



HAL
open science

Codesigned Communication and Data Analytics for Condition-Based Maintenance in Smart Buildings

Naveed Ahmad, Malcolm Egan, Jean-Marie Gorce, Jilles Dibangoye, Frédéric
Le Mouël

► **To cite this version:**

Naveed Ahmad, Malcolm Egan, Jean-Marie Gorce, Jilles Dibangoye, Frédéric Le Mouël. Codesigned Communication and Data Analytics for Condition-Based Maintenance in Smart Buildings. IEEE Internet of Things Journal, 2023, 10 (18), pp.15847-15856. 10.1109/JIOT.2023.3266029 . hal-04361059

HAL Id: hal-04361059

<https://hal.science/hal-04361059v1>

Submitted on 22 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Codesigned Communication and Data Analytics for Condition-Based Maintenance in Smart Buildings

Naveed Ahmad, Malcolm Egan, Jean-Marie Gorce, Jilles S. Dibangoye, and Frédéric Le Mouël

Abstract—With the proliferation of cheap sensors and the ubiquity of cloud and edge computing, predictive/condition-based maintenance is expected to play an important role in smart homes and buildings. Nevertheless, a key difficulty is ensuring that sensors provide data of sufficient quality in order to reliably detect building (e.g., heating system) degradation in systems or comfort. At the same time, sensor utilization should be limited as much as possible in order to minimize power consumption and increase the lifetimes of batteries. A solution to this problem requires careful codesign of sensor communication and data analytics. In this paper, we introduce a formulation of this codesign problem, which is based on an optimization problem to jointly design how often data is collected and compression levels in order to balance the quality of fault detection with the quantity of transmitted data. To solve the optimization problem, we apply a differentiable search algorithm based on a variant of stochastic gradient descent for discrete optimization problems. We apply our codesign framework and solve the resulting optimization problem using data obtained from a building comfort experiment known as the Twin House Experiment. We also provide an extension of our algorithm to a dynamic variant of the codesign framework, where comfort levels and power consumption penalties are time varying. Numerical results show that our algorithm rapidly finds an efficient tradeoff between classifier accuracy and sensor power consumption.

Index Terms—Codesign, Communications, Classification, Predictive Maintenance, Condition-based Maintenance

I. INTRODUCTION

A key challenge in buildings and homes is identifying degraded systems, ranging from heating, ventilation and air conditioning (HVAC) to water control. Predictive maintenance or Condition-Based Maintenance (CBM) aims to identify abnormal behavior potentially indicating a catastrophic failure in advance. The performance of condition-based maintenance depends on available sensors and the quality of the data analytics platform [1]. In buildings and homes, such failures can have severe human consequences; for example, a lack of heating in cold climates, or contamination of drinking water due to faulty water control.

With the increasing availability of low cost sensors and edge/cloud computing infrastructure for sophisticated data analytics, condition-based maintenance systems are becoming

more widespread and of a larger scale [2]. Moreover, often the sensors are not vendor specific; that is, the sensors are not installed directly within a system (e.g., HVAC), but only observe functionality indirectly through the impact of the system on physical observables (e.g., temperature or humidity) within the building.

On the other hand, as condition-based maintenance systems increase in scale, it is increasingly difficult to ensure that sensors are operational. Namely, for battery operated sensors connected wirelessly to the cloud via the Internet of Things (IoT), ensuring that each sensor has a long lifetime is critical. As energy consumption of sensors is often dominated by data collection and wireless transmission to an access point, the management of the predictive maintenance system must include strategies to minimize the collection and communication of sensor data.

A. Related Work

While there is now a vast literature on IoT-based predictive/condition-based maintenance, a common focus is on how to process data collected by wireless sensors. For example, algorithms have been developed for slitting machine monitoring [3], intelligent manufacturing [4], chiller monitoring [5], air compressors [6], gas turbines [7], renewable energy systems such as wind and hydro turbines [8] and personal care goods manufacturing [9]. In other work, communication aspects have been considered. For example, in [10] a messaging protocol for car maintenance was developed. In [11], an experimental study of power consumption and latency between sensors and gateways was carried out.

In the context of predictive maintenance for smart buildings the focus is on developing frameworks that can analyze or predict the potential faults in building systems [12]–[16]. A predictive maintenance framework along with implementation in buildings via an IoT device was discussed in [14]. The work in [15] discusses the predictive maintenance data framework and the devices used for implementation of the predictive maintenance systems. The work in [16] proposed a predictive maintenance system for buildings that is robust to faulty sensors. There are two key limitations of the work exemplified by [3], [5], [9]–[16]. First, the work focuses on the context of manufacturing, where the resulting data sets are not tailored to the sensor observations obtained within smart buildings.

N. Ahmad, M. Egan, J.-M. Gorce, J.S. Dibangoye, and F. Le Mouël are all with Univ Lyon, INSA Lyon, Inria, CITI. Email: {naveed.ahmad,jean-marie.gorce,jilles.dibangoye,frédéric.le-mouël}@insa-lyon.fr, malcolm.egan@inria.fr

Second, *either* data analytics *or* communication aspects are considered, but not both. As such, inefficiencies can be introduced where unnecessary power consumption arises due to excessive data collection or communication. In order to address both data analytics and communication for condition-based maintenance in smart buildings, a *codesign* approach is required.

At present, codesign of communication and data analytics has largely been addressed only in the context of predictive maintenance for manufacturing equipment. In particular, the work in [17] considered tradeoffs between energy consumption and the accuracy of the classifier for faults in a generic manufacturing context.

On the other hand, existing work in the context of smart buildings does not address optimization of sampling rates and compression of sensor data, which are the key sources of energy consumption in low cost sensors. For example, in [18] an error correction model and independent principle component analysis was applied to elevator management. In [19], feature selection was optimized in order to minimize building energy consumption in smart homes. In both cases, sensor power consumption was not considered. In [20], an autoencoder deep neural network was introduced for the purpose of classifying anomalies in HVAC installations in sport facility buildings. In [21], a variational autoencoder was utilized to detect power faults in electrical appliances (e.g., dishwashers, washing machines, and dryers) based on an aggregate power signal at a mains meter. Various other classification approaches based on simulation and regression methods have been proposed in [22]–[24].

Outside of the context of predictive maintenance, there has been extensive work in the building literature on energy modeling [25], [26] and sensor placement [27]–[32]. The literature on placement of wireless sensors focuses on the position and number of sensors required for energy modeling of buildings. Both white box (physical models) and statistical/data based models are used to optimize the number of sensors inside a building. The white box models require domain knowledge, high level of expertise and must be adapted for each building. The data based methods circumvent the complexities of white box models; however extra care is required when analyzing/implementing the results. Overall the focus of the sensors placement work is more on improvement in building energy modeling and occupant comfort but less on predictive maintenance in buildings.

B. Contributions

In this paper, we propose a strategy to codesign communication and data analytics for condition-based maintenance of smart buildings. In particular, we study the problem of reliably classifying when any part of a building is at an uncomfortable

temperature based on low cost temperature sensors equipped with wireless communication capabilities. This contrasts with existing work in the condition-based maintenance and building literature, which does not adopt a codesign approach. The main contributions of this work are summarized as follows:

- We formalize codesign of communication and data analytics in smart buildings for a network of low cost sensors that are capable of wirelessly communicating with the cloud via a single access point. In particular, we introduce a bi-level optimization problem, where both sensor configurations (sensor sampling rates, data compression, and sensor activity) and the classifier are optimized. The objective of the bi-level optimization problem incorporates a tradeoff between energy consumption and classifier accuracy via penalty parameters.
- We solve the bi-level optimization problem via a differentiable search procedure based on stochastic gradient descent, which bears a relation to architecture search algorithms in deep neural networks [33]. The algorithm is evaluated on a use case with experimentally obtained temperature measurements from the Twin House Experiment [34]. Numerical results show rapid convergence within 10 iterations to sensor configurations that yield a high accuracy and low battery power consumption.
- In real scenarios, the definition of a comfortable temperature may vary over time depending on the individuals present in the building or the outside temperature. Moreover, as sensors age, energy consumption may be increasingly important. In this setting, it is necessary to adapt the sensor configurations (e.g., data compression and utilized sensors) over time. To address this problem, we generalize the codesign problem to account for variations in comfortable temperatures and power consumption penalties. To solve the resulting optimization problem, we tune the parameters in the differentiable search algorithm to adapt to the changes in comfort levels or power consumption penalties. The algorithm is validated via the Twin House Experiment dataset [34].

The remainder of the paper is organized as follows: in Sec. II, we develop a framework to formalize the codesign problem, incorporating the performance of the classifier as well as power consumption. In Sec. III we introduce our differentiable stochastic search algorithm to solve the codesign problem. In Sec. IV, we evaluate our framework and algorithm on data obtained from the Twin House Experiment [34]. In Sec. V, we consider time-varying sensor configurations in response to changes in comfort levels and power consumption penalties. In Sec. VI, we conclude.

II. PROBLEM FORMULATION

Consider a building operated by a data service provider and equipped with N low-cost and battery-operated sensors that

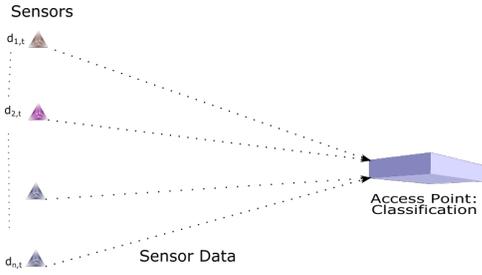


Fig. 1: Overview of the sensing and communication infrastructure for condition-based maintenance. Sensors collect data and, depending their configuration, compress and communicate the data to an access point. The access point identifies anomalies via a classifier tailored to the sensor configurations.

transmit data to an access point via wireless links, as illustrated in Fig. 1. The purpose of the sensor network is to identify degradation of systems in the building. In general, the sensors may be utilized to detect degradation in a range of systems; e.g., HVAC, lighting or water.

A. Sensor Configurations

Each sensor j is capable of collecting and communicating data every $T_c^j \in \mathcal{T}_c$ seconds, where the data is assumed to be a scalar quantity and $|\mathcal{T}_c| < \infty$. That is, every T_c^j seconds, each sensor can collect a single sample of a physical observable (e.g., temperature, humidity, light intensity) and send it to the access point. Sensor j is also capable of adapting the duration of each data transmission, with transmissions consisting of $n^j \in \mathcal{N}$ bits with $|\mathcal{N}| < \infty$.

Each bit has an energy cost of E_b Joules. As such, the average power consumption of sensor j is given by

$$P_j(n^j, T_c^j) = E_b \cdot n^j \cdot T_c^j \text{ Joules/sec.} \quad (1)$$

Observe that the power consumption P_j depends on both the duration of the transmission and how often transmissions occur. In low power wireless sensors, such as those in LoRa networks, power consumption arises from a range of functions; e.g., wake-up, data processing, data transmission, and sleep [35]. To incorporate these aspects, as in [35], we view E_b as the energy cost *per useful bit*. Alternatively, a more complex power consumption model could be utilized, possibly incorporating multiple hops, as in [36].

The duration of a transmission, n^j , is constrained by the level of data compression at sensor j . In particular, if the data packet consists of n information bits, then at most 2^n quantization levels are available. For example, if $n = 2$, and the scalar data samples lie in the interval $[0, 1]$, then all samples may be quantized to an element of $\{0, 1/3, 2/3, 1\}$.

More generally, if the data samples lie in an interval $\mathcal{I} = [d_{\min}, d_{\max}]$ of length $L = d_{\max} - d_{\min}$ and 2^n quantization levels are available, the quantization levels are separated by a distance

$$\Delta_j = \frac{L}{2^n - 1}. \quad (2)$$

Given a data sample $d_{t,j} \in \mathcal{I}$ at time t , sensor j transmits the compressed data

$$\hat{d}_{t,j} = \min_{i=0,1,\dots,2^n-1} |d_{t,j} - d_{\min} + i\Delta_j|. \quad (3)$$

If all sensors are utilized, then the total transmit power is given by $\sum_{j=1}^N P_j(n^j, T_c^j)$. However, over a given time period, not all the sensors may provide useful information. For example, when the sensors are observing the temperature of a building, nearby sensors may provide redundant information to the access point. In this case, it is desirable for the sensors to not be active in order to preserve their battery. To each sensor j , an activity variable $x_j \in \{0, 1\}$ is assigned, which indicates the sensor is off when $x_j = 0$. The total power is then defined as

$$P_{\text{tot}}(\mathbf{x}, \mathbf{n}, \mathbf{T}_c) = \sum_{j=1}^N x_j P_j(n^j, T_c^j), \quad (4)$$

where $\mathbf{x} = (x_1, \dots, x_N)$, $\mathbf{n} = (n^1, \dots, n^N)$, and $\mathbf{T}_c = (T_c^1, \dots, T_c^N)$.

B. Access Point Configuration

Given data available at the access point, the key problem is to identify whether or not it is anomalous. In particular, suppose at time t that sensor j transmits compressed data $\hat{d}_{t,j}$. The access point is then assumed to have an error-free observation of $\hat{\mathbf{d}}_t = (\hat{d}_{t,1}, \dots, \hat{d}_{t,N})$, where $\hat{d}_{t,j} = 0$ if sensor j is not active (i.e., $x_j = 0$).

Based on $\hat{\mathbf{d}}_t$, the access point makes a decision as to whether or not an anomaly is present. This is achieved via a classifier function $\Phi_{\theta} : \mathbb{R}^N \rightarrow \{1, \dots, M\}$, $\hat{\mathbf{d}}_t \mapsto \ell_t$, with parameters $\theta \in \Theta$, ℓ_t corresponding to the classifier label at time t , and M the number of system states ($M = 2$ for a single “normal” state and a single “anomalous” state). In particular, if $\ell_t = \Phi_{\theta}(\hat{\mathbf{d}}_t) \neq 1$, then an anomaly is detected.

In practice, given the functional form of the classifier, the parameter θ is optimized via a labeled training dataset $\{(\mathbf{d}_t, \ell_t)\}_{t=1}^{K_{\text{train}}}$ consisting of K_{train} samples. The framework in this paper is applicable to any supervised classification scheme; e.g., multinomial logistic regression, support vector machines and neural networks.

In particular, consider the multinomial logistic regression classifier with parameter $\theta = (\theta_1, \dots, \theta_M) \in \mathbb{R}^{N \times M}$. For

a data sample $\hat{\mathbf{d}}_t$, this classifier yields an estimate of the label $\hat{\ell}_t \in \{1, \dots, M\}$ given by

$$\hat{\ell}_t = \arg \max_{m=1, \dots, M} \boldsymbol{\theta}_m \cdot \hat{\mathbf{d}}_t. \quad (5)$$

In order to train $\Phi_{\boldsymbol{\theta}}$, a training data set $\{\mathbf{d}_{\text{train}, i}\}_{i=1}^{K_{\text{train}}}$ is used. The parameter $\boldsymbol{\theta}$ is obtained by minimizing the regularized empirical risk, given by

$$\boldsymbol{\theta} = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^{N \times M}} -\frac{1}{K_{\text{train}}} \sum_{t=1}^{K_{\text{train}}} \sum_{m=1}^M \mathbf{1}\{\ell_t = m\} \cdot \log \left(\frac{\exp(\tilde{\boldsymbol{\theta}}_m \cdot \mathbf{d}_{\text{train}, t})}{\sum_{k=1}^M \exp(\tilde{\boldsymbol{\theta}}_k \cdot \mathbf{d}_{\text{train}, t})} \right) + \frac{\gamma}{2} \|\tilde{\boldsymbol{\theta}}\|_F^2, \quad (6)$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\gamma \geq 0$ is the regularization parameter.

Associated with the classifier $\Phi_{\boldsymbol{\theta}}$ is an accuracy metric, $P_{\text{acc}}(\mathbf{x}, \mathbf{n}, \mathbf{T}_c)$, which is related to the probability that an anomaly is detected by the classifier. In practice, the accuracy is estimated with the aid of a test dataset consisting of K_{test} samples and is given by the F_1 -score, a number that gives a balanced accuracy score by taking into account both the false alarms (false positives) and the anomalies missed (false negatives). More precisely, the F_1 score is defined by

$$P_{\text{acc}}(\mathbf{x}, \mathbf{n}, \mathbf{T}_c) = \frac{1 - \alpha(\mathbf{x}, \mathbf{n}, \mathbf{T}_c)}{1 - \alpha(\mathbf{x}, \mathbf{n}, \mathbf{T}_c) + \frac{1}{2}(\alpha(\mathbf{x}, \mathbf{n}, \mathbf{T}_c) + \beta(\mathbf{x}, \mathbf{n}, \mathbf{T}_c))}, \quad (7)$$

where $\alpha(\mathbf{x}, \mathbf{n}, \mathbf{T}_c)$ and $\beta(\mathbf{x}, \mathbf{n}, \mathbf{T}_c)$ are the proportion of false positive and false negatives for the configuration $(\mathbf{x}, \mathbf{n}, \mathbf{T}_c)$, respectively, obtained from the test dataset.

C. Codesign Optimization Problem

The main focus of this paper is to address a codesign problem where the tradeoff between the energy consumption of the sensors and classification error is optimized. In particular, we formalize the codesign problem as

$$(\mathbf{x}^*, \mathbf{T}_c^*, \mathbf{n}^*) = \arg \max_{(\mathbf{x}, \mathbf{T}_c, \mathbf{n}) \in \{0, 1\}^N \times \mathcal{T}_c \times \mathcal{N}} -\lambda_1 P_{\text{tot}}(\mathbf{x}, \mathbf{n}, \mathbf{T}_c) + \lambda_2 P_{\text{acc}}(\mathbf{x}, \mathbf{T}_c, \mathbf{n}), \quad (8)$$

where $\lambda_1, \lambda_2 > 0$ are penalty parameters. Note that $\mathbf{x} \in \{0, 1\}^N$ and \mathbf{T}_c, \mathbf{n} lie in finite sets. As such, the problem is combinatorial.

We highlight that this problem is a bi-level optimization problem due to the fact that the computation of P_{acc} relies on the evaluation of the classifier on a test dataset. In particular, the classifier parameter $\boldsymbol{\theta}$ must be optimized based on a dataset consisting of only the data available to the access point, which is dependent on the sensor selection, the compression level, and the sampling rate. We also note that the optimization

procedure to obtain $\boldsymbol{\theta}$ is dependent on the choice of the classifier. In the case of the logistic classifier, parameter optimization involves solving (6).

In operation, the protocol utilized by the data service provider to optimize the system and classify new observations is summarized in Alg. 1. The remainder of this paper is concerned with efficient solutions to the problem in (8).

Algorithm 1 Classifier and Sensor Configuration Optimization and Deployment

- 1: The data service provider collects K_{train} labeled samples.
 - 2: The data service provider optimizes the sensor configuration and classifier via the codesign problem in (8).
 - 3: Sensors and AP are configured to collect, communicate, and classify observations. In particular, the AP transmits the sensor configurations to each sensor via a dedicated control link.
-

III. CODESIGN VIA DIFFERENTIABLE STOCHASTIC SEARCH

In this section, we develop an algorithm to solve the bi-level optimization problem in (8). To this end, suppose that the set of feasible configurations $\mathcal{C} = \{(\mathbf{x}, \mathbf{n}, \mathbf{T}_c)\} = \{0, 1\}^N \times \mathcal{T}_c \times \mathcal{N}$ has a cardinality of S . We denote the i -th element of the set of feasible configurations by $\mathbf{c}(i)$.

Although the optimization problem in (8) is discrete, it can be approximated by a smooth problem as follows. Let $\boldsymbol{\alpha} \in \mathbb{R}_+^S$ and consider the optimization problem

$$\boldsymbol{\alpha}^* = \arg \max_{\boldsymbol{\alpha} \in \mathbb{R}_+^S} \sum_{i=1}^S f(i) \frac{\alpha_i}{\sum_{j=1}^S \alpha_j}, \quad (9)$$

where

$$f(i) = -\lambda_1 P_{\text{tot}}(\mathbf{c}(i)) + \lambda_2 P_{\text{acc}}(\mathbf{c}(i)). \quad (10)$$

Note that in order to compute $P_{\text{acc}}(\mathbf{c})$ it is necessary to train the classifier $\Phi_{\boldsymbol{\theta}}$. An approximate solution to (8) can then be obtained via

$$i^* = \arg \max_{i=1, \dots, S} \alpha_i. \quad (11)$$

While the problem in (9) can in principle be solved by gradient descent, a drawback is that when S is large there are many terms in the sum. As such, computing each gradient is computationally expensive. Moreover, the computation of $f(i)$ requires optimization of the classifier to account for the data available to the access point under configuration $\mathbf{c}(i)$.

This issue can be resolved by noting that

$$\boldsymbol{\alpha}^* = \arg \max_{\boldsymbol{\alpha} \in \mathbb{R}_+^S} \frac{1}{S} \sum_{i=1}^S f(i) \frac{\alpha_i}{\sum_{j=1}^S \alpha_j}, \quad (12)$$

which can be viewed as a stochastic optimization problem. In particular,

$$\alpha^* = \arg \max_{\alpha \in \mathbb{R}_+^S} \mathbb{E}_I \left[f(I) \frac{\alpha_I}{\sum_{j=1}^S \alpha_j} \right], \quad (13)$$

where the random variable I is uniformly distributed on $\{1, \dots, S\}$. We remark that a similar observation is utilized for other discrete optimization problems in machine learning, such as deep neural architecture search [33].

A standard method to solve stochastic optimization problems of the form in (13) is projected stochastic gradient descent. Let

$$\mathcal{H} = \{\alpha \in \mathbb{R}^N : \alpha_i \geq 0, i = 1, 2, \dots, S\}. \quad (14)$$

In particular, as detailed further in Algorithm 2, the parameter α is updated recursively via

$$\alpha_{t+1} = \Pi_{\mathcal{H}} \left\{ \alpha_t + \epsilon_{t+1} f(i_{t+1}) \nabla_{\alpha} \frac{\alpha_{i_{t+1}}}{\sum_{j=1}^S \alpha_j} \Big|_{\alpha=\alpha_t} \right\}, \quad (15)$$

where $t \geq 0$, α_0 is an initialization of the iterates, $\{i_t\}$ are independent random variables with each uniform on $\{1, \dots, S\}$, and $\{\epsilon_t\} \subset \mathbb{R}_+$ is a positive step-size sequence. The operator $\Pi_{\mathcal{H}}\{\cdot\}$ is a metric projection onto \mathcal{H} ; i.e., $\pi_{\mathcal{H}}\{\alpha\}$ outputs the closest point (with respect to the Euclidean distance) to α in \mathcal{H} .

Algorithm 2 Differentiable Stochastic Search Algorithm

Input: $\alpha_0 \in \mathbb{R}_+^S$, $t = 0$, step-size sequence $\{\epsilon_t\}$.

Output: Configuration i^* .

- 1: **while** Not Stopped **do**
- 2: Sample i_{t+1} uniformly from $\{1, \dots, M\}$.
- 3: Compute

$$\alpha_{t+1} = \Pi_{\mathcal{H}} \left\{ \alpha_t + \epsilon_{t+1} f(i_{t+1}) \nabla_{\alpha} \frac{\alpha_{i_{t+1}}}{\sum_{j=1}^S \alpha_j} \Big|_{\alpha=\alpha_t} \right\},$$

where $f(i)$ is defined in (10), computed using the K_{train} training samples.

- 4: $t \rightarrow t + 1$.
 - 5: **end while**
 - 6: **return** $i^* = \arg \max_{i=1, \dots, S} \alpha_{t,i}$.
-

IV. VALIDATION IN THE TWIN HOUSE EXPERIMENT USE CASE

A key problem in building operation is to maintain thermal comfort. The conditions for thermal comfort are determined by variables such as air temperature, humidity, air velocity, and mean radiant temperature. The indoor air temperature typically takes precedence over the other parameters and it is generally accepted that a standard range should be maintained inside the building.

In particular, the American Society of Refrigeration and Air Condition (ASHRAE) and International Standards Organization (ISO) publish comfort standards that should be maintained inside buildings [37]. ASHRAE suggests that buildings should be at 22 °C in winters and 24.5 °C in summers with an allowable fluctuation range of [20 – 24] °C for winters and [22 – 26] °C for summers. In order to maintain comfort, it is necessary to detect when the temperature of the building lies outside the comfortable range; i.e., the operation of the heating system in the building is faulty.

In this section, we evaluate our codesign methodology and Algorithm 2 based on a dataset from the Twin House Experiment [34]. In the remainder of this section, we present a description of this use case and numerical results validating our approach.

A. Use Case Description

The Twin House Experiment [34] was performed on two stand alone houses installed with a number of sensors in each room. Each house consisted of a living room, a kitchen, a children’s room, a bedroom, a doorway, a corridor, an attic, and a basement. The experiments involved heating the houses at different levels and measuring variables such as the indoor temperature evolution, the power consumption, humidity, and heat losses from the walls. A weather station near the buildings was used to measure covariates including solar radiation, outdoor temperature, and wind speed.

The experiment was conducted for a period of 41 days, with data recorded every 10 minutes. The 41 day interval is subdivided as follows:

- Days 1-10:** An initialization period of 10 days where the temperature was kept constant at 30 °C;
- Days 10-25:** A 15 day period with random order logarithmic binary sequence heating input;
- Days 25-31:** A 7 day period of keeping temperature constant at 25 °C;
- Days 31-41:** A 10k day period of free floating temperature with no heating input.

These measurements (from the indoor sensors and weather station) along with the description of the houses are provided in [38] and are used to train, test and validate building energy and thermal characterization models [39].

The temperature measurements from several of the rooms are illustrated in Fig. 2. Observe that for the same house there are often significant differences in the measurements from each room. This suggests that certain sensors may be more important than others in classification tasks.

The observation that some sensors may be more important concerning the anomaly detection is reinforced in Fig. 3, which

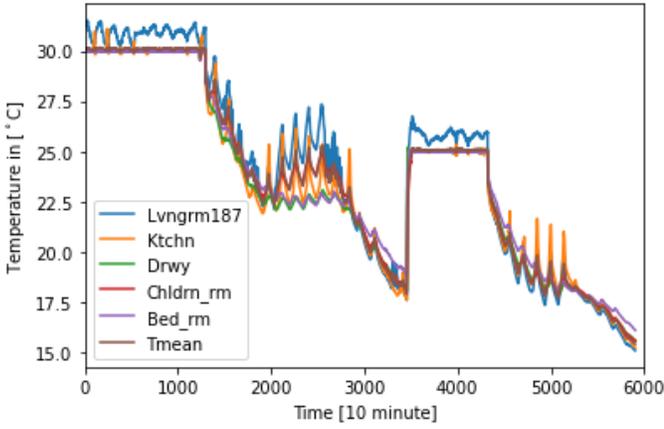


Fig. 2: The temperature evolution in different rooms of the Twin House. The data corresponding to Lvngm187, Ktchn, Drwy, Chldrm, Bedrm are the temperature data from the living room, kitchen, doorway and bedroom temperature sensors, respectively. Tmean is the average temperature of all rooms.

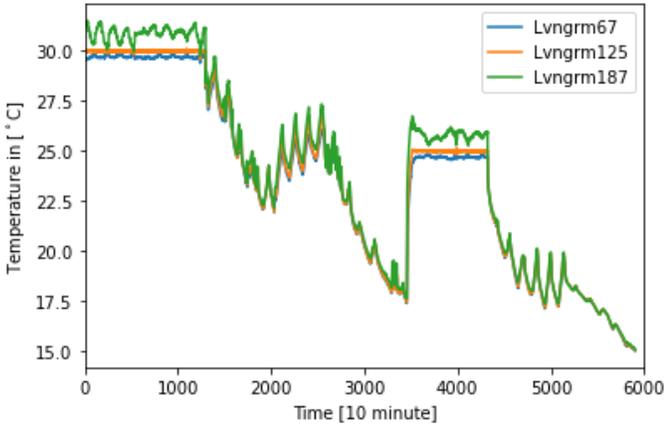


Fig. 3: Temperature measured in the same room with sensors at different heights in living room of the Twin House. Lvngm67, Lvngm125, Lvngm187 is the temperature data from the sensors at the height of 1.25 m, 1.67 m and 1.87 m, respectively [39]).

shows measurements from three sensors in the living room, but at different heights. Again, the sensors measurements have quite different behavior even though they are in the same area.

To validate our codesign framework, it is necessary to label the data in the Twin House Experiment dataset. To do so, we define an anomaly when the temperature measurement of *any* sensor in the house lies outside a predefined comfort range. In particular, the house is in “Hot” or “Too Hot” if any of the sensors record a measurement in these temperature ranges. For the purposes of the numerical study, the comfort zone i.e., “normal”, “hot”, and “too hot” ranges were defined as:

“Normal”: for any $T < 26$ °C,

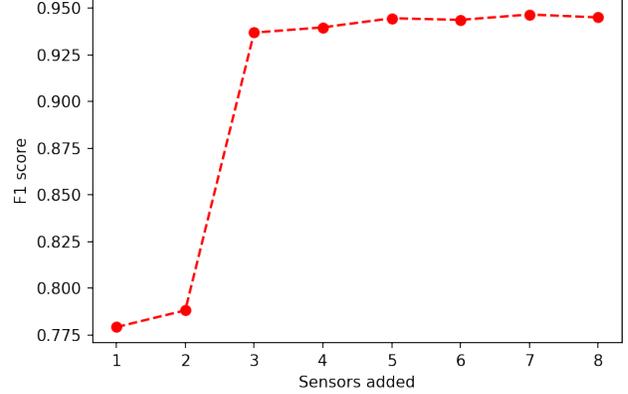


Fig. 4: Accuracy with increasing number of sensors.

“Hot”: for any 26 °C $\leq T \leq 30$ °C,
 “Too Hot”: for any $T > 30$ °C.

As a consequence, there are $M = 3$ classifier states. For the purposes of condition monitoring, we are interested when the building temperature is abnormal, but not “too hot” as this is the appropriate time to adapt the heating system. As a consequence, an anomaly is defined as when the building temperature is in the “hot range”. As “hot” temperatures arise rarely compared with the “normal” and “too hot” temperatures, it also forms a valuable test to ensure that the algorithm can reliably identify relatively rare anomalies.

In the sequel, we apply Alg. 2 with a constant stepsize (i.e., $\epsilon_t = \epsilon = 10$, $\forall t$) to solve the codesign problem in (8) for the Twin House Experiment dataset. In the following numerical results, training samples for the classifier are randomly selected from the 41 days duration with 5905 data points for each of the 8 sensors installed in different rooms in the building.

B. Numerical Results

In order to verify that the selection of sensors has a significant impact on the performance, we first examine how varying the number of sensors affects the F_1 -score. An exhaustive search is used to observe the impact of adding sensors. To train the classifier and compute the F_1 -score, the training and test split are 60% and 40%, respectively. In Fig. 4, the accuracy is plotted for a varying number of sensors. Observe that when one or two sensors are utilized there is a significant reduction in the F_1 -score compared with when data from all sensors is collected. These observations provide a strong justification for applying the codesign algorithm in Alg. 2 to search for high performance configurations.

To provide a baseline to evaluate Alg. 2, we considered a scenario consisting of 8 sensors with 4 different sampling rates

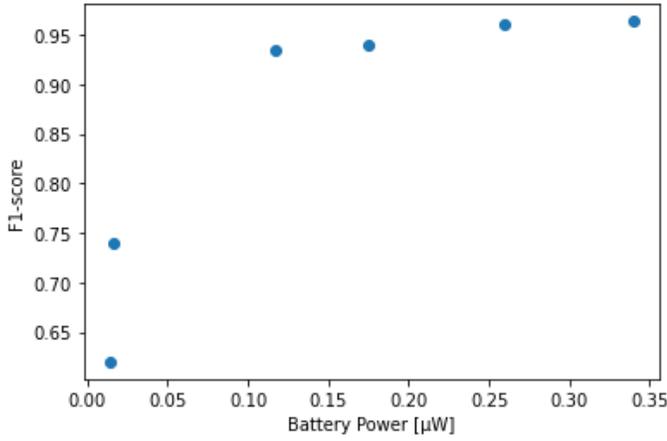


Fig. 5: Trade-off between accuracy and battery power consumption with optimized sensor configurations.

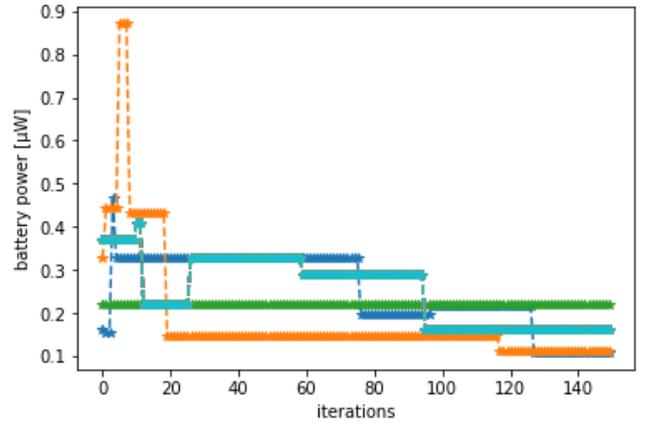


Fig. 7: Evolution of the battery power consumption with the number of iterations of the differentiable search algorithm with a constant step-size of 10. The different curves (colored) show the runs of the algorithm.

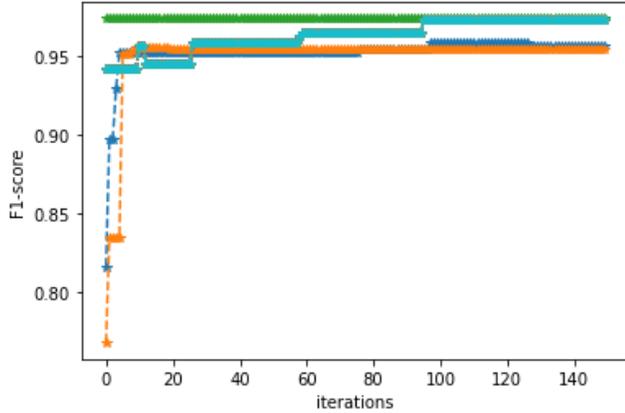


Fig. 6: Evolution of the F_1 score versus the number of iterations of the differentiable search algorithm with a constant step-size of 10. The different curves (colored) show the runs of the algorithm

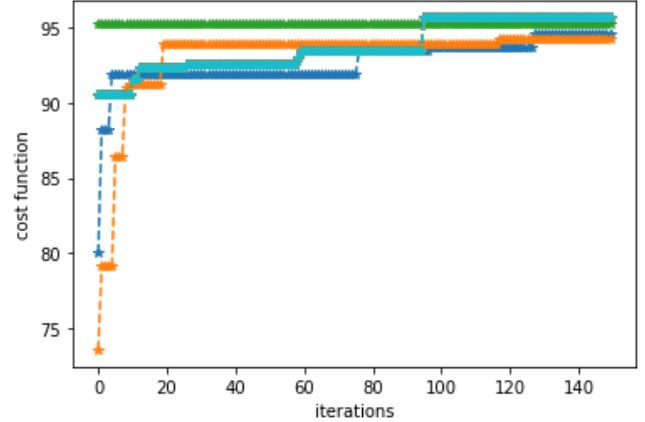


Fig. 8: Evolution of the total cost with the number of iterations of the differentiable search algorithm with a constant step-size of 10. The different curves (colored) show the runs of the algorithm.

and 4 data compression schemes, where each active sensor is provided with the same optimized values of the sampling rate T_c and duration n . This results in approximately 4080 possible configurations. In this case, it is feasible to perform an exhaustive search. In particular, Fig. 5 plots the optimal tradeoff between the F_1 -score and power consumption. Here, the different solutions were obtained by varying the penalty parameters λ_1 and λ_2 in (8). Observe that to achieve low power consumption levels, there is a significant reduction in the achievable F_1 -score.

The results in Fig. 4 and Fig. 5 are obtained through an exhaustive search of all possible communication strategies from a set of 4080 possible configurations. We now turn to the selection of communication strategy via Alg. 2, where the sensor configuration is optimized. In particular, solutions are computed based on 150 iterations of Alg. 2, which searches through

a combination of sampling rates [5905,4465,3025,1585], 8 sensors and 3 levels of data compression. As for the results in Fig. 4 and Fig. 5, the training and test split in the simulations are 60% and 40%, respectively.

Fig. 6, Fig. 7 and Fig. 8 plot the performance of Alg. 2 for varying numbers of iterations with $\lambda_1 = 10$ and $\lambda_2 = 100$. Observe that most trajectories rapidly converge to solutions with similar accuracy and battery power levels. Moreover, the selected configurations achieve a F_1 -score and power consumption level consistent with the optimal performance shown in Fig. 5.

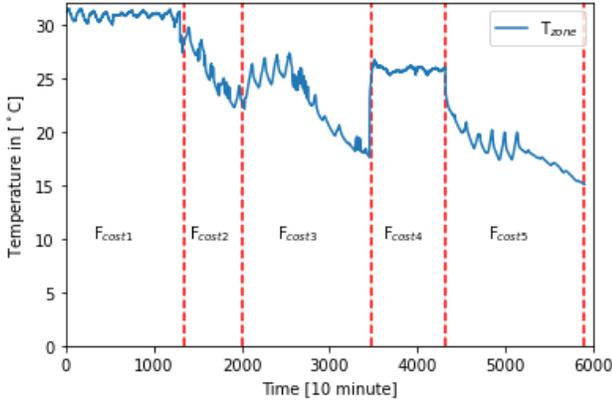


Fig. 9: The sectioning of twin house data in different zones (dashed vertical lines), each zone has a different cost function defined by penalties $\lambda_{t,1}$, $\lambda_{t,2}$ and different temperature limits for anomaly detection.

V. TIME-VARYING COMFORT LEVELS AND POWER COSTS

A. Problem Description

In Sec. III and Sec. IV, it was assumed that the classifier and sensor configurations did not vary over time after the initial optimization was performed. On the other hand, a comfortable temperature can vary between day and night or between seasons. As a consequence, the range of temperatures defining the labels “Normal”, “Hot”, “Too Hot” will also change over time. Similarly, the importance of power consumption in optimizing the sensor configurations may also vary; for example, as the sensor batteries approach the end of their lifetimes, it may be necessary to sacrifice some quality of the observations for an extended lifetime.

When label definitions and power costs vary over time, the classifier parameters and the sensor configurations must also be re-optimized. Moreover, it is highly desirable for a labeled data set to become available when this re-optimization is performed. The effect of time-varying label definition and power costs is that the objective in (10) now depends on time; that is,

$$f_t(i) = -\lambda_{t,1}P_{\text{tot}}(\mathbf{c}(i)) + \lambda_{t,2}P_{t,\text{acc}}(\mathbf{c}(i)), \quad (16)$$

where the penalties $\lambda_{t,1}$, $\lambda_{t,2}$ and the classifier accuracy $P_{t,\text{acc}}$ are time-dependent. An illustration of how the cost may change over time is illustrated in Fig. 9.

In particular, as illustrated in Fig. 10, we assume that a phase of sensor configuration optimization is fixed at K_{fixed} samples. Before this window, two sets of labeled samples of size K_{train} and K_{test} , respectively, are provided to the data service provider. The samples in K_{train} are utilized in order to optimize the classifier Φ_{θ} and the samples in K_{test} are used

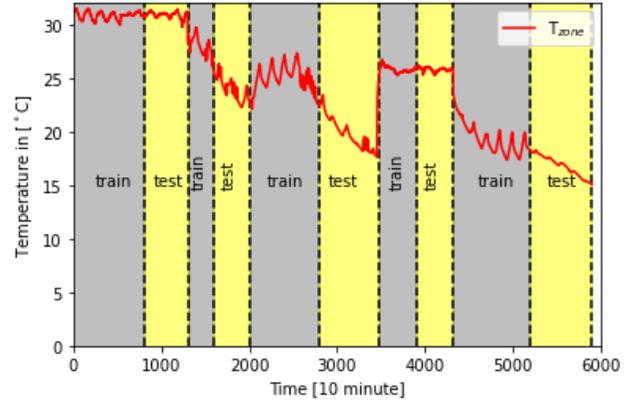


Fig. 10: The sectioning of twin house data into training and test data.

to optimize the sensor configuration. The resulting protocol is summarized in Alg. 3.

Algorithm 3 Time-Varying Classifier and Sensor Configuration Optimization and Deployment

- 1: The access point collects $K_{\text{train}} + K_{\text{test}}$ labeled samples.
 - 2: The access point optimizes the sensor configuration and classifier.
 - 3: Sensors and access point are deployed to collect, communicate, and classify data for K_{fixed} samples. For this period of K_{fixed} samples, the AP transmits the sensor configurations to each sensor via a dedicated control link.
 - 4: Repeat from Step 1.
-

A solution to the time-varying classifier and sensor configuration optimization problem is to apply Alg. 2 with a pre-defined initialization. For example, the configuration obtained from the previous window. However, this approach may be sensitive to the choice of step-size ϵ . Indeed, if the step-size is small, it will not be possible to quickly identify new solutions that are far from the previous one. On the other hand, if the step-size is too large, convergence to a good solution will be less likely. To address this problem, we performed a numerical study to investigate how the choice of the step-size impacts the selection of the sensor configuration. In the next section first we will show how the cost function changing with time impacts the performance of the Alg. 3 and how changing the values of step size changes the solutions.

B. Numerical Results

Anomaly detection with time varying comfort levels and penalty parameters as discussed in Alg. 3 is implemented by using data from the Twin House Experiment. In particular, the data is divided into 5 time zones. The communication policy is optimized in terms of the number of sensors and data compression level. Each period has different comfort levels

to classify data as “Normal”, “Hot” and “Too Hot”; i.e. the comfort levels for anomaly detection change between each data section, Table I. The cost function defined in (16) is adapted for each data zone by changing the penalties $\lambda_{t,1}$ and $\lambda_{t,2}$; i.e., the penalty for accuracy and battery power are different across each zone as detailed in Fig. 9 and Table I. Each zone is subdivided into training and test data as illustrated in Fig. 10.

Data zone	$\lambda_{t,1}$	$\lambda_{t,2}$	Anomaly detection temperature [C°]
zone-1	10	100	31
zone-2	80	20	28
zone-3	60	40	26
zone-4	40	60	27
zone-5	80	20	21

TABLE I: Comfort levels and penalty parameters for each data zone.

The key question is whether Alg. 3 rapidly identifies a good configuration in each data zone when the algorithm is initialized by the final configuration from the previous zone. To address this question, two scenarios are simulated in which the differentiable search algorithm is utilized with a step-size of 10.

In the first scenario the penalty parameters are kept constant ($\lambda_1 = 10$ and $\lambda_2 = 100$) whereas the temperature limits defining an anomaly are changing during each zone (over 30 iterations of the algorithm) as detailed in Table I. Fig. 11 plots the evolution of the cost over time in the presence of varying comfort levels and constant penalties. Observe that cost varies over time, showing that the algorithm is adapting to the changing cost function. Typically, the cost only varies immediately after a change in the zone, suggesting that the algorithm is rapidly converging. The relatively small cost in the third zone (iterations 60 to 90 in Fig. 11) is due to a low F_1 -score due to the small number of anomalies in this zone.

In the second scenario, both the penalty parameters (λ_1 and λ_2) and the comfort levels defining when an anomaly occurs vary between data zones. The F_1 -score, the battery power consumption and the cost function are plotted in Fig. 12, Fig. 13 and Fig. 14, respectively. As for the scenario of constant penalty parameters, the algorithm is converging rapidly within each data zone. Unlike the constant penalty scenario, the battery power consumption tends to vary less, suggesting that the same number of sensors are utilized during all data zones. The effect of varying penalty parameters is also visible in the F_1 -score, where sensors are all switched off due to the high penalty for battery power consumption.

The sensor configuration optimization may be sensitive to the selection of the step size. To investigate whether the step size value changes the solutions, simulations were performed at different values of the step-size ranging from $\epsilon = 0.01$ to

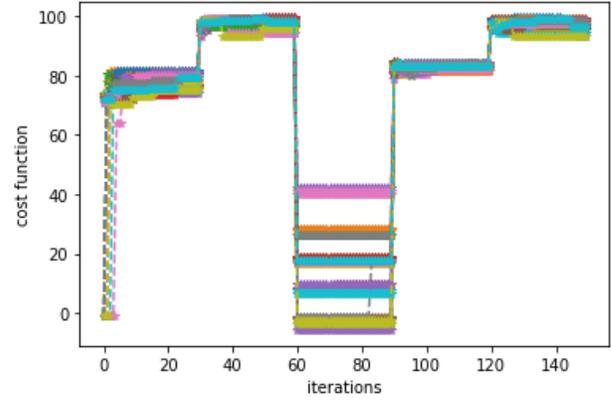


Fig. 11: Evolution of the cost function over time with varying comfort levels and constant penalties. The different curves (colored) show the runs of the algorithm

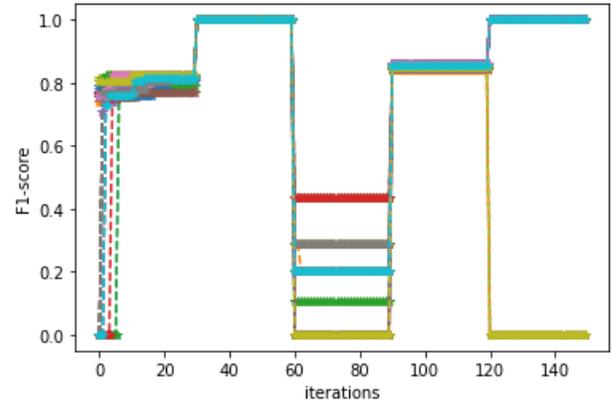


Fig. 12: Evolution of the F_1 -score over time with varying comfort levels and penalties. The different curves (colored) show the runs of the algorithm

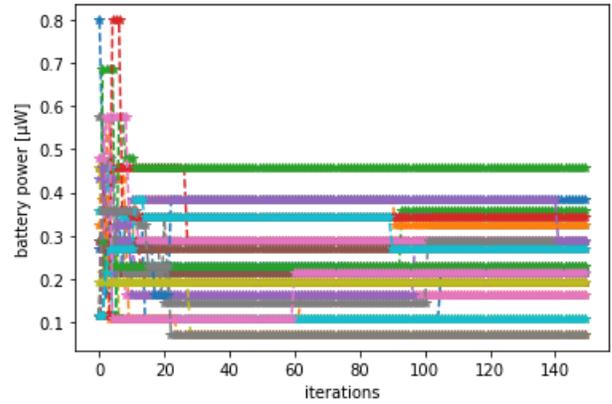


Fig. 13: Evolution of the battery power consumption over time with varying comfort levels and penalties λ_1 and λ_2 . The different curves (colored) show the multiple runs/realisations of the simulation.

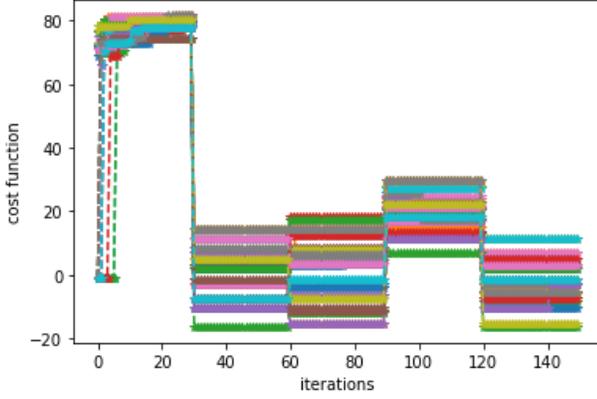


Fig. 14: Evolution of the cost function over time with varying comfort levels and penalties λ_1 and λ_2 . The different curves (colored) show the runs of the algorithm.

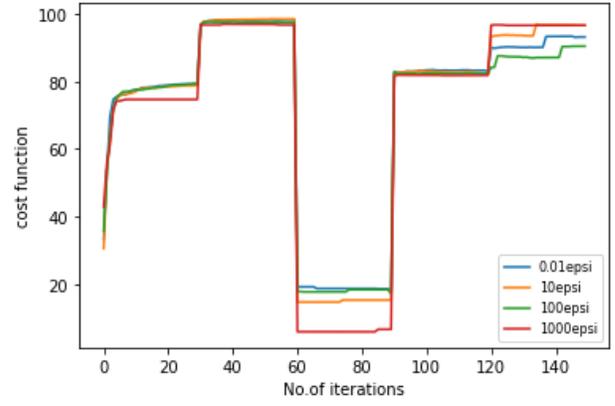


Fig. 17: The cost function for variable comfort levels and constant penalties averaged over 30 simulations for varying step-sizes.

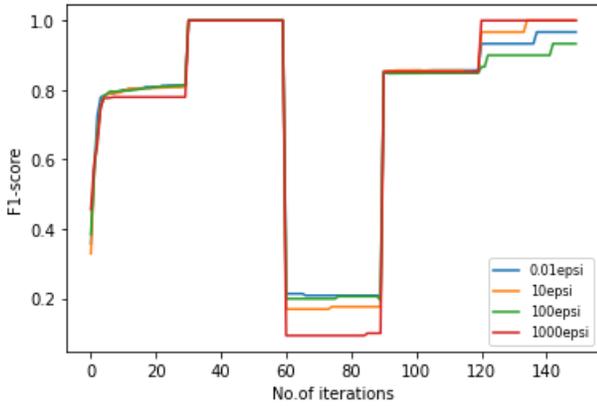


Fig. 15: The F_1 -score for variable set-point and constant penalties averaged over 30 simulations for varying step-sizes.

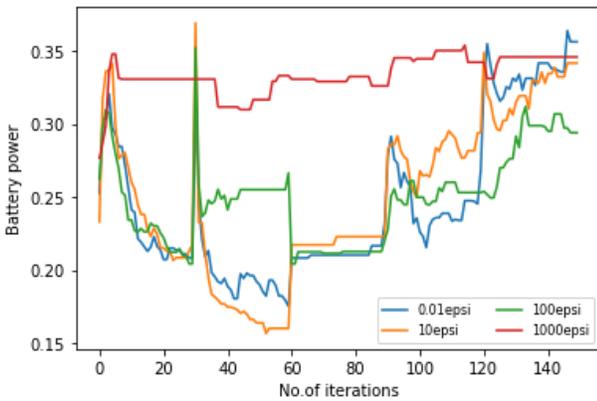


Fig. 16: The battery power for variable comfort levels and constant penalties averaged over 30 simulations for varying step-sizes.

$\epsilon = 1000$. In Fig. 15 for the F_1 -score and Fig. 17 for the cost function averaged over 30 simulations show that the results do not differ significantly for different step size values. However, Fig. 16 shows that the step size selection has a significant impact on the battery power. The effect of battery power is not visible from the cost function curves in Fig. 17 due to the small penalty ($\lambda_1 = 10$) for battery power and large reward for F_1 -score ($\lambda_2 = 100$). This suggests that there is a need for careful step-size optimization in this regime.

VI. CONCLUSIONS AND FUTURE WORK

A key challenge for smart buildings is to develop sensor networks that simultaneously provide data of a sufficient quality to allow for reliable classification of the degradation of heating system of the building, while also having long battery lifetimes. In this paper, we have addressed this challenge by formulating a codesign problem incorporating both communications and data analytics. Differentiable stochastic search algorithms were introduced both for static and time-varying comfort levels and power consumption penalties. The performance of the algorithms was verified in a use case based on data from the Twin House Experiment, which revealed that the algorithms rapidly identify efficient tradeoffs between classifier accuracy and power consumption. Simulation results show that the choice of the step-size can have a significant impact on the performance of the algorithm.

There are a number of avenues of future work. One avenue is to investigate the impact of different power consumption models, corresponding to specific low power sensor devices. Another avenue is to investigate the codesign problem with other classifiers (e.g., neural networks). While this is not expected to lead to significant improvements within the Twin House Experiment dataset, it is likely to be necessary in other use cases based on other data. A third avenue is to further

optimize the parameter section for the differentiable search procedure, particularly for time-varying comfort levels and energy penalties.

ACKNOWLEDGEMENTS

This work was supported by the INSA Lyon - SPIE ICS chairs on the Internet of Things and the Edge Artificial Intelligence. We also gratefully acknowledge support from the PSMN (Pôle Scientifique de Modélisation Numérique) of the Ecole Normale Supérieure de Lyon for the computing resources.

REFERENCES

- [1] R. Mobley, *An Introduction to Predictive Maintenance*. Woburn, MA: Elsevier Science, 2002.
- [2] T. Zonta *et al.*, "Predictive maintenance in the Industry 4.0: a systematic literature review," *Computers & Industrial Engineering*, vol. 150, 2020.
- [3] A. Kanawaday and A. Sane, "Machine learning for predictive maintenance of industrial machines using IoT sensor data," in *IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2017.
- [4] L. Wang and R. X. Gao, *Condition monitoring and control for intelligent manufacturing*. Springer Science & Business Media, 2006.
- [5] J. Cheng *et al.*, "Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms," *Automation in Construction*, vol. 112, 2020.
- [6] N. K. Verma, R. K. Sevakula, S. Dixit, and A. Salour, "Intelligent condition based monitoring using acoustic signals for air compressors," *IEEE Transactions on Reliability*, vol. 65, no. 1, pp. 291–309, 2015.
- [7] M. Tahan, E. Tsoutsanis, M. Muhammad, and Z. A. Karim, "Performance-based health monitoring, diagnostics and prognostics for condition-based maintenance of gas turbines: A review," *Applied energy*, vol. 198, pp. 122–144, 2017.
- [8] Y. Afridi, K. Ahmad, and L. Hassan, "Artificial intelligence based prognostic maintenance of renewable energy systems: A review of techniques, challenges, and future research directions," *International Journal of Energy Research*, vol. 46, 07 2021.
- [9] S. Ayvaz and K. Alpay, "Predictive maintenance system for production lines in manufacturing: a machine learning approach using IoT data in real-time," *Expert Systems with Applications*, vol. 173, 2021.
- [10] R. Dhall and V. Solanki, "An IoT based predictive connected car maintenance," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 4, no. 3, 2017.
- [11] F. Civerchia *et al.*, "Industrial Internet of Things monitoring solution for advanced predictive maintenance applications," *Journal of Industrial Information Integration*, vol. 7, pp. 4–12, 2017.
- [12] G. F. Huseien and K. W. Shah, "A review on 5g technology for smart energy management and smart buildings in singapore," *Energy and AI*, vol. 7, p. 100116, 2022.
- [13] F. Raco, M. Balzani, F. Planu, and A. Cittadino, "Inspire project: Integrated technologies for smart buildings and predictive maintenance," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 48, pp. 127–133, 2022.
- [14] A. Katona, P. Panfilov, and B. Katalinic, "Building predictive maintenance framework for smart environment application systems," in *Proceedings of the 29th DAAAM international symposium*, pp. 0460–0470, 2018.
- [15] C. Coupry, S. Noblecourt, P. Richard, D. Baudry, and D. Bigaud, "Bim-based digital twin and xr devices to improve maintenance procedures in smart buildings: A literature review," *Applied Sciences*, vol. 11, no. 15, p. 6810, 2021.
- [16] M. Lowin and C. Mihale-Wilson, "Towards predictive maintenance as a service in the smart housing industry," *INFORMATIK 2021*, 2021.
- [17] Y. Teoh, S. Gill, and A. Parlikad, "IoT and fog computing based predictive maintenance model for effective asset management in industry 4.0 using machine learning," *IEEE Internet of Things Journal*, 2021.
- [18] M. Razali *et al.*, "Big data analytics for predictive maintenance in maintenance management," *Property Management*, 2020.
- [19] S. Malik *et al.*, "A feature selection-based predictive-learning framework for optimal actuator control in smart homes," *Actuators*, vol. 10, no. 4, 2021.
- [20] Y. Bouabdallaoui *et al.*, "Predictive maintenance in building facilities: a machine learning-based approach," *Sensors*, vol. 21, no. 4, 2021.
- [21] M. Castangia *et al.*, "Anomaly detection on household appliances based on variational autoencoders," *Sustainable Energy, Grids and Networks*, vol. 32, 2022.
- [22] P. Gholami and A. Hafezalkotob, "Maintenance scheduling using data mining techniques and time series models," *Int. J. Manag. Sci. Eng. Manag.*
- [23] R.-Y. Kwak *et al.*, "Development of an optimal preventive maintenance model based on the reliability assessment for air-conditioning facilities in office buildings," *Build. Environ.*, vol. 39, pp. 1141–1156, 2004.
- [24] C. Mattera *et al.*, "A method for fault detection and diagnostics in ventilation units using virtual sensors," *Sensors*, vol. 18, p. 3931, 2018.
- [25] X. Li and J. Wen, "Review of building energy modeling for control and operation," *Renewable and Sustainable Energy Reviews*, vol. 37, pp. 517–537, 2014.
- [26] Y. Chen, M. Guo, Z. Chen, Z. Chen, and Y. Ji, "Physical energy and data-driven models in building energy prediction: A review," *Energy Reports*, vol. 8, pp. 2656–2671, 2022.
- [27] D. H. Tran, M. H. Nazari, A. Sadeghi-Mobarakeh, and H. Mohsenian-Rad, "Smart building design: a framework for optimal placement of smart sensors and actuators," in *2019 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, 2019.
- [28] A. D. Fontanini, U. Vaidya, and B. Ganapathysubramanian, "A methodology for optimal placement of sensors in enclosed environments: A dynamical systems approach," *Building and Environment*, vol. 100, pp. 145–161, 2016.
- [29] G. Suryanarayana, J. Arroyo, L. Helsen, and J. Lago, "A data driven method for optimal sensor placement in multi-zone buildings," *Energy and Buildings*, vol. 243, p. 110956, 2021.
- [30] "Optimal sensor placement strategy for office buildings using clustering algorithms," *Energy and Buildings*, vol. 158, pp. 1206–1225, 2018.
- [31] A. Guinard, A. McGibney, and D. Pesch, "A wireless sensor network design tool to support building energy management," in *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '09, (New York, NY, USA), p. 25–30, Association for Computing Machinery, 2009.
- [32] Z. Du, P. Xu, X. Jin, and Q. Liu, "Temperature sensor placement optimization for vav control using cfd-bes co-simulation strategy," *Building and Environment*, vol. 85, pp. 104–113, 2015.
- [33] H. Liu, K. Simonyan, and Y. Yang, "DARTS: differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.
- [34] M. Kersken and P. Strachan, "Twin house experiment IEA EBC annex 71 validation of building energy simulation tools-specifications and dataset," 2020.
- [35] T. Bouguera *et al.*, "Energy consumption model for sensor nodes based on LoRa and LoRaWAN," *Sensors*, vol. 18, no. 7, 2018.
- [36] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," in *IEEE SECON*, 2006.
- [37] *ANSI/ASHRAE Standard 55-2013: Thermal Environmental Conditions for Human Occupancy*. ASHRAE standard, ASHRAE, 2013.
- [38] Strachan, "Iea annex 58: Full-scale empirical validation of detailed thermal simulation programs," *Energy Procedia*, vol. 78, pp. 3288–3293, 2015.
- [39] C. Ghiaus and N. Ahmad, "Thermal circuits assembling and state-space extraction for modelling heat transfer in buildings," *Energy*, vol. 195, 2020.