A Knowledge-Based Proactive Intelligent System for Buildings Occupancy Monitoring

Marie Unmack Baerentzen[1], Jalil Boudjadar[1][a], Saif Ul Islam[2][b] and Carl Peter Leslie Schultz[1][c]

[1]*Aarhus University, Denmark*

[2]*Institute of Space Technology, Pakistan*

Keywords: Domain-Specific Language, Occupancy Analysis, Energy Efficiency, Optimization.

Abstract: Occupancy monitoring for buildings is a key component to enable cost-effective allocation of spaces and efficient resources utilization. The occupancy monitoring systems rely on networks of sensors and cameras to achieve high accuracy, however the main challenges are the privacy concerns and the computation cost. This paper proposes the design of an intelligent energy-efficient and privacy-aware system to track, monitor and analyze buildings occupancy. The core idea is that rather than collecting large amounts of sensor data to perform occupancy analysis post hoc, our proposal adopts a top-down approach where, using the knowledge about the activity expected to be taking place it proactively identifies the minimal data relevant to the actual state following the semantics of the expected activity. Thus, it switches on/off the sensors in accordance with such a subsequent dynamics and reduces the data amount to collect and the computation cost. The proposed system has been built using a domain-specific language, implemented in C++ and tested for a building case study. Our experimental results show that, while achieving a considerable reduction in computation cost (up to 35%) and energy consumption (up to 31%), our system maintains high accuracy for occupancy tracking compared to the state of the art solutions.

1 INTRODUCTION

Building occupancy monitoring enables to track the state of the different spaces in real-time (Azimi and O'Brien, 2022; Elkhoukhi et al., 2018; Pradeep Kumar, 2016; Perra et al., 2021). The primary goal of such systems is to provide a decision support to buildings management to achieve a better spaces allocation and resources-efficient operation (Salimi and Hammad, 2019; M et al., 2021). Particularly with the ever-increasing prices of electricity, the need to optimize energy efficiency is of capital interest (Lasla et al., 2019).

Occupancy tracking and analysis technology has the potential to transform buildings efficiency by automatically monitoring and identifying when and where resources utilization can be minimized (Elkhoukhi et al., 2018). However, current approaches for occupancy monitoring are driven by Big Data Analysis so that to collect large quantities of sensor data (video, audio, step panels, etc.) and employ predefined classifiers to detect patterns in the data (Zhang et al., 2019) (Jiang and Yin, 2015). Not only this analysis is slow, far to be real-time (Elkhoukhi et al., 2018), it has relatively low accuracy (Salimi and Hammad, 2020; Pan et al., 2014). Moreover, such comprehensive, brute-force data collection processes consume a high amount of energy and may raise privacy concerns, particularly in light of the new EU General Data Protection Regulations (GDPR) [1].

The primary aim of this paper is to design and implement a prototype for building occupancy monitoring and analysis using blind sensors, with the main goal being able to intelligently and proactively control system sensors. So that to actuate the power consumers in the building spaces, such as light and air conditioning, following the actual and predicted states.

Our key contribution is, rather than continuously (and blindly) monitoring and processing all available sensor data post-hoc to detect critical events (purely bottom-up, highly computationally inefficient), our approach is *knowledge-driven*: identify which high-level symbolic facts are needed to change the current state into a critical event, and trace these facts back through a well defined semantics of the adopted

[a] https://orcid.org/0000-0003-1442-4907

[b] https://orcid.org/0000-0002-9546-4195

[c] https://orcid.org/0000-0001-7334-6617

[1] https://eugdpr.org

Baerentzen, M., Boudjadar, J., Islam, S. and Schultz, C.
A Knowledge-Based Proactive Intelligent System for Buildings Occupancy Monitoring.
DOI: 10.5220/0012144000003538
In *Proceedings of the 18th International Conference on Software Technologies (ICSOFT 2023)*, pages 680-687
ISBN: 978-989-758-665-1; ISSN: 2184-2833

domain-specific language to identify which minimal combination of sensors can supply data to confirm or contradict these facts (top-down, then bottom-up). The start point of each iteration of this exploration is the event expected at that time point.

This intelligent sliding of the sensors operation stems from the problem that reactive systems, which gather and process massive amounts of data, drain a lot of power from the sensors. This challenge is much severe especially when using battery powered sensors.

The system we consider monitors a public building for academic and education activities. It has various activities such as *Lecture*; *Lesson*; *Seminar*; *Meeting*; *Empty*, which all have recognizable features, each of which is identifiable through a subset of the sensors with different confidence levels. The proactivation process consists in sliding the sensors frequency, including switch On/Off, to optimize data sampling in an intelligent way following the actual state and the semantics of the expected/identified activities. To automate the data analysis and the proactivation process, we defined a domain specific language formed by a set of events that have well-defined semantics. A prototype of the proposed system has been implement and tested against the state of the art (static frequency) occupancy monitoring systems in terms of computation cost and energy consumption.

Finally, Section 7 concludes the paper.

2 BACKGROUND

This section presents the background related to occupancy analysis and optimization.

2.1 Occupancy Analysis

Occupancy analysis is the process of identifying the occupancy state of a given building space, at different time points, by recognizing human presence and the activities conducted in such a building space (Ahmad et al., 2021). It relies on collecting data from a sensor network, or alternative technology, which usually captures the impact made by human presence through different features such as breathing, motion and noise.

This knowledge is mainly used to figure out how many people are in a building space and subsequently what is most likely ongoing activity at any time point. To leverage the benefit of using occupancy monitoring, the occupancy analysis needs to be real-time so that proper actions can be taken, either by a building manager or automated system, with respect to minimizing resources waste and improving the indoor comfort (Seghezzi et al., 2021).

Given that such systems are recently deployed on local micro-controllers (edge or embedded devices) using wireless battery-powered sensors, computation cost and energy consumption are two key performance metrics to optimize. One way of conducting such an optimization action is by reducing the data sampling and the underlying analysis, but this approach can have drastic impact on the outcomes accuracy and reliability.

To back up the occupancy analysis, usually a model is required (Yang et al., 2016). In fact, the model contains information about known data, activities and potentially the underlying semantics of such activities. Real-time data is then run through the model to identify the most likely ongoing activity. Occupancy analysis models are mostly domain-specific due to the given building functionality, sensors used, and building layout (Jiang et al., 2022). However, there are occupancy monitoring models that use extensively trained machine learning algorithms (Zhang et al., 2022) before being put into actual use. Either cases, the overall goal of occupancy analysis is to track how building spaces are utilized, and also to optimize the use of energy consumption and rooms allocation (Sun et al., 2020).

2.2 Optimization

Optimizing a system consists in minimizing and/or maximizing different performance metrics such as response time, resources utilization and energy consumption (Minoli et al., 2017). While improving some of the performance indicators, the optimization must not lead to deteriorating other metrics considerably neither impacting the system functionality (Boudjadar and Khooban, 2020), (Boudjadar and Tomko, 2022).

Building spaces are usually multi-functionality where a given space can be allocated to run different activities. Thus tailoring the building space towards the actual activities is of capital interest in the optimization of buildings functionality and resources. For occupancy analysis, this usually comes in the form of reducing the power spent on measurements and calculations while still maintaining low computation cost and high accuracy outcomes. The key driver for such an optimization is the real-time knowledge and estimation of the occupancy state so that a sliding of the sensors frequency (fewer data measurements, longer durations in between and even possibly turned off) can applied following the need of data to confirm/deny/investigate such states (Rault et al., 2014).

A trade-off between the accuracy and the sliding of the sensors functionality needs to be established and maintained through an objective function. Accordingly, building equipments (lights, ceiling fans, air conditioning, etc) can be tuned following the actual occupancy state.

3 RELATED WORK

Different occupancy monitoring and analysis models have been proposed in the literature (Luo et al., 2017; Rai et al., 2015; McKenna et al., 2015; Ahmad et al., 2021; Perra et al., 2021; Ortiz Perez et al., 2018; Abraham and Li, 2014). Mainly, the state of the art approaches differ in terms of the technology used to track the occupancy, analysis techniques and the features used in the optimization (accuracy, energy, sensors functionality, response time).

The authors of (Luo et al., 2017; Perra et al., 2021) proposed an agent-based occupancy analysis, that amounts at tracking the individual occupants rather than measuring the occupants impact. This occupancy monitoring alternative often models each occupant individually to allow capturing where occupants are expected to be at any time point. Although this approach can have high accurate estimation and confident occupancy predictability, it suffers from the scalability and the computation cost given that the system size is dependent on the total number of occupants.

Sanish *et al.* (Rai et al., 2015) combined the agent-based occupancy analysis with a graph-based model. This enabled to abstract away the detailed occupant behavior and interactions such as position tracking. Although this abstraction reduced the occupancy model complexity, scalability remains a challenge in a similar way to (Luo et al., 2017).

A state-based occupancy model has been proposed in (McKenna et al., 2015). It consists of creating stochastic data in terms of how probable a state will occur. This process enables to infer the location of people and their activities. Although the approach enables fine grained modeling and estimation of the occupancy state, the bottleneck can be related to the massive data gathering which drains sensor batteries and computation resources.

Knowledge-based occupancy analysis (Ahmad et al., 2021; Perra et al., 2021) amounts at designing a domain-specific model encoding the activities to be carried out in a given building space. By running actual data through the model, a potential activity is identified, thus either confirming or denying the expected schedule. The only concern is being domain specific, it requests expert knowledge to replicate the occupancy analysis for different applications.

We align with the work in (Ahmad et al., 2021) where we track occupancy by monitoring the impact occupants make in a given building space. Moreover, our events definition patterns are inspired from (Perra et al., 2021). However, our proposed system enables to identify the minimal data to look for, following the real-time state, in order to recognize the ongoing activity. This has led to reduce the operation cost, in terms of data gathering, computation cost and energy consumption.

Compared to the state of the art, our experimental results show that the proposed proactivation system outperforms existing occupancy monitoring techniques, where sensors are permanently active (Ortiz Perez et al., 2018; Abraham and Li, 2014), by reducing drastically the data and energy requirements to track occupancy. Moreover, it can achieve the same knowledge level as (Perra et al., 2021; Ahmad et al., 2021), where sensors operate with static sampling time intervals, but with 30% less data gathering. This results in less energy consumption for both data gathering and data processing.

4 OCCUPANCY MONITORING

This section describes the model of our occupancy monitoring system. The building spaces we consider consist of closed/structured rooms that can serve as offices, classrooms, meeting rooms, etc. Each individual space is monitored through a set of CO2 sensors, motion sensors, noise sensors and light sensors. The sensors are grouped into different hubs, distributed in the room space to monitor, each of which contains a sensor from each category. The light sensors measure data in *lux*, CO2 in parts per million (*ppm*), audio in decibels (*dB*), and motion in four levels: none (0), low (1), medium (2), and active (3). Moreover, the sensors can be configured on-the-fly to operate different frequencies.

To monitor and analyze the occupancy we define a domain-specific language. This includes the grammar and how to handle different levels of confidence to match the events the system can recognize. The language describes all potential events and features in a semantic way, thus enabling a thorough formal analysis of the occupancy state in real-time.

The proposed domain-specific language to model and track the occupancy state is defined as follows:

$$
\begin{aligned}
\mathcal{L} &\triangleq \quad \mathtt{E}1 \mid \mathtt{E}2 \mid \ldots \mid \mathtt{E}n \\
\mathtt{E} &\triangleq \quad (\mathtt{F}, c) \mid (\mathtt{F}, c) \wedge \mathtt{E} \\
\mathtt{F} &\triangleq \quad S \mid S \wedge \mathtt{F} \\
S &\triangleq \quad V \in [a, b] \\
c &\triangleq \quad H \mid M \mid L
\end{aligned}
$$

Where $\mathtt{E}$ is an event, $\mathtt{F}$ is a feature, S is a sensor, V is a sensor value that must be within a range $[a,b]$, and c is the confidence level of a feature to identify an event and can have values high (H), medium (M) or low (L). $\mathtt{E}$ and $\mathtt{F}$ are non-terminal elements of the grammar, which means they can create different constructs recursively. This is in fact practical as different events, respectively features, can have different numbers of features, respectively different sensors associated.

The sensor inputs are combined in different ways with different expected value ranges to identify the various features of the system. The features in turn, together with different confidence levels, are combined to define the system events.

Namely, the important features of our language, derived from observations on the case study, are the following:

- F1: 1 or 2 people in front,
- F2: Lecturing (consistent noise in front),
- F3: No lecturing,
- F4: Smartboard on,
- F5: No movement,
- F6: Smartboard off,
- F7: Many occupants present,
- F8: Few occupants present,
- F9: No occupant present,
- F10: Occupants quiet,
- F11: Occupants walking and talking,
- F12: Many people talking,
- F13: People talking and taking notes,
- F14: No or little noise,
- F15: Teacher sitting,
- F16: consistent sound from the back,
- F17: Front lights on,
- F18: Back lights off/dimmed,
- F19: No lights on.

A feature can be recognized using different sensors and for different values. As an example, $\mathtt{F}1 \triangleq S2 \in [410, 700ppm] \wedge S3 \in [2,3]$. One can see that some features, such as F11 and F19, may need specific interpretation in order to be specified and captured in the grammar. As an example, F11 feature can be recognized if people are walking, i.e. having a high value for the motion sensor and high value of the noise sensor. To improve the creditability of the events recognition, we consider predicates rather than single-point values, where an occupant is walking for example if the motion measure is within 70 to 100% of the maximum sensor value. However, for other features such F11, a feature is recognized only if the underlying sensor measurement is the highest value of the sample range. Each event of the system is then made up of a number of features, each with a confidence level c.

The system we define can specify a large set of events, recognize different features and incorporate a large number of sensors. However, for the sake of illustration in this work, we consider only 8 sensors grouped into 2 hubs each located in one end of the rooms to monitor. Moreover, we limit the set of events to analyze the five following events present in the use case: E1 is a lecture, E2 is a break, E3 is a meeting, E4 is a lesson (exercise session), and E5 refers to an empty building space. As an example, we specify a lecture event to be:

$$
\begin{aligned}
\mathtt{E}1 \triangleq \quad & < \mathtt{F}1, H > \wedge < \mathtt{F}2, H > \wedge < \mathtt{F}4, M > \wedge \\
& < \mathtt{F}7, M > \wedge < \mathtt{F}10, L > \wedge < \mathtt{F}18, L >
\end{aligned}
$$

We may need to state that the definition of the five events is not standard, however it is inspired from (Perra et al., 2021) and formalized based on the observations made during data gathering of the use case. The semantics of an event is then given by its set of sensors, inferred from the underlying features, together with confidence levels and measurement intervals that are required to recognize and confirm the occurrence of that event. The confidence levels of an event dictate the importance of each feature for the event, which is important for optimizing the sensor frequencies, for example sensors of features having low confidence for a given event e can be discarded while e is being confirmed through high confidence features. The calculation and optimization of sensors frequency will be defined in the next section.

A state of the system s is simply made up of a value (v(S_i)) from each of the eight sensors. Although sensors data is time stamped to ensure consistency, to simplify the state definition we omit to use time as part of the state, s(t). Thus s refers to the actual state, i.e. having the latest sensor readings, can simply be made as follows:

$$
\mathtt{s} = \langle v(S_1), v(S_2), \ldots, v(S_8) \rangle
$$

We overloaded V to be a valuation function that returns the value of each sensor.

5 PROACTIVATION: OPTIMIZED OCCUPANCY ANALYSIS

This section elaborates further on how the occupancy analysis in made together with how the sensors are planned on-the-fly and how the frequency sliding is computed following the expected event and the actual state.

To trigger the occupancy analysis, one may need to have an expected event ε. The expected event can be provided by the space schedule from the building management. From that event, our analysis process identifies the sensors to be used to capture ε. This is done according to the semantics of event ε as defined in Section 4.

It is important to trigger the analysis with an expected event ε, since this enables the algorithm to know what it has to look for, and thus operates the sensors needed only. The expected event can change dynamically either by outside means or following the knowledge built from actual sensors data, i.e. in case occupancy data does not match the expected event ε. In the *absence of an expected event*, our occupancy analysis can start with a brute force data sampling from all sensors then the event recognized can be set to be the expected one.

Sensing Plan. A sensing plan R dictates which sensors must be *On* and *how often* those sensors sample occupancy data. R is formally given by $R = \langle (S_x, r_x), (S_y, r_y), (S_z, r_z), \ldots \rangle$, where S_i are sensors and r_i are frequencies. The value of each r_i is derived from the confidence value of the feature each sensor belongs to. If a sensor belongs to multiple features of a given event, the highest confidence value is used then. As an example, if an event e_1 is given by $e_1 = < \mathtt{F1}, H > \wedge < \mathtt{F2}, L >$ where $\mathtt{F1} \triangleq S_1 | V(S_1) \in [a,b]$ and $\mathtt{F2} \triangleq (S_1, S_2) | S_1 \in [c,d]$ then S_1 will be assigned high confidence for both features, which means S_1 will run the same frequency for both F1 and F2, except that the value of S_1 has to satisfy both $[a,b]$ and $[c,d]$ ranges in order to confirm occurrence of event e_1.

5.1 Occupancy Analysis

To perform occupancy analysis, our algorithm compares the sensors readings to the features of the expected event, or any event of the domain-specific language L in case the expected event has already been disapproved or not provided at all. This can in fact lead to 3 cases: 1) good matching; 2) no matching; 3) partial matching.

Good Matching. In case that sensor readings match a single event e, then the expected event ε will be updated to this new event e, i.e. $\varepsilon := e$. From the semantics of e, a sensing plan is computed as follows:

$$R = \langle (S_1, r_1), .., (S_n, r_n) \mid \quad \forall i \; \exists c_j \; (S_i, c_j) \in ||e|| \wedge \; r_i = \begin{cases} 2 & \text{if } c_j = H \\ 4 & \text{if } c_j = M \\ 8 & \text{otherwise} \end{cases} \rangle$$

$||e||$ is the semantics of event e returning the sensors needed, data ranges and confidence levels to recognize e. The calculation of frequency measurement for each sensor is based on the confidence level of the event each sensor is assigned to. By default, we assign frequencies of 2, 4 and 8 minutes respectively to sensors having high, medium and low confidence levels respectively. Those are commonly adopted sensors frequencies for buildings occupancy monitoring solutions (Ortiz Perez et al., 2018; Abraham and Li, 2014; Beck et al., 2021).

In fact, if an event is being confirmed for few consecutive sampling iterations, an optimization of the sensing plan is needed so that we start attenuating sensor frequencies, or even switching off, those having lowest confidence levels as follows.

$$R = R \setminus \{(S_i, r_i) \mid \forall i \; (S_i, L) \in ||e||\}$$

Further elaboration on the optimization of sensing plans is provided in Section 5.2.

No Matching. No-match situations occur when the actual sensing plan leads to match no event of the given domain-specific language. In such a case, a new sensing plan should be to turn on all sensors with high frequency (2 minutes) to be able to get a starting point so that either a good matching or a partial matching of an event occurs.

$$R = \langle (S_1, 2), (S_2, 2), \ldots, (S_8, 2) \rangle$$

The no-match case is also a disapproval of the actual expected event ε. In practice, one has to be careful when re-configuring the sensors frequency as the no-match case is the most expensive in terms of energy consumption.

Partial Matching. This case corresponds to a partial matching of the actual state to a single, or more, event(s). We need to distinguish the two alternatives (single or multiple) as the processing is different for each case.

When the actual state partially matches a single event e, a new sensing plan is derived from the actual sensing plan R by including all the sensors belonging to the features with high confidence in event e. The sensors to be included will run with high frequency (2 minutes) since they are assigned high confidence H. The new sensing plan is calculated as follows:

$$R = R \cup \{(S_i, 2) | \forall i \ (S_i, H) \in ||e||\}$$

If this is still not enough to fully confirm the event partially matching the actual state, the sensing plan will be upgraded to include the sensors belonging to the features with medium confidence in event e. Such sensors will run with medium frequency (4 minutes) since they are assigned medium confidence M.

$$R = R \cup \{(S_i, 4) | \forall i \ (S_i, M) \in ||e||\}$$

If the partial matching situation persists, the actual sensing plan will be expanded to include all the sensor, being Off, belonging to the features with low confidence in event e. This will lead to either confirm the actual event e (*good matching* case), or disapprove the event e (*no-match* case). Thus, the actual sensing plan R is updated as follows:

$$R = R \cup \{(S_i, 8) | \forall i \ (S_i, L) \in ||e||\}$$

When the actual state partially matches multiple events, it is necessary to find a way to decide which event is actually ongoing without the need to activate all sensors as that would not be cost effective. This is done by calculating the difference between the events partially matching the actual state. If two events e_k and e_l partially match the actual state, then we calculate the set of features differentiating the two events e_k and e_l as follows: $D = ||e_k|| - ||e_l||$.

This difference is then used to update the sensing plan R, to be able to decide which of the events is actually ongoing, as follows:

$$R = R \cup D$$

Depending on how many features differentiate the two events e_k and e_l, low confidence ones can be excluded from the new sensing plan as to not turn on all sensors and only operate the sensors leading to distinguish among the two events. A recursive approach can be adopted in a similar way as for single partial matching case.

5.2 Optimization of Sensing Plans

Optimizing a sensing plan consists in tracking the actual state and matching it to an expected event, or potentially ending with a good match with an event even though it is not the expected one. Following the actual state, the optimization alternates between the different cases (good, partial, no-match) mentioned earlier while tuning the sensor frequencies accordingly.

Mainly, when the confirmation of an event occurs consecutively the number of sensors to use to track such an event will be reduced. The first time an event is identified, no optimization will be done. On the second consecutive confirmation, all sensors assigned to that confirmed event with low confidence will be turned off. Similarly, on the fourth consecutive approved occurrence of the expected event all sensors assigned to features with medium confidence will be turned off. On any consecutive occurrence confirmation (good match) beyond this, no optimization will be done.

6 IMPLEMENTATION AND EXPERIMENTAL RESULTS

The proposed system language, functionality, occupancy analysis and optimization algorithm have been implemented in C++. The implementation has been made bottom-up where the basic constructs are implemented as classes, thereafter such class types form the variable types within other classes.

Since the number of features within an event and the number of sensors in a feature are variable and need to be known when instantiating the classes to actual objects, we assigned a static array length for each type that is going to be the maximum entities and track the number of actual objects (for example adding or deleting a sensor to a sensing plan) at runtime.

To assess the performance achieved by our proactivation algorithm, we have conducted an actual occupancy experiment analysis of a University building (single) space for a 24-hour period. By performance, we mean the decision accuracy, the percentage of spared sensors and the energy saved due to some of the sensors being turned off/stretched frequency part of the time. The software implementation and data gathered are available here [2].

Figure 1 depicts the variable total number of sensors being active at runtime, following the actual state, for first 9 hours among the 24-hour period. The analysis for the last 15 hours is omitted as the only event recognized is `Empty`, where the number of sensors operating is static (2 sensors) and only the sensors with high confidence to that event are active i.e., namely motion and light sensors. This has saved con-

[2]https://e.pcloud.link/publink/show?code=kZw0sjZVvNS4RWIuoRSHhpTioRyMj3QF6NX

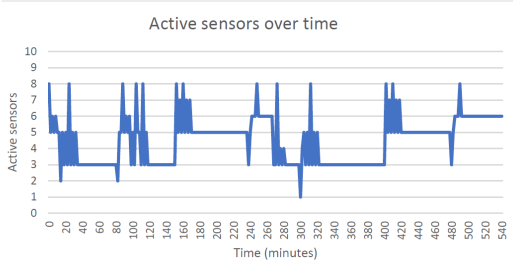

Figure 1: Number of sensors active at runtime.

siderable energy amount, that would have been consumed by 6 sensors over 15 hours if the proactivation is not used.

A single sensor sampling data every 2 minutes (2.5 minutes in (Perra et al., 2021)) consumes in average *0,13 kW/h*, this amounts to 0,0043 kW per sample, if we disregard the slight differences of the energy consumption between the different sensor types. Given that our use case operates 8 sensors, for 24 hours period the total energy consumption for data sampling when using an occupancy analysis algorithm that samples data statically every 2 minutes is: $p = 8 \times 24 \times 0,13 = 24.96 kw/day$.

The total number of samples would be then *5760*. The test run of our proactivation algorithm for the same setup and same data accumulated to 3942 samples, thus saving *31%* of the data sampling compared to the naive algorithm, while achieving the same accuracy and state knowledge level. This means that our intelligent occupancy analysis algorithms has led to save *7,89kw/day* compared to the static frequency algorithm, such as in (Perra et al., 2021; Aftab et al., 2013; Beck et al., 2021), that is mostly adopted in many occupancy monitoring solutions. This savings come from sensors only. The gain in energy saving is far higher one comparing the proactivation solution to the monitoring solutions where sensors are permanently actively, such in (Ortiz Perez et al., 2018; Abraham and Li, 2014). In fact, the real-time knowledge synthesized by the proposed solution can be used as a ground to optimize other energy consumption loads such as lights, heating and ventilation.

The matching between expected events and input data as well as sensing plan optimization percentage have also been analyzed. The results are depicted in Figure 2. This is to showcase the algorithm tradeoff between reliability, having more sensors active to identify whether an event is actually ongoing, and the performance optimization achieved by having fewer sensors active to save on energy.

As one can see from that figure, the sensing plan optimization (orange) kicks in a little while after the

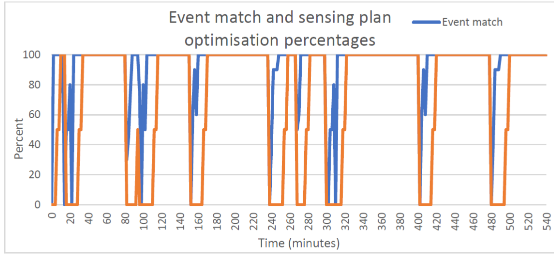

Figure 2: Expected event matching and sensing plan optimization.

event match (blue) has reached 100%. The sensing plan optimization also does not go to 100% right away but spends a couple of iterations at 50% first. In fact, the higher the event matching is, the higher the sensing plan optimization will be. This is justifiable as having the expected event confirmed many times leads to lower the sensors needed to track the continual availability of that event.

7 CONCLUSION

This paper proposed a proactivation system, that is an intelligent knowledge-driven real-time occupancy monitoring solution. The proposed analysis enables to tune the sensors frequency on-the-fly following the actual state so that to reduce data sampling and energy consumption.

The core idea is that rather than collecting large amounts of sensor data to perform occupancy analysis post hoc, we adopted a knowledge-driven approach where we proactively identify the minimal data relevant to the actual state following the semantics of the expected activities. The proposed contribution has been mathematically modeled and an early proof-of-concept prototype has been implemented in C++.

Our solution has been tested and compared to related occupancy analysis alternatives where it outperformed by reducing the sensors energy consumption with up to 31%.

As a future work, we plan to conduct an extensive experiment on actual use cases with larger set of events. Another future work would be to study the complexity and schedulability of the proposed algorithm on the actual multicore platform used for deployment (Boudjadar et al., 2014).

REFERENCES

Abraham, S. and Li, X. (2014). A cost-effective wireless sensor network system for indoor air quality monitoring applications. *Procedia Computer Science*, 34:165–171.

Aftab, M., Chau, S., and Armstrong, P. (2013). Smart air-conditioning control by wireless sensors: An online optimization approach. In *Proceedings of the 4th ACM International Conference on Future Energy Systems*, pages 225–236.

Ahmad, J., Larijani, H., Emmanuel, R., Mannion, M., and Javed, A. (2021). Occupancy detection in non-residential buildings–a survey and novel privacy preserved occupancy monitoring solution. *Applied Computing and Informatics*, 17.

Azimi, S. and O'Brien, W. (2022). Fit-for-purpose: Measuring occupancy to support commercial building operations: A review. *Building and Environment*, 212.

Beck, M. M., Boudjadar, J., and Chougui, Y. (2021). Energy efficient real-time calibration of wireless sensor networks for smart buildings. In *Advances in Model and Data Engineering in the Digitalization Era*.

Boudjadar, J., David, A., Kim, J. H., Larsen, K. G., Nyman, U., and Skou, A. (2014). Schedulability and energy efficiency for multi-core hierarchical scheduling systems. In *International conference on Embedded Real Time Systems and Software ERTS*.

Boudjadar, J. and Khooban, M. H. (2020). A safety-driven cost optimization for the real-time operation of a hybrid energy system. In *Proceedings of the 27th International Conference on Systems Engineering*.

Boudjadar, J. and Tomko, M. (2022). A digital twin setup for safety-aware optimization of a cyber-physical system. In *In Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics*.

Elkhoukhi, H., NaitMalek, Y., Berouine, A., Bakhouya, M., Elouadghiri, D., and Essaaidi, M. (2018). Towards a real-time occupancy detection approach for smart buildings. *Procedia Computer Science*, 134:114–120.

Jiang, J., Wang, C., Roth, T., Nguyen, C., Kamongi, P., Lee, H., and Liu, Y. (2022). Residential house occupancy detection: Trust-based scheme using economic and privacy-aware sensors. *IEEE Internet of Things Journal*, 9(3).

Jiang, W. and Yin, Z. (2015). Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of the 23rd ACM International Conference on Multimedia*.

Lasla, N., Doudou, M., Djenouri, D., Ouadjaout, A., and Zizoua, C. (2019). Wireless energy efficient occupancy-monitoring system for smart buildings. *Pervasive and Mobile Computing*, 59:101037.

Luo, X., Lam, K. P., Chen, Y., and Hong, T. (2017). Performance evaluation of an agent-based occupancy simulation model. *Building and Environment*, 115.

M, J., A, K., MZ, A., SS, W., and MA, H. (2021). Iot-based occupancy monitoring techniques for energy efficient smart buildings. *Turkish Online Journal of Qualitative Inquiry*, 12-3.

McKenna, E., Krawczynski, M., and Thomson, M. (2015). Four-state domestic building occupancy model for energy demand simulations. *Energy and Buildings*, 96.

Minoli, D., Sohraby, K., and Occhiogrosso, B. (2017). Iot considerations, requirements, and architectures for smart buildings-energy optimization and next-generation building management systems. *IEEE Internet of Things Journal*, 4.

Ortiz Perez, A., Bierer, B., Scholz, L., Wöllenstein, J., and Palzer, S. (2018). A wireless gas sensor network to monitor indoor environmental quality in schools. *Sensors*, 18(12).

Pan, S., Bonde, A., Jing, J., Zhang, L., Zhang, P., and Noh, H. Y. (2014). Boes: Building occupancy estimation system using sparse ambient vibration monitoring. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 9061.

Perra, C., Kumar, A., Losito, M., Pirino, P., Moradpour, M., and Gatto, G. (2021). Monitoring indoor people presence in buildings using low-cost infrared sensor array in doorways. *Sensors*, 21.

Pradeep Kumar, H. (2016). Multi-sensor-based occupancy monitoring for energy efficient smart buildings based on internet of things. *ProQuest Dissertations and Theses*.

Rai, S., Wang, M., and Hu, X. (2015). A graph-based agent-oriented model for building occupancy simulation. In *Proceedings of the Symposium on Agent-Directed Simulation*.

Rault, T., Bouabdallah, A., and Challal, Y. (2014). Energy efficiency in wireless sensor networks: A top-down survey. *Computer Networks*, 67.

Salimi, S. and Hammad, A. (2019). Critical review and research roadmap of office building energy management based on occupancy monitoring. *Energy and Buildings*, 182.

Salimi, S. and Hammad, A. (2020). Sensitivity analysis of probabilistic occupancy prediction model using big data. *Building and Environment*, 172:106729.

Seghezzi, E., Locatelli, M., Pellegrini, L., Pattini, G., Di Giuda, G. M., Tagliabue, L. C., and Boella, G. (2021). Towards an occupancy-oriented digital twin for facility management: Test campaign and sensors assessment. *Applied Sciences*, 11.

Sun, K., Zhao, Q., and Zou, J. (2020). A review of building occupancy measurement systems. *Energy and Buildings*, 216.

Yang, J., Santamouris, M., and Lee, S. E. (2016). Review of occupancy sensing systems and occupancy modeling methodologies for the application in institutional buildings. *Energy and Buildings*, 121:344–349.

Zhang, H.-B., Zhang, Y.-X., Zhong, B., Lei, Q., Yang, L., Du, J.-X., and Chen, D.-S. (2019). A comprehensive survey of vision-based human action recognition methods. *Sensors*, 19.

Zhang, W., Wu, Y., and Calautit, J. K. (2022). A review on occupancy prediction through machine learning for enhancing energy efficiency, air quality and thermal comfort in the built environment. *Renewable and Sustainable Energy Reviews*, 167:112704.