# Privacy Sensitive Building Monitoring Through Generative Sensors

Angan Mitra[a], Denis Trystram[b] and Christopher Cerin[c]
*University of Grenoble Alpes, France*

Keywords: Smart Buildings, Sensor Combinatorial Optimization Problem in IoT, Ecological Sustainability, Evolutionary Computing, Recommender Systems and Location-Awareness, Data Analytics.

Abstract: A building equipped with sensors collects heterogeneous data, distributed naturally across zones. The lack of spatiotemporal awareness can lead to excessive sensors or non-optimal distribution across a building. We introduce a novel approach to reduce the friction between high smartness cost and ecological sustainability by proposing virtual sensors as an artifact to estimate the environmental benefit for the planet of doing the "same with less." The key idea behind the contribution is to inject data from virtual sensors to determine if an actual sensor can be replaced, followed by a sub-grouping of sensors. As a first contribution, our work exploits the concept of "less is more" to bring down the capital investment (CAPEX) and recurring expense (OPEX) of the smart-building solutions. This fact opens the door to new research for an eco-responsible deployment of sensors by revisiting the current approach of blind systematic deployment of sensors. We aim to deploy the necessary amount (according to actual, simulated, or virtual uses) and not every room with all possible sensors. As a second contribution, our experiments show a trade-off between virtualization accuracy and active monitoring. Additionally, we validate our insights with 40-60% savings on sensor reduction for a 7-storied Thailand building.



## 1 INTRODUCTION

### 1.1 Context

The know-how of designing and making buildings has seen tumultuous scales of updates, from huts to skyscrapers. Before electricity advent, buildings were conceived as a mere brick-and-mortar rendition of habitable and workable spaces. When electrical appliances started populating households, the notion of passive space turned into a controllable environment using sensors and actuators. However, most of the existing buildings were already constructed before the apparition of the Internet and the World Wide Web in 1990. This observation means building architectures were not developed according to the sensors' quantity, type, and location.

Firstly, the popularity of Internet of Things (IoT) devices led to ad-hoc dissemination in buildings, where environments of dynamic parameters like temperature, CO2, wind, etc., characterize buildings. Such an approach can lead to a naive zonal distribution of sensors due to the obscurity of spatiotemporal importance.

Secondly, a streaming IoT sensor can act as a data source of sensitive patterns raising privacy concerns among stakeholders.

Thirdly, the cost of equipping spaces with embedded hardware over a large commercial area is non-negligible and comes with recurring payments for powering up the solution. In this work, we investigate if there is a way to determine a minimalist sensing solution for non-intrusive spatiotemporal coverage to lower the capital cost and energy footprint of a smart building solution.

In the realm of smart buildings, the convergence of sensor combinatorial optimization problems within the Internet of Things (IoT) landscape presents a significant opportunity for advancing ecological sustainability. Through the lens of evolutionary computing, complex algorithms can be harnessed to optimize sensor placement, maximizing efficiency while minimizing environmental impact.

This approach not only enhances the functionality of IoT systems but also aligns with principles of ecological responsibility. Leveraging recommender systems and location-awareness technology further re-

[a] https://orcid.org/0000-0002-4581-808X
[b] https://orcid.org/0000-0002-2623-6922
[c] https://orcid.org/0000-0003-0993-9826



Mitra, A., Trystram, D. and Cerin, C.
Privacy Sensitive Building Monitoring Through Generative Sensors.
DOI: 10.5220/0012728100003705

In *Proceedings of the 9th International Conference on Internet of Things, Big Data and Security (IoTBDS 2024)*, pages 107-118
ISBN: 978-989-758-699-6; ISSN: 2184-4976

fines data collection processes, ensuring that insights derived from data analytics are both relevant and actionable.

However, amidst these advancements, the paramount concern remains privacy-sensitive building monitoring. By integrating generative sensors, which prioritize data anonymization and encryption, the integrity of individual privacy is preserved without compromising the efficacy of smart building operations. Thus, this holistic approach fosters a symbiotic relationship between technological innovation and ethical considerations, laying the foundation for a sustainable and privacy-respecting built environment.

### 1.2 Problem Statement and Outline of Contributions

Given a temporal stream of data produced by IoT sensors, we investigate the question of what subset of sensors can be reliably powered off. We propose a methodology for pre-integration and plan to place sensors optimally within a building. The methodology considers both virtual sensors (avatars) and non-virtual sensors. The motivation for a virtual avatar envelope over sensors in a building improves non-intrusive sensing and reduces capital and operational costs. The idea of an avatar to simulate a more extensive IoT infrastructure than the current one is one of the main lines of our proposal. Since the markets for smart buildings[1] and IoT[2] are overgrowing, it is urgent to take into account as soon as possible, in an eco-design approach, the need for a reasoned approach to the digitization of buildings.

In Section 2 we introduce the related works. In Section 3 we propose a method to discover a logical grouping of sensors at the edge and formulate encodings to orchestrate a data-sharing policy. We solve the underlying problem using a multi-objective optimization algorithm to locate the edge network structures and identify distinct semantic collections. As per experiments in Section 5, we empirically analyze the policy evidence, discovery, and the lifelong mechanism of checking for optimal data-sharing topologies at the edge. Finally, summarizing in Section 6, we argue that, for environmental issues, it is better to pre-calculate the number of sensors and then buy and deploy them rather than purchasing and deploying an overestimation of the number of sensors.

[1] https://www.fortunebusinessinsights.com/industry-reports/smart-building-market-101198

[2] https://iot-analytics.com/number-connected-iot-devices/

## 2 RELATED WORK

Historically, buildings were not designed to cater to forms of ambient intelligence, instead somewhat optimized spatially for acceptable levels of thermal comfort, indoor ventilation, and privacy. Over time, the building became a composite of observable and controllable elements. In this section, we highlight the limitations of the current situation in mastering the placement of sensors, both at the technology level, model level, and assumption level. Then we conduct a literature survey of the domain, such as sensor approximation and optimal sensor placement. We also introduce the machine learning and combinatorial optimization problem-solving concepts used in our work.

### 2.1 Smart Building Technology Acceptance

Smart applications (Wong et al., 2005) for buildings have been developed mainly for monitoring, analysis, and control of thermal units like Heating Ventilation Air Conditioning (HVAC) units, illumination channels, etc. A 2019 review (Jia et al., 2019) of the smart building industry states the major pain points towards technological adaption. High installation costs, obscurity on data storage policies, and privacy concerns impede the acceptance (Hojjati and Khodakarami, 2016) of the Internet of Things in buildings. Typically smart building applications thrive on real-time sensor data for monitoring or actuation. Research shows that analyzing sensor streams can reveal sensitive patterns about occupancy (Garg and Bansal, 2000) or usage. Consequently, privacy becomes a significant concern for occupants in a building due to the non-zero possibility of a data leak. The cost of constructing (Ma et al., 2017) a smart building is usually 1.2-1.8 times a non-smart counterpart. This initial capital poses the second barrier for a stakeholder (Xu et al., 2019) to overcome before system installation. But before a technical deployment (Ma et al., 2016), the smart solution needs to go through a pre-evaluation stage before finalizing the bill of materials.

### 2.2 Edge Learning for Smart Buildings

The incorporation of the Internet of Things (IoT) has shaped machine learning-driven outcomes for predictive maintenance, anomaly detection, resource optimization, and much more. Sustainability goals have put the spotlight on optimization possibilities for exploring frugality in the training of models (Gong et al., 2021), inference on physical devices, etc.

The vision to enable battery-less computing (Nirjon, 2018) for platforms is shown to be capable of running predefined machine learning tasks via intermittent learning. Usually in the domain of IoT, the data is on the move either between the cluster of devices or to and fro from remote servers. Long-distance information propagation is energy-intensive for which edge computing is growing to be a sustainable computing partner for IoT.

Algorithmic developments and localized data handling techniques at edge (Medeiros and Fernandes, 2020) to develop distributed learning models align with a system-oriented approach (Thrun, 1995) towards machine learning where one focuses on knowledge representation and inferring meaningful information against a stream of productive tasks (Chen and Liu, 2016). Notably for buildings, the generated data contains sensitive information regarding activity patterns and this makes data sharing difficult. This opens the scope for federated learning (Mitra et al., 2021) or decentralized techniques (Mitra et al., 2022) to promote building intelligence by strictly adhering to inhouse data policy. This line of work provides critical insight into the role of communication topology and utilization on a real-life smart building data set.

### 2.3 Sensor Allocation Problem

Multiple cyber-physical systems like sensors and actuators work in cohesion to maintain the desired quality of ambiance and indoor comfort of a building. Some examples of non-intrusive ambient sensors are temperature, humidity, and luminosity. Data values recorded by a type of sensor are usually dissimilar across different buildings or separate zones in a building. Empirical Mode Decomposition (EMD) (Fontugne et al., 2012) of a continuous variable such as temperature, humidity, or luminosity yields Intrinsic Mode Functions (IMF)(Ayenu-Prah and Attoh-Okine, 2010). This model has been helpful for structural health monitoring (Barbosh et al., 2020) for buildings. K means clustering over the space of IMF for all the sensors is shown to be effective (Hong et al., 2013) in identifying non-identical sensors. This approach is further extended (Yoganathan et al., 2018) by using information loss to eliminate weak candidate points from a cluster to obtain a sensor placement solution.

Generally, choosing the globally optimal placement within the search space of a large-scale complex system is an intractable computation, in which the number of possible placements grows combinatorially with the number of candidates (Ko et al., 1995). Py-Sensors (de Silva et al., 2021) is a software package published in 2021 that includes state-of-the-art algorithms on scalable optimization of sensor placement from data. It is to be noted that the basis on which one represents measurement data can have a pronounced effect (Manohar et al., 2018) on the sensors that are selected and the quality of the reconstruction. The task of classifying sites for sensor placement for benchmarking is the Sparse Sensor Placement Optimization for Classification (SSPOC) algorithm (Brunton et al., 2013). The algorithm is related to compressed sensing optimization (Emmanuel et al., 2005) but identifies the sparsest set of sensors that reconstructs a discriminating plane in a feature subspace. Regarding reconstruction problems, the package implements methods for efficiently analyzing the effects that data or sensor quantity have on reconstruction performance (Manohar et al., 2018). Often different sensor locations impose variable costs, which are taken into account during sensor selection via a built-in cost-sensitive optimization routine (Clark et al., 2018). Above mentioned methods are neither incremental nor self-aware to attempt corrective measures. Hence it is obscure how they will detect changes in building patterns and correspondingly adjust the sensor allocation/placements.

## 3 PROBLEM MODELING

The question ”How many are too few or too many sensors” is often an undermined topic when installing sensors in a building or multiple spaces. The work addresses data privacy for smart buildings and proposes in-house data circulation as a backbone to power off redundant sensors. In this context, we introduce the Virtual Sensor Field, a mixed basket of physical and computable sensors that creates a virtual avatar over a set of sensors distributed over multiple spaces.

Figure 1 represents a high-level overview of the virtual field. The principal idea is grouping the set of sensors the formulation is twofold:

1. Learn a methodology to partition the sensor set $S$ and provide insight on which sensors are most likely to stay active or be replaced by virtual counterparts.
2. Figure out how grouping sensors can leverage data proximity at the edge, following a strictly inhouse data retention policy.

Assume that the notation $n^X$ means the number of elements in set X. So in a building let there be $n^S$ sensors of $n^K$ types distributed over $n^F$ floors. Let $G$ be a collection of $n^G$ groups/sets, where for each group $g$, active and virtual sensors are denoted by $\mathcal{A}^g$

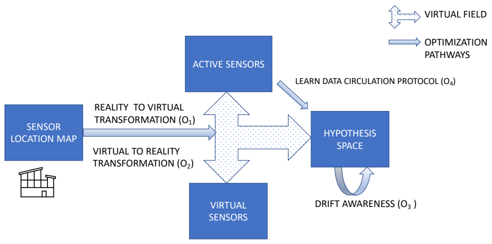



Figure 1: Schematic of Virtual Sensor Field with optimization pathways.

and $\mathcal{V}^g$ respectively.

$$\underbrace{S}_{All\ Sensors} = \bigcup_{g=1}^{g=n^G} \underbrace{S^g}_{Sensors\ in\ group\ g} \tag{1}$$

$$\underbrace{S^g}_{Group\ g\ Sensors} = \underbrace{\mathcal{A}^g}_{Active\ Sensors} \cup \underbrace{\mathcal{V}^g}_{Virtual\ Sensors} \tag{2}$$

Now we break down the virtualization process into three steps as follows:

- **Reconstruction** of hidden sensor data from real or virtual deployed sensors. (Section 3.1)
- **Classifying** sensors as real or virtual and fixing where the sensors are to be placed. (Section 3.2)
- **Re-calibration** of virtual sensor field with an incremental data feed from real sensors. (Section 3.3)

## 3.1 Regressing Signal Reconstruction

The signal reconstruction mechanism reconstructs virtual sensor data from actual sensors and vice versa by creating a set of machine-learned regressors. Since this mechanism works for every group before optimization, one faces the cold start problem where the optimal group size is unknown. Most importantly, we bring to the reader's attention that the classification for real and virtual sensors is a priori not known. To resolve the issue, the system creates the following seed groupings:

1. Bucketing sensors by the same type, thus ending up with $n^K$ groups.
2. Grouping sensors by space or same floor, therefore creating $n^F$ groups.

Now for all of the cold start groups, the system learns all possible pairwise regressors between sensors. For example, in the case of spatial grouping, the stream of a $CO_2$ type sensor can be reconstructed using a luminous intensity sensor placed within the same floor. Likewise, for a type-wise grouping, a temperature sensor placed in the $2^{nd}$ floor can be approximated using a similar type of candidate sensor from the top floor. Thus, the learning complexity for $n^G$ logical groups is $O(n^G W^2)$ models, where $W$ is the maximum group size.

The bidirectional transformation function between $\{\mathcal{A}^g, \mathcal{V}^g\}$ is learned through per group hypothesis space $H^g$ defined by Equation 3.

$$H^g = \begin{bmatrix} H_f^g & : \mathcal{A}^g \to \mathcal{V}^g \\ H_b^g & : \mathcal{V}^g \to \mathcal{A}^g \end{bmatrix} \tag{3}$$

$H_f^g$ refers to forward feature space that translates from reality to the virtual world, while the subscript $b$ in $H_b^g$ denotes the reverse backward mapping between hidden sensors and real-life deployment. The quality of $H^g$ is evaluated through a cost function $L$ (such as Root Mean Square, L2, L1 norms) executed over all possible pair-wise interaction pairs $(u,v)$ $\forall u \in S, v \in S, u \neq v$. The error in predicting channel $v$ using a sensor $u$ is recorded at the $[u,v]^{th}$ index of an error matrix $E^g$ as per Equation 4.

$$E^g[u,v] = L(v, \underbrace{H^g[u,v]}_{ML\ model}(u)) \tag{4}$$

Note that $E^g[u,v] \neq E^g[v,u]$ implies that the two losses generated by swapping the dependent and independent variables may not be equal.

To estimate the sensor value $y_v$ of a channel $v \in \mathcal{V}^g$, we first select the optimal channel $(u^*)$ to predict by using the $[u^*,v]^{th}$ entry of hypothesis library $H_f^g$ as per Equation 5.

$$\begin{aligned} u^* \leftarrow &\quad \arg\min_{g \in G} E^g[u,v] \\ y_v = &\quad H_f^g[u^*,v](u^*) \end{aligned} \tag{5}$$

This technique bounds the maximum observable error since another optimal mapping $H^*$ can exist using more than one feature for prediction.

## 3.2 Classifying Sensor Placement

Next, we introduce grouping sensors for answering: *How can we leverage intra-zone patterns to optimize data flows between sensors for virtualization?* The solution to such problems is typically a set of 'non-dominated' solutions where an objective can not be improved without decreasing the other objectives. We define the first two objectives to measure the prediction error due to forward $H_f^g$ and backward $H_b^g$ hypothesis, respectively. For both equations 6 and 7, symbols **u** and **v** shall stand for real and virtual sensors, respectively.

$$\underbrace{O_1(G)}_{Virtual\ loss} = \Sigma_{g \in G, u \in \mathcal{A}^g, v \in \mathcal{V}^g} E^g[u,v] \tag{6}$$

$$\underbrace{O_2(G)}_{Inv\ Virtual\ loss} = \Sigma_{g \in G, u \in \mathcal{A}^g, v \in \mathcal{V}^g} E^g[v,u] \quad (7)$$

### 3.3 Re-Calibrating with Episodic Data

The system experimentally investigates the quality of the sensor configuration to power up the virtual sensor field optimally. Once optimal configurations are deployed, valid sensor data at certain zones are missing. Over time, predictions may lead to blind spots where the ground truth may vastly deviate, or installing sensors can become necessary.

Thus the affinity grouping $\mathcal{A}^g, \mathcal{V}^g$ can be re-calibrated with the availability of additional data, but such a process must consider the historical performance. For any $t \in T$, reconstruction loss at sensor position $i$ is the absolute difference between the actual $(y_i)$ and predicted value $\hat{y}_i$ of a sensor for a virtual mask is given as per Equation 8.

$$\underbrace{O_3(G,t_s,t_e)}_{Reconstuction\ Feed} = \frac{\Sigma_{t \in [t_s:t_e]} \Sigma_{i \in S}}{|t_e - t_s| n^S} |y_i(t) - \hat{y}_i(t)|$$

$$\hat{y}_i(t) \in \begin{cases} H_f^g(\mathcal{A}^g) & if\ s_i \in \mathcal{A}^g \\ H_b^g\big(H_f^g(\mathcal{A}^g)\big) & if\ s_i \in \mathcal{V}^g \end{cases} \quad (8)$$

### 3.4 Data Network Sparsity

The policymaker additionally models the network topology of sensors to minimize the number of data-sharing links. Let every node $i$ have $e_i^I$ number of incoming edges and $e_i^O$ outgoing connections. Equation 9 gives $O_4$ defined as the ratio between the number of edges in $M$ to the total edges in a complete graph. In a spectral space spanned by $n^G$ entries, the representation for any sensor u is given by $\alpha_u \in \mathbf{R}^{n^G}$ and the net non-randomness$(O_4)$ of the underlying connectivity network is simply the sum of all possible pairwise dot product between nodes.

$$\underbrace{O_4(S)}_{Networking\ Volume} = \Sigma_{i \in S} \frac{e_i^I}{e_i^O + e_i^I} \quad (9)$$

The non-randomness of an edge tends to be small when the two nodes linked by that edge are from two different communities. The quality of the Virtual Sensor Field is further tracked through a relative measure that indicates to what extent the data sharing/connectivity graph differs from random graphs in terms of probability.

When $O_4$ is close to 0, the graph tends to be more likely generated by an Erdos Renyi model.

## 4 SENSOR FIELD VIRTUALIZATION SOLVER

We now present the solver that combines all 4 objectives as formulated above, under Section 3 to optimize the sensor placement incrementally. Each placement strategy is encoded as a vector of numbers, and such a vector shall be referred to as a policy. A policy assigns a group number and a 0/1 tag indicating every sensor's virtual or physical presence. For example, if a sensor belongs to group $i$, and has a virtualization tag $j \in \{0,1\}$, then the encoding is given as $2 \times i + j$. The search space for all possible placements of $n^S$ sensors is exponential in the order of $2^{n^S}$.

### 4.1 Policy Building Routine

Let $\mathcal{P}$ be a set of policies modeling non-identical sensor placement configurations. *How to cherry-pick robust positions and create partial ordering amongst multiple strategies?* Algorithm 1 creates an ordered front ($\phi$) of positioning strategies which accelerates decision-making. It uses non-dominated sorting to obtain solutions superior to other configurations. This enables incrementally adding sensors Pareto-optimally while keeping track of the number of solutions inferior to a policy within the pool. The candidate solutions with maximal superiority are included in the first front/batch to build up the sensor blanket bottom-up.

### 4.2 Policy Exploration Routine

*How to ascertain if there is a relative advantage in switching from one configuration to another?* The answer is a Gain Matrix ($GN$) of size $M \times N$ with M policies and N objectives. The core intuition behind Algorithm 2 is the density of solutions within a policy's neighborhood. The policy pool is sorted for every objective in ascending order, and the corresponding objective weight initializes the starting objective value for each sensor configuration. Each element of Gain Matrix $GN$ is updated with the differential margin of the objective scores between the policies at the $i-1$ and $i+1$ index. Note that if all the objectives' values are co-linear, the gain term is 0.

**Policy Explorer** spans the configuration space through two well studied genetic operators (Umbarkar and Sheth, 2015). Mutation operates per group and randomly toggles a sensor from the active group ($\mathcal{A}^g$) to the virtual group ($\mathcal{V}^g$) and vice-versa. The second operator, the Random Crossover, is performed amongst two randomly chosen affinity groups within a policy mask.

Algorithm 1: Sensor Front Builder.

**Input**: Policy set $P$
**Output**: Ordered Sensor Front $\phi$

**for** every policy $p \in P$ **do**
  **for** every policy $q \in P$ **do**
    **if** $p \succ q$ **then**
      $S_p \leftarrow S_p \cup \{q\}$ { $\triangleright$ absorb policy $q$ since every objective in $q$ is better than $p$ }
    **else if** $q \succ p$ **then**
      $n_p = n_p + 1$ { $\triangleright$ Count how many solutions are superior in $S$ to $q$. }
    **end if**
  **end for**
  **if** $n_p$=0 **then**
    $\phi_1 = \phi_1 \cup \{p\}$ { $\triangleright$ Select only non-dominating solutions as the first front. }
  **end if**
**end for**
i= 1
**while** $\phi_i \neq 0$ **do**
  $\mathbf{C} = \phi$ { $\triangleright$ For every front, incrementally add sensors starting from zero.}
  **for** each p $\in \phi_i$ **do**
    **for** each q in $S_p$ **do**
      $n_q = n_q - 1$
      **if** $n_q = 0$ **then**
        $\mathbf{C} = \mathbf{C} \cup \{q\}$ { $\triangleright$ Add non-dominant sensors to a placement configuration}
      **end if**
      $i = i + 1; \phi_i = \mathbf{C}$
    **end for**
  **end for**
**end while**

Algorithm 2: Differential Gain Estimator.

**Input**: Policy pool $P$ of size $M$,
N objective functions
**Output**: Gain Matrix $GN$ of size $M \times N$

**for** every objective $j \in 1$ to $N$ **do**
  $GN_j[p] = O_j(p) \forall p \in P$. { $\triangleright$ Number of entries in $GN_j = M$}
  $GN_j[0] = GN_j[M] = 0$
  **for** i = 2 to $M - 1$ **do**
    $GN_j[i] += (GN_j[i-1] - GN_j[i+1])$
  **end for**
**end for**

## 4.3 Policy Optimizer Routine

In Algorithm 3 NSGA II (Deb et al., 2002) is modified to search the configuration vector space, with time complexity of $O(2^{n^S})$, to optimize our objectives,

Algorithm 3: Policy Optimizer.

**Input**: Initial policy pool $P_{t=0}$ of size $M$
N objective functions,
Iteration Limit $T_{max}$
**Output**: M best policies

Initialize $t \leftarrow 0$, $Q_{t=0} = \phi$
**while** $t \leq T_{max}$ **do**
  $R_t \leftarrow P_t \cup Q_t$
  $\phi$ = **Sensor Front Builder**($R_t$)
  $i \leftarrow 0$ {$\triangleright$ Incrementally add configurations}
  **while** $i < |\phi|$ **do**
    $P_{t+1} = P_t \cup \phi_i$
    $GN \leftarrow$ **Differential Gain Estimator**($\phi_i$)
    Sort($P_{t+1}$) based on $GN$
    $P_{t+1} = P_{t+1}[1:M]$ {$\triangleright$ Get top M policies}
    $i \leftarrow i + 1$
  **end while**
  $Q_{t+1}$ = **Policy Explorer**($P_{t+1}$)
  $t \leftarrow t + 1$
**end while**

and as follows:

1. Take as input the learned hypothesis space and error matrix $\{H^g, E^g\} \forall g \in G$.
2. Initialize a fixed-sized sample pool ($P_t$) of policies as a random string of 0's and 1's.
3. For every policy, evaluate the objective set $[O_1, O_2, O_3, O_4]$.
4. If the maximum number of generations is reached or incremental gain is lower than a threshold, the algorithm stops; else, a child population $Q_t$ is created using steps 5, 6, and 7.
5. Policy sorting is used to incrementally identify Pareto optimal solutions till the entire population is exhausted.
6. Policy Gain Estimator is used to check the density around individual solutions to prevent the algorithm from terminating in a local optimum. Policies within the rectangular field spanned by the nearest adjacent solutions are discarded.
7. Alteration in the encoding is achieved through genetic operators: Random Crossover (Umbarkar and Sheth, 2015) implemented as the policy sampler.
8. Finally populations $P_t$ and $Q_t$ are combined to generate the parent population at time $t + 1$ using steps 5 and 6 in order.
9. Go back to Step 3 and iterate with the generation count decreased by 1.

# 5 EXPERIMENTS

## 5.1 Data Set, Settings and Experimental Plan

We consider the dataset from (Pipattanasomporn et al., 2020) for the experiments. It comes from a seven-floor building in Thailand, including 24 smart zones with 1.5 years of data collected at a 1-minute resolution. The analysis highlights three key decomposition steps to build up a Virtual Sensor Field:

- **Evidence Investigation** of error matrices ($\{E^g\}$) to judge the quality of virtualization accuracy as per sub-section 5.2.
- **Policy Encoding** ($M$) for generating a virtual sensor field according to sub-section 5.3.
- **Policy Re-calibration** to incrementally build up a policy from the bottom up by optimizing the hypothesis space as given in sub-section 5.4.

## 5.2 Evidence Investigation

Table 1: Error Matrix with Spatial Grouping when predicting a sensor of type *t* and floor *f*, all the sensors from *f* are used for forecasting.

| Zone | Power | | | Ambience | | |
|---|---|---|---|---|---|---|
| | AC | Light | App | Temp | RH | Lux |
| FL-2Z1 | 0.15 | 0.14 | 0.13 | **0.18** | 0.53 | 0.15 |
| FL-2Z2 | **0.08** | 0.07 | 0.15 | **0.11** | 0.36 | **0.06** |
| FL-2Z4 | 0.33 | 0.31 | 0.73 | 0.33 | 0.66 | 0.31 |
| FL-3Z1 | 0.32 | 0.23 | 0.38 | 0.24 | 0.45 | 0.26 |
| FL-3Z2 | 0.35 | 0.25 | 0.27 | 0.29 | 0.4 | 0.27 |
| FL-3Z4 | 0.34 | 0.23 | 0.25 | 0.22 | 0.61 | 0.22 |
| FL-3Z5 | 0.42 | 0.25 | 0.27 | 0.28 | 0.63 | 0.24 |
| FL-4Z1 | 0.28 | 0.24 | 0.19 | 0.2 | 0.53 | 0.26 |
| FL-4Z2 | 0.34 | 0.27 | 0.48 | 0.25 | 0.59 | 0.25 |
| FL-4Z4 | 0.28 | 0.25 | 0.29 | 0.24 | 0.53 | 0.24 |
| FL-4Z5 | 0.36 | 0.18 | 0.35 | 0.23 | 0.46 | 0.17 |
| FL-5Z1 | 0.23 | 0.2 | 0.19 | 0.15 | 0.45 | 0.22 |
| FL-5Z2 | 0.29 | 0.19 | 0.28 | 0.19 | 0.35 | 0.19 |
| FL-5Z4 | 0.33 | 0.36 | 0.31 | 0.3 | 0.58 | 0.3 |
| FL-5Z5 | 0.43 | 0.26 | 0.29 | 0.31 | 0.64 | 0.26 |
| FL-6Z1 | 0.26 | 0.23 | 0.25 | 0.29 | 0.37 | 0.22 |
| FL-6Z2 | 0.36 | 0.28 | 0.22 | 0.28 | 0.38 | **0.3** |
| FL-6Z4 | 0.26 | 0.17 | 0.27 | 0.22 | 0.41 | 0.21 |
| FL-6Z5 | 0.47 | 0.22 | 0.26 | 0.23 | 0.58 | 0.23 |
| FL-7Z1 | 0.34 | 0.28 | 0.43 | 0.31 | 0.65 | 0.48 |
| FL-7Z2 | 0.31 | 0.3 | 0.59 | 0.33 | 0.61 | 0.23 |
| FL-7Z4 | 0.28 | 0.21 | 0.28 | 0.23 | 0.41 | 0.2 |
| FL-7Z5 | 0.44 | 0.38 | 0.71 | 0.34 | **0.61** | 0.36 |

*What is the trade-off in terms of accuracy between keeping a sensor powered on and alternately switched off within a group g*? Once a grouping strategy is fixed, the system trains a forecasting model between two sensor channels $(u, v)$ for every sensor map group $(\mathcal{A}^g, \mathcal{V}^g)$. $n^G$ disjoint groups enable computing the hypothesis space $H^g$ and the error matrix table in parallel. For every prediction task between $u, v$, the computed center generates an error matrix for each model, for example, linear regression, random forest, and XGBoost. The training step ingests 90 days of sensor data feed from every type of sensor in each place. The evidence described corresponds to one month per season train feed picking three months from each of the:

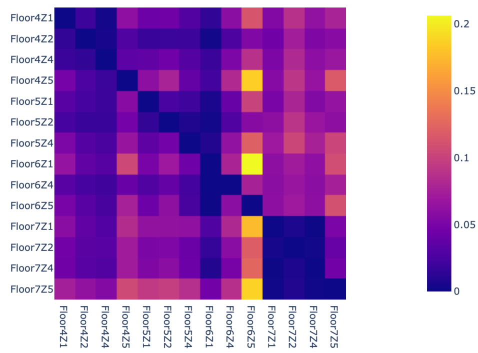
- Summer (March - June). Hottest time of the year with an average low of 25 degrees to an average high of 35 degrees.
- Rainy Season (July – October). Average minimum 24 degrees and average high 32 degrees.
- Winter (November – February). Average minimum 20 degrees and average high 29 degrees.

Tables 1 reflect the maximum error recorded with space-wise grouping, respectively. For every channel, the minimum and the maximum training errors are presented as a tuple. Power consumption patterns are best learned from a similar category of sensors (min = 0.03, max = 0.27, grouping = type) rather than combining data from multiple heterogeneous sensors in the same place (min = 0.2, max = 0.71, grouping = space). This observation is consistent for predicting light, AC, and appliance power consumption levels as seen from Figure **2**.

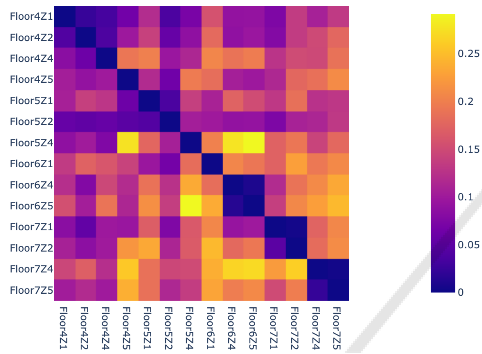
Luminosity (lux) levels have the best type-wise grouping approximation (min=0.03, max=0.14), although, for floor 2 Zone 2, we observe spatial grouping lower error $E$ (spatial) $= 0.06 < E$ (domain) = 0.09. For indoor temperature prediction, domain-wise grouping (min=0.04, max=0.14) yields an error lower than (min=0.19, max=0.71, grouping = space) for all floors except in floor 2 with zone 1, where Error (spatial) = 0.18 < Error (domain) = 0.25, and for zone 2 Error (spatial) =0.11 < E (domain) = 0.24. The relative humidity is most challenging to approximate (min=0.37, max=0.66, grouping = space) and (min=0.14, max=0.63, grouping = type) across all zones. Overall we see that domain-wise grouping performs better on average, which confirms the intuitiveness of being guessed *easily* by similar peers as seen from Figure **3**.

## 5.3 Policy Discovery

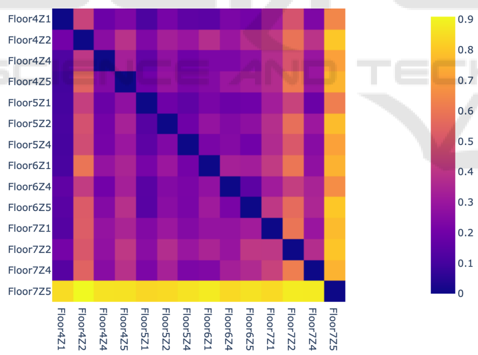
The policy is a 1D vector made up of $n^S = 138$ positive numbers where each integer encodes the data affinity group number and a Boolean instruction (0/1) indicating whether to be powered off or on, respec-




(a) AC Power.

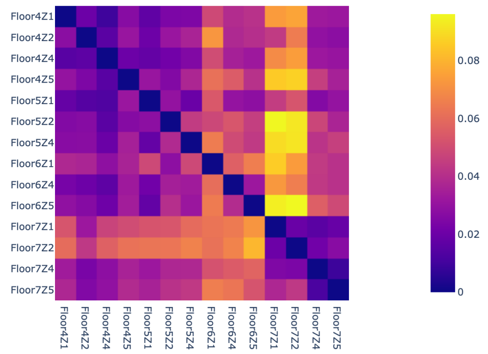



(b) Light Power.

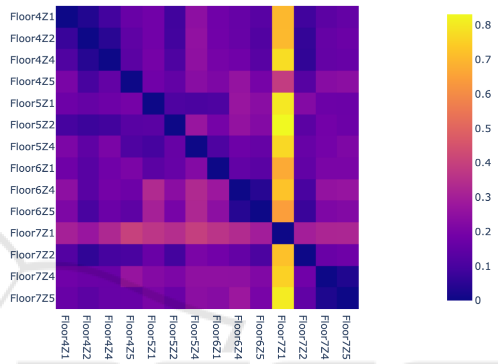



(c) Appliance Power.

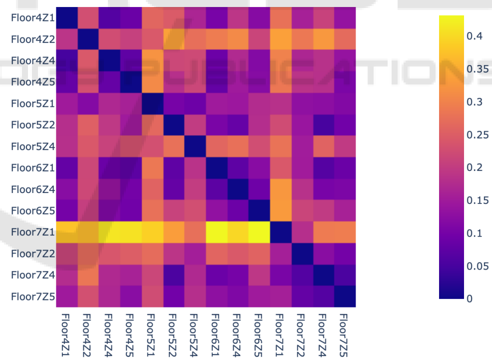
Figure 2: Virtualization prediction on treating an identical type of power meters as one group.




(a) Temperature.




(b) Luminosity.

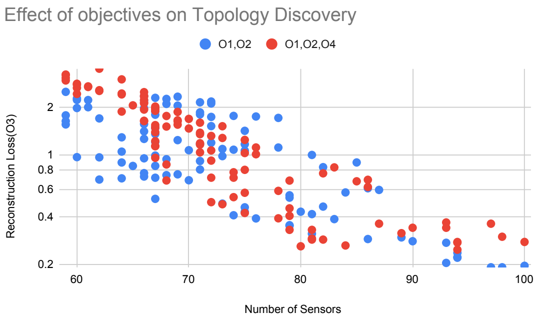



(c) Humidity.

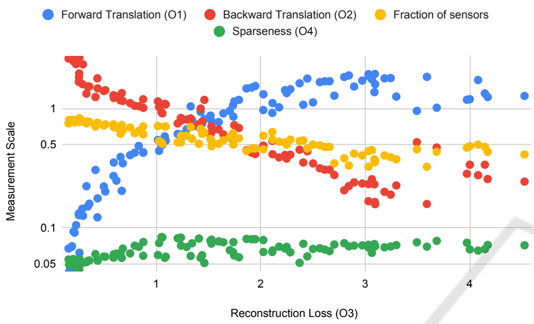
Figure 3: Virtualization prediction on processing similar ambiance channels as one group.

tively. Given an exhaustive set of policy evidence, the task is to optimize the bi-partition of $\{\mathcal{A}^g, \mathcal{V}^g\}$ for every group $g$ belonging to the affinity mask $(A)$. A pool of 50 candidate policies is randomly generated and acts as an input to Algorithm 3, optimized over $n^G$ sensor groups.

Figure 4a displays the variation in reconstruction loss $O_3$ on the test data when training the system with only $O_1, O_2$ losses and then additionally plugging topology loss $(O_4)$. It is seen that the fraction of sensors bounded by the policy region of $O_3 \in (1,2) \wedge O_1 \in (0.5, 1)$ is around 45-65% and has a backward translation error between (0.4, 1.2). Regarding $O_4$, from Figure 4b, it is observed that 5-10 % of total possible edges or data flow paths are sufficient for a policy to be competitively accurate. Indeed, policy configurations exist where the observed error difference is bounded by within 1.5 units of deviance for ambiance and energy monitoring sensor groups.




(a) Discovering Policy Space.



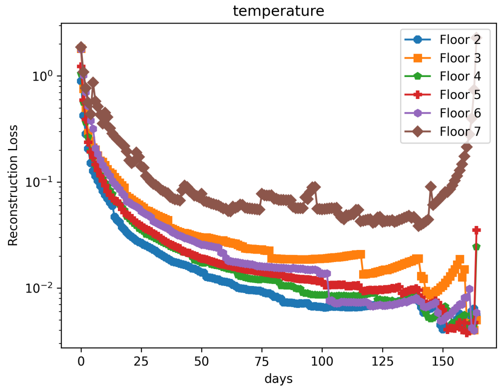

(b) Evaluating Objectives.

Figure 4: Characteristics of the Policy Space.

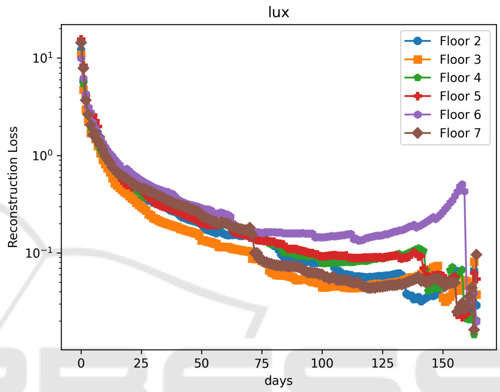
## 5.4 Policy Re-Calibration

This subsection answers how to incrementally build up a policy mimicking the situation where a temporary sensor collects data and updates the policy on the fly. It is desired for a re-calibrated sensor placement configuration to have high confidence in detecting relatively more challenging spatiotemporal sensor patterns. This helps in deciding which sensors to include, thereby generating the most negligible reconstruction loss at run time. Once Pareto Optimal Sensor configurations are generated using the train data, the system tracks their performance over time on the hold-out data, assuming every policy is exclusively deployed. The data set per sensor be split into $B$ batches, where a batch $i$ for a sensor $k$ placed at zone $z$ is denoted by $D^i_{k,z} \equiv [\frac{T_{max}}{B} : \frac{T_{max}}{B} + B]$. On receiving $D^i_{k,z}$ at $i^{th}$ time-step, the learning system evaluates 4 objectives denoted by Equations 6 - 8 to generate better data transfer topologies. Figures 5 and 6 show the reconstruction error $O_3$ on test year for each of 6 sensor types.

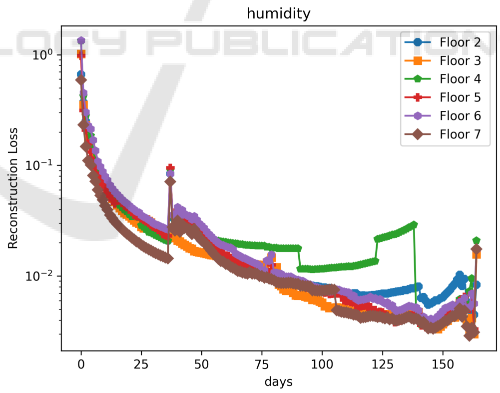
We discover that the temperature at the topmost floor of the building is susceptible to the maximum environmental fluctuations, and expressed by the diverging nature of $O_3 > 10\%$ in Figure 5a. Some of the




(a) AC Power Prediction.



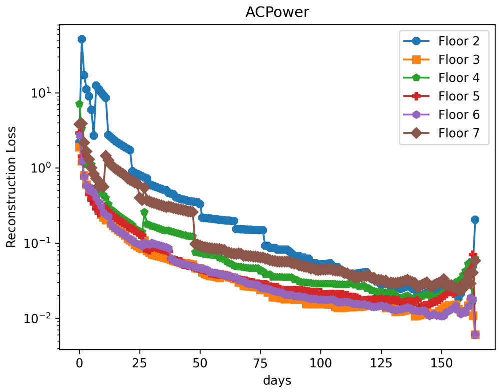

(b) Illumination Power Prediction.



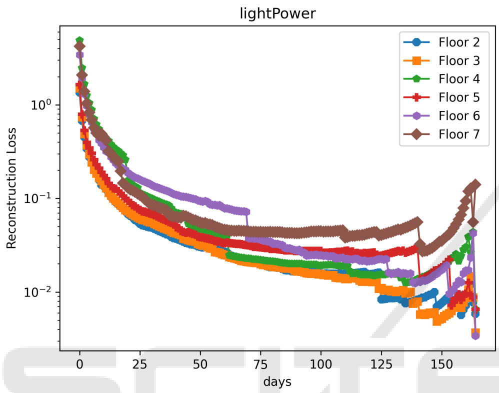

(c) Humidity.

Figure 5: Variation of reconstruction loss $O_3$ in ambiance sensing group with increasing data feed (X-axis).

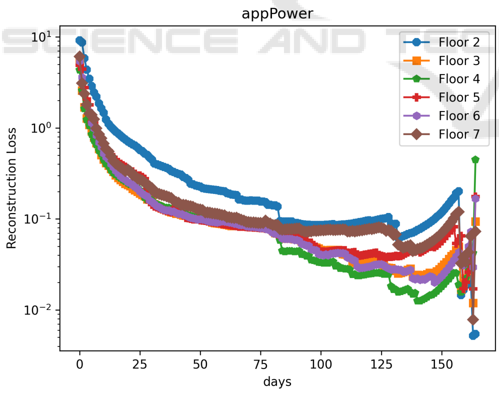
significant factors that influence the luminosity level at a spot are natural lighting, artificial illumination, and occlusion. The interaction between the three elements is more complicated to model than controlling the power for lighting. It is revealed by comparing reconstruction loss ($O_3$) between luminosity levels and power consumption in Figures 5b and 6b, respectively. Due to unknown spatial orientation, it is




(a) AC Power Prediction.




(b) Illumination Power Prediction.




(c) Humidity.

Figure 6: Variation of reconstruction loss $O_3$ in energy consumption group with increasing data feed (X-axis).

hard to tell which zones have windows. On average, the approximating power consumption shows close to 10 times lower reconstruction loss than ambient sensors like temperature, luminosity, and humidity. As per Figure 6a, 75 days or 2.5 months of data collection suffices to keep the approximation error below 10% for all the six floors, the probable reason being controlled power consumption by an AC. Notably, the approximation ability of light power and lux is close to 98% accurate for floor 2 compared to 90% correct for the top two floors (6,7). In a continual setting, the system updates the hypothesis space and auto-re-calibrates to stabler sensor placement configurations with the availability of more data. Table 2 gives the optimal sensor placement distribution that uses 45 sensors instead of 138, bringing in a 67% sensor reduction.

Table 2: Optimal installation suggestions to ecologically monitor the seven-storied buildings in Thailand as covered by the data set.

| Type | # | Save | Installation Sites | Approximated Locations |
|---|---|---|---|---|
| Temperature | 9 | 0.61 | 'FL-4Z4', 'FL-2Z2', 'FL-4Z2', 'FL-3Z1', 'FL-7Z5', 'FL-3Z2', 'FL-5Z1', 'FL-7Z1', 'FL-3Z5' | FL-4Z5', 'FL-6Z4', 'FL-6Z5', 'FL-2Z1', 'FL-6Z1', 'FL-2Z4', 'FL-6Z2', 'FL-4Z1', 'FL-7Z4', 'FL-5Z5', 'FL-5Z4', 'FL-7Z2', 'FL-3Z4', 'FL-5Z2' |
| Humidity | 6 | 0.74 | 'FL-4Z4', 'FL-3Z1', 'FL-7Z2', 'FL-7Z1', 'FL-3Z5', 'FL-5Z2' | FL-4Z5', 'FL-2Z2', 'FL-6Z4', 'FL-6Z5', 'FL-2Z1', 'FL-6Z1', 'FL-4Z2', 'FL-2Z4', 'FL-6Z2', 'FL-4Z1', 'FL-7Z4', 'FL-7Z5', 'FL-5Z5', 'FL-3Z2', 'FL-5Z4', 'FL-5Z1', 'FL-3Z4' |
| Luminosity | 8 | 0.65 | 'FL-2Z2', 'FL-2Z1', 'FL-6Z1', 'FL-4Z2', 'FL-3Z1', 'FL-7Z5', 'FL-3Z2', 'FL-7Z2' | 'FL-4Z5', 'FL-4Z4', 'FL-6Z4', 'FL-6Z5', 'FL-2Z4', 'FL-6Z2', 'FL-4Z1', 'FL-7Z4', 'FL-5Z5', 'FL-5Z4', 'FL-5Z1', 'FL-7Z1', 'FL-3Z5', 'FL-3Z4', 'FL-5Z2' |
| lightPower | 7 | 0.7 | 'FL-2Z2', 'FL-6Z5', 'FL-4Z1', 'FL-3Z1', 'FL-7Z5', 'FL-7Z2', 'FL-3Z4' | 'FL-4Z5', 'FL-4Z4', 'FL-6Z4', 'FL-2Z1', 'FL-6Z1', 'FL-4Z2', 'FL-2Z4', 'FL-6Z2', 'FL-7Z4', 'FL-5Z5', 'FL-3Z2', 'FL-5Z4', 'FL-5Z1', 'FL-7Z1', 'FL-3Z5', 'FL-5Z2' |
| ACPower | 10 | 0.57 | 'FL-2Z1', 'FL-6Z1', 'FL-7Z4', 'FL-3Z1', 'FL-7Z5', 'FL-3Z2', 'FL-5Z4', 'FL-7Z2', 'FL-5Z1', 'FL-5Z2' | 'FL-4Z5', 'FL-4Z4', 'FL-2Z2', 'FL-6Z4', 'FL-6Z5', 'FL-4Z2', 'FL-2Z4', 'FL-6Z2', 'FL-4Z1', 'FL-5Z5', 'FL-7Z1', 'FL-3Z5', 'FL-3Z4' |
| appPower | 5 | 0.78 | FL-4Z4', 'FL-2Z4', 'FL-4Z1', 'FL-5Z4', 'FL-5Z1' | 'FL-4Z5', 'FL-2Z2', 'FL-6Z4', 'FL-6Z5', 'FL-2Z1', 'FL-6Z1', 'FL-4Z2', 'FL-6Z2', 'FL-7Z4', 'FL-3Z1', 'FL-7Z5', 'FL-5Z5', 'FL-3Z2', 'FL-7Z2', 'FL-7Z1', 'FL-3Z5', 'FL-3Z4', 'FL-5Z2' |

## 5.5 Comparative Study

Now we test the performance of the virtual sensing field in comparison to a random distribution, Support Vector Decomposition guided placement, and sparse sensor placement optimization for classification (SSPOC) (de Silva et al., 2021). When the number of sensors is gradually incremented, Figure 7 shows the performance gain in accuracy using our approach. The benchmark methods show a low virtualization quotient thereby needing more sensors to maintain comparable levels of accuracy. The key highlights of our approach to monitoring a building are as follows:

- Evidence investigation measures the virtualization capacity at every place and displays the error if a sensor were to be powered off at that spot.
- Sensor placement configuration is augmented with in-house data circulation pathways. We ob-

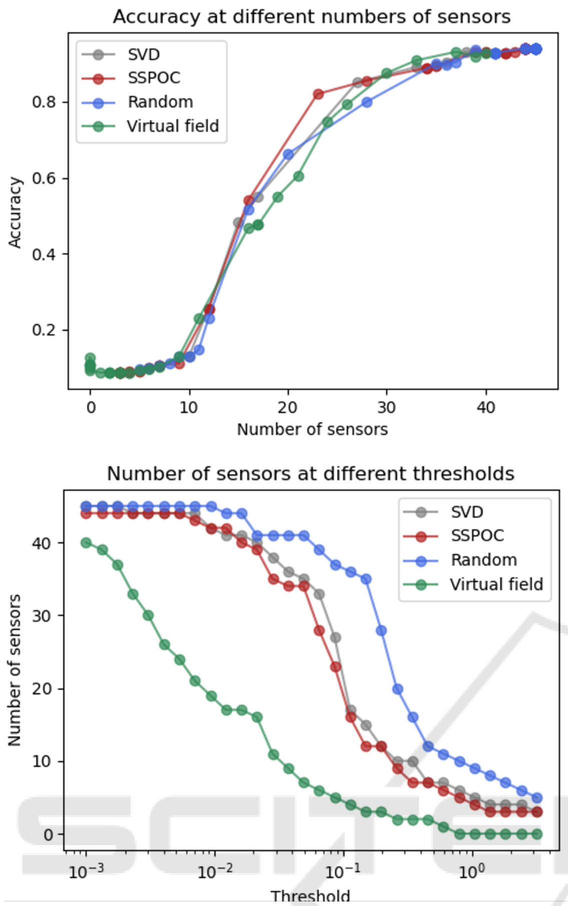



Figure 7: Gain in accuracy using virtual sensing field in comparison to state of art SSPOC and SVD methods.

serve the topological gain in obtaining a better estimator through linking sensors of similar type rather than constraining to floor-specific only.

- The system, in a nutshell, segregates sensor data stream into more intricate and more straightforward predictable patterns. The procedure shows a lifelong re-calibration strategy to affirm the intuition that placing sensors mostly at places with low virtualization capacity can provide 100 % coverage with less than 10 % error.

For example, the behavior of a group of temperature sensors situated across multiple zones can probably be learned by an optimal fraction of embedded devices. For example, a sensor with a power rating of 50 watts consumes $0.05 \times 365 \times 24 = 438$ units yearly. Now imagine 100 such operating sensors, therefore needing, $43,800 kWh$ of energy annually. One can argue about lowering the energy need by powering up a fraction of the sensors only.

## 6 CONCLUSION

In this paper, we demonstrate that, according to a general methodology, too many sensors are usually deployed in buildings. Thus, this work emphasizes the utility of spatiotemporal knowledge in bringing down the operating cost of building management systems. With explainable insights, the missing sensor approximation can be kept competitively accurate with bidirectional power-ambiance converters. The extension of the work can be studying the Utopian sensor placement across zones with theoretical learning guarantees.

Future works include the following insights. First, evaluating the model drift in an online learning setting is a benefit, which can be the next step toward auto-updating spatiotemporal models. Second, the experimental results call for another way to deploy sensors in a building. As part of a sustainable approach to reducing the number of sensors, a facility undergoing renovation could be temporarily equipped with sensors, according to the "sensors everywhere" methodology, to understand the uses of the building. Then, thanks to our methods, we can list the sensors that are in excess, which can be dismantled, and then redeployed in another building under renovation.

Thirdly, in a slightly orthogonal way, we could imagine physically deploying a small number of sensors in a building renovation and then introducing virtual sensors behaving like the sensors next to them. This information increase would allow us to study whether the sensor is essential to the building model or whether we can do without it. In this context, one can utilize temporal graph neural networks to capture the dynamics between rooms or between a room and a sensor.

## ACKNOWLEDGEMENTS

This work has been partially supported by the Multidisciplinary Institute on Artificial Intelligence (MIAI) at Grenoble Alpes (ANR-19-P3IA-0003) and the Resource manager for the Cloud of Things project (Greco – ANR-16-CE25-0016). Angan Mitra is supported by a convention CIFRE-2018/0874 with ANRT.

## REFERENCES

Ayenu-Prah, A. and Attoh-Okine, N. (2010). A criterion for selecting relevant intrinsic mode functions in empirical mode decomposition. *Advances in Adaptive Data Analysis*, 2(01):1–24.

Barbosh, M., Singh, P., and Sadhu, A. (2020). Empirical mode decomposition and its variants: a review with applications in structural health monitoring. *Smart Materials and Structures*, 29(9):093001.

Brunton, B. W., Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2013). Optimal sensor placement and enhanced sparsity for classification. *arXiv preprint arXiv:1310.4217*.

Chen, Z. and Liu, B. (2016). Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(3):1–145.

Clark, E., Askham, T., Brunton, S. L., and Kutz, J. N. (2018). Greedy sensor placement with cost constraints. *IEEE Sensors Journal*, 19(7):2642–2656.

de Silva, B. M., Manohar, K., Clark, E., Brunton, B. W., Brunton, S. L., and Kutz, J. N. (2021). Pysensors: A python package for sparse sensor placement. *arXiv preprint arXiv:2102.13476*.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

Emmanuel, C., Romberg, J., and Tao, T. (2005). Stable signal recovery from incomplete and inaccurate measurements.

Fontugne, R., Ortiz, J., Culler, D., and Esaki, H. (2012). Empirical mode decomposition for intrinsic-relationship extraction in large sensor deployments. In *Workshop on Internet of Things Applications, IoT-App*, volume 12.

Garg, V. and Bansal, N. K. (2000). Smart occupancy sensors to reduce energy consumption. *Energy and Buildings*, 32(1):81–87.

Gong, Z., Cui, Q., Chaccour, C., Zhou, B., Chen, M., and Saad, W. (2021). Lifelong learning for minimizing age of information in internet of things networks. In *ICC 2021-IEEE International Conference on Communications*, pages 1–6. IEEE.

Hojjati, S. N. and Khodakarami, M. (2016). Evaluation of factors affecting the adoption of smart buildings using the technology acceptance model. *International Journal of Advanced Networking and Applications*, 7(6):2936.

Hong, D., Ortiz, J., Whitehouse, K., and Culler, D. (2013). Towards automatic spatial verification of sensor placement in buildings. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pages 1–8.

Jia, M., Komeily, A., Wang, Y., and Srinivasan, R. S. (2019). Adopting internet of things for the development of smart buildings: A review of enabling technologies and applications. *Automation in Construction*, 101:111–126.

Ko, C.-W., Lee, J., and Queyranne, M. (1995). An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691.

Ma, Z., Badi, A., and Jorgensen, B. N. (2016). Market opportunities and barriers for smart buildings. In *2016 IEEE Green Energy and Systems Conference (IGSEC)*, pages 1–6. IEEE.

Ma, Z., Billanes, J. D., and Jørgensen, B. N. (2017). A business ecosystem driven market analysis: The bright green building market potential. In *2017 IEEE Technology & Engineering Management Conference (TEMSCON)*, pages 79–85. IEEE.

Manohar, K., Hogan, T., Buttrick, J., Banerjee, A. G., Kutz, J. N., and Brunton, S. L. (2018). Predicting shim gaps in aircraft assembly with machine learning and sparse sensing. *Journal of manufacturing systems*, 48:87–95.

Medeiros, D. R. d. S. and Fernandes, M. A. (2020). Distributed genetic algorithms for low-power, low-cost and small-sized memory devices. *Electronics*, 9(11):1891.

Mitra, A., Ngoko, Y., and Trystram, D. (2021). Impact of federated learning on smart buildings. In *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pages 93–99. IEEE.

Mitra, A., Thang, N. K., Nguyen, T.-A., Trystram, D., and Youssef, P. (2022). Online decentralized frank-wolfe: From theoretical bound to applications in smart-building. *arXiv preprint arXiv:2208.00522*.

Nirjon, S. (2018). Lifelong learning on harvested energy. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pages 500–501.

Pipattanasomporn, M., Chitalia, G., Songsiri, J., Aswakul, C., Pora, W., Suwankawin, S., Audomvongseree, K., and Hoonchareon, N. (2020). Cu-bems, smart building electricity consumption and indoor environmental sensor datasets. *Scientific Data*, 7(1):1–14.

Thrun, S. (1995). Lifelong learning: A case study. Technical report, Carnegie-Mellon Univ Pittsburgh pa Dept of Computer Science.

Umbarkar, A. J. and Sheth, P. D. (2015). Crossover operators in genetic algorithms: a review. *ICTACT journal on soft computing*, 6(1).

Wong, J. K., Li, H., and Wang, S. (2005). Intelligent building research: a review. *Automation in construction*, 14(1):143–159.

Xu, Y., Ahokangas, P., Turunen, M., Mäntymäki, M., and Heikkilä, J. (2019). Platform-based business models: Insights from an emerging ai-enabled smart building ecosystem. *Electronics*, 8(10):1150.

Yoganathan, D., Kondepudi, S., Kalluri, B., and Manthapuri, S. (2018). Optimal sensor placement strategy for office buildings using clustering algorithms. *Energy and Buildings*, 158:1206–1225.