermore, we evaluated the performance when applying no pre-processing at all to the input signal. Applying some pre-processing to the data is motivated by the circumstance that neural networks usually benefit from a standardized input [32]. The first technique we applied, is a *z-score* normalization to adjust the input to have zero mean and unit variance by applying:

$$x_{norm} = \frac{x - \mu}{\sigma}$$

To estimate the standard deviation σ, we take the standard deviation over all samples in the dataset. In addition to this, we take the mean of each appliance window to estimate the expected value μ for the normalization. This is a small variation to the procedure proposed by [19], that has shown to produce more normalized input signals. Instead of normalizing the signal, we also only scaled the input by using two different min-max scaling procedures. Therefore, we evaluated scaling the data into two value ranges, i.e., $[0, +1]$ and $[-1, +1]$ respectively. The transformation to the first interval is given by

$$x_{scaled} = \frac{x - min(x)}{max(x) - min(x)}$$

In order to scale the data into the second interval, we used the formula below.

$$x_{scaled} = 2 \times \frac{x - min(x)}{max(x) - min(x)} - 1$$

Besides the presented techniques, we also evaluated the classifier without transforming the input data at all. In contrast to other approaches like [22], no down-sampling is applied before feeding the data into the classifier.

B. Data Augmentation

One of the challenges in applying Deep Learning models to NILM datasets like WHITED and PLAID is the small number of samples per class. Collecting sub-metered data usually requires a high effort and special metering devices [35], especially when sampling at high frequencies [4]. In order to support the training of neural networks, the number of samples can be increased. In addition to this, training the classifier with augmented data has a regularization effect on the model, helping it to generalize better to new samples [32]. Special care must be given to DA transformations to avoid changing the class of the sample. In order to increase the dataset size by an order of magnitude, we applied two transformations to the samples:

1) *Phase Shift*: Shift the activation window by one phase.
2) *Half-Phase Flip*: Shift the activation window by a half-phase and flip the signal, i.e., change the signs.

A phase shift increases the time in-variance, whereas a half-phase flip increases the flexibility of the classifier to variations in the signal. Figure 1 shows the DA process. First, the original window is shifted one phase to the left to generate a sample. Afterwards, the signal is shifted by a half-phase and the signs of the measurement points are inverted.

C. Model Architecture

Appliance identification can be modelled as a multi-class classifications problem, where each instance gets assigned exactly one out of many possible classes. To increase the flexibility of the classifier and to facilitate parallel training and inference, we reformulated the multi-class problem into multiple binary ones for an One-versus-All (OvA) approach. We trained one binary classifier per appliance type and aggregated the results afterwards. This results in n classifiers for the n appliance types we want to distinguish, with the n_i^{th} classifier's positive output belonging to appliance class $c_i \in C$. Each of the binary classifiers outputs two probabilities, one for the positive and one for the negative class (i.e., all other classes). In the aggregation step we then take the highest positive probability over all classifiers and assign it to the particular sample. If, for each classifier, the negative class probability is higher then the positive one, we assign the sample to a "none" class. Using a "none" class allows us to group and further investigate samples which are unknown to the classifier. As appliance identification is one of the base steps for a lot of applications, miss-classifications can have a huge effect on the whole process. The conservative treatment of samples for which the network is uncertain about helps to detect problems early in the analytical pipeline. Each binary classifier is a fully DCNN with the following architectural components: The non-linearities we used in the hidden layers of the networks are Rectified Linear Units (ReLU). Furthermore, we chose to use the Nesterov Accelerated Gradient optimizer, with a momentum weight of 0.9 [36]. For parameter initialization we used a Glorot initializer [37]. Works like [33], [38] have shown how one can learn useful features from high-dimensional input signals by gradually down-sampling the input when it passes through the network. To prevent the network from overfitting the data, we used L2-regularization and early stopping. Motivated by the findings of Dai et al. [33] on the similar audio classification tasks, we used large receptive fields in the convolutional layers and multiple pooling layers to handle the input. We then designed three DCNN architectures and evaluated them (Table I). Some of the hyperparameters differ slightly between the datasets due to the different size of the input vectors. The notation for the convolutional layers is as follows: (filter,kernel,stride) x number. For the pooling layers, we denoted the kernel and the stride we used.

To obtain variations of the base architecture we stacked different amounts of convolutional layers on one another. Besides this, we evaluated multiple hyperparameters (e.g., stride and kernel sizes) to obtain a configuration that produces the best results over all binary classifiers. All the pooling and the convolutional layers use *same* zero-padding, except the Avg-Pooling layer, which uses *valid* padding. It takes the average value of every feature map at the end and produces, after flattening the output, a one-dimensional representation for the appliances. We apply the *softmax* function at the end of the pipeline to condition the output signal. The cost function we use is the cross-entropy cost function. When building multiple binary classifiers, the distribution of the class labels in the training data becomes highly imbalanced. For each classifier n_i the amount of data for the class c_i is much smaller then the count of samples labelled with the negative class. Many real-world datasets are highly imbalanced, therefore a lot of solutions have been proposed by researchers from different fields [39]. This extrinsic between-class-imbalance can be solved by various methods, one type being cost-sensitive approaches. In general, one uses a cost matrix to assign different cost values to the classes in order to adjust for the imbalance. In the binary case, one typically uses two values Cost(*Maj,Min*) and Cost(*Min,Maj*), with the first being the cost to classify one sample from the minority class as a majority one and the latter one exactly the other way round. To prevent imbalances, we assigned a higher cost to miss-classifying minority class samples, i.e., Cost(*Maj,Min*) $>$ Cost(*Min,Maj*) [39]. In [40] the authors evaluated multiple cost-sensitive approaches for neural network classifiers. Among the proposed ones, we decided to choose an approach that aims to adapt the network output by assigning class specific costs because of its simplicity. The factor we used to scale the outputs stems from the relative class imbalance, i.e. we use the proportion of minority class to majority class samples.

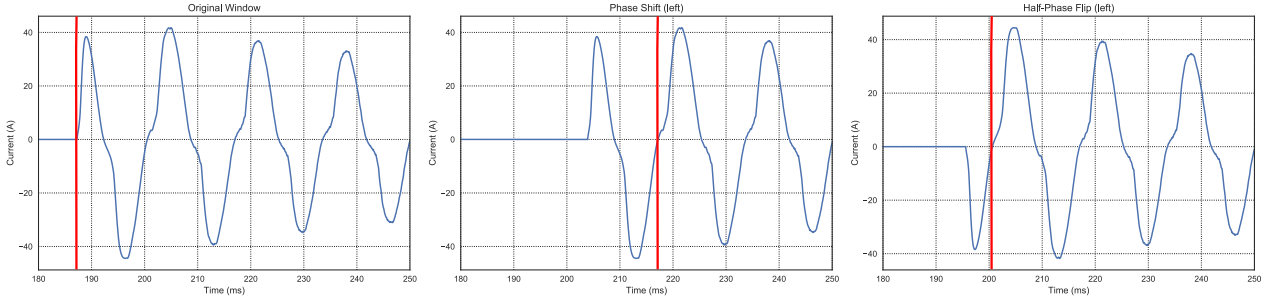


Fig. 1. Augmentation techniques applied to the activation window of a vacuum from PLAID [8]

TABLE I
BASIC DCNN ARCHITECTURES FOR WHITED AND PLAID

	WHITED			PLAID		
	DCNN4	DCNN5	DCNN6	DCNN4	DCNN5	DCNN6
Input Layer	22050			15000		
Convolutional Layer	(64, 80, 2) x 1			(64, 80, 2) x 1		
Max-Pooling Layer	5, 5			3, 3		
Convolutional Layer	(64, 3, 1) x 1	(64, 3, 1) x 2	(64, 3, 1) x 2	(64, 2, 1) x 1	(64, 2, 1) x 2	(64, 2, 1) x 2
Max-Pooling Layer	3, 3			4, 4		
Convolutional Layer	(128, 3, 1) x 1	(128, 3, 1) x 1	(128, 3, 1) x 2	(128, 3, 1) x 1	(128, 3, 1) x 1	(128, 3, 1) x 2
Max-Pooling Layer	5, 5			5, 5		
Convolutional Layer	(256,3,1)			(256,3,1)		
Max-Pooling Layer	2, 2			5, 5		
Avg-Pooling Layer	49, 1			25, 1		
Softmax-Output Layer	2			2		

IV. EXPERIMENTAL METHODOLOGY

We conducted several experiments to select the best performing DCNN architecture and hyperparameter settings and to evaluate the pre-processing and DA techniques. For model selection, we trained three binary classifiers on three representative appliances. We then used the results from these experiments to train one DCNN per appliance type in the dataset. To compare the performance of our models to the ones using handcrafted features, we adapted the experimental setup from an extensive feature study by Kahl et al. [9].

A. Datasets

The DCNNs are trained on start-up events of WHITED [7] and PLAID [8]. Both datasets are sampled with high frequencies: WHITED with 44.1 kHz [7] and PLAID with 30 kHz [8]. Similar to [9] we further adapted the 500ms activation window size and used subsets of the two datasets. This activation window size results in input vectors containing 22050 measurement points for WHITED and 15000 for PLAID. For WHITED, we used a typical household subset with 27 appliance types. PLAID on the opposite contains multiple models per appliance type. We used all of the 11 appliance types in the dataset. Therefore, we trained 27 binary classifiers for WHITED and 11 for PLAID. To increase the sample sizes of the datasets, we applied the previously described Data Augmentation techniques. We first applied the Phase Shift four times and afterwards the Half-Phase Flip transformation to each sample. This results in a 10 factor increase of the amount of data. For the experiments, we split the data into 80% for training and divided the remaining data into equal parts for validation and testings.

B. Metrics

To evaluate the performance of the appliance classifiers, we apply the commonly used F1-Score. The F1-Score is based on using the amount of True Positives (*TP*), False Positives (*FP*), True Negatives (*TN*), and False Negatives (*FN*). The F1-Score is defined as follows:

$$F1\text{-}Score = \frac{2 \times recall \times precision}{recall + precision}$$

$$precision = \frac{TP}{TP + FP} \quad recall = \frac{TP}{TP + FN}$$

To compute the performance over all appliance models for a given architectural configuration, we take the non-weighted average score over all appliances and compute a macro F1-Score. We do not report the Accuracy metric, because it can be deceiving in settings with highly imbalanced data [39].

V. EXPERIMENTAL RESULTS

Looking at the results of the experiments for both datasets, one can see that the effect of the normalization procedures and the model architectures are highly dataset dependent. For the final model, we selected the model configuration that performs best over all appliance modes to obtain a general setting. Another approach is to select different configurations for the individual binary models, possibly leading to better

classification results. We chose the first approach, because it generalizes better and requires less human interference. The best F1-Scores for the respective datasets and model architectures are shown in Table II. For WHITED, the DCNN4

TABLE II
BEST F1-SCORES ON THE AUGMENTED DATASETS

Dataset	Metric	DCNN4	DCNN5	DCNN6
WHITED	F1-Score	**1**	0.91	0.59
PLAID	F1-Score	0.65	**0.69**	0.58

architecture turned out to perform best on the augmented data without normalizing it, using the corresponding hyper-parameter configurations in Table III. The kernel and filter size parameters alter the respective values in the first layer of the architecture and subsequently the upper layers according to the base architecture (Table I). The model configurations

TABLE III
HYPERPARAMETER CONFIGURATIONS FOR THE BEST MODELS

Model	Learning Rate	Batch Size	Kernel	Filter
WHITED DCNN4	0.01	25	80	64
PLAID DCNN5	0.01	40	126	128

with a high learning rate and a small batch size performed best for both datasets. When looking at the normalization methods we applied, one can see that the z-normalization and applying no normalization clearly outperformed both min-max scaling approaches. We chose to apply no normalization in the final model instead of using z-normalization, because it follows the declared minimal pre-processing approach and preserves all information in the raw data. The best model achieved a macro F1-Score of **1** (Figure 2). Over all parameter configurations, the models on the augmented data have a higher average F1-Score compared to the models trained on the non-augmented data. Despite this, some configurations still achieved a F1-Score of 1, even on the non-augmented data. The best performing approach in [9] also achieves a macro F1-Score of 1, but requires extensive feature engineering to do so. For PLAID, the best model, our DCNN5 model, achieved a macro F1-Score of **0.69** (Figure 3). For both of the confusion matrices it has to be noted that the "none" class is omitted from the results, because the information is intrinsically contained in the matrices. In contrast to the results on WHITED, the min-max scaling outperforms the other normalization techniques, followed by applying no normalization to the data. The effect of the Data Augmentation techniques is significant over all architectures. The best DCNN4 architecture achieved a F1-Score of 0.65 on the augmented data and only a score of 0.28 on the non-augmented one. The other classifiers showed similar results, further supporting the effect of the proposed Data Augmentation techniques. When comparing the results to the ones by Kahl et al. on PLAID [9], one sees that our approach is outperformed. The authors' handcrafted feature based k-Nearest Neighbor approach achieved a macro F1-

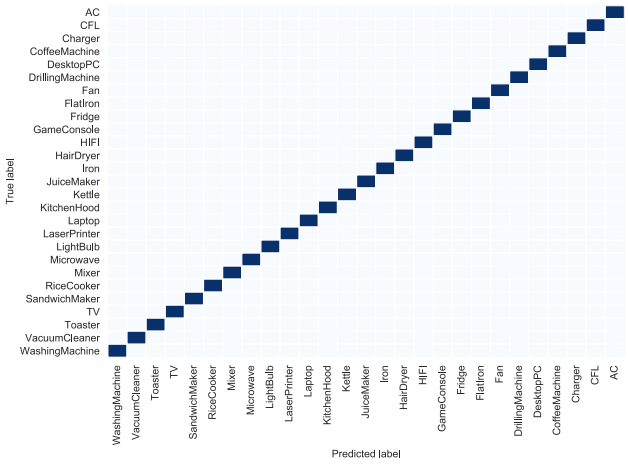


Fig. 2. Confusion Matrix of the DCNN4 architecture on WHITED

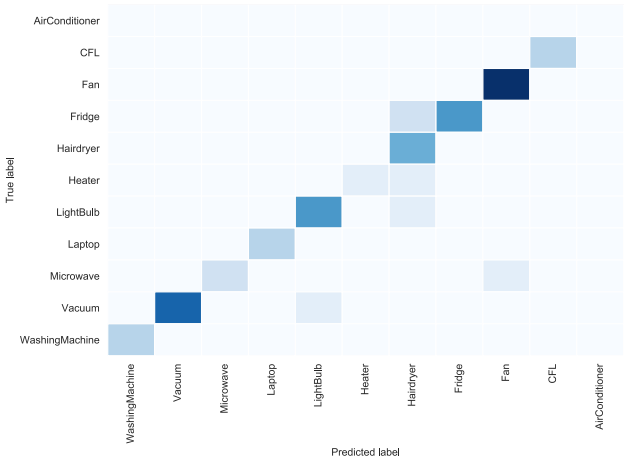


Fig. 3. Confusion Matrix of the DCNN5 architecture on PLAID

Score of 0.89 [9]. Despite this, our experiments clearly showed the feasibility of using the raw current signal as an input to perform the classification task.

VI. CONCLUSIONS

In this paper, we developed Deep Convolutional Neural Networks to identify electrical appliances from raw, high frequency activation events. The results show that by applying Data Augmentation techniques and by carefully selecting among different pre-processing techniques, we achieved state of the art results on WHITED and good ones on PLAID. We performed as well as classifiers based on extensive feature engineering on WHITED, and showed the feasibility of our approach on PLAID. Our approach has the advantage that no explicit feature engineering by a domain expert is necessary. Different appliance types require different sets of features to identify them [9] and therefore significant feature engineering by a domain expert. Our approach circumvents this, because it

provides a generic approach to the problem. This is particularly useful in volatile settings with a lot of different appliances.

ACKNOWLEDGMENT

This research was partially funded by the Alexander von Humboldt Foundation established by the government of the Federal Republic of Germany.

REFERENCES

[1] M. R. Asghar, G. Dán, D. Miorandi, and I. Chlamtac, "Smart meter data privacy: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2820–2835, 2017.

[2] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey," *Sensors*, vol. 12, no. 12, pp. 16 838–16 866, 2012.

[3] J. M. Alcalá, J. Ureña, Á. Hernández, and D. Gualda, "Assessing Human Activity in Elderly People Using Non-Intrusive Load Monitoring," *Sensors*, vol. 17, no. 2, p. 351, 2017.

[4] K. Carrie Armel, A. Gupta, G. Shrimali, and A. Albert, "Is Disaggregation the Holy Grail of Energy Efficiency? The Case of Electricity," *Energy Policy*, vol. 52, pp. 213–234, 2013.

[5] T. DeNucci, R. Cox, S. B. Leeb, J. Paris, T. J. McCoy, C. Laughman, and W. C. Greene, "Diagnostic Indicators for Shipboard Systems using Non-Intrusive Load Monitoring," in *IEEE Electric Ship Technologies Symposium*. Piscataway, NJ: IEEE, 2005, pp. 413–420.

[6] G. W. Hart, "Nonintrusive Appliance Load Monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[7] M. Kahl, A. U. Haq, T. Kriechbaumer, and H.-A. Jacobsen, "WHITED - A Worldwide Household and Industry Transient Energy Data Set," in *3rd International Workshop on Non-Intrusive Load Monitoring*, 2016.

[8] J. Gao, S. Giri, E. C. Kara, and M. Bergés, "PLAID: A Public Dataset of High-resolution Electrical Appliance Measurements for Load Identification Research," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. New York, NY, USA: ACM, 2014, pp. 198–199.

[9] M. Kahl, A. U. Haq, T. Kriechbaumer, and H.-A. Jacobsen, "A Comprehensive Feature Study for Appliance Recognition on High Frequency Energy Data," in *Proceedings of the 8. ACM e-Energy conference*, vol. 5. New York, NY, USA: ACM, 2017, pp. 121–131.

[10] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng, "Load Signature Study—Part II: Disaggregation Framework, Simulation, and Applications," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 561–569, 2010.

[11] S. Makonin, "Investigating the Switch Continuity Principle assumed in Non-Intrusive Load Monitoring (NILM)," in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. Piscataway, NJ, USA: IEEE, 2016, pp. 1–4.

[12] T. Kriechbaumer and H.-A. Jacobsen, "BLOND, a building-level office environment dataset of typical electrical appliances," *Scientific Data*, vol. 5, 2018.

[13] J. Kelly and W. Knottenbelt, "The UK-DALE Dataset: Domestic Appliance-Level Electricity Demand and Whole-House Demand from Five UK Homes," *Computing Research Repository*, vol. abs/1404.0284, 2014.

[14] Z. J. Kolter and T. Jaakkola, "Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation," in *Proceedings of Machine Learning Research*, 2012.

[15] M. Zhong, N. Goddard, and C. Sutton, "Signal Aggregate Constraints in Additive Factorial HMMs, with Application to Energy Disaggregation," in *Advances in Neural Information Processing Systems*, 2014, pp. 3590–3598.

[16] J. A. Mueller, A. Sankara, J. W. Kimball, and B. McMillin, "Hidden Markov models for nonintrusive appliance load monitoring," in *2014 North American Power Symposium (NAPS)*, 2014, pp. 1–6.

[17] O. Kramer, T. Klingenberg, M. Sonnenschein, and O. Wilken, "Non-Intrusive Appliance Load Monitoring with Bagging Classifiers," *Logic Journal of IGPL*, vol. 23, no. 3, pp. 359–368, 2015.

[18] L. Du, Y. Yang, D. He, R. G. Harley, T. G. Habetler, and B. Lu, "Support Vector Machine Based Methods for Non-Intrusive Identification of Miscellaneous Electric Loads," in *38th Annual Conference on IEEE Industrial Electronics Society*, 2012, pp. 4866–4871.

[19] J. Kelly and W. Knottenbelt, "Neural NILM," in *Proceedings of the 2nd ACM BuildSys*, D. Culler, Ed. New York NY: ACM, 2015, pp. 55–64.

[20] T.-T.-H. Le, J. Kim, and H. Kim, "Classification Performance using Gated Recurrent Unit Recurrent Neural Network on Energy Disaggregation," in *2016 ICMLC*. Piscataway, NJ: IEEE, 2016, pp. 105–110.

[21] H. Lange and M. Berges, "The Neural Energy Decoder: Energy Disaggregation by Combining Binary Subcomponents," in *3rd International Workshop on Non-Intrusive Load Monitoring*, 2016.

[22] L. de Baets, J. Ruyssinck, C. Develder, T. Dhaene, and D. Deschrijver, "Appliance Classification using VI Trajectories and Convolutional Neural Networks," *Energy and Buildings*, vol. 158, pp. 32–36, 2018.

[23] L. de Baets, C. Develder, D. Deschrijver, and T. Dhaene, "Automated Classification of Appliances using Elliptical Fourier Descriptors," in *8th IEEE International Conference on Smart Grid Communications (SmartGridComm 2017)*, 2017, pp. 1–6.

[24] H.-H. Chang, K.-L. Lian, Y.-C. Su, and W.-J. Lee, "Energy Spectrum-Based Wavelet Transform for Non-Intrusive Demand Monitoring and Load Identification," in *2013 IEEE Industry Applications Society annual meeting*. Piscataway, NJ, USA: IEEE, 2013, pp. 1–9.

[25] K. S. Barsim, L. Mauch, and B. Yang, "Neural Network Ensembles to Real-time Identification of Plug-level Appliance Measurements," in *3rd International Workshop on Non-Intrusive Load Monitoring*, vol. 2, 2016.

[26] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng, "Load Signature Study—Part I: Basic Concept, Structure, and Methodology," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 551–560, 2010.

[27] Y.-H. Lin and M.-S. Tsai, "A Novel Feature Extraction Method for the Development of Nonintrusive Load Monitoring System based on BP-ANN," in *2010 International Symposium on Computer, Communication, Control and Automation* , Q. Luo, Ed. Piscataway, NJ: IEEE, 2010, pp. 215–218.

[28] D. Srinivasan, W. S. Ng, and A. C. Liew, "Neural-Network-Based Signature Recognition for Harmonic Source Identification," *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 398–405, 2006.

[29] W. He and Y. Chai, "An Empirical Study on Energy Disaggregation via Deep Learning," in *2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016)*, ser. Advances in Intelligent Systems Researc. Paris, France: Atlantis Press, 2016.

[30] L. Mauch and B. Yang, "A New Approach for Supervised Power Disaggregation by using a Deep Recurrent LSTM Network," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. Piscataway, NJ: IEEE, 2015, pp. 63–67.

[31] J. Kim, T.-T.-H. Le, and H. Kim, "Nonintrusive Load Monitoring Based on Advanced Deep Learning and Novel Signature," *Computational Intelligence and Neuroscience*, vol. 2017, no. 2-3, pp. 1–22, 2017.

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, Massachusetts and London, England: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org/

[33] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very Deep Convolutional Neural Networks for Raw Waveforms," *2017 IEEE ICASSP*, pp. 421–425, 2017.

[34] J. G. Roos, I. E. Lane, E. C. Botha, and G. P. Hancke, "Using Neural Networks for Non-Intrusive Monitoring of Industrial Electrical Loads," in *IEEE Instrumentation and Measurement Technology Conference, 1994*. Piscataway, NJ,USA: IEEE, 1994, pp. 1115–1118.

[35] T. Kriechbaumer, A. Ul Haq, M. Kahl, and H.-A. Jacobsen, "MEDAL: A Cost-Effective High-Frequency Energy Data Acquisition System for Electrical Appliances," in *Proceedings of the 8th International Conference on Future Energy Systems*. New York, NY, USA: ACM, 2017, pp. 216–221.

[36] Y. Nesterov, "A Method for Unconstrained Convex Minimization Problem with the Rate of Convergence O(1/k^2)," *Doklady AN USSR*, vol. 269, pp. 543–547, 1983.

[37] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *Proceedings of the 13. International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[38] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *CoRR*, 2016.

[39] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[40] M. Kukar and I. Kononenko, "Cost-Sensitive Learning with Neural Networks," in *Proceedings of the 13th European Conference on Artificial Intelligence*. John Wiley & Sons, 1998, pp. 445–449.