Learning from Correlated Events for Equipment Relation Inference in Buildings

Dezhi Hong, Renqin Cai, Hongning Wang, Kamin Whitehouse
Department of Computer Science, University of Virginia
{dh5gm,rc7ne,hw5x,whitehouse}@virginia.edu

ABSTRACT

Modern buildings produce thousands of data streams, and the ability to automatically infer the physical context of such data is the key to enabling building analytics at scale. As acquiring this contextual information is currently a time-consuming and error-prone manual process, in this study we make the first attempt at automatically inferring one important contextual aspect of the equipment in buildings — how each equipment is functionally connected with another. The main insight behind our solution is that functionally connected equipment is exposed to the same events in the physical world, creating correlated changes in the time series data of both equipment. Because events are of indeterminate length in time series, however, identifying them requires solving a non-polynomial combinatorial data segmentation problem. We present a solution that first extracts latent events from the sensory time series data, and then sifts out coincident events with a customized correlation procedure to identify the relationship between equipment. We evaluated our approach on data collected from over 1,000 pieces of equipment from 5 commercial buildings of various sizes located in different geographical regions in the US. Results show that this approach achieves 94.38% accuracy in relation inference, compared to 85.49% by the best baseline.

KEYWORDS

Relation inference, smart buildings, time series analysis, event detection, probabilistic modeling

ACM Reference Format:
Dezhi Hong, Renqin Cai, Hongning Wang, Kamin Whitehouse. 2019. Learning from Correlated Events for Equipment Relation Inference in Buildings. In *The 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '19), November 13–14, 2019, New York, NY*. ACM, New York, NY, USA, Article 4, 10 pages. https://doi.

1 INTRODUCTION

Building analytics is a nascent but growing industry that has the potential to save 15% or more of the energy consumption in commercial buildings [25], which accounts for almost 20 percent of the total energy use in the U.S. [7]. Modern buildings produce tens of thousands of data streams that record the speed of every fan, the pressure in every duct, and the power consumption on every circuit. Among this data, as an example, when diagnosing an over-heated/over-cooled room, an analytical tool needs to locate a chain of equipment related to the room, from the Variable Air Volume (VAV) Box serving that room to the upstream Air Handling Unit (AHU) that serves the VAV, as illustrated in Figure 1. However, despite the emerging standardized schema [1], acquiring accurate physical context (e.g., the VAV-AHU connection relationship) required by building analytics is currently a *manual* process. This process anecdotally can take a week or even longer for each building, often requiring a site visit and manual inspection of the equipment [8]. Yet, errors still occur, e.g., the connection might not be updated after equipment upgrade. This laborious manual process is clearly not scalable to the nearly 100 million commercial buildings across the world. New automatic techniques are needed to fully realize the potential of building analytics at scale.

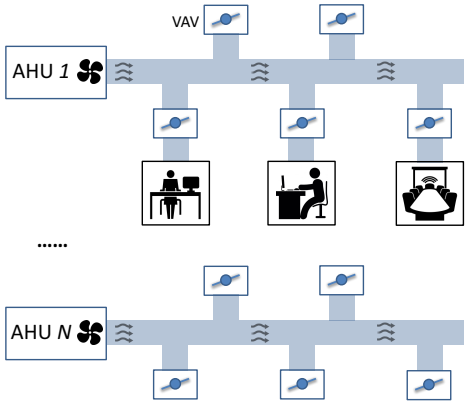


Figure 1: A typical building contains 5-15 air handling units (AHUs), each heating/cooling the air and circulating it to 10-30 variable air volume (VAV) boxes. Each VAV fine tunes the air flow for a single room. Building analytics requires the physical context of the vast amounts of data streams (e.g., which VAV connects to which AHU), which is currently acquired via a time-consuming, error-prone manual process.

In this paper, we address the key problem of automatically inferring the functional relationships between AHU and VAV, i.e., to assign each VAV to one of the AHUs in a building, based on which an analytics engine can perform designed tasks such as model predictive control [24], or fault detection and diagnosis [18]. The research community has been trying to address the issue of inferring various kinds of relationships among equipment and sensors. However, these works either build upon problem-specific domain knowledge that does not generalize in order to extract relations [14, 20, 33], or

BuildSys'19, November 13–14, 2019, New York, NY

ACM ISBN 978-1-1111-1111-1/19/11…$15.00
DOI: 10.1145/3360322.3360852

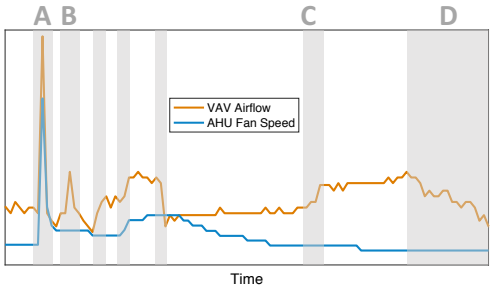


Figure 2: Example sensor readings from a VAV and its connected AHU. The VAV events (in grey) have many different causes—it can result from a change in the AHU (coincident event A) or changes in the served space (unique event B,C,D). Only the coincident ones help identify the functional connection between equipment.

still involve a manual process that must be applied to each building individually, thus not scalable [2, 29]. For example, a recent study infers the relations between VAVs and AHUs by perturbing the operation of each AHU and observing the responses in VAVs [29]. However, this approach takes weeks to execute and requires knowledge of when and how to perturb operations in a way that does not interfere with building needs. Another study shows that contextual information can be extracted based on the names of data streams and equipment controllers [2]. However, this approach also requires manual effort to interpret the names of sensors and equipment in each building. Additionally, mechanical relationships are often not encoded in the equipment names and sometimes the names can even be missing entirely. In contrast, the method proposed in this paper will automatically determine the functional relations between equipment with *minimal* manual setup and configuration effort.

The key intuition behind our solution is that functionally connected equipment is exposed to the same real-world events (e.g., an AHU's supply air fan turning on in Figure 1 will cause air flow to increase in connected VAVs), and thus will exhibit correlated changes in their time series data. Figure 2 shows example readings from a VAV air flow meter and its connected AHU's fan speedometer in a known office building: we observe a coincident event A in both equipment, which happened due to abrupt cooling demand after a group of people entered the room, while we also see events (C and D) in VAV only due to sun effect on the room. Observing this wide range of causes for events, we therefore design a solution where we first detect events in the sensor time series data and then sift out the correlated ones to help identify relations between equipment.

In our context, an event does not necessarily manifest as data outlier(s) in the traditional sense, but is rather a short subset of time series that may have exactly the same values as a steady state operation but with different variances, and/or different rates of change, and/or different second order derivatives. Because events are subsets of time series of indeterminate length, differentiating them from normal operation requires solving a complex combinatorial data segmentation problem — there are K possible states at each of the N timestamps and this gives a total of K^N possible combinations. To effectively locate events, we develop a layered Markovian Event Model (MEMO) that uses a bank of kinematic models (each akin to a Kalman Filter [17]) to characterize the transition patterns of different states in time series data, and introduce a set of latent variables to govern the switching between these models, reflecting the fact that the latent mechanical state underlying each data point is not observable. Our model also incorporates explicit modeling of rates of change in data and enforces domain structure such as sensor readings cannot change abruptly, which together contribute to more accurate estimation of the latent states in the data.

Once the events are detected, we identify the relations between equipment by correlating these events with a search-based procedure. A key challenge is that the data streams are affected by a mixture of latent factors, not just by the equipment relationships. For example, false correlations can be created by diurnal weather patterns, changes in room occupancy, the opening of a window, and many other factors. We design a search-based correlation procedure to filter out irrelevant events, which also allows for time lags in events across different streams, and only look at coincident events in both pieces of equipment. This dramatically reduces the number of spurious relations and helps to sift out pairs of equipment that are truly connected. Moreover, as we search for more of these correlated events over time, the probability of two pieces of equipment being correlated by random chance drops exponentially.

To evaluate our solution, we consider the functional relation between a particular pair of equipment in buildings discussed earlier: which VAV box is connected to which AHU. We evaluated the proposed solution on one-month data from 5 commercial buildings of various sizes which are geographically located across the U.S. Particularly, we performed evaluation in two different settings: 1) when we have knowledge about what type of sensors to use for the relation inference (e.g., only measuring the correlation between air flow volume in VAV and fan speed in AHU for the inference); Our approach identifies the AHU-VAV functional relationships with a 94.38% accuracy on average across all test buildings, compared to 85.49% by the best baseline. 2) when we do not have any knowledge about the type to use, i.e., we need to *automatically* select sensors for the analysis, our method achieves an average accuracy of 91.19%, compared to the possibly best accuracy of 95.64%. To the best of our knowledge, we make the *first* attempt at automatically inferring the functional relationships between equipment in buildings. Our solution is fully automatic, only requiring some knowledge about from which type of sensor streams to look for correlations. This fundamentally makes it possible to quickly apply building analytics involving the functional relation information about equipment at scale. We believe this is a promising result, and expect our method to be generally applicable to event detection and relation inference in broader domains.

2 RELATED WORK

Among the first works attempting to automatically infer the relations between different sensors using time series data, Smith et al. [33] used linear models to infer the relationship between temperature sensors and heating vents, and Hong et al. [14] used a correlation-based metric over time series to infer which sensors are in the same room. Koc et al. [20] later also used a correlation-based method to infer spatial relationships between discharge air and zone temperature sensors from different rooms. In this work, we focus on a different type of relation — the functional connection between equipment, and so all these approaches do not directly

apply. Additionally, these approaches assume a known structure of building design or building equipment, whereas our solution is expected to be building-independent. A recent work [29] focuses on the same type of relation as our study, where the solution relies on manually turning off each AHU and leveraging the responses in VAVs to identify the connections. This approach takes weeks to execute, yet still contains significant errors and must still be applied to each individual building manually. The same manual perturbation-based mechanism is also explored to identify the connections between sensing points and a VAV [21]. In contrast, the goal of this work is to enable more accurate inference using sensory measurements, yet with *minimal* manual intervention. Another study develops a data-driven solution [27] by cross-correlating the raw measurements from a particular pair of sensors in the equipment and taking the majority match over a period of time. Our method will exploit a more effective differentiator – the physical events – in sensor measurements to identify the relations between equipment.

In addition, there are also approaches that extract sensor relations from the textual metadata [2, 22, 32]; however, the metadata might not always encode the relations of interest (e.g., the AHU-VAV connection studies in this work), and sometimes, the metadata simply might not exist [15].

As an intermediate step in our solution, we seek an unsupervised event detection method, as our solution looks for correlated events among connected equipment. However, because the events in our context are real-world activities (e.g., fans turning on) that can be aperiodic, irregular, and do not necessarily manifest as outliers, prior works on outlier [13] and anomaly detection [4], and unsupervised change/event detection [16, 19, 28, 30, 34] become ineffective in our context. Instead, we take a "macro" view of each entire stream and holistically learn how data progresses in different operation states of the equipment, using a layered latent state model. We note that the use of switching variables is a key to effectively capturing the data progression, and that standard latent models fail to model the types of data we consider (as will be discussed in evaluation), as they typically assume that all data points in a stream share the same set of state parameters and thus progress in the same way. Additionally, the correlated events in our sensor streams caused by the mechanical connection usually exhibit time lags, as it takes time for the change in an AHU to propagate to its connected VAVs. But such time lags are unknown ahead of time. Consequently, stream pattern mining techniques [26] that are subjective to such time lags will likely produce spurious correlations. We address the time lags by duplicating detected events in short nearby time windows, which is effective and more computationally efficient than cross correlation [31] or dynamic time warping [5]. Moreover, in our problem, the existence of groups of time series with unknown sizes makes the recent advances in simultaneous segmentation and clustering [12] inapplicable, which requires knowledge about the size of each group beforehand.

3 APPROACH

The main insight behind our solution is that functionally connected equipment is exposed to the same real-world events, thus exhibiting correlation in the attached sensors' readings. As we search for more of these correlated events over time, the probability of two streams being correlated by chance will drop exponentially and sift out the true relations. Following the intuition, our approach first detects the latent events in sensor time series by employing a bank of kinematic models – each modeling one of the latent states – and a latent variable to govern the switch between these models. Once the events are detected, we use a customized procedure to further pinpoint the correlated set of events that are indicative of the equipment relations.

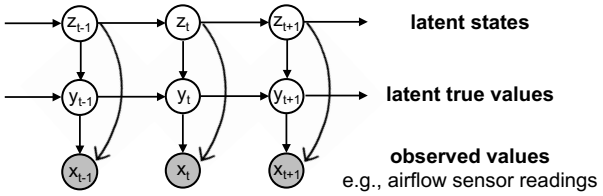


Figure 3: Graphical model representation of the proposed Markovian Event Model.

3.1 The Markovian Event Model

An important first step of our approach is to detect events in the time series data from sensors measuring the physical world, e.g., an air flow meter in a VAV that conditions a room. As Figure 2 shows, events can appear in different forms, from outlier values (A and B) to gradual changes (C and D), or discontinuities. Observing this variety, we desire a unified model that can differentiate normal operation from other modes of operation, by characterizing the transition and noise patterns. Overall, there are three layers in our proposed event model, as illustrated in Figure 3 (from top to bottom): 1) a sequence of switching variables that determine the states (e.g., steady or event) behind the data; 2) a sequence of latent variables modeling the "true" values of the data, whose parameters are determined by the switching variable at the top; and 3) the observed primitive sensor readings. We model the latent true values of time series based on the observed values as we assume the observed time series is contaminated by unknown noise. Particularly, for the latent true values, we model both the value itself and its "rate of change" of the time series readings. Considering the rate of change makes the model more expressive in that it can better characterize the time series in cases of abrupt change, e.g., a swarm of persons entering a room will cause a sudden increase in the air flow of HVAC system. In addition, we also assume the current true values are dependent on the previous ones, and the transition mode (e.g., how fast the transition is) would essentially reveal the underlying state. Furthermore, using the top layer, we follow the intuition that each latent true value is inherently generated from an underlying latent state. We shall note that, different from previous works, the parameters (e.g., variance) of each state in our model are learned globally and apply to all the data points in time, thus allowing data in seemingly different states (e.g., the two steady plateaus in Figure 2) to be modeled as in the same state.

Given the model structure specified in Figure 3, let $\mathbf{x}_t = [\mathbf{x}(t), \mathbf{v}_x(t)]^\top \in \mathbb{R}^2$ be the observed continuous readings from a particular sensor, where $\mathbf{x}(t)$ is the observed sensor value at timestamp t, and $\mathbf{v}_x(t)$ is the corresponding rate of change, for $t \in \{1, 2, \ldots, N\}$. Likewise, we define the latent true values $\mathbf{y}_t = [\mathbf{y}(t), \mathbf{v}_y(t)]^\top \in \mathbb{R}^2$, where $\mathbf{y}(t)$ and $\mathbf{v}_y(t)$ are the latent true sensor value and rate of change at

time t, respectively. For the latent discrete state variables $\mathbf{z}_{1:N}$, each $\mathbf{z}_t$ takes on a value from $\{1, 2, \ldots, K\}$, implying a total of K possible states in data at each time t. Therefore, the joint distribution over the observed values $\mathbf{x}_{1:N}$, the latent true values $\mathbf{y}_{0:N}$, and the latent states $\mathbf{z}_{0:N}$ is given by

$$p(\mathbf{x}_{1:N}, \mathbf{y}_{0:N}, \mathbf{z}_{0:N}) = p(\mathbf{z}_0)p(\mathbf{y}_0|\mathbf{z}_0)\prod_{t=1}^{N} p(\mathbf{z}_t|\mathbf{z}_{t-1})p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{z}_t)p(\mathbf{x}_t|\mathbf{y}_t, \mathbf{z}_t), \quad (1)$$

where $\mathbf{z}_0$ and $\mathbf{y}_0$ stand for the initial latent state and latent true values, respectively.

For the latent states $\mathbf{z}_{1:N}$, we assume they form a first-order Markov chain and the transition probability is given by $p_{ij} = p(\mathbf{z}_t = j|\mathbf{z}_{t-1} = i), 1 \le i, j \le K$. Thus, p_{ij} is modeled by a K-dimensional multinomial distribution. As we assume the observations are transformed from the latent true values subject to underlying noise, we assume the observed values are drawn from a Gaussian distribution centered at the latent true values. As a result, conditioned on the latent state $\mathbf{z}_t$, the probability of observing sensor reading data $\mathbf{x}_t$ is given by

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_t, \mathbf{z}_t) &= N(\mathbf{H}_{\mathbf{z}_t}\mathbf{y}_t, \mathbf{R}_{\mathbf{z}_t}) \quad (2)\\ &\propto \exp\left[\left(\mathbf{H}_{\mathbf{z}_t}\mathbf{y}_t - \mathbf{x}_t\right)^\top \mathbf{R}_{\mathbf{z}_t}^{-1}\left(\mathbf{H}_{\mathbf{z}_t}\mathbf{y}_t - \mathbf{x}_t\right)\right] \\ &= \exp\left[\left(\mathbf{y}_t - \mathbf{H}_{\mathbf{z}_t}^{-1}\mathbf{x}_t\right)^\top \left(\mathbf{H}_{\mathbf{z}_t}^{-1}\mathbf{R}_{\mathbf{z}_t}\left(\mathbf{H}_{\mathbf{z}_t}^{-1}\right)^\top\right)^{-1}\left(\mathbf{y}_t - \mathbf{H}_{\mathbf{z}_t}^{-1}\mathbf{x}_t\right)\right], \end{aligned}$$

where $N(\cdot, \cdot)$ denotes a Gaussian distribution, $\mathbf{H}_{\mathbf{z}_t} \in \mathbb{R}^{2\times 2}$ is a matrix that linearly maps the hidden true values $\mathbf{y}_t$ to the expectation of the observation $\mathbf{x}_t$, and $\mathbf{R}_{\mathbf{z}_t} \in \mathbb{R}^{2\times 2}$ is the corresponding covariance matrix capturing the noise embedded in this transformation. For the latent true values $\mathbf{y}_t$, we assume they also form a Markov chain and the transition probability follows Gaussian as well, i.e.,

$$p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{z}_t) = N(\mathbf{F}_{\mathbf{z}_t}\mathbf{y}_{t-1}, \mathbf{Q}_{\mathbf{z}_t}), \quad (3)$$

where $\mathbf{F}_{\mathbf{z}_t} \in \mathbb{R}^{2\times 2}$ is the state transition matrix between $\mathbf{y}_t$ and $\mathbf{y}_{t-1}$, and $\mathbf{Q}_{\mathbf{z}_t} \in \mathbb{R}^{2\times 2}$ is the covariance matrix capturing noise in the state transition.

3.2 Posterior Inference and Model Estimation

The key in applying the proposed model for event extraction is to infer the latent states $\mathbf{z}_{1:N}$ in a given time series $\mathbf{x}_{1:N}$. Based on our first-order Markovian assumption in state transition, we develop an efficient posterior inference algorithm based on Gibbs sampling techniques [10]. Particularly, given our model structure in Figure 3, we can obtain the conditional probability of $\mathbf{y}_t$ as

$$\begin{aligned} &p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{y}_{t+1}, \mathbf{x}_t, \mathbf{z}_t) \\ &\propto p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{z}_t)p(\mathbf{y}_{t+1}|\mathbf{y}_t, \mathbf{z}_{t+1})p(\mathbf{x}_t|\mathbf{y}_t, \mathbf{z}_t), \end{aligned} \quad (4)$$

where the first and second terms on the right-hand side follow Eq. (3), and the third term is given in Eq. (2). Due to the Gaussian assumption about the emission and transition probabilities, the product term here can still be analytically derived as a composite Gaussian distribution [3]. We can hereby sample the latent variable $\mathbf{y}_t$ from the composite Gaussian distribution, whose parameters are derived based on the above three Gaussian density functions. For sampling the latent state $\mathbf{z}_t$, we similarly have the conditional probability as follows:

$$\begin{aligned} &p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{z}_{t+1}, \mathbf{x}_t, \mathbf{y}_t) \quad (5)\\ &\propto p_{\mathbf{z}_{t-1},\mathbf{z}_t}p_{\mathbf{z}_t,\mathbf{z}_{t+1}}p(\mathbf{x}_t|\mathbf{y}_t, \mathbf{z}_t)p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{z}_t). \end{aligned}$$

We can then draw samples for each of the variables by sweeping through all the posterior conditionals. The theory of MCMC guarantees that the stationary distribution of the samples simulated under the Gibbs sampling algorithm is the target joint posterior that we are interested in [11]. Therefore, we run for a sufficient number of iterations of the sweeping process for each variable, and use the converged results to approximate the target posteriors. The aforementioned posterior inference procedures depend on the availability model parameters, i.e., $\{\mathbf{F}_i, \mathbf{H}_i, \mathbf{Q}_i, \mathbf{R}_i\}_{i=1}^K$. In theory, we can estimate the parameters using Maximum Likelihood Estimation (MLE). However, as the sensor time series data in our problem reflects the physical property of the world, it is straightforward to assume the values (i.e., $\mathbf{x}_t$ and $\mathbf{y}_t$) follow kinematics — sensor reading values at the next timestamp are based on the current one plus the product of rate of change and time difference. As a result, for our problem, we fix $\mathbf{F} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ and $\mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ regardless of the value of $\mathbf{z}_t$, to reflect kinematics.

In order to estimate $\{\mathbf{Q}_i, \mathbf{R}_i\}_{i=1}^K$, we maximize the complete data log-likelihood function defined by Eq. (1) based on the sampled results of $\mathbf{z}_{1:N}$ and $\mathbf{y}_{1:N}$. As the state transition follows a Gaussian distribution, the maximization problem has a closed form solution; and for brevity, we only give the conclusion below:

$$\hat{\mathbf{Q}} = \frac{1}{N}\sum_t (\mathbf{y}_t - \mathbf{F}\mathbf{y}_{t-1})(\mathbf{y}_t - \mathbf{F}\mathbf{y}_{t-1})^\top. \quad (6)$$

Note that, we dropped the subscription $\mathbf{z}_t$ in $\mathbf{y}$ and $\mathbf{Q}$ to avoid cluttered notations. Specifically, when estimating $\mathbf{Q}$ for each state (i.e, $\mathbf{Q}_i$), one should only collect the inferred $\mathbf{y}_{1:N}$ values in this latent state. Similarly, we can estimate $\mathbf{R}$ with

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_t (\mathbf{x}_t - \mathbf{H}\mathbf{y}_t)(\mathbf{x}_t - \mathbf{H}\mathbf{y}_t)^\top. \quad (7)$$

We can estimate the transition probability p_{ij} for $\mathbf{z}$ by collecting the sufficient statistics

$$p_{ij} = \frac{\sum_{\{t|\mathbf{z}_{t-1}=i, \mathbf{z}_t=j\}} 1}{N} \quad \text{for } 1 \le i, j \le K, \quad (8)$$

which basically summarizes the percentage of transitions from state i to j in the sampled $\mathbf{z}_{1:N}$ sequence.

Putting together the aforementioned posterior inference and parameter estimation procedures, we develop a stochastic Expectation-Maximization algorithm to iteratively refine the model parameters and the latent variable inference:

E-Step: At each timestamp t, infer the latent variables $\mathbf{y}_t, \mathbf{z}_t$ by Gibbs sampling based on the current model parameters (i.e., $\{\mathbf{Q}_i, \mathbf{R}_i\}_{i=1}^K$ and $\{p_i\}_{i=1}^K$) using Eq. (4)-(5).

M-Step: Update the model parameters based on the collected sufficient statistics of each latent variable, following Eq. (6)-(8).

We shall note that the sampling procedure for the latent variables in the E-step can be performed in time linearly with the length of time series N and further efficiently expedited via *parallelism*: because each variable depends only on the adjacent nodes before and after, we can update all the variables associated with even timestamps in parallel with all their neighbors at odd timestamps

fixed, or vice versa. We alternate between these two groups and such parallelism helps us empirically achieve more than 30x speedup for the inference in our evaluation datasets.

3.3 State-dependent Transitions

In practice, when the data stabilizes within one state, it is reasonable to have the velocity of change follow the first-order model we specified earlier, i.e., the velocity at the current timestamp depends on the one at a previous timestamp. However, when the time series transits between states, e.g., from steady into events, or vice versa, a sudden change in velocity is expected. Still enforcing the first-order transition as in Eq. (3) will cause unnecessary early reaction or delay in latent velocity inference. In other words, the first-order dependency will enforce the velocity to start increasing ahead of events when transiting from steady into events (for example, from the first plateau to C in Figure 2), or stay decreasing post-events after exiting from events to steady.

To account for possible sudden changes in velocity across state transition boundaries, it is necessary to eliminate the dependency on its neighbors for velocity modeling in these boundary cases, i.e., the velocity at a timestamp such that at least one of its two adjacent readings is in a different latent state than its own. This can be intuitively understood as the system being perturbed by some sudden external stimulus, e.g., the AHU has been shut down. Basically, we have two different scenarios: 1) one of the adjacent readings is in a different state, and 2) both of the adjacent readings are in a different state.

For the first scenario, when modeling the posterior of $\mathbf{y}_t$, we need to modify the form of its conditional probability by only preserving the emission probability and one direction of the transition probability. First, when $\mathbf{z}_{t+1}$ and $\mathbf{z}_t$ are different, we use the emission probability $p(\mathbf{x}_t|\mathbf{y}_t)$ and the transition probability $p(\mathbf{y}_t|\mathbf{y}_{t-1})$, i.e., (note the difference below from Eq. (4))

$$p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{y}_{t+1}, \mathbf{x}_t, \mathbf{z}_t \neq z_{t+1}) \propto p(\mathbf{x}_t|\mathbf{y}_t)p(\mathbf{y}_t|\mathbf{y}_{t-1}), \quad (9)$$

where the first term on the right-hand side is given in Eq. (2), and the second term is given in Eq. (3). We shall note that the above function only applies to the sampling for the velocity dimension (recall that $\mathbf{y}_t \in \mathbb{R}^2$). Due to the Gaussian assumption about the emission and transition probabilities, the posterior distribution in the transition case here can still be analytically derived as a composite Gaussian distribution.

When $\mathbf{z}_{t-1}$ and $\mathbf{z}_t$ are different, to capture the unknown external stimulus that affect the sampling of $\mathbf{y}_t$, we introduce a global prior distribution of the latent true velocity $p(\mathbf{y}_t|\theta_{\mathbf{z}_t})$ that is also state-dependent:

$$p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{y}_{t+1}, \mathbf{x}_t, \mathbf{z}_{t-1} \neq \mathbf{z}_t) \propto p(\mathbf{x}_t|\mathbf{y}_t)p(\mathbf{y}_{t+1}|\mathbf{y}_t)p(\mathbf{y}_t|\theta_{\mathbf{z}_t}). \quad (10)$$

For the second scenario, when modeling the posterior of $\mathbf{y}_t$, we adjust the conditional probability to include only the emission probability, together with the prior probability, i.e.,

$$p(\mathbf{y}_t|\cdot) \propto p(\mathbf{x}_t|\mathbf{y}_t)p(\mathbf{y}_t|\theta_{\mathbf{z}_t}). \quad (11)$$

3.4 Inferring Relations via Detected Events

The search space for equipment relation inference in a typical commercial building is huge. For a set of N sensors, the number of true relationships of interest is on the order of $O(kN)$, given the fact that each sensor is associated with k other sensors with respect to a particular relation; while the total number of pairwise relationships among them is on the order of $O(N^2)$. As we scale to thousands of sensors in a building, the total number of possible relations quadratically outnumbers the true relations of interest.

To facilitate the inference process, our basic assumption for identifying the relations is that connected equipment will be exposed to the same events in the physical world, and thus will exhibit correlated changes in the data. Yet, a key challenge is that the data streams are affected by a mixture of unobserved factors, not just the equipment relationships. For example, false correlations can be created by diurnal weather patterns, changes in room occupancy, and other external events. Thus, we need to filter the detected events and only look for co-incident events between a pair of equipment. To this end, we exploit the observation that, whenever there are events in an AHU, e.g., the outlet fan speed increases, these events will affect the downstream VAVs that connect to it. In other words, we should only look at the events in each VAV caused by the AHU for correlation, as they are the ones that reveal the true functional relation between them.

We shall also note that, as it often takes time for a VAV to respond to changes from the connected AHU, for each detected event in a VAV we make duplicates in its precedent 15-minute long time window, in order to compensate for potential *time lags* in events. We will later experimentally justify the choice in the evaluation. Assuming that the resultant events in a VAV would align temporally with its connected AHU's events, we therefore filter spurious events by *masking* each VAV event sequence with AHU event timestamps — we only preserve the (detected) events in a VAV that align in time with the events in a given AHU, and remove the others, i.e.,

$$\mathbf{z}^{(i,j)}_{VAV}(t) = \mathbf{z}^{i}_{VAV}(t) \,\&\, \mathbf{z}^{j}_{AHU}(t)\,, \text{for } t \in \{1, \ldots, N\}, \quad (12)$$

where $\mathbf{z}^{(i,j)}_{VAV}(t)$ is the value at time t in the event sequence of VAV i masked by AHU sequence j, $\mathbf{z}^{j}_{AHU}(t)$ is the value at time t in the event sequence of AHU j, and & is the bitwise AND operation. To be more specific, when correlating the event sequence from a VAV sensor i with each of the candidate AHU sensor j, we first mask the VAV event sequence using the given candidate AHU event sequence, and then calculate the cosine similarity sim_{ij} by

$$sim_{ij} = \frac{\sum_t \mathbf{z}^{(i,j)}_{VAV}(t)\, \mathbf{z}^{j}_{AHU}(t)}{\|\mathbf{z}^{(i,j)}_{VAV}\| \cdot \|\mathbf{z}^{j}_{AHU}\|}, \quad (13)$$

where $\|\cdot\|$ is the L-2 norm of a vector. After that, we assign each VAV to the AHU with which it has the highest similarity score. The masking step helps to further concentrate the comparison between VAV and AHU around periods when there are events in the AHU, which we believe drive the events in its connected VAVs and are the key to identifying the relations. This can dramatically reduce the number of spurious relations and help to sift out pairs of equipment that are truly connected. Presumably, a masked VAV event sequence is expected to best correlate with its truly connected AHU among all the candidate AHUs.

4 EVALUATION

4.1 Dataset

We evaluate our approach using the data from 5 commercial buildings distributed across the U.S.: the number of VAVs varies from 100 to over 500, and the number of AHUs ranges from 5 to 13. The ground truth for VAV to AHU association is obtained from

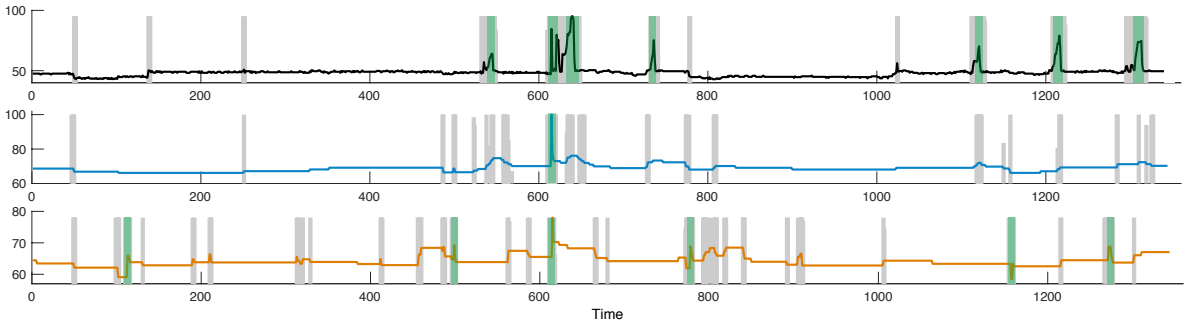


Figure 4: Examples of detected events for state number K=3 (events are marked with colored bars and the height of each bar indicates its probability) for a VAV (top), its associated AHU (mid), and a non-associated AHU (bottom).

Table 1: Details of the buildings used in our study, including the number of air handling units (AHU), variable air volume boxes (VAV), floors, and attached sensors.

Building ID	10312	10320	10596	10606	10642
# of AHU	12	8	5	13	8
# of VAV	261	113	195	510	259
# of Floor	9	1	5	4	8
# of Sensor	2754	1318	2010	4811	2379

the vendors of these buildings. The details of each building are summarized in Table 1. The typical number of sensing and control points attached to each equipment is between 6 and 13 (see Table 4 for example points), and the smallest building in our set has over 1,300 points. The number of connected VAVs for each AHU falls in anywhere from a handful to more than 80. We shall note that, oftentimes, the connected VAVs to the same AHU can span over different floors within a building, which leaves the manual perturbation-based solution [29] inefficient, inaccurate, albeit considerably time-consuming. For the time series data from each sensing point, it is recorded every 15 minutes. In each building we have collected 8-week worth of time series data for all sensors.

4.2 Baselines

Pearson Correlation Coefficient (CC): As a simple baseline, we calculate the pairwise Pearson Correlation Coefficient between each VAV and AHU, and assign a VAV to the AHU with the highest score.

StreamMine: This approach adaptively computes the principal components (PC) for a data stream and treats changes in the number of PCs as events, based on which it correlates streams [26].

Inspired by [6], we also compare our method with several baselines that combine the idea of time series clustering and bipartite matching. In particular, we first cluster the VAVs (using k-means with k set to the number of AHUs) and calculate the average linkage degree between every AHU and each group of VAVs. We then perform bipartite matching to assign each group of VAVs to one AHU, using the Ford-Fulkerson Maximum Flow Algorithm [9]. The linkage degree is defined as

$$l(AHU_i, C_j) = \frac{1}{|C_j|} \sum_{k \in \{C_j\}} d(AHU_i, VAV_k), \tag{14}$$

where $C_j, \{C_j\}, |C_j|$ refer to the j-th cluster of VAVs, the VAVs assigned to the cluster, and the size of the cluster, respectively. $d(a, b)$ calculates the cosine similarity between the two input streams, and we experiment with two different kinds of input streams – the primitive time series (TS) and the event sequence (ES). This gives us two baseline combinations, i.e., **KMeans-TS** and **KMeans-ES**.

As detecting events is an important intermediate step in our approach, we also consider the following baselines as alternatives for event detection, and the subsequent relation inference step remains the same as our approach.

Hidden Markov Model (HMM): We apply a K-state HMM—sensor observations depend on state changes and past sensor observations—to infer the latent state in the sensor time series data.

Kalman Filter (KF): We adopt a standard KF that models the rate of change in the time series data, and take as events the residues between the filtered values and observed values.

Adaptive Event Detection (AED): AED models a time series by combining two Poisson distributions – one for the normal periodic portion and the other for the rare event portion [16]. We replace the original Poisson distributions with Gaussians, to cater the continuous sensor readings in our problem.

Sliding Window Likelihood (SWL): We also compare with a method that employs sliding windows and estimates the likelihood of having an event in each window sequentially [28].

4.3 Experimental Setup

For the number of possible states K in sensor data, we set K=2 by default, if not specified otherwise: we assume the time series data is either in a steady state or an event state. For the first dimension of the input variable $\mathbf{x}$ to our algorithm (i.e., $\mathbf{x}(t)$), we use the observed sensor readings; and for the second dimension (the rate of change, $\mathbf{v}_x(t)$), we take the difference between two successive readings in the observed sensor time series. For the initial values of the latent true values $\mathbf{y}$ in our Gibbs sampling based posterior inference, we pass the raw sensor readings $\mathbf{x}$ through an Exponentially Weighted Moving Average filter (EWMA) [23] with a look-back window length (L = 5), and take the output as the initial values for $\mathbf{y}$. For the latent state variable $\mathbf{z}$, we assign random values from $\{0, 1\}$. Starting from the initial assignments, The sampled values by Gibbs sampling in the first several iterations may not necessarily represent the actual posterior distribution; and thus, it is common to discard these samples. The discarded iterations are often referred to as the "burn-in" period, and we set the number of burn-in we discard the samples from first $M_b(= 20)$ iterations. For the total number of samples M for each variable in Gibbs sampling, we set $M = 200$ and it is usually large enough to obtain converged results

Table 2: Our algorithm (MEMO, K=2) consistently outperforms the baselines w.r.t the VAV assignment accuracy (%).

Building ID	10312	10320	10596	10606	10642
CC	42.53	58.41	57.95	35.29	49.42
StreamMine	82.88	73.30	71.57	78.44	81.56
KMeans-TS	41.00	35.40	38.46	47.06	42.08
KMeans-ES	52.88	53.10	51.28	58.82	54.83
HMM	18.77	11.50	21.54	31.76	34.78
KF	90.04	80.53	87.69	80.29	88.91
AED	72.38	61.42	78.72	70.98	63.24
SWL	79.69	69.91	86.67	74.71	68.50
MEMO (K=4)	93.28	86.59	93.81	89.24	91.57
MEMO (K=3)	95.43	88.72	94.16	91.35	94.67
MEMO-NM (K=2)	96.17	84.96	95.38	89.41	94.59
MEMO (K=2)	**96.93**	**91.15**	**95.90**	**92.55**	**95.37**

on our dataset. For the stopping criterion for the EM procedure, we set a threshold on the log data likelihood difference between two successive iterations, with a sufficiently small value.

As there are often multiple sensors attached to each piece of equipment, we need to choose a pair of sensing points — one from VAV and one from AHU — to run our algorithm. In other words, knowledge about what type of sensors in different equipment best correlate and indicate the relations is required, and such knowledge can be readily provided by a building manager. This is the *only* input required from a human by our algorithm. In particular, in our experiments, we use `AirFlowVolume` in VAVs and `SupplyFanSpeed` in AHUs for evaluation, as the two are known to be physically correlated. We shall note that, in later section, we will demonstrate that it is possible to completely *automate* the selection of sensors for each equipment, with only minor degradation in performance.

4.4 Results & Analysis

• **Relation Inference Accuracy.** We first pass each time series through our Markovian Event Model (MEMO) to obtain a sequence of detected events. Figure 4 shows examples of detected event sequences from the air flow measurement of a VAV, the fan speed measurement of the associated AHU, and that of a non-associated AHU. To illustrate the power of MEMO in detecting events, we set K=3 in this case study, i.e., the model identifies three latent states. We label them as steady state, mild event state (in grey bars), and wild event state (in green bars). MEMO is able to detect the significant events, while ignoring lots of relatively strong noise (such as the small change around timestamp 100). By visually comparing three detected event sequences, we are able to recognize that the VAV event sequence is more closely aligned with the event sequence from the associated AHU, than the one from the non-associated AHU.

Formally, the assignment *accuracy* is defined as the proportion of VAVs that are correctly assigned to the associated AHU in a building. The main results are summarized in Table 2. On average, our algorithm achieves 94.38% accuracy across all the buildings, compared to 85.49% by the best baseline. First, we see that the combination of clustering and bipartite matching (KMeans-TS/ES) could not produce competitive results as our method, in large because the clustering step generates spurious groups of VAVs in the first place, and hence the significant matching errors. Yet, we do still see an average improvement of ~14% by KMeans-ES over KMeans-TS, where the former performs clustering over the event sequence. This shows that the detected events provide more accurate information about the correlation between equipment than the primitive time series. For StreamMine, it relies heavily on PCA to detect events, which is not robust to outliers and noise, and thus results in significant spurious correlations.

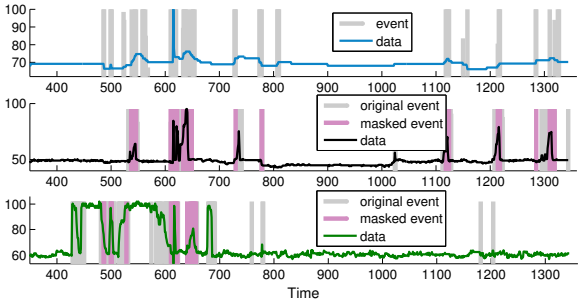
Table 3: Ranking quality of the true AHUs for our method (MEMO) and the best baseline (KF) by mean reciprocal rank (1 is perfect): for each building we measure the AHU ranking quality for all VAVs (first number) and for mis-assigned VAVs only (second number).

Building ID	10312	10320	10596	10606	10642
MEMO	0.89/0.44	0.85/0.36	0.88/0.43	0.86/0.39	0.91/0.48
KF	0.80/0.38	0.75/0.29	0.82/0.33	0.82/0.30	0.83/0.40

Among the baselines using alternative event detection methods, as expected, HMM does not perform well because it simply models the state at each point based on raw sensor reading value, thus producing almost random state assignments. KF performs the closest to ours, as it also considers the rate of change in the data, yet it only encodes the influence of past observations on the current values, but not the influence from future data (as standard KF only performs forward computation of expected values). For AED, because it relies on a normal periodic component to model the steady states, e.g., daily patterns, any slight deviations from the norm, such as stronger cooling due to temperature rise on a hotter day in our data, will be modeled as events, thus resulting in too many false positive events. In contrast with these methods, our algorithm complements the strength of a kinematic model by also taking into account the backward temporal information, i.e., the influence of future data, and holistically learns how data progresses in each state.

As we assign each VAV to the AHU with largest correlation score, and therefore, the true AHU for each VAV should ideally rank the first among all the AHUs. To take a closer look at the results, we measure the AHU *ranking quality* for VAVs in each building by Mean Reciprocal Rank, defined as $MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}$, where $rank_i$ is the ranking of the true AHU that VAV_i connects to, and N is the total number of VAVs in the building. We see from Table 3 that when we look at all VAVs, the ranking quality (the first numbers) is close to 1 and is consistent with our relation inference accuracy. If we look at the mis-assigned VAVs only, the ranking quality (the second numbers) is all above 0.36, meaning that the true AHUs are mostly ranked as the second best for these mis-assigned VAVs. This indicates that, although our solution makes mistakes, if someone were to manually figure out the connections, he/she would only need to inspect mostly two candidates, which still saves a significant mount of manual work. Comparing to the best baseline method, i.e., Kalman filter, it generally places the correct AHU for a VAV at a lower position, which leads to more manual inspection to find the correct alignment.

Overall, we attribute the better performance of our algorithm to its more accurate event detection, which produces less spurious candidates to compare against. The results imply that our algorithm could have profound impacts: with only limited knowledge about what type of sensors to use in the analysis, the algorithm is able to *automatically* identify the correct relations for about 94% of the equipment. Our solution can thus significantly reduce the manual



m

Figure 5: Event sequences from an AHU (top), a connected VAV (mid), and a non-connected VAV (bottom): preserving VAV events that overlap in time with AHU events (marked in pink) helps to better pinpoint the correlated set of events and infer the relations.

effort required in the relation identification process, as the current state-of-the-art industrial solution stands. One should note that, due to the absence of the event ground truth for the dataset, we were unable to directly evaluate the event detection accuracy. However, here we indirectly demonstrate the accuracy of event detection with a proxy, namely, the relation inference accuracy.

• **Effect of Masking.** In the proposed correlation analysis, we take the masking operation on detected event sequences, i.e., to preserve only the events in a VAV that overlap in time against a given AHU. To demonstrate the effect of masking, we show an illustrative example in Figure 5, where from top to bottom are readings from an AHU, its connected VAV, and a non-connected VAV, respectively (we set $K = 2$ to simplify the illustration). We see that, after masking the original events (in grey), the remaining events (in pink) in the connected VAV (mid) clearly better correlate with the AHU (top), than the non-connected VAV (bottom) does, which fails to correlate in the later part of sequence. More specifically, before and after the masking operation, the similarity score is 0.73 vs 0.87 between the top and mid, while the score is 0.59 vs 0.31 between the top and the bottom. As a result, masking clearly helps to concentrate on the potentially correlated set of events. As a comparison, we also experiment with using the primitive event sequences between VAV and AHU without the masking operation for the correlation step, i.e., we compute

$$sim_{ij} = \frac{\sum_t \mathbf{z}^i_{VAV}(t)\, \mathbf{z}^j_{AHU}(t)}{\|\mathbf{z}^i_{VAV}\| \cdot \|\mathbf{z}^j_{AHU}\|}, \tag{15}$$

and assign a VAV to the AHU with the highest correlation score. The results are reported in Table 2 (MEMO-NM), and we can observe clear degradation, especially in building 10320 and 10606.

Upon a closer inspection, we note that the VAV placement in these two buildings is significantly denser than the others – for these two buildings there are more than 100 VAVs on each floor, whereas for the others the number is between 20-30. The denser VAV placement in these two buildings means that each individual conditioning zone is essentially smaller, and thus more subject to the thermal influences from other nearby zones. This results more complex changes in the VAV data stream. As a result, using the masking operation on each VAV's event sequence, which is designed to filter out irrelevant events with respect to the equipment connection, our algorithm is able to better reveal the functional connections among the equipment.

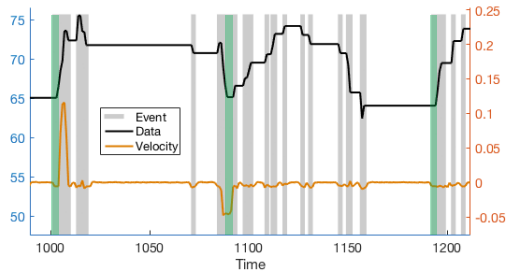


Figure 6: Without the independence constraint, the modeled velocity incurs early reactions and delays, consequently producing false positive events (marked in green).

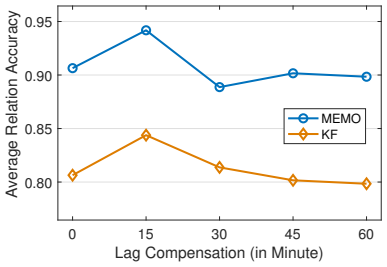


Figure 7: The effect of time lag compensation for events on average inference accuracy across all the test buildings: properly choosing a window size (15 minutes) helps to achieve the best results.

• **Effect of Velocity Independence.** We also examine the effect of relaxing dependency in velocity modeling at the transition periods, as explained in Section 3.3. To this end, we instead preserve the dependency between velocity during transition periods and inspect the inferred events and velocity. We see from Figure 6, the velocity reacts early or delays when dramatic changes occur, resulting in false positive events (marked in green). We conclude that adding independence into velocity modeling with respect to the latent state change benefits the event detection.

• **Effect of Time Lag Compensation for Events.** Before feeding the detected events into the correlating procedure, we compensate for potential time lags in the event sequence by duplicating every detected event in a nearby 15-minute long time window, as described in §3.4. We do so because we believe that, in practice, it usually takes time for changes in an AHU to reach and affect the connected VAVs, as the equipment often resides reasonably apart (e.g., on different floors or sides) in the building. To examine how the choice of lag compensation time window affects the relation inference accuracy, we vary the window size from 0 to 1 hour in 15-minute multiples, as our data sampling interval is 15 minutes. From Figure 7, we see that compensating for the time lag in events helps when the window size is 15 minutes. Allowing for too long a delay (e.g, 30 minutes or longer) actually degrades the performance, as it introduces too many spurious events for the correlation. And we observe the same pattern for the best baseline (KF). In addition, we speculate that the best compensation window size would be smaller than 15 minutes as changes are not expected to lag more than that amount. However, we are not able to verify this experimentally since the data is only reported every 15 minutes.

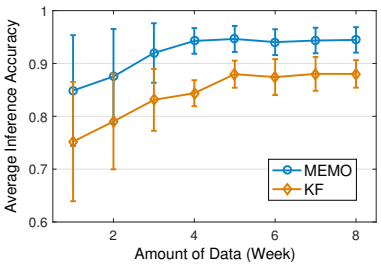


Figure 8: The effect of amount of data on average inference accuracy across all the test buildings.

• **Effect of Number of States.** For the set of results above, we set the number of states in the event detection model K=2, i.e., we assume there is either an event or none. Here we further inspect how the choice of K affects the relation inference performance. Particularly, we increased K to 3 and 4 (see MEMO (K=3, 4) in Table 2) and found that choosing a larger K decreased the relation inference accuracy. When the number of states increases, the types of inferred events become inconsistent across streams (see Figure 4); a wild event in one sequence might look like a relatively mild event in another, or vice versa. In other words, the alignment between different types of identified states is not guaranteed in our model, as the model parameters (i.e., $\{\mathbf{Q}_i, \mathbf{R}_i\}_{i=1}^K$) are not shared across sequences. We need to emphasize that as $\{\mathbf{Q}_i, \mathbf{R}_i\}_{i=1}^K$ specify the inherent noise in modeling the time series data from different pieces of equipment, it is not reasonable to share them across streams, which will enforce the same noise distribution on all streams. The joint modeling across sequences has to be perform from other perspectives, e.g., sharing the latent event sequence; and we leave it as our future work.

• **Effect of Amount of Data.** To examine the effect of the amount of training data on the relation inference accuracy, we vary the amount of data passed to our proposed algorithm from 1 week to 8 weeks. From Figure 8, we find that, as we increase the amount of time series data for model training, both our model (MEMO) and the best baseline improve in the inference accuracy and plateau after reaching around 4 to 5 weeks' data.

This also verifies our intuition that, as we search for more correlated events between points over a longer time period, the probability of two points being correlated by random chance drops, thus resulting in higher accuracy of relation inference. We also observe a similar trend for the best baseline (KF), yet its performance is still worse than our method.

4.5 Fully Automation for General Applicability

A premise in the previous evaluation is that we are given the knowledge about which points to use for correlation and that pair of points are consistently available across all buildings. However, such a premise might not always stand in practice, as the type of measurements available could vary from site to site, namely, in different buildings. For example, Table 4 summarizes the VAV assignment accuracy using different pairs of points in one of our test buildings. We see that the accuracy of using different pairs of points varies drastically, as some of the measurements do not necessarily correlate directly — for example, the air pressure of duct and the temperature of the room. Consequently, we need a strategy to automatically decide which pair of points to use, in order to make it generally applicable across sites, or even problem areas. The question raised here is, "*Can we automatically decide which pair of points to use to best identify the relations?*" As a futuristic direction, we make a preliminary attempt at automating the selection of points to use for identifying the relations.

Table 4: VAV assignment accuracy (%) by using different pairs of points in one of the buildings: the performance varies drastically from pair to pair.

VAV \AHU	Supply AirPressure	Supply AirTemp	Supply FanSpeed
AirFlowVolume	75.22	33.63	91.15
DischargeAirTemp	57.52	42.48	36.28
SpaceTemp	13.27	15.04	31.86

Table 5: Assignment accuracy using automatically selected pair of sensors (AutoPair) for correlation against the oracle accuracy.

Building ID	10312	10320	10596	10606	10642
Oracle	96.55	91.15	97.95	92.55	100.00
AutoPair	95.79	84.88	89.72	85.98	99.60

Intuitively, the points representative of the relations are expected to yield a "deviant" correlation score for the pair of truly associated equipment, compared to the other candidates. In other word, ideally, the correlation score between the pair of equipment that are truly associated should as much separate from the rest of pairs as possible. Following such an intuition, we design a metric to measure the deviation of the highest score produced by a particular pair of points. Given the m-th point from the i-th VAV and the n-th point from the j-th AHU, we obtain the similarity scores between that particular VAV and all candidate AHUs:

$$s^{m,n} = \left\{ sim_{i,j}^{m,n} \right\}, \text{ for } 1 \le j \le L, \tag{16}$$

where $sim_{i,j}^{m,n}$ stands for the correlation score between the i-th VAV with the j-th AHU using the m-th sensing point in VAV and the n-th sensing point in AHU, and j sweeps through all L AHU candidates in a given building. Then we measure the following quantity for each pair of points:

$$p(s^{m,n}) = \sigma(s^{m,n}) \times \left(1 - N\left(s_{\max}^{m,n} | \mu(s^{m,n} \setminus s_{\max}^{m,n}),\ \sigma(s^{m,n} \setminus s_{\max}^{m,n})\right)\right),$$

where the notation "\" denotes set difference, and $N(v|\mu, \sigma)$ gives the likelihood of observing v under the Gaussian distribution parameterized by the mean μ and standard deviation σ. Then for each VAV, we decide a pair of points to use by $\text{argmax}_{m,n} p(s^{m,n})$, and the rest of the assignment process remains the same as the original approach.

To evaluate the approach, we first need the oracle accuracy of using automatically chosen pair. To this end, we calculate the relation inference accuracy enumerating all pairs of points, such as presented in Table 4, and pick the highest accuracy achieved among all pairs as the oracle accuracy. This can be understood as the upper bound accuracy of our algorithm in this dataset. In Table 5, we summarize the accuracy using the pairs decided with our proposed metric, compared with the oracle accuracy. On average, while the

oracle hits 95.64%, our approach achieves 91.19%, which degrades only slightly compared to the 94.38% accuracy when given the pair of points to use a priori. We believe our approach is promising in automating the relation inference process and making it generally applicable across sites, or even domains.

5 DISCUSSION AND FUTURE WORK

We have demonstrated the effectiveness of our proposed solution in equipment relation inference. The intuition behind our solution is general: when groups of measurements exist such that they respond to common events, our technique can be applied to detect such groupings. Thus, we would expect our technique to apply to other domains or systems, for example, to infer the functional connections between the water-related and air-related HVAC equipment (e.g., cooling tower - AHU). Furthermore, the solution could be used in the reverse direction — to detect anomalies or failures when streams are known to be correlated but not observed so. For example, sensor failures or mis-configuration in building equipment could be detected if coincident changes are only observed in some sensor readings, but not in all.

As future work, we plan to perform joint learning for event detection and relation inference, so that the errors in relation inference can be propagated back to improve the event detection. In addition, We currently model each sensor time series independently for event detection, and it would be beneficial to incorporate the dependency between sequences into the modeling, such as VAVs connected to the same AHU might respond to the same set of events. This could further refine the inference results. We can also use the feedback in relation inference to guide the event detection, so as to improve both components.

6 CONCLUSION

In this study, we tackle the problem of automatically inferring the functional relations between equipment in buildings, a critical piece of information to various building analytics for energy savings, which is currently acquired via costly, error-prone manual processes. We build upon the intuition that connected equipment is exposed to the same real-world activities, thus exhibiting correlated changes in their time series data. Our solution first identifies the latent events in the time series data of equipment, and then sifts out the correlated set of events for inferring the underlying relations among equipment. We evaluated our approach on data from 5 commercial buildings and the approach achieves 94.38% accuracy, compared to the 85.49% by the best baseline. We believe the solution is promising and can be generally applied to event detection and relation inference in broader domains.

ACKNOWLEDGMENTS

We thank our shepherd, Yanyong Zhang, and the anonymous reviewers for helpful comments. This work was supported by National Science Foundation IIS-1718216 and Department of Energy DE-EE0008227.

REFERENCES

[1] Bharathan Balaji, Arka Bhattacharya, Gabriel Fierro, Jingkun Gao, Joshua Gluck, Dezhi Hong, Aslak Johansen, Jason Koh, Joern Ploennigs, Yuvraj Agarwal, et al. 2016. Brick: Towards a unified metadata schema for buildings. In *BuildSys*.

[2] Arka A Bhattacharya, Dezhi Hong, David Culler, Jorge Ortiz, Kamin Whitehouse, and Eugene Wu. 2015. Automated metadata construction to support portable building applications. In *BuildSys*. ACM, 3–12.

[3] Paul Bromiley. [n. d.]. Products and convolutions of Gaussian probability density functions. ([n. d.]).

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 15.

[5] Yueguo Chen, Mario A Nascimento, Beng Chin Ooi, and Anthony KH Tung. 2007. Spade: On shape-based pattern detection in streaming time series. In *ICDE*.

[6] Inderjit S Dhillon. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*. ACM, 269–274.

[7] DOE. 2015. Total annual cost of energy in the commercial and industrial sector. (2015).

[8] Bing Dong and Khee Poh Lam. 2014. A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting. In *Building Simulation*, Vol. 7. Springer.

[9] LR Ford and DR Fulkerson. 2009. Maximal flow through a network. In *Classic papers in combinatorics*. Springer, 243–248.

[10] Stuart Geman and Donald Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), 721–741.

[11] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. 1995. *Markov chain Monte Carlo in practice*. CRC press.

[12] David Hallac, Sagar Vare, Stephen Boyd, and Jure Leskovec. 2017. Toeplitz inverse covariance-based clustering of multivariate time series data. In *KDD*.

[13] Victoria Hodge and Jim Austin. 2004. A survey of outlier detection methodologies. *Artificial intelligence review* 22, 2 (2004), 85–126.

[14] Dezhi Hong, Jorge Ortiz, Kamin Whitehouse, and David Culler. 2013. Towards automatic spatial verification of sensor placement in buildings. In *BuildSys*.

[15] Dezhi Hong, Hongning Wang, Jorge Ortiz, and Kamin Whitehouse. 2015. The building adapter: Towards quickly applying building analytics at scale. In *BuildSys*.

[16] Alexander Ihler, Jon Hutchins, and Padhraic Smyth. 2006. Adaptive event detection with time-varying poisson processes. In *KDD*. ACM, 207–216.

[17] Simon J Julier and Jeffrey K Uhlmann. 1997. New extension of the Kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, Vol. 3068. International Society for Optics and Photonics, 182–194.

[18] Srinivas Katipamula and Michael R Brambley. 2005. Methods for fault detection, diagnostics, and prognostics for building systemsâĂŤa review, part I. *Hvac&R Research* 11, 1 (2005), 3–25.

[19] Yoshinobu Kawahara and Masashi Sugiyama. 2009. Change-point detection in time-series data by direct density-ratio estimation. In *ICDM*. SIAM, 389–400.

[20] Merthan Koc, Burcu Akinci, and Mario Bergés. 2014. Comparison of linear correlation and a statistical dependency measure for inferring spatial relation of temperature sensors in buildings. In *BuildSys*. ACM, 152–155.

[21] Jason Koh, Bharathan Balaji, Vahideh Akhlaghi, Yuvraj Agarwal, and Rajesh Gupta. 2016. Quiver: Using control perturbations to increase the observability of sensor data in smart buildings. *arXiv preprint arXiv:1601.07260* (2016).

[22] Jason Koh, Bharathan Balaji, Dhiman Sengupta, Julian McAuley, Rajesh Gupta, and Yuvraj Agarwal. 2018. Scrabble: transferrable semi-automated semantic metadata normalization using intermediate representation. In *Proceedings of the 5th Conference on Systems for Built Environments*. ACM, 11–20.

[23] James M Lucas and Michael S Saccucci. 1990. Exponentially weighted moving average control schemes: properties and enhancements. *Technometrics* 32, 1 (1990), 1–12.

[24] Yudong Ma, Francesco Borrelli, Brandon Hencey, Brian Coffey, Sorin Bengea, and Philip Haves. 2012. Model predictive control for the operation of building cooling systems. *IEEE Transactions on control systems technology* 20, 3 (2012).

[25] Microsoft. 2016. Data analytics and smart buildings increase comfort and energy efficiency. *https://tinyurl.com/yac59pqx* (2016).

[26] Spiros Papadimitriou, Jimeng Sun, and Christos Faloutsos. 2005. Streaming pattern discovery in multiple time-series. In *VLDB*. VLDB Endowment, 697–708.

[27] June Young Park, Bertrand Lasternas, and Azizan Aziz. 2018. Data-Driven Framework to Find the Physical Association between AHU and VAV Terminal Unit–Pilot Study. (2018).

[28] Dan Preston, Pavlos Protopapas, and Carla Brodley. 2009. Event discovery in time series. In *ICDM*. SIAM, 61–72.

[29] Marco Pritoni, Arka A Bhattacharya, David Culler, and Mark Modera. 2015. Short paper: A method for discovering functional relationships between air handling units and variable-air-volume boxes from sensor data. In *BuildSys*. ACM, 133–136.

[30] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In *WWW*.

[31] Yasushi Sakurai, Spiros Papadimitriou, and Christos Faloutsos. 2005. Braid: Stream mining through group lag correlations. In *SIGMOD*. ACM, 599–610.

[32] Anika Schumann, Joern Ploennigs, and Bernard Gorman. 2014. Towards automating the deployment of energy saving approaches in buildings. In *BuildSys*.

[33] Virginia Smith, Tamim Sookoor, and Kamin Whitehouse. 2012. Modeling building thermal response to HVAC zoning. *ACM SIGBED Review* 9, 3 (2012), 39–45.

[34] Kenji Yamanishi and Jun-ichi Takeuchi. 2002. A unifying framework for detecting outliers and change points from non-stationary time series data. In *KDD*.