Annoticity: A Smart Annotation Tool and Data Browser for Electricity Datasets

Benjamin Völker
voelkerb@informatik.uni-freiburg.de
Institute for Computer Architecture,
University of Freiburg

Marc Pfeifer*
Philipp M. Scholl*
pfeiferm@informatik.uni-freiburg.de
pscholl@informatik.uni-freiburg.de
Institute for Computer Architecture,
University of Freiburg

Bernd Becker
becker@informatik.uni-freiburg.de
Institute for Computer Architecture,
University of Freiburg

ABSTRACT

The growing request for eco-feedback and smart living concepts accelerated the development of Non-Intrusive Load Monitoring (NILM) algorithms during the last decade. Comparing and evaluating these algorithms still remains challenging due to the absence of a common benchmark datasets, and missing best practises for their application. Despite the fact that multiple datasets were recorded for the purpose of comparing NILM algorithms, many researchers still have to record their own dataset in order to meet the requirements of their specific application. Adding ground truth labels to these datasets is a cumbersome and time consuming process as it requires an expert to visually inspect all the data manually. Therefore, we propose the Annoticity inspection and labeling tool which simplifies the process of visualizing and labeling of electricity data. We use an event detector based on the log likelihood ratio test which achieved an F_1 score of 90.07 % in our experiments. Preliminary results indicate that the effort of generating event labels is reduced by 80.35 % using our tool.

CCS CONCEPTS

• **Human-centered computing** → **Visualization systems and tools**; • **Information systems** → *Data exchange*; *Web interfaces.*

KEYWORDS

datasets, electricity monitoring, NILM, automatic labeling

ACM Reference Format:
Benjamin Völker, Marc Pfeifer, Philipp M. Scholl, and Bernd Becker. 2020. Annoticity: A Smart Annotation Tool and Data Browser for Electricity Datasets. In *The 5th International Workshop on Non-Intrusive Load Monitoring (NILM'20), November 18, 2020, Virtual Event, Japan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3427771.3427844

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
NILM'20, November 18, 2020, Virtual Event, Japan
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8191-8/20/11…$15.00
https://doi.org/10.1145/3427771.3427844

1 INTRODUCTION

The global per capita electricity consumption was 3132 kW h in 2014 [20]. Producing this massive amount of electricity contributes to 27 % of the global greenhouse gas emissions [1]. Hence, reducing the global electricity consumption is a significant mission that starts in our homes. A meta-study by Ehrhardt-Martinez et al. [4] indicates that electricity savings of up to 13.7 % can be achieved if real-time per device electricity feedback is provided to the home's owner. Algorithms like Intrusive- (ILM) or Non-Intrusvie Load Monitoring (NILM) can be used to generate such device specific consumption and have shown to be effective in providing eco-feedback [7, 14, 17]. While ILM requires a dedicated electricity meter at each appliance, NILM requires only a single electricity meter that measures the composite load of all appliances. This composite load is disaggregated into the load of each individual appliance using specially designed and often individually trained algorithms. In particular event-based NILM algorithms detect events in the aggregated power consumption and relate them to state changes of an particular appliance, to reconstruct its individual consumption.

In addition to simply providing eco-feedback, smart home agents can be utilized to save electricity. Such agents could learn routines of the home's inhabitants by exploring when and how a user interacts with appliances. Based on such observations, these systems could derive advanced automations to directly save electricity or actively support the users by recommending advanced energy saving strategies. To learn these routines, information about state changes of, and interactions with an appliance are required. Smart devices like newer *SmartTVs* or smart light systems can directly provide such events (state changes or interactions). A retrofitable solution for non-smart devices is to automatically extract these events from the power consumption of an appliance (e.g. provided by a NILM or ILM algorithm).

The development of event-based NILM and smart agents depends on datasets with precise ground truth data, i.e., labeled events in the power consumption of each appliance.

Some electricity datasets were specifically recorded with event-detection in mind and deliver some appliance events as part of their ground truth data (e.g. switch-on and -off events in the BLUED dataset [5]). Adding such ground truth information after the recording remains challenging, as it requires human inspection of the data and expert knowledge. Therefor, researchers have developed semi-automatic labeling approaches such as [15] or [18]. Further crowd sourcing and gamification techniques have been examined to enable collaborative ground truth labeling and applying the "wisdom of

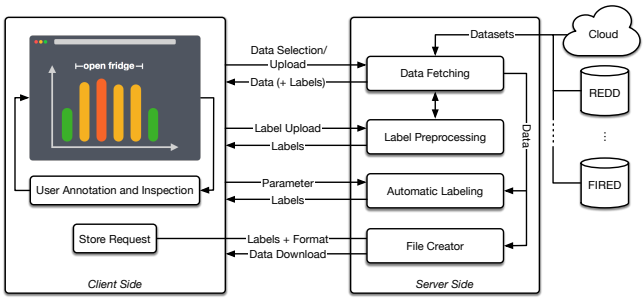

Figure 1: Flow of the Annoticity labeling tool. Data fetching, automatic labeling and file creation is performed on the server side, while plain labeling and user interaction is performed on the client side.

the crowd" [3]. Tools to label electricity data have been published by [16] and [8]. Although these tools exist, the inspection and labeling of public electricity datasets or newly recorded data still remains cumbersome. Datasets have to be downloaded from the internet and must be stored locally first. The shear size of some datasets (e.g. 35 TB for BLOND [11]) requires careful preparation. As no common file format is established, each dataset requires specific code to load data into memory, before even an initial inspection of the data is possible. Adding labels to those datasets is a challenging and time consuming task as several appliances change their state regularly resulting in hundred's of events per day. Pereira et al. evaluated their labeling tool in [16] and were able to find 94 % of the events automatically. Some shortcomings are that only appliances which draw more than 30 W are supported and that no textual descriptions can be added to events.

We present a novel annotation tool for the task of generating precise text based ground truth labels for electricity data. It is realized as a web application and provides direct access to various publicly available electricity datasets without requiring to download the data to disk. Users can add labels manually, review automatically generated labels, or modify existing label sets. As we detail in this paper, the automatic labeling significantly reduces the manual labeling effort. The labels can be downloaded in various file formats including an option to store the labels with the original data.

2 ANNOTICITY TOOL

The Annoticity labeling tool is implemented as an interactive web application. Manual labeling and inspection is performed on the client side while data fetching and automatic labeling is performed on the server. The workflow is depicted in Figure 1.

The server backend is written in *python* using the *Django* framework [6]. The backend's main purpose is to load the data and prepare it for visualisation, perform the task of automatic labeling and provide file downloads. Data can be uploaded through the web application. Currently, only Matroska [12] multimedia containers (*mkv*) are supported. The REDD [10], UK-DALE [9], BLOND [11], ECO [2] and the FIRED [19] datasets can be directly selected (more will be added). The backend resamples the data to a reasonable sampling rate according to the current time-span selected by the user. If the dataset already contains labels they will be displayed

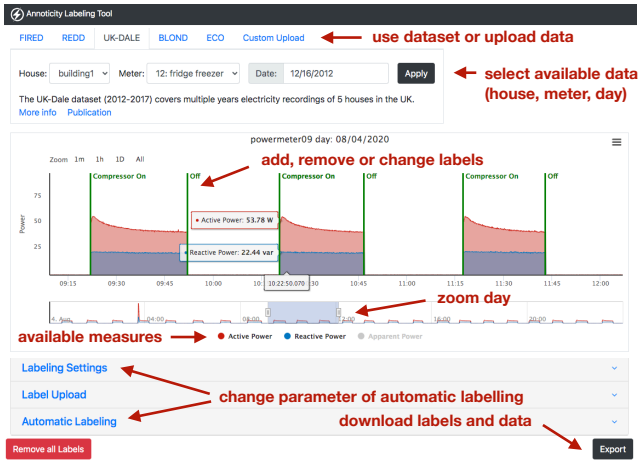

Figure 2: The graphic user interface of the Annoticity labeling tool. The fridge events were generated and clustered automatically. Each text description (*Off*, *Compressor On*) was only set once, other occurrences were labeled accordingly.

to the user. Additionally, a file containing labels can be uploaded and modified. The supported formats are *csv*, *srt* and *ass*. The automatic labeling, described in Section 3, generates labels from the data by identifying events in the data. These events are clustered, pre-labeled and sent to the client side for inspection and validation.

The client side is implemented in *HTML* and *JavaScript* and provides the frontend to the user. Annoticity's graphical user interface is shown in Figure 2. After either uploading a file or selecting a timespan and device of an available dataset, the user can visually inspect the data. Different measures (e.g. active and reactive power) can be selected, and data can be zoomed in which leads to a data download at a higher sampling rate. The user can add a label by clicking at the slope where an event occurs, remove the label by clicking on the vertical bar, or modify the label. Each label consists of a start time and a (possibly empty) text description. The frontend also allows to set the parameters of the automatic labeling algorithm. Labels are stored either as plain *csv*, *ass* or *srt* files or embedded into a *mkv* file together with the original data.

As Annoticity is designed as a web application, access to the data is provided from anywhere. The only requirements are a modern browser and an internet connection. With the user management directly built-in to the Django framework, the Annoticity tool is already primed for collaborative labeling and gamification mechanisms in a future version (e.g. as realized by [3]). Even though the tool is optimized to label electricity data, it can be applied to other time series data as well.

3 AUTOMATIC LABELING

The identification and labeling of events is split into three steps: event detection, a unique event identification and a high variance filtering. These steps are applied to the data's apparent or active power depending on availability. If neither apparent nor active power is available, but raw voltage and current waveforms are, the apparent power (S) is calculated from the these waveforms.

Event Detection: The event detector is based on previous work from Pereira et al. [13] and Völker et al. [18]. It uses an adapted version of the Log-Likelihood Ratio (LLR) test applied to the power signal to detect a change in the signals mean value. For each power sample (j) we calculate the likelihood ($L(j)$) that an event has happened at sample j. The calculation applies a detection window over the power signal. This detection window is split into two sub-windows, the *pre-event* window $[i, j[$ and the *post-event* window $[j, k]$. $L(j)$ is calculated as

$$L(j) = \ln\left(\frac{\sigma_{[i,j[}}{\sigma_{[j,k]}}\right) + \frac{\left(S(j) - \mu_{[i,j[}\right)^2}{2 \cdot \sigma^2_{[i,j[}} - \frac{\left(S(j) - \mu_{[j,k]}\right)^2}{2 \cdot \sigma^2_{[j,k]}}. \quad (1)$$

$\sigma_{[i,j[}$ and $\sigma_{[j,k]}$ are the standard deviations of the pre-event and post-event window while $\mu_{[i,j[}$ and $\mu_{[j,k]}$ are the means, respectively. To further clean this signal, $L(j)$ is set to zero, if the change of the mean value between the pre- and post-event is below a certain threshold ($thres_j$). Ultimately, the likelihood of an event present at sample j is calculated as

$$L(j) = \begin{cases} L(j), & if\ \left|\mu_{[i,j[} - \mu_{[j,k]}\right| > thres_j \\ 0, & otherwise \end{cases}. \quad (2)$$

$thres_j$ linearly increases with the mean power of the pre-event window as

$$thres_j = thres_{min} + m \cdot \mu_{[i,j[}, \quad (3)$$

where $thres_{min}$ is the minimum power change of interest and m is a coefficient that increases the threshold for larger mean values.

The reason for having an increasing threshold is that high power appliances typically also show a higher variance due to components such as motors or processing units which consume variable power depending on the current load (resistance or calculations). If $thres_j$ is set to a small fixed value, a high variance causes many falsely classified events. If $thres_j$ is set to a higher threshold (e.g. 30 W as in [15]), events of low power devices such as smartphone chargers or even smaller televisions are not detected at all. Therefore, $thres_j$ linearly increases with the power consumed around sample j to adapt to possibly larger fluctuations, which prevents false positives and false negatives.

If an event is detected at sample j, L is also non-zero around sample j as the change in mean is still observable in close proximity to the event. To identify the exact sample at which the event occurred, a *voting window* is applied to the signal L. Inside the window only the maximum of the absolute value of L is kept. We further restrict the minimum distance between two events with an additional parameter l.

Summarized, the algorithm has six adjustable parameters: the duration of the pre- and post-event windows, the duration of the voting window, the minimum detection threshold $thres_{min}$, the linear coefficient m and the minimum distance between two events l. These can be individually adjusted in the Annoticity tool. The standard values where chosen based on the findings of [18] as: pre-event window: 1 s, post-event window: 1.5 s, voting window: 2 s, $thres_{min} = 3$ W and $m = 0.005$. A user should specifically adjust the parameters $thres_{min}$ and l according to a-priory knowledge of the data: a low threshold $thres_{min}$ is required if events with small mean changes are expected, and a short l should be chosen if events can happen close in time.

Unique Event Identification: After detecting events, we calculate the mean power between all consecutive events. These mean values are used to identify identical events as most of the appliances draw a different power after an event (e.g. the kettle after it is switched on). Some appliances show a high rush-in power followed by a power settling due to e.g. motors. These high peaks would bias the mean value calculation. Hence, we remove the 10 % of the highest and lowest values before calculating the mean value. The set of these mean values is clustered using hierarchical clustering with a distance threshold determined by $thres_{min}$.

High Variance Filtering: For appliances which draw variable power depending on the current context (e.g. calculations of a PC, content of a television, or audio played by a hifi system, etc.) event detection based on the log likelihood causes a large number of false positives. To prevent those, we detect regions which show an increased signal variance, since this is typical for these appliances. We therefor calculate the mean and variance on a sliding window. If the variance is larger than 2 % of the mean value, high variance is assumed in this window. If consecutive windows of high variance exceed a length of 4 s, all events found in these windows are removed.

4 EVALUATION AND RESULTS

The proposed event detector was evaluated using the publicly available datasets REDD [10] and FIRED [19]. We used 8 days from each dataset and labeled all events manually by visually inspecting the data. In particular, we labeled 4030 events in total to serve as the ground truth for the following evaluation.

The results are presented in terms of the following metrics: True Positives (*TP*), False Positives (*FP*) and False Negatives (*FN*). A TP is defined as a detected event that is reflected within 2 s in the ground truth data. Accordingly, a FP is defined as a detected event without a corresponding event in the ground truth, and a FN is an event in the ground truth data which is not found during detection. To sum up the results of TP, FP and FN, we calculated the F_1 measure as

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}. \quad (4)$$

4.1 REDD dataset

The Reference Energy Disaggregation Dataset (REDD) was introduced by Kolter et al. in 2011 [10]. The authors used a custom built system to record the electricity consumption of six different homes in the US over a time period of 35 days. Apparent power measurements are available at mains, socket and sub-circuit-level at a sampling frequency of approximately 1/3 Hz. We used the socket- and sub-circuit-level data of eight days (from April 19th 2011 to April 26th 2011) from house 2 for evaluation and resampled the data to 1 Hz. Due to the low sampling rate of the REDD dataset, we used the following parameters for evaluation: pre-event window and post-event window = 3 s, voting window length = 4 s, $thres_{min} = 4.5$ W, $m = 0.009$ and $l = 2$ s.

The results are shown in Table 1. Overall, an F_1 score of 86.73 % was achieved. 1627 out of 1944 events (83.69 %) were identified correctly. The *refrigerator*, which exhibits the most events, shows

Table 1: Event detector performance on REDD. The *washer dryer* was not used during the evaluation time period.

Appliance	Events	TP	FP	FN	F_1
dishwaser	59	51	34	8	70.83
disposal	19	11	0	8	73.33
kitchen outlets #1	72	49	3	23	79.03
lighting	156	85	7	71	68.55
stove	47	40	6	7	86.02
microwave	158	130	20	28	84.42
washer dryer	0	0	2	0	0.00
kitchen outlets #2	525	464	5	61	93.36
refrigerator	908	797	104	111	88.11
sum	1944	1627	181	317	86.73

the highest number of FP. These stem from short defrosting cycles which have not been treated as relevant events during the manual ground truth labeling process. The *lighting* and the *kitchen outlets #2* show high numbers of FN due to events close or below the minimum mean change threshold of 4.5 W.

4.2 FIRED dataset

The Fully-labeled hIgh-fRequency Electricity Disaggregation (FIRED) dataset was introduced by Völker et al. in 2020 [19]. The authors used custom built hardware to record high frequency main's and socket level consumption data of an apartment building in Germany over a time period of 52 days. Power measurements are available for 21 appliances at a sampling rate of 50 Hz. We selected eleven of these appliances and evaluated the event detector on eight days of data (from July 28th 2020 to August 4th 2020). The following parameter are used for evaluation: pre-event window and post-event window length = 1 s and 1.5 s, voting window length = 2 s, $thres_{min}$ = 3 W, $m = 0.005$ and l=2 s.

The results are summarized in Table 2. The overall F_1 score for the FIRED dataset is 93.04 %. 1964 out of 2086 events are recognized correctly. The *coffee grinder* and the *hifi system* show a high number of FP due to high variance while grinding or playing audio. A high variance filtering as explained in Section 3 would have potentially removed these, but our evaluation focused solely on the event detection process. The *espresso machine* has very short heating cycles and a pump which happen to cause events close in time, i.e. smaller than $l = 2$ s, which are not detecting. These account for the comparably large amount of FN.

5 DISCUSSION

Manually adding 4030 labels for the evaluation took a decent amount of effort despite being just two weeks of data. The proposed automatic labeling of the Annoticity tool was able to correctly place 3591 of these labels. Correcting the results of the automatic labeling would consist of removing 353 FP and manually adding 439 FN. Even if all automatically generated labels are to be inspected visually, the bare labeling effort, i.e., the number of required clicks, is already reduced by 80.35 %. Considering that a fixed parameter set was used per dataset to simplify evaluation, the number of FP and FN could further be reduced by adjusting the parameters individually for a particular device. Moreover, as the labeling tool also identifies identical events, the additional workload of adding textual labels is reduced to changing the name of one event (e.g. *Compressor On*, *Door Open*, *Off* for the fridge).

Table 2: Evaluation results of the event detector for the FIRED data.

Appliance	Events	TP	FP	FN	F_1
baby heat lamp	6	6	0	0	100.00
fridge	574	559	16	15	97.30
coffee grinder	225	186	94	39	73.66
espresso machine	1095	1035	8	60	96.82
kettle	16	16	5	0	86.49
hairdryer	12	12	0	0	100.00
hifi system	28	28	42	0	57.14
television	47	39	2	8	88.64
kitchen spot light	4	4	0	0	100.00
oven	50	50	3	0	97.09
fume extractor	29	29	2	0	96.67
sum	2086	1964	172	122	93.04

The better results for the FIRED dataset indicate that higher data resolution and sampling rate are beneficial for event recognition methods. An additional evaluation could determine a sweet spot for both data parameters but is beyond the scope of this work.

Annoticity is made available to the public for everyone to discover electricity data or label datasets[1].

6 CONCLUSION

We presented Annoticity, a data browser and annotation tool designed for labeling electricity datasets. To decrease the effort of manual labeling we used an event detector which is able to recognize and pre-label power events in the data. Therewith the process of labeling data is changed from repetitively selecting annotating regions of data to reviewing proposals of the automatic event detection and skimming the time-series for missed events. We evaluated the event detector using a fixed parameter set across devices of two publicly available datasets. This evaluation has shown that events could be identified with an overall F_1 score of 90.07 %. Using this detector, Annoticity can pre-label a large amount of events automatically, hence significantly reducing the amount of human labeling. By reducing the effort required to label electricity datasets, researchers can focus on improving disaggregation algorithms, which are crucial to saving earth's energy resources. Future work should focus on improving the event detection algorithm, and to automatically devise the algorithm's parameter from statistical summaries of the data.

REFERENCES

[1] United States Environmental Protection Agency. 2014. Sources of Greenhouse Gas Emissions. https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions. Online; accessed 24-January-2018.

[2] Christian Beckel, Wilhelm Kleiminger, Romano Cicchetti, Thorsten Staake, and Silvia Santini. 2014. The ECO data set and the performance of non-intrusive load monitoring algorithms. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. ACM, 80–89.

[1] https://earth.informatik.uni-freiburg.de/annoticity

[3] Hông-^An Cao, Tri Kurniawan Wijaya, Karl Aberer, and Nuno Nunes. 2015. A collaborative framework for annotating energy datasets. In *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2716–2725.

[4] Karen Ehrhardt-Martinez, Kat A Donnelly, Skip Laitner, et al. 2010. Advanced metering initiatives and residential feedback programs: a meta-review for household electricity-saving opportunities. American Council for an Energy-Efficient Economy Washington, DC.

[5] Adrian Filip. 2011. BLUED: A fully labeled public dataset for event-based non-intrusive load monitoring research. In *2nd Workshop on Data Mining Applications in Sustainability (SustKDD)*. 2012.

[6] Django Software Foundation. 2020. DjangoThe web framework for perfectionists with deadlines. https://www.djangoproject.com/

[7] Jon Froehlich, Leah Findlater, and James Landay. 2010. The design of eco-feedback technology. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1999–2008.

[8] Jana Huchtkoetter, Andreas Reinhardt, and Sakif Hossain. 2019. ANNO: A Time Series Annotation Tool to Evaluate Event Detection Algorithms. In *International Workshop on Simulation Science*. Springer, 70–87.

[9] Jack Kelly and William Knottenbelt. 2015. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Scientific data* 2 (2015), 150007.

[10] J Zico Kolter and Matthew J Johnson. 2011. REDD: A public data set for energy disaggregation research. In *Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA*, Vol. 25. 59–62.

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

[12] Non-Profit Organization Matroska. 2020. The Matroska File Format. https://www.matroska.org/

[13] Lucas Pereira. 2017. Developing and evaluating a probabilistic event detector for non-intrusive load monitoring. In *2017 Sustainable Internet and ICT for Sustainability (SustainIT)*. IEEE, 1–10.

[14] Lucas Pereira and Nuno Nunes. 2020. Understanding the practical issues of deploying energy monitoring and eco-feedback technology in the wild: Lesson learned from three long-term deployments. *Energy Reports* 6 (2020), 94–106.

[15] Lucas Pereira and Nuno J Nunes. 2015. Semi-automatic labeling for public non-intrusive load monitoring datasets. In *Sustainable Internet and ICT for Sustainability (SustainIT), 2015*. IEEE, 1–4.

[16] Lucas Pereira, Miguel Ribeiro, and Nuno Nunes. 2017. Engineering and deploying a hardware and software platform to collect and label non-intrusive load monitoring datasets. In *2017 Sustainable Internet and ICT for Sustainability (SustainIT)*. IEEE, 1–9.

[17] Filipe Quintal, Lucas Pereira, Nuno J Nunes, and Valentina Nisi. 2015. What-a-Watt: exploring electricity production literacy through a long term eco-feedback study. In *2015 Sustainable Internet and ICT for Sustainability (SustainIT)*. IEEE, 1–6.

[18] Benjamin Völker, Philipp M Scholl, and Bernd Becker. 2019. Semi-Automatic Generation and Labeling of Training Data for Non-Intrusive Load Monitoring. In *Proceedings of the Tenth International Conference on Future Energy Systems* (Phoenix, USA) *(e-Energy '19)*. ACM.

[19] Benjamin Völker, Philipp M. Pfeifer, Marc amd Scholl, and Bernd Becker. 2020. FIRED: A Fully-labeled hIgh-fRequency Electricity Disaggregation Dataset. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys), Yokohama, Japan*, Vol. 7. ACM.

[20] Worldbank. 2014. Electricity Consumption Per Capita. https://data.worldbank.org/indicator/EG.USE.ELEC.KH.PC. Online; accessed 29-August-2020.