Power-Based Device Recognition for Occupancy Detection

Azkario Rizky Pratama[1,2], Widyawan[2], Alexander Lazovik[1], and Marco Aiello[1]

[1] Distributed Systems Group, Johann Bernoulli Institute
for Mathematics and Computer Science, University of Groningen, the Netherlands
{a.r.pratama,a.lazovik,m.aiello}@rug.nl
[2] Department of Electrical Engineering and Information Technology,
Universitas Gadjah Mada, Indonesia
widyawan@ugm.ac.id

Abstract. Each person using electrical devices leaves electricity fingerprints in the form of power consumption. These can be very useful for understanding the context of that person in, for instance, a smart office. A device that is highly correlated with the presence of a person in an office is the computer monitor; the correlation is in the range 83–96%. Therefore, it is useful to recognize from an aggregated power load the portion that is due to computer monitors. In this paper, we propose an event-based device recognition approach. After studying several predictors and features for device classification, we build a prototype for the classification. We evaluate the approach with actual power measurement of seven office monitors used by four workers in an office environment. Our experiments show that the approach is feasible and the per-day accuracy ranges in the range 69–80% for seven and five physical devices, respectively.

Keywords: ·Device recognition ·Load disaggregation ·Occupancy detection

1 Introduction

Smart buildings operate efficiently by being aware of their actual use and environmental conditions [1]. One of the biggest challenges to achieve smart buildings is the automatic classification of human activities and state within the building. Low intrusive approaches are generally preferred due to privacy and economic concerns [2]. The aggregated measurement of power consumption is an important feature to consider for context mining and human activity classification.

Human presence detection is one of the necessary components in a smart building system to provide a custom service, specific to present occupants. Some common examples are automated lighting and HVAC (heating, ventilation, and cooling) systems that automatically tune their operations on the users' occupancy [3]. These systems require the environment contexts (such as human presence) to be updated accordingly. Since the context of the building is highly dynamic, we need to keep the building systems up to date, in order to assure composed services achieve building operation objectives, such as providing occupant satisfaction and efficient energy consumption. Furthermore, for privacy and performance reasons, the processing of such context should happen at the edge of the network, before further sending selected data and knowledge to cloud infrastructures. Such a *fog computing* paradigm supports the demands of quick and efficient data processing while having connected service-based systems. In the present work, we do not focus on the service-oriented architecture and cloud components, as they are rather standard, while instead we focus on the IoT and Smart Building aspects. In particular, we investigate the feasibility of load disaggregation from a unique power consumption reading with the final goal of correctly classifying the human presence in an office. We collect large amounts of data using a global power meter/smart meter. These meters are increasingly installed by utilities, are relatively inexpensive, and do provide basic energy readings with reasonable sampling rates. Such an approach opens the possibility of recognizing personal occupancy using global room-level or department-level meters. In other terms, by recognizing particular devices associated to a particular person, we obtain the reduced-size, finer-grain occupancy information which further can be forwarded to the cloud for capturing the bigger picture about building occupancy. The chosen office devices for the present study are computer monitors as there is evidence that most of the time people are in offices they are engaged in computer related activities. E.g., in the US, workers spend on average of more than six hours per day at the computer and an additional hour at home [4]. A first indication that the computer use is closely related to office presence and work.

To learn which approach works best, what values to provide to our models, and to evaluate the performance of the approach, we experimented for two and a half months in our own offices at the University of Groningen. We collected power consumption data using global power meters. We deployed power meters in a single point measurement in incoming electrical line as well as per-appliance plugs to collect ground truth information and observe characteristics of every monitor screen. To make the approach more flexible and portable, we define synthetic aggregate data by applying superposition of several monitors that are owned by the same person in an office. The rationale for this choice is that no significant differences are found between the synthetic signals and the composite loads measurement on the electrical line. From each device, we extract and explore several features that possibly characterize turning on/off events. Such descriptors are used to train classification models to infer which are the active devices at any time. We develop a device recognition approach which is based on event detection. Events are triggered when potential switching occurs.

The contribution of the present work can be summarized as follows:

1. We propose a novel device recognition system (specifically, computer monitors) based on energy load disaggregation;

2. We compare and identify meaningful features to describe switching events of monitor instances recovered from a single electricity measurement (i.e. active power);
3. We evaluate experimentally the approach; and
4. We propose the concept of *virtualdevice* to detect multi appliances running simultaneously and improve the recognition performance.

The remainder of the paper is organized as follows. The overview of previous, related work is provided in Section 2. We describe the system's design and implementation in Section 3. Experimental setup and evaluation are provided in Section 4. In Section 5, experimental results are reported and discussed. Finally, conclusions are drawn in Section 6.

2 Related work

Diverse sensor types have been used to obtain occupancy context both in residential and commercial buildings. These include RFID, passive infrared PIRs, door sensors, GPS, WiFi, temperature, humidity, and other environmental sensors [3, 5, 6]. Binary occupancy detection, based on electricity consumption has also been proposed, e.g. [7] [8]. Lu *et al.* make use of historical occupancy and indication of human presence (through PIR and door sensors) to infer occupancy states in homes [3]. By using a Hidden Markov Model, they show that 88% occupancy accuracy can be achieved. In this work, we aim to address the more general case of multiple people detection, rather than individual home occupancy inference.

Load disaggregation, also referred to as device separation, deals with the identification of consumptions of individual devices from a global electricity consumption signal. The most common approach is based on recognizing appliance signatures. Liang *et al.* define two signature forms [9]. First, a signature can be recognized in *snapshot* form: an observation of load behavior at any fixed time intervals. Second, the signature can be formed as *delta* form: taking parameter changes into account when a state transition occurs.

In [10], the authors observe appliance switching events to learn characteristic of several devices in an office, such as monitors and printers. They observe and describe the behavior of several type of monitors, without trying to supervise models and classify the fresh data. Low power appliances recognition using 120 changing states of appliances is presented in [11]. The performance shown is 90% and 76% for individual and multiple appliance recognition, respectively. To achieve these results feature-rich, high-resolution power meters were employed. Due to National regulations on smart meters [12], to keep the study realistic with standard installations, in the present study we decide to utilize generic power meters with only active power measurement capability.

An effort to classify personal occupancy in the office was developed by [13]. The authors deploy one power measurement on each work desk and classify whether the respective occupant is present, away, or in standby. By using two weeks worth of data, they show 93% accuracy with a KNN-based classification method.

3 Design and Implementation

Smart and energy-aware buildings [1], rely on a number of components, such as a context inference and repository component, an AI Planning and Scheduling one, and an orchestrator one [14]. Figure 1 shows an architecture derived from our previous work. The present contribution aims at offering improved context information which is in turn essential to determine the current state of the building. The context is inferred on-site to enable efficient data processing and reduce the amount of data to be transported to the cloud. The current state of the building, possibly with the bigger picture of how the persons occupy the building (e.g. processed in the cloud), affect the plan composing. The plans then go to the orchestrator which is responsible for the actuation and for evaluating possible failures or execution deviations, in turn affecting the context again.

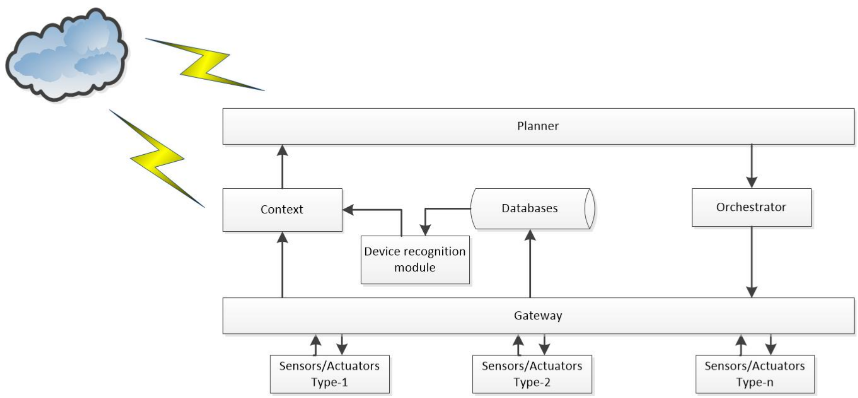

Fig. 1. Software architecture of a smart building, derived from [14].

The context is a point in a possibly infinite feature space of relevant building context variables. The move from one point to another one is defined as an *event*. In the specific case of load disaggregation, an event is a significant peak or slope occurring in power consumption waveforms. These are typically associated with a device (electricity load) switch going from ON to OFF, or vice-versa. We develop a mechanism to detect candidate events using thresholds and validate them according to empirical data (i.e., 10 Watt difference, 60 seconds between two consecutive events). These values are derived from previous experimentation. In particular, the watt-difference parameter is based on the study of monitors of different brands and types with the lowest power consumption for them positioned at 12 Watt. The time interval parameter is based on the fact that people typically work in burst higher than one minute. The precise processing is presented as Algorithm 1.

For each combination of validated events, we extract the relevant features. Inspired by the field of dynamic systems [15] and statistics [8], we consider: rise-

Algorithm 1 Event detection from an aggregated data

procedure EVENT DETECTION AND EVENT VALIDATION
    $X \leftarrow aggregateddata$
    set window in moving windows
get events:
    compute *range* in a *window*
    if $range > wattThreshold$ **then**
        if $durationBetweenEvent > durationThreshold$ **then**
            $event \leftarrow window$
get delta power:
    compute *mean* after and before an event
    $\Delta P \leftarrow (mean_{after} - mean_{before})$
validate events:
    if $\Delta P > wattThreshold$ **then**
        $validatedEvent \leftarrow event$

time, overshot, steady level, variable variance, and mean of absolute difference, described as follows.

Delta P **(**ΔP**)** is the difference of average power before a detected event and average power after the event. We consider five samples for both before and after events; illustrated as black arrows in Figure 2.

Steady level is the value of a device (or set of devices) in a stable state; represented as dashed line in Figure 2.

Rise time is the time needed for a transition from 10% to 90% of the reference levels; represented as a grey shaded rectangle in Figure 2.

Overshoot is the percentage of the difference between state levels. It is defined as Eq. 1, where y_{max} is the maximum value (indicated by downward-pointing triangle in Figure 2), $level(s_k)$ is the steady state level, and $|A|$ is the amplitude [16].

$$Overshoot = \frac{(y_{max} - level(s_k))}{|A|} 100\% \tag{1}$$

Mean of Absolute Difference (MAD) captures the ripples during a device's active period, Eq. 2 [8].

$$MAD = \frac{1}{N-1} \sum_{i=2}^{N} |y_i - y_{(i-1)}| \tag{2}$$

Variance measures how far a set of values are spread out from the steady level, i.e.,:

$$var = \frac{1}{N-1} \sum_{i=1}^{N} |y_i - \bar{y}|^2 \tag{3}$$

Features such as Power Level, MAD, and Variance satisfy the criterion of so-called additive features [9], therefore it is possible to compute the delta value of an event; see Algorithm 2.

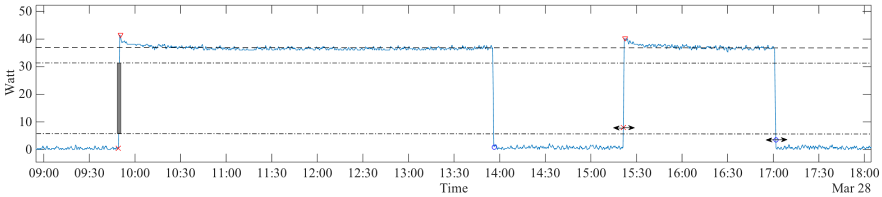

Fig. 2. Example of a day worth of feature values

Algorithm 2 Feature extraction from aggregated data

procedure FEATURE EXTRACTION
get features between validatedEvents:
  for all combination *validatedEvents* **do**
    compute {*RiseTime; Overshoot; Level; Peak; MAD; Variance*};
get delta features
  $\Delta_{features} \leftarrow \Big(Level_t; MAD_t; Var_t\Big)_{t-1}^{t};$

The extracted features contribute to the classification of the event and new context state. Several classification methods are possible:

k-Nearest Neighbor is one of the simplest learning techniques that works by finding the predefined number of labeled samples nearest to a query and predict the class label with the highest votes [17].

Naive Bayesian is a simple probabilistic classifier that assumes features are independent given a class label [18]. We choose this technique with an assumption that each feature contributes independently to the probability of class labels, regardless of any correlations between the features.

Neural network is a nonlinear statistical model for regression or classification, typically represented by a network diagram [19]. It works by deriving hidden features Z from linear combination of the inputs X and then modeling the target classification Y as a function of linear combination of the Z.

Due to the flexibility of the input features and the easy extensibility of the network structure, we choose single layer Neural networks and extend to multiple layers, as illustrated in Figure 3.

We further define *virtualdevice* as a combination of two or more physical devices belonging to a specific person. *virtualdevice* are useful when the composing devices change their state concurrently. In the present setup, *virtualdevice* is denoted by device index number 24, 25, and 26.

4 Evaluation

To evaluate the approach, we experiment in our own offices located in Groningen on the fifth floor of the Bernoulli building on the Zernike Campus of the

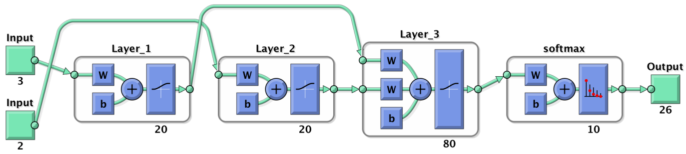

Fig. 3. Multiple hidden layer, with two inputs and three hidden layers.

University of Groningen. The experiment took place from the 13th to the 31st of March 2017 and from the 17th April to the 22nd June 2017.

Setup We consider two office rooms occupied by four people (PhD students). To collect the ground truth, we equip all electric devices of the rooms with Plugwise Circle, the single power consumption sensors from Plugwise[3]. Each Circle utilizes the wireless ZigBee protocol. We use Raspberry Pi[4] to pool the data from the plugs and forward them to a server for processing.

We sample data at 10 second intervals to assure there is enough time for the pooler to receive data from all plugs. Furthermore, this value is set to comply with the Dutch National regulation on smart meters [12]. If due to some failure, we miss a reading, we keep the previous valid one. This approach is common to mimic the constant consumption of simple devices, such as LCD monitors [13]. We then analyze the recorded data to attest the system performance. From each individual power load, we extract features of events for teaching learning models and construct ground truths from known switching events. We supply two weeks data to train models and use two months fresh data to examine the classification performance of the models. The details of number training and testing set is summarized in Table 1. From the table there is an indication that the number of training instances depends on the considered training labels. It can also be seen that the number of test data relies on target devices. The more target devices are included, the more frequent occupant presence should be detected. Thus the number of available test instances is also increased.

Table 1. Summary of number of training and testing set

No	#instances		#traininglabels	target devices
	Training	Test		
1	252	78 (10days)	8	[4;10]
2	252	160 (27 days)	8	[4;7;10]
3	252	188 (27 days)	8	[4;5;7;10]
4	252	298 (31 days)	8	[4;5;7;10;14]
5	252	274 (31 days)	8	[4;5;7;10;14;(24)]
6	317	241 (31 days)	10	[4;5;7;8;10;11;14;(24;25;26)]

[3] https://www.plugwise.com

[4] https://www.raspberrypi.org/

The actual presence of people to populate the ground truth is taken manually, based on paper based diary and human observation.

Metrics: Event detection To evaluate the experiment, we measure how accurate the proposed approach is in classification. We resort to standard metrics, such as *Precision*, the rate of True Positive over all events detected by system regardless the truth, and *Sensitivity*, the proportion of real events that are correctly identified.

Metrics: Device classification The precision of classification is defined on the basis of the actual beloning of devices identified by the k *most-probable* classe to the correct class. We use the average of how many classes are correctly inferred, Eq. 4, using k = 2 with an constant weight.

$$Accuracyperday = \frac{1}{n_{events}} \sum_{i=1}^{n_{events}} (y = \hat{y_1} \cap y = \hat{y_2}) \quad (4)$$

The average, overall accuracy, can then be computed as Eq. 5:

$$Accuracyaverage = \frac{1}{d_{days}} \sum_{d=1}^{d_{days}} accuracyday_d \quad (5)$$

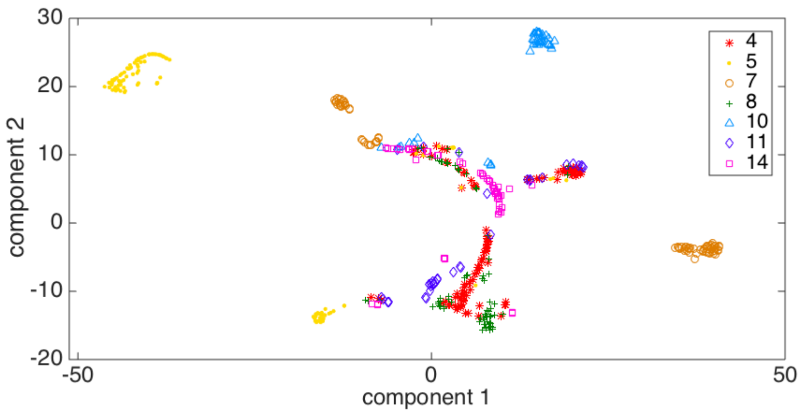

Fig. 4. Visualization of seven device classes.

5 Results and Discussion

Following Algorithm 1), we perform device detection on the acquired data set. The result is shown in Table 2. The devices mentioned in the table are monitor screens (i.e., monitor 4 and 5; 7; 10; and 14 belong to Worker W1, W2, W3, and W4, respectively) and a *virtualdevice* (i.e., device 24 which is a combination of device 4 and 5). The number of days shows how many days these devices are used or activated during observation. The best performance of event detection is when the system only detects two devices. The precision and sensitivity reaches 87.9% and 97.3%, respectively. As the number of involved devices increases, the performance declines, reaching 70% precision and 89% sensitivity. A significant drop occurs when device 14 is added to the aggregated power, while adding other devices gradually changes the performance by just 1%. Device 14 worsen the overall event detection. The reason for this is in the short interval transitions that occur in the dataset for this device (i.e., 2 consequent transitions, ON-OFF-ON, in less than 5 minutes), thus resulting in possible undetected events.

Table 2. Device events detection.

No	#days	Devices	Precision	Sensitivity
1	10	[4;10]	0.879	0.97333
2	27	[4;7;10]	0.87361	0.94193
3	31	[4;7;10;14]	0.71002	0.92854
4	31	[4;5;7;10;14]	0.71575	0.85877
5	31	[4;5;7;10;14;(24)]	**0.70157**	**0.8912**

Events-extracted features can be visualized using t-Distributed Stochastic Neighbor Embedding, Figure 4 [20]. The features taken into account for the analysis are ΔP, steady level, rise time, overshoot, mean of absolute difference, and variance. One can be observe the challenge of device classification; in fact, classes are not easily separable.

We compare several feature sets and three different techniques to classify particular devices. We also observe the significance of number of recognized devices, starting from two (devices 4 and 10) and up to six devices (physical devices 4,5,7,10,14, and virtual device 24). The comparisons are summarized in Figure 5.

Feature set 1 considers the difference in power before and after an event (ΔP). The accuracy is 32% and 45% using NeuralNet and NB techniques, respectively. For these methods, the performance seems not to be affected by the number of devices. By using the same feature, the accuracy with kNN reaches 50%. However, increasing the number of appliances does result in considerably decreasing performance using kNN.

The rise time and overshoot in Feature set 2 also give fluctuations in terms of per-day accuracy, depending on the number of devices to be classified, i.e., 15-21%; 10-34%; and 10-60% for NeuralNet, NB, kNN, respectively. This feature set brings the lowest performance compared to the other sets. It is also shown that the performance of kNN and NB method depends on the number of device. The higher number of devices considered, the less performance can be achieved.

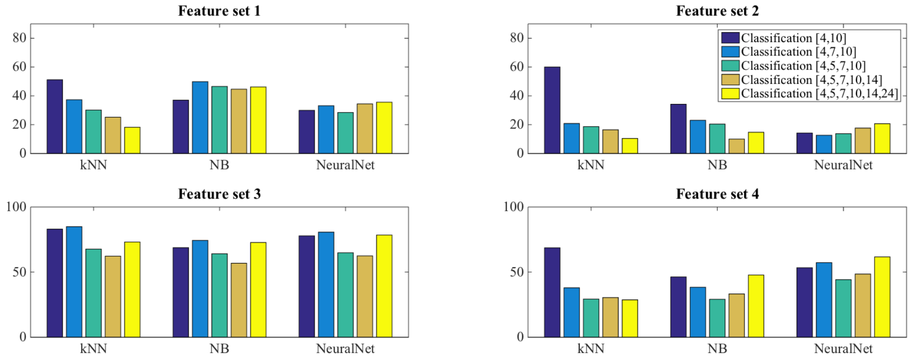

Fig. 5. Comparison of feature sets.

With respect to NeuralNet, it shows stable results below 21% accuracy with this feature set.

The combination of steady-level, MAD, and Variance in Feature set 3 delivers the highest performance among the other feature combinations, up to 81%, 74%, and 84% for the NeuralNet, NB, and kNN, respectively. It is worth noting that these results would degrade as the number of classification device increases, reaching 62% for both NeuralNet and kNN, and 56% for NB. Our proposed concept of $virtualdevice$ can improve the performance by 16%, 16%, and 11% for NeuralNet, NB, and kNN, respectively, by introducing $virtualdevice$ to the classification models. Such improvements are presented as an upward trend from the 4^{th}- to 5^{th}-bar for three models in Feature set 3, in Figure 5.

Feature set 4—a combination of Feature set 2 and 3—does not deliver a better result than the others. However, the average performance of NN outperforms NB and kNN in recognizing of 6 devices by 14% and 33%, respectively. This result is in accordance with the result of experiment with feature set 3 where the NeuralNet slightly outperforms the others.

In addition to physical devices, it is useful to also consider virtual ones, resulting from the combination of measurements from physical ones. In the evaluation, let us consider seven physical devices owned by four people (three have multiple screens). By introducing three virtual devices (i.e., 24, 25, 26), we classify these devices with Feature set 4 in a modified network structure, as shown in Figure 3. The device recognition result is shown in Figure 6.

Device set 1 that consists of five physical devices can be recognized correctly with 63.81% accuracy per-day. By adding one virtual device that represents two devices activated almost simultaneously, 80.1% accuracy per-day is achieved. The accuracy of recognizing device set 2 with seven physical devices drops significantly to 28.65% accuracy per-day. By introducing three virtual devices represents six devices, the performance improves, reaching 69.13% accuracy per-day.

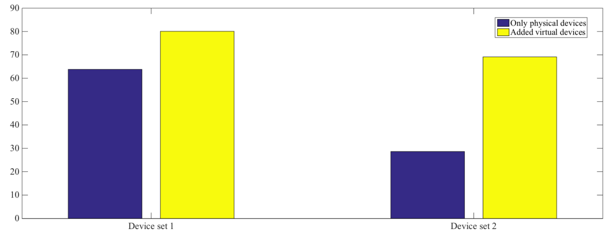

Fig. 6. Simultaneous devices classification

5.1 Relation of monitor screen activation and occupancy

The observation of occupancy during working hours (set from 8 am to 9 pm, due to the different working times of individuals) is shown in Table 3. The monitors reveal the accurate occupancy of people up to 96.8%. It is lower, about 83.5%, for the person who is present at the office for 4 days of a week observation. The reason is that the monitor needs to wait its timeout to automatically put on sleep mode after plugged out from the laptop/sources. This does not happen to worker W1 and W2 due to different hardware specifications.

Table 3. Occupancy accuracy over a 5-minute interval

Worker	Presence days	Accuracy
W1	7d	96.7949
W2	5d	89.8718
W5	4d	83.4936

5.2 Discussion

Based on the evaluation in an actual office space, we conclude that the proposed event detection approach has very promising performance. It achieves 90% sensitivity with a 70% precision. In other words, the system is good at the detection of the actual events, yet of all inferred events, some are misread. In fact, the system misinterprets oscillations on the waveform as switching events. This happens as the considered devices have a low-power consumption, making harder to discern the events from common, regular fluctuations.

The Feature set 2 (rise time and overshoot) are not describing the devices very well. The reason could be in the time required for the positive-going transition not being captured by the 10 seconds data sampling. On the contrary, Feature set 3 shows the best performance among the others. This set is applicable to the three methods with comparable results, up to 84% accuracy per-day. The combination of Feature set 2 and 3 does not contribute to improving the performance of kNN, NB, and NeuralNet. However, with the same features, the classification works better in multiple hidden layers network (Figure 3) than in single hidden layer network.

The classification performance of kNN suffers from the dependencies of the number of devices. It can be moderately dropped as a number of considered devices increase. Conversely, the performance of NeuralNet and NB is more robust to the number of involved devices. This is because kNN works by finding the nearest labeled sample. As the sample of training set during 2 weeks does not cover very well for all devices, the results of kNN become worse.

Based on our empirical observation during a week, the personal occupancy classification shows acceptable performance in relation to the monitor activation. However, some factors might affect the relation, such as whether the person is working using electrical devices, a personal habit to consistently deactivate the device while being away, and hardware configuration (auto sleep mode).

6 Concluding remarks

Even with simple aggregated power consumption, it is possible to recognize device usage and turn that information into building-user context knowledge. We have proposed an approach based on neural networks and evaluated over ten days in an actual office. We used various power features, such as ΔP, steady power level, rise time, overshoot, MAD, and variance.

The experimental evaluation shows that it is possible to recognize low-power devices from composite energy loads, achieving 84%, 81%, and 74% accuracy per-day using kNN, NeuralNet, and NB, respectively. The kNN performance will show a downward trend as the number of devices increased, while NeuralNet and NB seem more robust in the addition a number of devices. We notice that steady-level, MAD, and Variance features give a good description to the classifiers while adding rise time and overshoot features not always give a positive impact. It is also validated that the proposed *virtualdevice* can improve the performance by 40% in the recognition of 10 classes (7 physical devices and 3 virtual devices), reaching 69.13%.

7 Acknowledgement

Azkario Rizky Pratama is supported by the Indonesia Endowment Fund for Education (LPDP). This research has been partially sponsored by the EU H2020 FIRST project, Grant No. 734599, FIRST: vF Interoperation suppoRting buSiness innovaTion.

References

1. T. A. Nguyen and M. Aiello, “Energy intelligent buildings based on user activity: A survey,” *Energy and Buildings*, vol. 56, pp. 244 – 257, 2013.
2. T. Nguyen and M. Aiello, *Beyond indoor presence monitoring with simple sensors*, pp. 5–14. 2012.

3. J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse, "The smart thermostat: Using occupancy sensors to save energy in homes," in *Proc. of the 8th ACM Conf. on Embedded Networked Sensor Systems*, SenSys '10, pp. 211–224, 2010.
4. "Microsoft: Us workers spend 7 hours on the computer a day on average." `https://www.onmsft.com/news/microsoft-us-workers-spend-7-hours-computer-day-average`, 2013. (Accessed on 26/09/2017).
5. J. Scott, A. Bernheim Brush, J. Krumm, B. Meyers, M. Hazas, S. Hodges, and N. Villar, "Preheat: Controlling home heating using occupancy prediction," in *Proc. of the 13th Int. Conf. on Ubiquitous Comp.*, UbiComp, pp. 281–290, 2011.
6. C. Koehler, B. D. Ziebart, J. Mankoff, and A. K. Dey, "Therml: Occupancy prediction for thermostat control," in *Proc. of the 2013 ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing*, UbiComp, pp. 103–112, 2013.
7. S. Mamidi, Y.-H. Chang, and R. Maheswaran, "Improving building energy efficiency with a network of sensing, learning and prediction agents," in *Proc. of the 11th Int. Conf. on Autonomous Agents and Multiagent Systems - Vol. 1*, AAMAS '12, pp. 45–52, 2012.
8. M. Jin, R. Jia, and C. Spanos, "Virtual occupancy sensing: Using smart meters to indicate your presence," *IEEE Trans. on Mobile Computing*, pp. 1–1, 2017.
9. J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng, "Load signature study x2014;part i: Basic concept, structure, and methodology," *IEEE Trans. on Power Delivery*, vol. 25, pp. 551–560, April 2010.
10. B. Kalluri, S. Kondepudi, K. H. Wei, T. K. Wai, and A. Kamilaris, "Opld: Towards improved non-intrusive office plug load disaggregation," in *2015 IEEE Int. Conf. on Building Efficiency and Sustainable Technologies*, pp. 56–61, 2015.
11. A. Zoha, A. Gluhak, M. Nati, and M. A. Imran, "Low-power appliance monitoring using factorial hidden markov models," in *2013 IEEE Eighth Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 527–532, 2013.
12. "P1 companion standard." `http://www.netbeheernederland.nl/_upload/Files/Slimme_meter_15_32ffe3cc38.pdf`, March 2014. (Accessed on 02/08/2017).
13. A. Akbar, M. Nati, F. Carrez, and K. Moessner, "Contextual occupancy detection for smart office by pattern recognition of electricity consumption data," in *2015 IEEE Int. Conf. on Communications (ICC)*, pp. 561–566, 2015.
14. I. Georgievski, T. A. Nguyen, F. Nizamic, B. Setz, A. Lazovik, and M. Aiello, "Planning meets activity recognition: Service coordination for intelligent buildings," *Pervasive and Mobile Computing*, vol. 38, pp. 110 – 139, 2017.
15. G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Pearson, 7th ed.
16. "Ieee standard for transitions, pulses, and related waveforms - redline," *IEEE Std 181-2011 (Revision of IEEE Std 181-2003) - Redline*, pp. 1–71, 2011.
17. G. Shakhnarovich, T. Darrell, and P. Indyk, *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing)*. The MIT Press, 2006.
18. J. F. Trevor Hastie, Robert Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd ed., 2009.
19. H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural Network Design*. USA: Martin Hagan, 2nd ed., 2014.
20. L. van der Maaten, G. Hinton, and Y. Bengio, "Visualizing data using t-sne," in *Journal of Machine Learning Research 9*, 2008.