2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

INFUSING DOMAIN KNOWLEDGE INTO GAUSSIAN PROCESS REGRESSION: PORTABLE PREDICTION MODEL FOR INDOOR TEMPERATURE

Lu Wan[1,2], Xiaobing Dai[3], Mai Chatani[1], Jan Gall[1], Ekaterina Petrova[2], and Pieter Pauwels[2]
[1] Advanced Energy Systems, Robert Bosch GmbH, Germany
[2] Department of the Built Environment, Eindhoven University of Technology, Netherlands
[3] Chair of Information-oriented Control, Technical University of Munich, Germany

Abstract

Accurate indoor temperature prediction is key to smart building control, yet standard Gaussian Process Regression (GPR) faces scalability and manual configuration challenges. We apply LoG-GP, a distributed GPR variant, to enable real-time model updates and fast predictions. Through backward selection and sensitivity analysis, we identify minimal input features across building instance types. We propose a metadata-driven workflow that automates feature selection and dataset retrieval. Tested on three building instances, our method maintains accuracy while reducing computational cost. This work advances GPR usability in real-world settings by combining scalability, automation, and domain knowledge for portable and efficient model deployment.

Introduction

For deploying model predictive control (MPC) to operate the building energy systems more efficiently, constructing a system model that makes accurate predictions is essential. However, a lot of domain knowledge and manual effort are required to configure such a model, which impedes the broad adoption of MPC algorithms (Blum et al., 2022). Since machine learning techniques rely mainly on collected measurements to approximate system dynamics, they have gained increasing popularity in predicting the thermal response of buildings coupled with Heating, Ventilation, and Air Conditioning (HVAC) systems (Stoffel et al., 2024; Wan et al., 2023). Among different learning approaches, Gaussian Process Regression (GPR) is a supervised learning approach frequently used in real-world model-based control applications due to its modeling flexibility of continuous dynamics and performance quantification (Dai et al., 2024; Maddalena et al., 2022).

It is acknowledged that prediction accuracy improves with the larger data set collected during the building operation (Stoffel et al., 2024; Massa Gray and Schmidt, 2018). Nevertheless, the computational resources required for training GPR grow quadratically as data points grow (Rasmussen and Williams, 2006). Increasing the computational efficiency of GPR is critical to identifying online building dynamics for real-time control tasks. Moreover, reducing the size of input space by including only the necessary predictors or input features can increase data density, thereby achieving more accurate predictions (Lederer et al., 2020). Yet, the selection of optimized input features and the corresponding dataset depends on concrete building systems and may differ among buildings. If domain understanding of the building systems could be provided, an integrated workflow based on such information could be developed to select input features and the dataset smartly.

Thanks to the rapid development of Semantic Web technologies (Berners-Lee et al., 2001) in the building domain, the context of building topology, energy systems as well as telemetric sensor measurements can be described in a machine-interpretable way via formal knowledge representation, addressed as ontologies or metadata schemas (Pauwels and Fierro, 2022). Using the metadata of buildings, standard pipelines can be built to access technical system information and historical as well as streamed measurement data, as suggested. This forms the basis for fast deployment and easier use of the GPR algorithm among different buildings, namely the portability.

The main contribution of this paper is the automated generation of GPR models for indoor temperature prediction using semantics. We adopt LoG-GP (Lederer et al., 2021), a computationally efficient online GPR algorithm. To design the blueprint of the algorithm, sensitivity analysis of input features are performed. Based on that, we propose a workflow that utilizes the metadata to select the input features accordingly, and generate GPR models automatically.

The remainder of this paper is structured as follows. It begins with a review of related work and an explanation of the methodology. This is followed by a detailed description of the knowledge-based workflow designed to configure GPR models automatically. Next, the test building instances and experimental setup are outlined, and the performance of the generated models is analyzed and compared to Naive-GP for benchmarking purpose. Finally, the main findings of this study are summarized.

Related work

Among previous studies on zone temperature prediction, naive-GP methods (Massa Gray and Schmidt, 2018; Stoffel et al., 2024) were mainly adopted. Previous research also revealed that a larger training dataset prompts better prediction accuracy, and the prediction model should

be retrained with newly collected data on a regular basis, e.g., every 24 hours (Stoffel et al., 2024; Massa Gray and Schmidt, 2016). Moreover, Maddalena et al. (2022); Stoffel et al. (2024) emphasized that the computational speed of GPR needs to be increased to allow a longer prediction horizon. Many advanced algorithms, such as sparse GP and distributed GP, are proposed for handling the big data regime to achieve fast prediction of online GPR. By extracting a given number of virtual interior points from the original data set, sparse GP is a commonly used method to train and infer based on the interior points. However, the reconstruction of the interior points blocks its online application (Snelson and Ghahramani, 2005). Distributed GP, as another accelerating technique, chooses a subset of the original data set for training and prediction, allowing fast adaptation to streamed data while inheriting the performance quantification from naive GPR (Lederer et al., 2021). Specifically, this study considers locally growing random trees of GPs (LoG-GP) (Lederer et al., 2021), a distributed GP method, because it utilizes the entire dataset, enables incremental online learning with streamed data, preserves fast prediction speed and the consistent prediction error bound. LoG-GP algorithm could address the requirement of fast and accurate prediction, as it achieves a logarithmic computational complexity increase to the training data size for model update and inference (Lederer et al., 2021). Despite a few successful applications in robotics (Lederer et al., 2022, 2021), LoG-GP has yet not been utilized in the building domain, probably because the deployment of the algorithm demands consistent data exchange with the existing Building Management System (BMS) and has higher configuration complexity.

As for the selection of input features, few existing studies have reported sensitivity analysis of input features. Regarding indoor temperature prediction using data-driven approaches, selecting input features is conducted empirically, while the selected features vary from study to study. This is likely because 1) the prediction models were developed for individual buildings instead of for specific types of systems; 2) the selection of the input features depends intrinsically on the building and energy systems, which requires domain knowledge of the system. Since the increase of input features can increase computational time, introduce overfitting, and degrade the prediction accuracy (Guenther and Sawodny, 2019), determining the necessary minimal subset of input features for specific systems accordingly is beneficial for developing a more portable algorithm. Focusing on commercial buildings with a radiator heating system, and a Variable-Air-Volume (VAV) system, we summarize the common input features for temperature prediction into the following parts: time (Massa Gray and Schmidt, 2016, 2018), last step(s) of zone temperature (Maddalena et al., 2022; Ramadan et al., 2021; Stoffel et al., 2024), estimated HVAC power (Massa Gray and Schmidt, 2016, 2018; Di Natale et al., 2022; Ramadan et al., 2021), states of the field actuators (Maddalena et al., 2022; Stoffel et al., 2024), ambient conditions (Massa Gray and Schmidt, 2016, 2018; Maddalena et al., 2022; Di Natale et al., 2022; Stoffel et al., 2024), and occupants (Wan et al., 2023).

To provide a standardized description of disparate systems in buildings, ontologies such as Brick, upcoming ASHRAE Standard 223P and so on are proposed. Encoded in the same format using Resource Description Framework (RDF), metadata from the diverse sources can be synthesized for more comprehensive services with less manual efforts, such as to onboard load prediction algorithm of different chillers (Kan and Xiao, 2023), and streamline data-driven MPC configuration (Chamari et al., 2024). This paper utilizes a comprehensive semantic model of building geometry, HVAC systems' topology, and telemetric measurements built in our previous work (Wan et al., 2025b), aiming to streamline the GPR configuration. The semantic model uses mainly BOT, FSO, BRICK, SOSA/SSN, and TD ontologies.

Fast Incremental Learning with GPR

Gaussian Process Regression

Gaussian Process (GP) assumes that an unknown function $f(\cdot) : \mathbb{X} \to \mathbb{R}$ follows a Gaussian distribution over function, i.e., $f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$, where $m(\cdot) : \mathbb{X} \to \mathbb{R}$ denotes the prior mean function and $k(\cdot,\cdot) : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_{0,+}$ is the covariance function, also addressed as kernel function. The prior mean function encodes all known information of $f(\cdot)$, which is set as $m(\boldsymbol{x}) = 0, \forall \boldsymbol{x} \in \mathbb{X}$ for completely unknown $f(\cdot)$. To model any continuous function with arbitrary accuracy, universal kernels are commonly used, such as the Squared Exponential (SE) kernel in the form of $k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp(-\|\boldsymbol{x}-\boldsymbol{x}'\|^2/(2l^2))$, where variance $\sigma_f \in \mathbb{R}_+$ and length-scale $l \in \mathbb{R}_+$ are hyperparameters lumped as $\boldsymbol{\theta}_{\text{GP}} = [\sigma_f, l]^T$. The value of $\boldsymbol{\theta}_{\text{GP}}$ is chosen from hyperparameter optimization to maximize the log marginal likelihood over the training dataset (Rasmussen and Williams, 2006). Specifically, a training dataset is denoted as $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) | i = 1, 2, \cdots, N\}$ with $N \in \mathbb{N}$ data pairs, where $\boldsymbol{x}_i \in \mathbb{X}$ and the measurement noise follows an identical independent zero-mean Gaussian distribution $y_i = f(\boldsymbol{x}_i) + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$.

For the inference of $f(\boldsymbol{x}^*)$ at the inquiry point $\boldsymbol{x}^* \in \mathbb{X}$, Gaussian process assumes the concatenation $[\boldsymbol{y}^T, f(\boldsymbol{x}^*)]^T$ follows a joint Gaussian distribution (Rasmussen and Williams, 2006), i.e.,

$$\begin{bmatrix} \boldsymbol{y} \\ f(\boldsymbol{x}^*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0_{N\times 1} \\ 0 \end{bmatrix}, \begin{bmatrix} \boldsymbol{K} + \sigma_n^2 \boldsymbol{I}_N & \boldsymbol{k}(\boldsymbol{x}^*) \\ \boldsymbol{k}^T(\boldsymbol{x}^*) & k(\boldsymbol{x}^*, \boldsymbol{x}^*) \end{bmatrix} \right), \tag{1}$$

where $\boldsymbol{K} \in \mathbb{R}^{N\times N}$ is the kernel gram matrix with the element on the i-th row and j-th column as $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the kernel vector $\boldsymbol{k}(\cdot) : \mathbb{X} \to \mathbb{R}^N$ is defined as $\boldsymbol{k}(\boldsymbol{x}^*) = [k(\boldsymbol{x}_1, \boldsymbol{x}^*), k(\boldsymbol{x}_2, \boldsymbol{x}^*), \cdots, k(\boldsymbol{x}_N, \boldsymbol{x}^*)]^T$. The conditional distribution of $f(\boldsymbol{x}^*)$ denoted as $f(\boldsymbol{x}^*)|\mathcal{D}$ is Gaussian,

which is written as $f(\boldsymbol{x}^*)|\mathcal{D} \sim \mathcal{N}(\mu(\boldsymbol{x}^*), \sigma^2(\boldsymbol{x}^*))$ with

$$\mu(\boldsymbol{x}^*) = k^T(\boldsymbol{x}^*)(\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}_N)^{-1}\boldsymbol{y}, \tag{2a}$$

$$\sigma^2(\boldsymbol{x}^*) = k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{k}^T(\boldsymbol{x}^*)(\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}_N)^{-1}\boldsymbol{k}(\boldsymbol{x}^*). \tag{2b}$$

The posterior mean $\mu(\boldsymbol{x}^*)$ is used as the prediction of $f(\boldsymbol{x}^*)$, and the posterior standard deviation $\sigma(\boldsymbol{x}^*) \in \mathbb{R}_{0,+}$ is adopted to quantify the prediction performance.

To improve prediction performance, collecting more data is a promising method resulting in a dataset $\mathcal{D}$ with larger N. However, the computing cost for GP prediction also increases with N. Specifically, the calculation of the posterior mean $\mu(\cdot)$ and variance $\sigma^2(\cdot)$ requires $\mathcal{O}(N)$ and $\mathcal{O}(N^2)$ computations, respectively, with an offline calculated $(\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}_N)^{-1}$. The increased computational burden may deteriorate the control performance, making naive GP challenging to apply in cases with high accuracy requirements.

Locally growing random tree of GPs

As the update complexity of naive GPR grows quadratically with the length of the training dataset N, Lederer et al. (2021) propose Locally Growing random trees of GPs (LoG-GP) that preserves uniform prediction error bounds as naive GP, yet can make the inference with only a logarithmical growth of computational complexity $\mathcal{O}(\log(N))$. In contrast to naive GP and sparse GP approaches that construct one single model, LoG-GP employs a dynamically evolving tree structure to organize the training dataset and train a series of local GPR models accordingly, which are hosted in the leaf nodes of the tree and used for predictions. LoG-GP starts with a single GP model at the root node $\mathbb{D}_0$, which is incrementally updated with streamed data until the quantity of training samples reaches a predetermined threshold $\bar{N}$. Then, the dataset $\mathbb{D}_0$ is partitioned into two subsets $\mathbb{D}_i$, where $i = 1, 2$, according to prescribed data allocation rule Π with a probability function p^0. Consequently, a tree with two leaf nodes containing all training data is generated, whose individual local GP models can be computed efficiently using Eq. 2. The splitting process continues recursively: newly arriving streamed data will be allocated to a subset $\mathbb{D}_i$ until either of the subsets obtains the limit $\bar{N}$; Then, the data points in the complete subset are divided to another two subsets, following Π characterized by its probability function p^i. Eventually, a tree structure of GP models is constructed, where each leaf node represents a local GP model. The depth of the tree grows logarithmically with the number of training samples, ensuring lower computational complexity $\mathcal{O}(\log(N))$. For making a prediction using LOG-GP, the streamed data x^* is assigned to the appropriate local GP model(s), following the same data allocation rule Π. The probability weight for the individual leaf node $w_l(x^*)$ is calculated by the product of probabilities along the path P_l from the root node to the leaf node l: $w_l(x^*) = \prod_{i \in P_l} p^i(x^*)$. The aggregated mean and variance of the prediction made by LoG-GP are computed as (Lederer et al., 2021):

$$\tilde{\mu}(\boldsymbol{x}^*) = \sum\nolimits_{l \in \mathbb{L}} \omega_l(\boldsymbol{x}^*)\tilde{\sigma}^2(\boldsymbol{x}^*)\sigma_l^{-2}(\boldsymbol{x}^*)\mu_l(\boldsymbol{x}^*), \tag{3a}$$

$$\tilde{\sigma}^2(\boldsymbol{x}^*) = \sum\nolimits_{l \in \mathbb{L}} \omega_l(\boldsymbol{x}^*)\sigma_l^{-2}(\boldsymbol{x}^*), \tag{3b}$$

where $\mathbb{L}$ denotes the set of all leaf nodes, μ_l and σ_l^2 are the predicted mean and variance of the local GP model in the leaf node l, respectively. Note that only a few leaf nodes are active and have a positive probability weight by defining Π and p^i properly, whereas most of the leaf nodes are inactive, and their weights $w_l(x)$ become 0. Thus, the aggregated prediction can be efficiently calculated based on the active local GP(s), realizing a reduced computational complexity in $\mathcal{O}(\log(N)^2)$. To ensure the smoothness of the LoG-GP, an overlapping ratio between the neighboring leaf nodes is defined (Lederer et al., 2021). The design of the LoG-GP algorithm makes it highly suitable for large training datasets and online learning. It also maintains the consistent error bounds inherited from the local naive GPRs.

According to (Lederer et al., 2021), we set $\bar{N}$ to 400 data points for each leaf node, overlap ration to 1%, and a simple allocation rule Π that the data in the root set is split into two subsets by a hyperplane, which is at the median of the input feature with the biggest range.

Knowledge-informed GPR Configuration

Input feature selection

This study focuses on the zone temperature prediction of commercial buildings equipped with hydraulic heating system and VAV ventilation system, where terminals, i.e., radiators and VAV boxes, are available at the room side for realizing both the cooling and heating. Based on the literature review, the input features often used are summarized in Table 1. Considered input features $\boldsymbol{x}^T$ include: hour of the day HD, day of week DW, the status of AHU Θ_{AHU}, zone temperature $\vartheta_{\text{z,2}}$, HVAC thermal power $\dot{Q}_{\text{HVAC}}$, supply water temperature of the radiator $\vartheta_{\text{rad,1}}$, thermal valve position of the radiator α_{rad} or water volume flow rate $\dot{V}_{\text{rad}}$, supply air temperature $\vartheta_{\text{z,1}}$, damper opening $\alpha_{\text{z,1}}$ or supply air volume flow rate $\dot{V}_{\text{z,1}}$, external ambient temperature ϑ_{amb}, global irradiation H_{glo}, estimated occupants N_{occ} at the current time step k. The prediction output is the zone temperature $\vartheta_{\text{z,2}}$ at the next time step $k+1$.

Following the wrapper input feature selection method (El Aboudi and Benhlima, 2016), sensitivity analyses are designed to further investigate how the concrete knowledge about buildings influences the selection of these features. The input features are divided into four groups, as displayed in Table 1 to identify the minimal subset of input features $\mathcal{J}$ dependent on the types of building instances. Group 1 (G1) is selected as the baseline, consisting of time predictors, HVAC operating status, field actuators, exterior ambient conditions, and the number of occupants. Group 2 (G2) includes the estimated HVAC thermal power used in the zone instead of the field actuators. Group 3 (G3) excludes the ambient conditions. Group 4 (G4) excludes the

Table 1: design of the input feature groups
'+' indicates inclusion in the group, and '-' indicates exclusion. Grey cells highlight differences compared with G1.

Predictors	G1	G2	G3	G4
HD	+	+	+	+
DW	+	+	+	+
Θ_{AHU}	+	+	+	+
$\vartheta_{\text{z,z}}$	+	+	+	+
$\dot{Q}_{\text{HVAC}}$	-	+	-	-
$\vartheta_{\text{rad,1}}$	+	-	+	+
$\alpha_{\text{rad}} \mid \dot{V}_{\text{rad}}$	+	-	+	+
$\vartheta_{\text{z,1}}$	+	-	+	+
$\alpha_{\text{z,1}} \mid \dot{V}_{\text{z,1}}$	+	-	+	+
θ_{amb}	+	+	-	+
H_{glo}	+	+	-	+
N_{occ}	+	+	+	-

estimated number of occupants. The design of G2, G3, and G4 is to determine the influence of the estimated HVAC power, ambient conditions, and occupants on the prediction accuracy, especially for different types of building instances, respectively.

Automated data selection using metadata

To automate the data selection required in GPR, semantics are used in three consecutive steps:

1. Querying the topological properties of the zone or building modeled, i.e., the floor area A, the interior wall area A_{in}, the exterior wall area A_{ext} and window area A_{w} exposed to the ambient, to decide the necessary input features $\mathcal{I}$;
2. Querying the topology of the distribution systems to determine the corresponding terminals and locate the corresponding sensors for the input features $\mathcal{I}$;
3. Retrieving both training dataset $\mathcal{D}$ and newly arriving testing data $\mathcal{X}^*$ using metadata of the sensors, e.g., index for historical and streamed measurements.

The metadata manipulation (Steps 1 and 2) is realized via a group of SPARQL queries. Listing 1 demonstrates an example of how to query the sensor ID of the input feature $\vartheta_{\text{rad,1}}$ in Step 2 if the modeling object is recognized as the entire building in Step 1. The queries for other input features involved are constructed similarly. With the metadata about input features $\mathcal{I}$, training data is retrieved from the BMS. Since the metadata of all instances is organized consistently, data required for GPR generation can be allocated individually automatically. More details about queries of room instances and the connection with existing BMS can be referred to in our previous work (Wan et al., 2025b).

```
PREFIX fso-ext:
    ↪ <http://bosch-cr-aes//hsbc/fso-extension#>
PREFIX bot: <https://w3id.org/bot#>
PREFIX fso: <https://w3id.org/fso#>
PREFIX ssn: <http://www.w3.org/ns/ssn/>
PREFIX sosa: <http://www.w3.org/ns/sosa/>
PREFIX brick: <https://brickschema.org/schema/Brick#>
PREFIX td: <https://www.w3.org/2019/wot/td#>
PREFIX ref:
    ↪ <https://brickschema.org/schema/Brick/ref#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX tso: <https://w3id.org/tso#>
select (COUNT(DISTINCT ?step) as ?distance)
    ↪ ?tempSenID ?tempIoT
where {
   ?building a bot:Building;
             rdfs:label '{Building_Name}'.

   ?pump a fso-ext:Pump.
   ?comp1  a fso-ext:SensorFitting;
           a sosa:FeatureOfInterest.

   ?supSys a tso:HeatingSystem;
        fso:exchangesHeatWith ?building;
        fso:hasComponent ?comp1;
        fso:hasComponent ?pump.
    ?pump fso:feedsFluidTo* ?step.
    ?step fso:feedsFluidTo+ ?comp1.
    ?comp1 ssn:hasProperty ?tempProperty.
    ?tempProperty a sosa:ObservableProperty.
    ?tempSensor sosa:observes ?tempProperty;
       a brick:Temperature_Sensor;
       ref:hasTimeseriesID ?tempSenID;
       td:name ?tempIoT.
}
GROUP BY ?tempSensor ?com1 ?tempSenID ?tempIoT
ORDER BY ?distance
LIMIT 1
```

Listing 1: SPARQL Query for the input feature $\vartheta_{rad,1}$ of the hydraulic heating system, for modeling a given building.

Workflow

Figure 1 depicts the process for the automated configuration of GPR models using the semantic graph guided by a pre-defined model template. Firstly, important properties of the modeling object are analyzed, as described in the previous section. The relevant features $\mathcal{I}$ are then selected following the conclusion drawn from the sensitivity analysis. Secondly, given the input features $\mathcal{I}$, the workflow executes several SPARQL queries to determine the measurement ID and retrieve the historical measurements accordingly to construct the training dataset $\mathcal{D}_{\mathcal{I}}$. Thirdly, the GPR model, denoted as $\mathcal{GP}(\mu, k(x, x^{'}))$, is trained using the predefined model templates. Both naive-GP and LoG-GP are considered in our study. Fourthly, the generated GPR model is used for making the inference $\boldsymbol{y}^* = \mathcal{GP}(\mathcal{D}_{\mathcal{I}}, \boldsymbol{x}^*)$, as a new measurement $\boldsymbol{x}^*$ arrives. Note that LoG-GP model is generated with the incremental learning enabled, which means $\boldsymbol{x}^*$ updates $\mathcal{D}_{\mathcal{I}}$ and tree structure of GPs during the run-time, as highlighted in Figure 1. However, incremental learning is not implemented for naive GP due to the intensive computational resources required at each step.

Experiments

We conducted the experiments using real measurement data in year 2022-2023 of a building that sits in the Bosch Research campus in Renningen, Germany. In total, 6 instances within the building were considered, which include four meeting rooms (R1, R2, R6, R7), one open-plan office

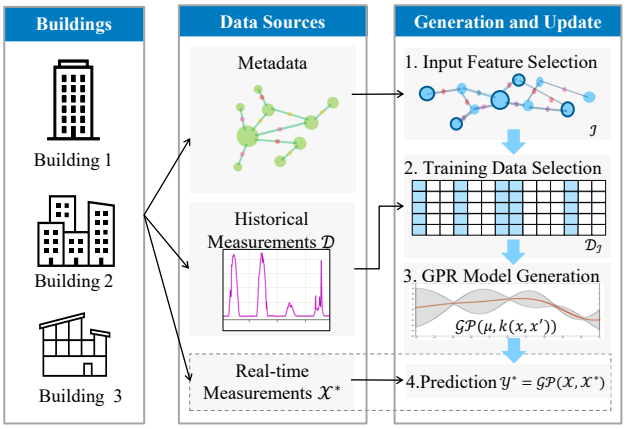

Figure 1: Proposed workflow for portable GPR model generation using domain knowledge.

(R3), and the entire building, whose floor plan is detailed in (Wan et al., 2025a).

Experiments A and B were designed to examine the input feature selection rule and the portability of the proposed workflow. In Experiment A, we performed sensitivity analysis of input features on three rooms (R1, R2, and R3), to determine the optimal input features for temperature prediction depending on the instance type. R1, R2, and R3 represent rooms with different kinds of boundaries: exterior walls of a small window-wall ratio (Type 1), complete interior walls (Type 2), and exterior walls of a large window-wall ratio (Type 3), respectively. We selected the LoG-GP algorithm due to its fast prediction speed. In Experiment B, we generated GPR models (LoG-GP and Naive-GP) for three other zones (R6, R7, and the entire Building). The input features of the individual room instances are selected according to the conclusion drawn from the sensitivity analysis. Eventually, we analyze the performance of the generated GPR models. For benchmarking purposes, we compare the LoG-GP with Naive-GP models. Note that all the GPR models were automatically generated using the proposed workflow.

Details about tested building instances can be found in Table 2. We implemented the LoG-GP algorithm based on the open-source code in (Lederer et al., 2021), Naive-GP using GPyTorch, and hyperparameter optimization of length-scale and noise using Adam (Kingma and Ba, 2014). The time-series data is stored in Influx DB Version 2.7, metadata in Graph DB Version 10.2. The proposed workflow is implemented using Python 3.12. Experiments were conducted on a machine with Windows 10, Intel Core i5, and 16 GB RAM. The sampling interval in both training and testing datasets is 15 minutes.

Results and Discussion

To evaluate the accuracy of GPR models, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) of the predicted temperature y^* compared with the measured value $\hat{y}^*$ were used. In preliminary investigations, we optimized the hyperparameters with monthly data and repeated the optimization iteratively for an entire year. The optimized σ_f and l remained almost unchanged for the individual instances (R1, R2, R3). Therefore, θ_{GP} of all generated GPR models were by default optimized with training data of the first month. The performance of GPRs is evaluated with multiple steps' prediction, in a way that the prediction model is called within an MPC algorithm. This means GPR makes a series one-step-ahead prediction using the last predicted value, in a time horizon T with N discrete steps: $\boldsymbol{y}^* = [\vartheta_{\text{z},2,k+1}, \vartheta_{\text{z},2,k+2}, \cdots, \vartheta_{\text{z},2,k+N}]^T$, and only feeds the measured room temperature into GPR model every N steps. The measuring accuracy of the temperature sensor is 0.1 °C.

Experiment A: Sensitivity Analysis

Experiment A compared the accuracy of GPR models trained with four different input feature groups (G1 to G4) on three rooms (R1, R2, and R3). The temperature prediction models using LoG-GP were trained on data collected from April to December 2022, with 26,400 records. G1, G2, and G3 were tested from January to December 2023 (35,040 records), with results in Table 3. G4 and a baseline G1 were only tested in July 2023 (2,976 records), as in Table 4, because the occupants were observed to vary actively during the period in all three instances. Incremental learning were enabled in all groups, and prediction horizon T was set to 12 hours.

Throughout the experiments, the MAE consistently exhibits lower values than RMSE. This discrepancy observed in RMSE is attributed to a small number of extreme errors, which are greater than 3 °C. The possible reason for such extreme errors is that the local GP model(s) activated for the prediction lack enough representative samples,thus. the predicted temperature drops to the prior mean.

Comparing G1 and G2 in Table 3, GPR models with field actuators (G1) demonstrate better accuracy regarding RMSE, and MAE in R1 and R2, while same RMSE and MAE in R3. Given the fact that R1 has one supply air damper and one radiator, whereas R2 contains only one supply air damper, the difference is probably because that field actuators depicts the dynamics of different HVAC systems and their individual influences on the temperature, while estimated HVAC power is an aggregated value and holds some simplification of the real heat transfer. R3 has multiple air dampers and radiators, however the prediction results are not much different between G1 and G2. This could result from insufficient measurement points in the zone of such an large volume – only a single temperature sensor is installed in the return duct and the air is not always well mixed in reality in the volume. Another important lesson learned is that when considering field actuators as input features, using a weighted average value of the same type of input feature leads to better prediction, due to the fact the former dataset contains higher information density. For example, using a volume-flow weighted damper position $\bar{\alpha}_{\text{z},1}$ predicts better than considering $\alpha_{\text{z},1,i} i \in 1, 2, \cdots, 13$ all as input features.

Table 2: Details about the geometry and HVAC systems of the building instances involved in the experiments.

Rooms	R1	R2	R3	R6	R7	Building
Experiments	A: Input feature selection			B: Performance evaluation		
Algorithm	LoG-GP			LoG vs. Naïve-GP		
V (m^3)	74	73	909	153	32	13393
A (m^2)	18	18	227	38	8	3162
A_{IW} (m^2)	57	78	149	94	52	7653
A_{EW} (m^2)	18	-	82	15	-	635
A_{W} (m^2)	4	-	88	8	-	643
Damper (-)	1	1	13	1	1	36
Reheat Coil (-)	1	1	-	1	-	9
Radiator (-)	1	-	10	1	-	69

Table 3: Results of Experiment A: Influence of prediction accuracy from $\mathcal{I}$ of the field actuator, the estimated thermal power and ambient data

Groups	**G1**	**G2**	**G3**
R1			
RMSE (°C)	0.37	0.48 (+30%)	0.53 (+43%)
MAE (°C)	0.25	0.28 (+12%)	0.27 (+8%)
R2			
RMSE (°C)	0.42	0.48 (+14%)	0.52 (+24%)
MAE (°C)	0.29	0.32 (+10%)	0.34 (+17%)
R3			
RMSE (°C)	0.66	0.66 (-)	1.17 (+77%)
MAE (°C)	0.29	0.29 (-)	0.35 (+21%)

Table 4: Results of Experiment A: Influence of prediction accuracy from $\mathcal{I}$ of the occupants

Groups	**G1**	**G4**
R1		
RMSE (°C)	0.36	0.38 (+5%)
MAE (°C)	0.27	0.27 (-)
R2		
RMSE (°C)	0.32	0.33 (+3%)
MAE (°C)	0.24	0.24 (-)
R3		
RMSE (°C)	0.47	0.48 (+2%)
MAE (°C)	0.35	0.39 (+11%)

Turning to results of the input feature group for ambient factors (G3), Table 3 reveals that R3, which has large area of exterior walls and windows, is most strongly negatively impacted by missing input features θ_{amb} and H_{glo}. Surprisingly, the prediction accuracy for R2 also drops, which is not exposed to the outdoor ambience. This is probably because there exist no fully isolated zones in real buildings, and adjacent rooms that can be exposed to the outdoor influence the air temperature. This suggests that the ambient factors provide critical information to room temperature prediction regardless of the boundary conditions, should, thus, be considered as input features if possible. Note that we purposely did not consider room temperatures of the adjacent zones in the input features, because during the run-time it is difficult to provide an accurate predicted trajectory for the adjacent zone temperatures, which may result in a reduced performance of GPR deployed in the MPC algorithm.

Regarding the influence of occupants, Table 4 demonstrates no significant difference between G1 and G4 for all three instances. The sensitivity of GPR models to N_{occ} is lower than the initial anticipation. Three main possible reasons are: 1) the building investigated was not fully occupied during year 2022 - 2023, since working from home was still recommended during the post-COVID period. 2) In majority of the dataset, N_{occ} is zero, because the office building is only occupied from roughly 08:00 – 18:00 on working days (less than 50 % of the complete dataset). 3) We also noticed that there was a weak negative correlation between N_{occ} and HVAC power controlled by the legacy control, which lacks system excitation. Measurement data with higher excitation of the HVAC system and occupants needs to be prepared in future research. N_{occ} can be omitted from the input features based on the test on this building.

Overall, these results of sensitivity analysis indicate that the optimal input features $\mathcal{I}$ for predicting the zone temperature include: time, current zone temperature $\vartheta_{\text{z},2,k}$, state of field actuators, and external ambient factors, regardless of the geometrical boundaries of the zone instance.

Experiment B: Performance Analysis

Using the optimal input features identified in the previous section and proposed workflow, Naive-GP and LoG-GP models of three other instances (R6, R7 and Building) were automatically generated.

The performance of GPRs were compared in **Experiment B-1**. Because the computation time of Naive-GP becomes extremely long as the dataset grow and the inference performance degrades as the prediction horizon increases, we limit the training dataset to 3 months (8,832 records) and testing dataset to the following month (2,976 records), with the prediction horizon T first set to 3 hours. Experiments were conducted for both summer and winter, which demonstrated similar trends. Due to limited space, only results of the summer time are presented in this section.

To make a fair comparison, the incremental learning ability is disabled in this comparison experiment. Table 5 lists the errors of GPRs for all three instances during the testing period, and Figure 2 elaborates detailed predictions on August 2nd and 3rd, 2023. The results show that LoG-GP outperforms the Naive-GP model, demonstrating lower RMSE and MAE. As shown in Figure 2, predictions made by LoG-GP follow closely the ground truth, while Naive-GP displays error propagation within the horizon. Although Naive-GP uses a larger training dataset for making each inference, two possible reasons can explain the better accuracy achieved by LoG-GP. First, the LoG-GP limits the size of each local GP to a maximum number (400 data points in this study) and predicts with the local GPs of the highest similarity, which may enhance predictive accuracy by focusing on the most relevant data. Second, Naive-GP utilizes a large dataset, which nevertheless is likely accompanied with a high portion of noisy data e.g. non-working hours of the HVAC system. Thereby, the prediction accuracy is degraded due to extraneous or misleading information. Further empirical studies are necessary to rigorously validate these claims.

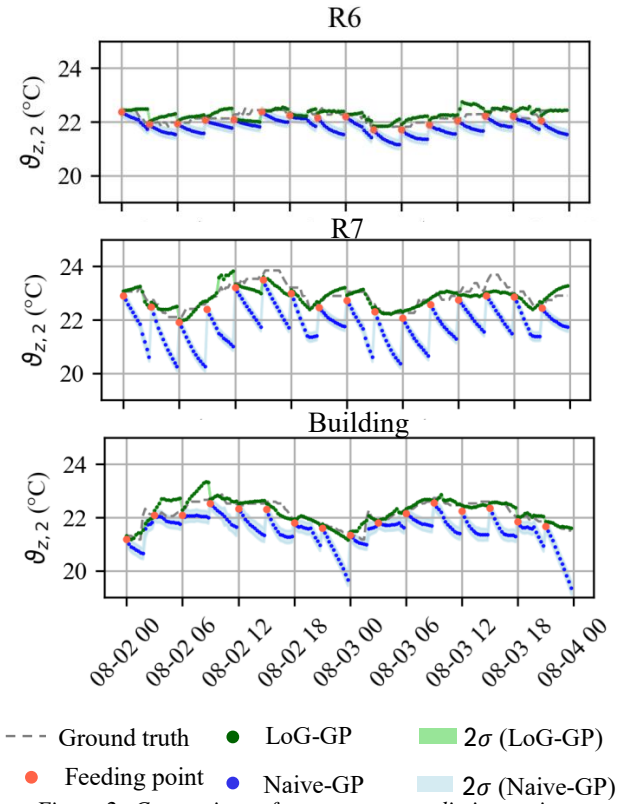

Figure 2: Comparison of temperature predictions using LoG-GP and Naive-GP for R6, R7 and Building (Training from May to July 2023, testing in August 2023).

To further validate the full potential of the generated LoG-GP, **Experiment B-2** investigated the performance of the generated GPR models of R6, R7 and Building with a training period of April to December 2022, testing period of January to December 2023, prediction horizon T of 12 hours, and incremental learning. Table 6 shows the prediction accuracy of three instances. 12-hours-ahead temperature prediction RMSEs of R6, R7 and Building are 0.36 °C, 0.84 °C, and 0.97 °C, while all the MAEs are below 0.5 degree. For each data point, the inference takes 0.003 seconds, and incremental training about 0.01 seconds. This proves the computational efficiency, prediction accuracy and transferability of the proposed approach.

Table 5: Results of Experiment B-1: Performance comparison of 3-hours-ahead temperature prediction between LoG-GP and Naive-GP in August 2023.

Algorithms	**Log-GP**	**Naive-GP**
R6		
RMSE (°C)	0.21	0.61
MAE (°C)	0.15	0.51
R7		
RMSE (°C)	0.62	1.2
MAE (°C)	0.22	1.05
Building		
RMSE (°C)	0.41	0.98
MAE (°C)	0.21	0.83

Table 6: Experiment B: Performance analysis of LoG-GP for 12-hour-ahead temperature prediction

	R6	**R7**	**Building**
RMSE (°C)	0.36	0.84	0.97
MAE (°C)	0.22	0.27	0.32

Conclusions

This study presented a knowledge-informed workflow to configure GPRs, in which metadata of the buildings is used to first determine the input features, and then allocate the training dataset as well as the streamed data accordingly. The workflow was applied to six different instances in an office building, and successfully generated LoG-GP models that can perform fast incremental learning for predicting the zone temperature. Sensitivity analysis on the input features shows that regardless of the geometrical boundaries of the zone instances in the studied building, input features such as the state of field actuators, and external ambient factors play important roles, whereas number of occupants demonstrates low influences. Using the generated LoG-GP models, the 12-hours-ahead temperature predictions of all the instances achieve MAE beneath 0.5 °C and RMSE beneath 1 °C over a testing period of 1 year, which is comparable to previous studies (Massa Gray and Schmidt, 2016; Stoffel et al., 2024). By infusing domain knowledge into GPR configuration, we are able to generate computational efficient, accurate and portable data-driven models. As the current study only demonstrates the portability of different instances within a single building, we envision that future studies validate the proposed workflow on different buildings, and investigate further the adoption of the generated LoG-GP model in MPC design.

Acknowledgments

The authors would like to thank Dr. Philipp Kotman from group CR/AES2, Robert Bosch GmbH for organizational supports and proof-reading.

References

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. Scientific American, 284(5):pp. 34–43.

Blum, D., Wang, Z., Weyandt, C., Kim, D., Wetter, M., Hong, T., and Piette, M. A. (2022). Field demonstration and implementation analysis of model predictive control in an office hvac system. Applied Energy, 318:119104.

Chamari, L., Walker, S., Wan, L., Petrova, E., and Pauwels, P. (2024). Portable Model Predictive Controller System Design for Demand Side Management using Semantic Web Technologies. Marrakesh, Morocco.

Dai, X., Yang, Z., Zhang, S., Zhai, D.-H., Xia, Y., and Hirche, S. (2024). Cooperative Online Learning for Multiagent System Control via Gaussian Processes With Event-Triggered Mechanism. IEEE transactions on neural networks and learning systems, PP.

Di Natale, L., Svetozarevic, B., Heer, P., and Jones, C. (2022). Physically Consistent Neural Networks for building thermal modeling: Theory and analysis. Applied Energy, 325:119806.

El Aboudi, N. and Benhlima, L. (2016). Review on wrapper feature selection approaches. In 2016 International Conference on Engineering & MIS (ICEMIS), pages 1–5, Agadir, Morocco. IEEE.

Guenther, J. and Sawodny, O. (2019). Feature selection and Gaussian Process regression for personalized thermal comfort prediction. Building and Environment, 148:448–458.

Kan, X. and Xiao, F. (2023). Development and application of brick schema based multi-sources data integration for smart building energy management.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. In International Conference on Learning Representations.

Lederer, A., Capone, A., Umlauft, J., and Hirche, S. (2020). How training data impacts performance in learning-based control. IEEE Control Systems Letters, 5(3):905–910.

Lederer, A., Conejo, A. J. O., Maier, K. A., Xiao, W., Umlauft, J., and Hirche, S. (2021). Gaussian Process-Based Real-Time Learning for Safety Critical Applications. In Meila, M. and Zhang, T., editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 6055–6064. PMLR.

Lederer, A., Zhang, M., Tesfazgi, S., and Hirche, S. (2022). Networked Online Learning for Control of Safety-Critical Resource-Constrained Systems based on Gaussian Processes. _eprint: 2202.11491.

Maddalena, E. T., Müller, S. A., dos Santos, R. M., Salzmann, C., and Jones, C. N. (2022). Experimental data-driven model predictive control of a hospital HVAC system during regular use. Energy and Buildings, 271:112316.

Massa Gray, F. and Schmidt, M. (2016). Thermal building modelling using Gaussian processes. Energy and Buildings, 119:119–128.

Massa Gray, F. and Schmidt, M. (2018). A hybrid approach to thermal building modelling using a combination of Gaussian processes and grey-box models. Energy and Buildings, 165:56–63.

Pauwels, P. and Fierro, G. (2022). Annex 81 Activity A3-Survey of metadata schemas for data-driven buildings. Technical report.

Ramadan, L., Shahrour, I., Mroueh, H., and Chehade, F. H. (2021). Use of Machine Learning Methods for Indoor Temperature Forecasting. Future Internet, 13(10):242.

Rasmussen, C. E. and Williams, C. K. I. (2006). Gaussian processes for machine learning. Adaptive computation and machine learning. MIT Press, Cambridge, Mass. OCLC: ocm61285753.

Snelson, E. and Ghahramani, Z. (2005). Sparse gaussian processes using pseudo-inputs. Advances in neural information processing systems, 18.

Stoffel, P., Berktold, M., and Müller, D. (2024). Real-life data-driven model predictive control for building energy systems comparing different machine learning models. Energy and Buildings, 305:113895.

Wan, L., Dai, X., Welfonder, T., Petrova, E., and Pauwels, P. (2023). Semi-automated thermal envelope model setup for adaptive model predictive control with event-triggered system identification. In Building Simulation 2023, volume 18, pages 3664–3671. IBPSA.

Wan, L., Esmaeili, P., Fjerbæk, E. V., Gall, J., Welfonder, T., Petrova, E., and Pauwels, P. (2025a). Generating building performance simulation models using semantic graphs and sensor measurements. In CLIMA 2025: the 15th REHVA HVAC World Congress.

Wan, L., Rossa, F., Welfonder, T., Petrova, E., and Pauwels, P. (2025b). Enabling scalable model predictive control design for building hvac systems using semantic data modelling. Automation in Construction, 170:105929.