Scalable Causal Learning for Predicting Adverse Events in Smart Buildings

Aniruddha Basak
Carnegie Mellon University, Silicon Valley Campus
Moffett Field, CA 94035
abasak@cmu.edu

Ole Mengshoel
Carnegie Mellon University, Silicon Valley Campus
Moffett Field, CA 94035
ole.mengshoel@sv.cmu.edu

Stefan Hosein
University of the West Indies, St. Augustine
St Augustine, Trinidad & Tobago
stefan.hosein2@my.uwi.edu

Rodney Martin
NASA Ames Research Center
Moffett Field, CA 94035
rodney.martin@nasa.gov

Abstract

Emerging smart buildings, such as the NASA Sustainability Base (SB), have a broad range of energy-related systems, including systems for heating and cooling. While the innovative technologies found in SB and similar smart buildings have the potential to increase the usage of renewable energy, they also add substantial technical complexity. Consequently, managing a smart building can be a challenge compared to managing a traditional building, sometimes leading to adverse events including unintended thermal discomfort of occupants (too hot or too cold). Fortunately, todays smart buildings are typically equipped with thousands of sensors, controlled by Building Automation Systems (BASs). However, manually monitoring a BAS time series data stream with thousands of values may lead to information overload for the people managing a smart building. We present here a novel technique, Scalable Causal Learning (SCL), that integrates dimensionality reduction and Bayesian network structure learning techniques. SCL solves two problems associated with the naive application of dimensionality reduction and causal machine learning techniques to BAS time series data: (i) using autoregressive methods for causal learning can lead to induction of spurious causes and (ii) inducing a causal graph from BAS sensor data using existing graph structure learning algorithms may not scale to large data sets. Our novel SCL method addresses both of these problems. We test SCL using time series data from the SB BAS, comparing it with a causal graph learning technique, the PC algorithm. The causal variables identified by SCL are effective in predicting adverse events, namely abnormally low room temperatures, in a conference room in SB. Specifically, the SCL method performs better than the PC algorithm in terms of false alarm rate, missed detection rate and detection time.

Introduction

NASA Ames Sustainability Base[1] (SB) is a green building that provides a research testbed for different sustainable technologies and concepts. The SB is designed with a Net Zero Energy objective. Detailed monitoring of the BAS is required at regular intervals. SB is instrumented with 2636 sensors, which perform physical or logical measurements.

One major area of consumption is the building heating and cooling system. From Jan 2014 to May 2014 many alarms were acknowledged from sensors specific to the heating system. An alarm is initiated whenever a sensor value goes beyond the desired range, indicating an anomalous behavior of the heating system. Some alarms, e.g room temperature going outside the predefined range, cause occupants' discomfort and others can lead to short-term or long-term damage to the system. In order to eliminate these alarms, it is crucial to identify the causes. An attempt via human inspection is time consuming and can take several weeks or months. As the sensor data captures enough information about the system, an alternative for cause-detection is a data driven approach which does not require intervening with the BAS of SB building.

Causality is mainly described by two methods: counterfactuals or causal graphs (Pearl 2000). Here we primarily consider the second one. A causal graph is based on the principle that each variable is independent of its non-effects, conditional on its direct causes. PC, FCI, RFCI and GES are commonly used algorithms to learn causal graphs from data (Spirtes, Glymour, and Scheines 2000; Chickering 2003; Colombo et al. 2012). Bayesian networks have also been shown to convey causal interpretation (Pearl, Verma, and others 1991).

A Bayesian network (BN) is a directed acyclic graph (DAG) which represents a family of distributions over its nodes. The structure of a BN represents the conditional independence relations among the nodes. Hence learning the independence relations by conditional independence tests is a main approach in learning the structure of a Bayesian network from data (Margaritis and Thrun 1999). Another structure learning approach is to perform a search over directed graphs based on a score function (Heckerman, Geiger, and Chickering 1995). Hybrid techniques have also been developed to learn BN structure (Tsamardinos, Brown, and Aliferis 2006). However identifying the causal Bayesian network structure only from observations is challenging because it is possible to model one distribution by two distinct directed graphs which have different causal meaning of the variables (Ellis and Wong 2008).

In this work we propose a novel amalgamation (Figure

[1] http://www.nasa.gov/ames/facilities/sustainabilitybase

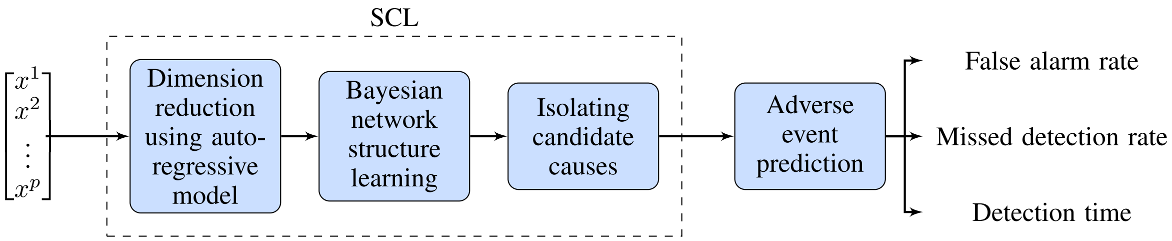

SCL
x^1
x^2
$\vdots$
x^p
Dimension reduction using autoregressive model
Bayesian network structure learning
Isolating candidate causes
Adverse event prediction
False alarm rate
Missed detection rate
Detection time

Figure 1: Data processing steps

1) of autoregressive model and Bayesian network structure learning techniques to learn the candidate causes of a target from observational data. Using only autoregressive model for causal learning can lead to spurious causes and learning a Bayesian network over all variables demands huge compute time. Our novel integration solves both problems enabling accurate identification even in large scale data sets. Thus we call this method Scalable Causal Learning (SCL).

We test the SCL method by attempting to identify the causes of too low room temperatures in an SB conference room (target effect). Due to the HVAC system in SB, there are strong correlations among the room temperature sensor readings of the building. Further, what a building manager can directly control is the HVAC system, not room temperature. Consequently, we remove all room temperature data, except the temperature for the target conference room, from the SB dataset. Applying SCL to this SB dataset, we find that the isolated causes make functional sense according to the building's operation model. As the true causes are unknown, we measure the quality of the causal set by using them as features in adverse event prediction of the target variable. We have seen that the causes identified by our method performs better than the causes identified by PC algorithm in terms of false alarm rate, missed detection rate and detection time (Osborne et al. 2012).

Related Research

Bayesian networks are often used for anomaly detection or fault detection in dynamical systems. Matsuura et al. used a Bayesian network to model a dynamical system operating under normal conditions and thereby predicting low probability events as faults (Matsuura and Yoneyama 2004). Their approach performed better than the Luenberger observer technique. A similar technique was proposed for multivariable systems/agents (Guo et al. 2011). Guo et al. used a Dynamic Bayesian network to detect faults in heating, ventilation and air-conditioning (HVAC) systems. The authors showed that the method was advantageous over rule-based fault detection and could detect most faults in the physical system. Unlike conditioning, blocking operator has also been introduced for causal discovery in relational data sets (Rattigan, Maier, and Jensen 2011).

Isolating cause effect relations: Machine learning techniques have been used for automated debugging through isolation of cause-effect chains (Jiang and Su 2005). Qui et al. used Granger graphical models to detect anomaly in time-series data (Huida Qiu, Subrahmanya, and Li 2012). They used temporal causal graphs to identify instances deviating from the normal pattern of the data sequence.

Bayesian networks were also used for causal discovery in multivariate time series (Wang and Chan 2011). The authors applied Bayesian network learning algorithm to construct structural vector autoregression (SVAR) from time series data. This captured both temporal and contemporaneous causal relations. Although our work addresses a similar problem, the use of VAR and Bayesian networks are very different in our case. Moreover, Wang et al. used SVAR model for causal discovery but we developed a method to directly use Bayesian networks to isolate causes for a target.

Notation

We denote i^{th} time series of the data set ($\mathcal{D}$) by x^i. And x^* represents the target variable for which candidate causes are identified. To indicate time-lagged signals we add a subscript. For instance, x^i_{-n} indicates the i^{th} time series shifted n steps backward in time. We use X_{-n} to represent the set of all signals lagged by n time steps. The number of variables in $\mathcal{D}$ is denoted by p and the set of indices for these variables is represented by $\mathcal{D}_I$. $\mathbb{Z}$ indicates the set of natural numbers.

Structure learning from high dimensional data

Learning the structure of a Bayesian network is a very computationally intensive task (Daly, Shen, and Aitken 2011). Figure 2 shows the execution time of learning a Bayesian network structure using two algorithms, Hill climbing (HC) (Heckerman, Geiger, and Chickering 1995) and PC (Spirtes, Glymour, and Scheines 2000), on random data sets (normally distributed with zero mean and unit variance) with 25000 observations each. The experiment was performed in a machine with eight core 2.20GHz Intel Core i7 processor and 8GB of RAM. We see that the execution time increases super-linearly with increasing number of variables and the curves almost resembles exponential rise.

The Proposed Method

We propose a new technique to learn candidate causes of one target variable using a score based structural learning algorithm as opposed to discovering all causal relations between all variables in the data. A flow diagram consisting of all our data analysis steps is presented in Figure 1.

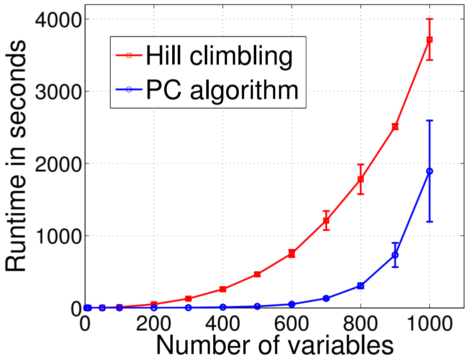


Figure 2: Comparison of execution times of 2 structure learning algorithms.

Reducing Dimensionality

In an attempt to reduce the number of nodes in the network, we filter some variables in the data using linear regression. As candidate causes should help predicting future values of the target, we use a regularized regression technique to remove non-informative variables. This step makes our approach scalable to large scale data sets.

Autoregressive (AR) models are often used in economics and for modeling time-varying natural processes (Kelejian and Prucha 2010; Chakraborty et al. 2012). We used an autoregressive model of order τ, AR(τ), to express the target as a linear combination of all time-lagged variables,

$$\begin{aligned} y_t^{(j)} &= \mathbf{a}_1^T \mathbf{y}_{t-1} + \mathbf{a}_2^T \mathbf{y}_{t-2} + \cdots + \mathbf{a}_\tau^T \mathbf{y}_{t-\tau} + e_t^{(j)} \\ &= \beta^T \mathbf{Y}_{t-1,t-\tau} + e_t^{(j)} \end{aligned} \qquad (1)$$

where $y_t^{(j)} = x^*$ is the target, $\mathbf{y}_t \in \mathbb{R}^p$ is a vector containing the values of all variables at time t and a_t is the corresponding weight vector. $\mathbf{Y}_{t-1,t-\tau} \in \mathbb{R}^{p\tau}$ and $\beta \in \mathbb{R}^{p\tau}$ concatenates the variables and weights respectively. To reduce non-informative variables we train the model with a sparsity constraint. Our optimization formulation is as follows,

$$\hat{\beta}_L = \underset{\beta \in \mathbb{R}^{p\tau}}{\arg\min} \sum_{t=\tau}^{T} \left(y_t^{(j)} - \beta^T \mathbf{Y}_{t-1,t-\tau} \right)^2 + \lambda ||\beta||_L \qquad (2)$$

where the regularization norm L is chosen to be 1 or 2; and λ controls the amount of shrinkage. We select the first k variables, sorted in decreasing order of the weights in the trained model $\hat{\beta}_L$, as *informative variables*. Only these variables (say $\bar{X} \in \mathbb{R}^k$ with $k < p$) are used to construct a Bayesian network. Thus the autoregressive model serves as a dimension reduction step and reduces the computational cost of structure learning step.

Learning Bayesian Networks

As causes must precede the effect in time, we attempt to construct a Bayesian network with all variables shifted backwards in time and the target in real time. The nodes of the network are $\{\bar{X}_{-1}, \bar{X}_{-2}, \cdots, \bar{X}_{-\tau}, x^*\}$. To restrict the search space towards feasible causal structures, we introduce a few temporal constraints to the search. No outgoing edge for the target and no direct edge going backwards in time for a particular variable is permissible in the network. Hence, the following types of edges are avoided:

$$\begin{aligned} & x^* \to \bar{x}^i_{-t} \quad \forall i \in [1,k] \\ & \bar{x}^i_{-t+j} \to \bar{x}^i_{-t} \quad \forall i \in [1,k], j \in [1,t] \subset \mathbb{Z} \end{aligned} \qquad (3)$$

As our objective is not to learn a fully causal network, we reduce the number of constraints by enforcing the second constraint to individual variables as opposed to all pairs of variables. Preventing all possible direct edges going backwards in time require $O(k^2\tau^2)$ constraints. In our method, only $O(k\tau^2)$ or $O(p\tau^2)$ (as $k < p$) constraints are needed.

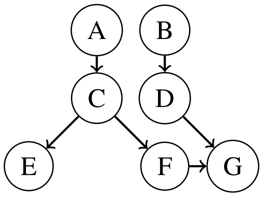


Figure 3: An example Bayesian network

Isolating Candidate Causes

The parents of the target variable are natural choices for candidate causes. However, these parents can be the effect of target's grandparents and so forth. Events that create cause-effect chains spanning multiple time-steps may lead to this kind of structure in a directed graph. Thus we consider all variables in all paths from target to root nodes as candidate causes. For instance, the candidate causes for G in the network shown in Figure 3 are $\{A, B, C, D, F\}$. The set of candidate causes is denoted by C_{CD} according to the causal discovery algorithm CD.

Predicting Adverse Events

Once the candidate causes are identified, we train models to predict adverse events in the target variable using feature set C_{CD}. For model training, Adverse Condition and Critical Event Prediction Toolbox (ACCEPT)[2] (Martin et al. 2015) is used. ACCEPT's main goal is to provide the ability to predict or forecast adverse events in time series data. Moreover, it provides a single, unifying framework to compare multiple combinations of algorithms concurrently, without having to rerun the system for each new algorithm and gather results separately. Further, its architecture is patterned after MSET (Multivariate State Estimation Technique) since it represents the current state-of-the-art in prediction technologies and is used ubiquitously in nuclear applications, as well as aviation and space applications (Bickford 2000). Lastly, ACCEPT produces results in the form of missed detection (false negative), false alarm (false positive) rate and detection time (the number of time steps in advance the system can predict an anomalous event), all of which is essential in

[2] http://ti.arc.nasa.gov/opensource/projects/accept/

validating our approach to identifying candidate causes (Osborne et al. 2012).

The need to identify potential causes is important to ACCEPT since it will enhance the classification potential as it will remove misleading data and thus improving accuracy. Another reason is that it improves the computational efficiency of ACCEPT as there is less data for each algorithm to work with and thus drastically improving the time to produce results. Furthermore, these candidate causes gives us a better understanding of the domain.

The architecture of ACCEPT has the same basic structure as MSET but involves both a regression step and a detection step where the ground truth is used to aid the detection methods in determining the false alarm and missed detection rates. The regression step is implemented with the aid of machine learning techniques and the detection step tests a set of fixed hypotheses relating to the statistical properties of the resulting residual, using a variety of models.

Experimental Results

Data and Methods

Our data set consists 26,493 samples (Nov 2014 to Feb 2015) from 2,636 sensors of the BAS of NASA SB building. These sensors measure various physical and logical quantities and record in 5 minutes interval. We used 60% of total samples for training models, 10% for validation and 30% for testing.

The target variable (x^*) in our analysis is a room temperature sensor associated with the *cold complaints* scenario in NASA SB building. The Wet Bulb Globe Temperature (WBGT) index is a proxy for four measureable environmental factors (air temperature, air relative humidity, air velocity and mean radiant temperature) associated with Fangers famed Predicted Mean Vote (PMV) index (Atthajariyakul and Leephakpreeda 2005). The target sensor, x^*, is designed to characterize thermal comfort based on the WBGT index and thus computed a weighted average of dry bulb, wet bulb and mean radiant (black bulb) temperature.

As a preprocessing step of our experiments, we centered and scaled every sensor data to make the mean 0 and variance 1. Moreover, we discretized all continuous sensor values before Bayesian network learning step. The discretization was performed with different number of levels for different sensors.

While training autoregressive model, we observed that strong correlations exist among room-temperature sensors measurements. Thus for proper causal discovery, we removed all temperature sensors, except the target, from the dataset.

Identifying Ground Truth

An empirical approach was taken to determine the ground truth for the *cold complaints* prediction scenario. We estimated the distribution of the target room temperature sensor (x^*) and found that it was a unimodal distribution with mean 71.7 and standard deviation 1.8. Hence, a 95% confidence interval around the mean corresponds to 68.1°F and 75.3°F. Considering this range as nominal room temperature values, we established 68.1°F as the upper threshold for cold regions. In our problem, we are only concerned with anomalous drops in temperature. Thus, we considered any temperature value below 68.1°F as an *adverse event* (cold). And there are multiple adverse events as shown in Figure 4.

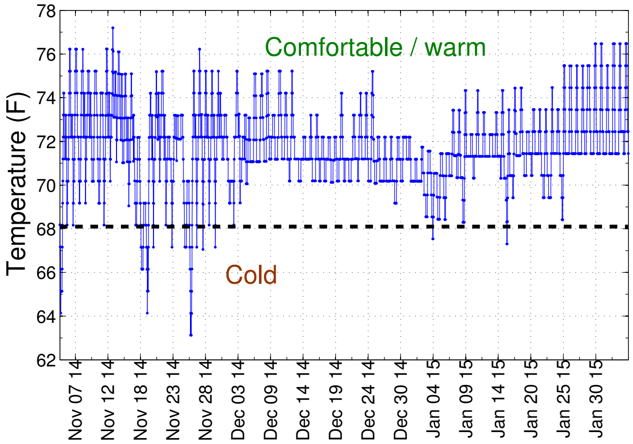


Figure 4: Cold and warm temperature regions according to empirical analysis.

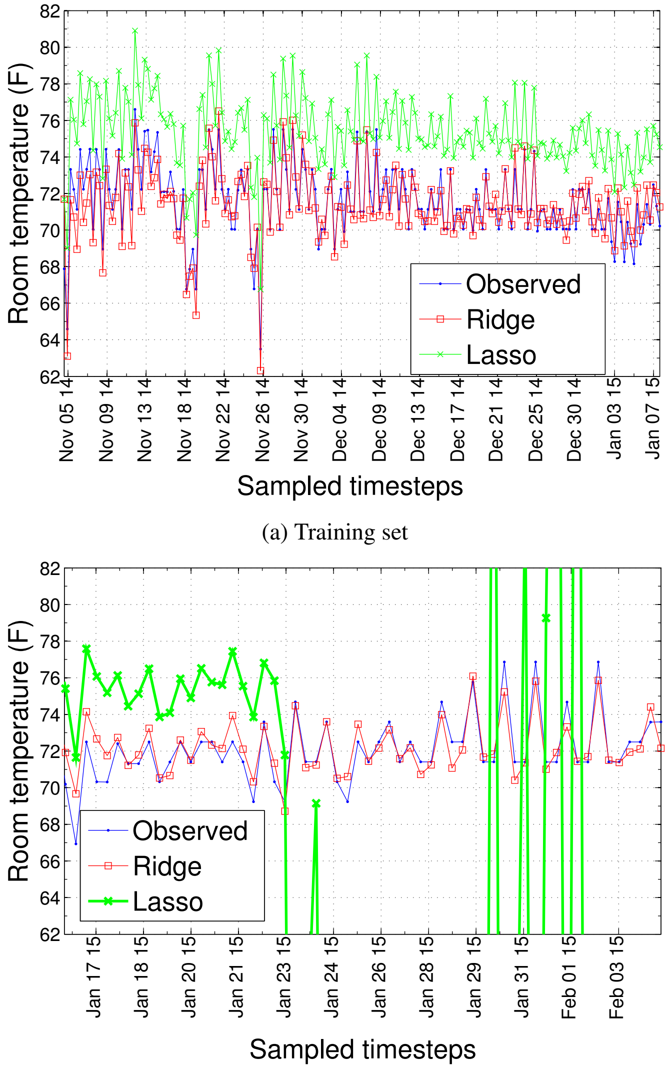
Training Autoregressive Model

The goal of this experiment was to find the most appropriate autoregressive model for dimension reduction. We started with an $AR(1)$ to predict room temperature (as in equation 1). We compared L_1 (lasso) and L_2 (ridge) penalties in terms of prediction error on test set, to find which one is more suitable to the training data. Figure 5 shows the prediction of trained models on both training and test data sets. Clearly the model trained using ridge penalty has more accurate predictions than its lasso counterpart. Therefore we used ridge regression for the dimension reduction step in the remaining experiments.

Next we compared autoregressive models of orders $\tau = 1, 2, \cdots, 5$ based on prediction error on the test data. Surprisingly we found, as shown in Figure 6a, the first order model AR(1) achieves minimum prediction error. The limited training data was possibly insufficient to train the more complicated models. It is worth mentioning here that with increasing AR order, the training time increased superlinearly. Hence for this data set, AR(1) model is superior in both statistical and computational measures.

Moreover we inspected the capability of the AR model to predict a few steps ahead in time. We trained the following models (equation 4) compared their prediction errors.

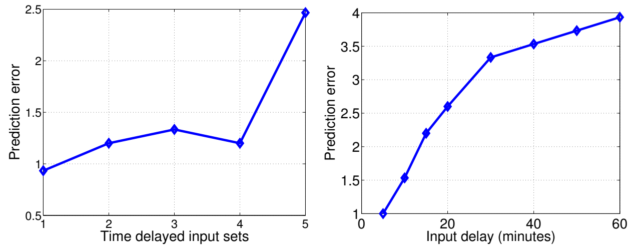
$$\begin{aligned} y_t^{(j)} &= \mathbf{a}_1^T \mathbf{y}_{t-1} + e_t^{(j)} \\ y_t^{(j)} &= \mathbf{a}_2^T \mathbf{y}_{t-2} + e_t^{(j)} \\ \cdots & \\ y_t^{(j)} &= \mathbf{a}_\tau^T \mathbf{y}_{t-\tau} + e_t^{(j)} \end{aligned} \qquad (4)$$



(a) Training set

(b) Test set

Figure 5: Prediction of linear model trained with ridge and lasso regularization on DCTN240T room temperature sensor.



(a) Prediction errors (on test data) of AR models with orders $\tau = 1, 2, \cdots, 5$.

(b) Prediction errors (on test data) of AR models with increasing τ.

Figure 6: Tuning parameters of autoregressive model.

From Figure 6b we observe that the prediction error increases at a higher rate for smaller input delays compared to larger delays. In Figure 7 we present the prediction of the trained model (ridge penalty) with 1 hour lagged input signals. We see that the predicted values follow the observed values closely until Jan 22. Thereafter we see high prediction error. This indicates the need to retrain the models after every few days (4-5) while predicting with lagged input signals.

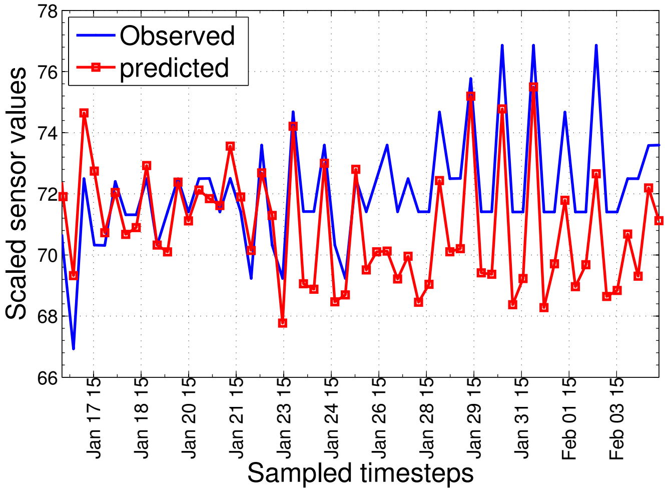


Figure 7: Prediction of trained (order 1) auto regressive model with 1 hour lagged input signals on test set.

Selecting Informative Variables

As the data is scaled and centered before training, all weights in the AR model are in same scale. Thus the predictive variables should have higher weights than the rest. We sorted the variables according to decreasing absolute weights of the trained AR(1) model and picked the top k variables as informative variables. Only this set is used for causal graph learning. The parameter k is a design choice. We demonstrate our results using $k = 10$.

Learning Bayesian Network and Isolating Causes

A Bayesian network structure is learned over the informative variables according to our proposed method. Although we found AR(1) model to perform well, the causes need not be limited to 1 time-step in past. We demonstrate our results for variables with two time-steps delay. Thus for k informative variables, we learn a Bayesian network with $2k + 1$ nodes where the last variable is the target without any time delay. As we are using $k = 10$, we learn the network structure with 21 nodes. However this step can be extended to informative variables with multiple delays.

For structure learning we used the Hill Climbing algorithm as implemented in *bnlearn* package of R (Scutari 2010). The algorithm performs a greedy search over the space of directed graphs guided by a predefined score function. We used Bayesian Information Criterion (BIC) to score each directed acyclic graph. The constraints in equation 3 are provided to the hill climbing search as blacklisted edges.

The learned Bayesian network structure for the target room temperature sensor is presented in Figure 8a. The numbers after the node labels indicate the delay of that variable. The candidate causes of the target variables are:

$$C_{SCL} = \{\text{room's dew point, heat pump 2 current (HPP2 current), CRCP (ceiling radiant cooling panel) valve, supply air temperature, heat pump flow switch (HP4C)}\}. \quad (5)$$

The elements of C_{SCL} have direct functional relationship with x^*. The dew point is located at the same place as x^* measures humidity which is a part of the weighted sum computed by the target sensor. The heat pump 2 current drives the pump for heating and the HP4C sensor is an indicator of flow for heat pump 4. As both heat pump 2 and 4 are designated for heating in the building, they affect x^*. The CRCP valve and supply air temperature set point are also associated with the heating system of the building.

Moreover, we observed that learning a BN structure with 1.5% fewer training samples leads to fewer edges in the learned graph. Hence, an attempt to reduce training time by decreasing the training set size results in incorrect identification.

Comparison of Hill climbing and the PC Algorithm

A causal graph, as shown in Figure 8b, was also constructed using the R implementation of the PC algorithm (Kalisch et al. 2012) over the informative variables for x^*. The identified causes are:

$$C_{PC} = \{\text{DX1 start/stop, heat pump current, N2 average temperature, heat pump flow switch}\}. \quad (6)$$

Similar to the graph learned by Hill Climbing algorithm, heat pump current and flow switch were also detected as direct caused of the target. Two distinct causal variables, N2 average temperature and DX1 start/stop, were also detected by PC algorithm and both are closely related to room temperature. A comparison of these two causal sets is performed in the next section.

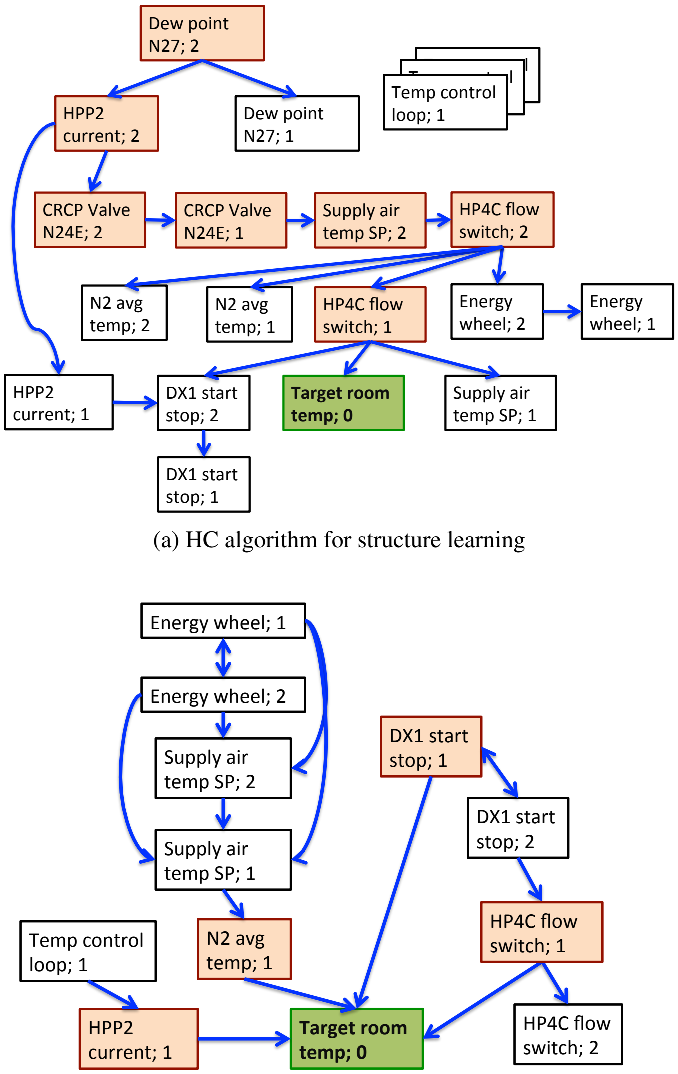
Adverse Event Prediction

The goal of this experiment is to compare different causal learning techniques where the true set of causes is unknown. We use the causal features C_{CD} to predict the adverse events of x^*. As ACCEPT works with only continuous features, we discard the discrete features from C_{CD} for predictions by ACCEPT. The feature set, with only continuous features, is denoted by $\bar{C}_{CD}$.

The first two set of sensors $\bar{C}_{SCL}$ and $\bar{C}_{PC}$ are defined by Equations 7 and 8 respectively.

$$\bar{C}_{SCL} = \{\text{room's dew point, heat pump 2 current (HPP2 current), supply air temperature}\} \quad (7)$$

$$\bar{C}_{PC} = \{\text{heat pump current, N2 average temperature}\}. \quad (8)$$



(a) HC algorithm for structure learning

(b) PC algorithm for structure learning

Figure 8: Causal graphs learned over the informative variables of target room temperature using two structure learning methods. The target variable (x^*) is highlighted in green and the causal variables (C_{CD}) are highlighted in orange.

To test the necessity of all causes in $\bar{C}_{SCL}$, we created a third set ($\bar{C}_{SCL-rand}$) by randomly replacing three causal sensors by three other random sensors. Additionally, to test the effect of non-causal sensors, we created a fourth set ($\bar{C}_{SCL+rand}$) by randomly adding three causal sensors to those in $\bar{C}_{SCL}$.

To compare these four feature sets we used false alarm rate, missed detection rate and detection time. The detection time is defined as the number of timesteps in advance, a warning in generated. For ACCEPT we used linear and extreme learning machine (Huang, Zhu, and Siew 2006) as regression methods. Moreover, we used three detection methods: redline, predictive and optimal (Martin 2010).

Table 1 shows the performance of the models trained with

Feature Set	False Alarm Rate	Missed Detection Rate	Detection Time
$\bar{C}_{PC}$	18%	20%	0
$\bar{C}_{SCL}$	**8%**	**0%**	**50 minutes**
$\bar{C}_{SCL-rand}$	36%	26%	0
$\bar{C}_{SCL+rand}$	16%	15%	0

Table 1: Results for each feature set

the above-mentioned four sets of sensors. For the sets with random features, 10 independent sampling runs were performed and the mean statistics are reported. The feature set $\bar{C}_{SCL-rand}$ produced worst results. There was a very high false alarm rate of 36% and missed detection rate of 26%, while the detection time is 0 indicating that this set was insufficient to give any advanced warning. The performance of $\bar{C}_{PC}$ was much better than $\bar{C}_{SCL-rand}$. Its false alarm rate is much lower (18%), the missed detection rate is also lower (20%) but the detection time is again 0. While the PC algorithm gave an improvement, it is not sufficient to implement into our system as both the false alarm and missed detection rates are too high and there is no prediction capabilities. We also observe that $\bar{C}_{SCL+rand}$ performed better than both $\bar{C}_{PC}$ and $\bar{C}_{SCL-rand}$ with a lower false alarm rate of 16% and a missed detection rate of 15%.

In contrast, $\bar{C}_{SCL}$ produced very low false alarm rate of 8% and kept the missed detection rate to 0%. This is significantly better than the previous feature sets. Also, the detection time is 10 timesteps or 50 minutes (one timestep is 5 minutes in our data set). We can implement this feature set into our system for adverse event prediction due to the extremely low false alarm rate, no missed detections and a decent detection time. These metrics show that our causal sensors are necessary for good modeling and prediction of the target temperature sensor.

This comparison indicates that even in presence of all sensors in $\bar{C}_{SCL}$, the additional random sensors can impede the predictive capabilities. Moreover, the inferior performance of $\bar{C}_{PC}$ compared to $\bar{C}_{SCL}$ and $\bar{C}_{SCL+rand}$ implies that SCL is more effective in learning causes of a target sensor compared to the PC algorithm.

Conclusion and Future Work

In this work, we present a novel integration of Bayesian network structure learning algorithm and autoregressive model to to isolate candidate causes from observational time-series data. As structure learning algorithms, typically, do not scale towards networks with large number of nodes, we perform dimension reduction on the original data set to filter out non-informative variables. We train an autoregressive model with sparsity constraint on the parameters and select the variables with high weights as informative variables. Thus, a Bayesian network is learned only with the informative variables instead of all variables. This step makes our method, Scalable Causal Learning (SCL), applicable to large-scale data sets.

We test SCL on a time series data set from a building automation system. We find that a first order autoregressive model trained with ridge penalty performed better on the data set compared to its lasso counterpart. We present the causal networks and the sets of candidate causes learned the HC and PC algorithm.

To compare these sets of causes, we feed the variables to an adverse event prediction system, ACCEPT. We find that the causes isolated by SCL produced better results in prediction (lower false alarm and missed detection rates; higher detection time), compared to the causes identified by the PC algorithms.

Future work will be directed towards developing theoretical guarantees of our method. We also intend to add alternate validation of our method using root cause analysis. Moreover, we plan to extend our method towards data sets consisting of both discrete and continuous variables.

References

Atthajariyakul, S., and Leephakpreeda, T. 2005. Neural computing thermal comfort index for hvac systems. *Energy Conversion and Management* 46(15):2553–2565.

Bickford, R. 2000. MSET Signal Validation System Final Report. Technical report, NASA Contract NAS8-98027.

Chakraborty, P.; Marwah, M.; Arlitt, M. F.; and Ramakrishnan, N. 2012. Fine-grained photovoltaic output prediction using a bayesian ensemble. In *AAAI Conference on Artificial Intelligence*.

Chickering, D. M. 2003. Optimal structure identification with greedy search. *The Journal of Machine Learning Research* 3:507–554.

Colombo, D.; Maathuis, M. H.; Kalisch, M.; Richardson, T. S.; et al. 2012. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics* 40(1):294–321.

Daly, R.; Shen, Q.; and Aitken, S. 2011. Learning bayesian networks: approaches and issues. *The Knowledge Engineering Review* 26(02):99–157.

Ellis, B., and Wong, W. H. 2008. Learning causal bayesian network structures from experimental data. *Journal of the American Statistical Association* 103(482):778–789.

Guo, Y.; Wall, J.; Li, J.; and West, S. 2011. A machine learning approach for fault detection in multi-variable systems. In *ATES in conjunction with Tenth Conference on Autonomous Agents and Multi-Agent Systems (AAMAS) AAMAS*.

Heckerman, D.; Geiger, D.; and Chickering, D. M. 1995. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning* 20(3):197–243.

Huang, G.-B.; Zhu, Q.-Y.; and Siew, C.-K. 2006. Extreme learning machine: Theory and applications. *Neurocomputing* 70(13):489 – 501.

Huida Qiu, Y. L.; Subrahmanya, N. A.; and Li, W. 2012. Granger causality for time-series anomaly detection.

Jiang, L., and Su, Z. 2005. Automatic isolation of cause-effect chains with machine learning. Technical report, Tech. rep., Technical Report CSE-2005-32, University of California, Davis.

Kalisch, M.; Mächler, M.; Colombo, D.; Maathuis, M. H.; and Bühlmann, P. 2012. Causal inference using graphical models with the r package pcalg. *Journal of Statistical Software* 47:1–26.

Kelejian, H. H., and Prucha, I. R. 2010. Specification and estimation of spatial autoregressive models with autoregressive and heteroskedastic disturbances. *Journal of Econometrics* 157(1):53–67.

Margaritis, D., and Thrun, S. 1999. Bayesian network induction via local neighborhoods.

Martin, R.; Das, S.; Janakiraman, V.; and Hosein, S. 2015. ACCEPT: Introduction of the adverse condition and critical event prediction toolbox. Technical Report NASA/TM-2015-218927, National Aeronautics and Space Administration (NASA).

Martin, R. 2010. A state-space approach to optimal level-crossing prediction for linear gaussian processes. *Information Theory, IEEE Transactions on* 56(10):5083–5096.

Matsuura, J. P., and Yoneyama, T. 2004. Learning bayesian networks for fault detection. In *Machine Learning for Signal Processing. Proceedings of the 2004 14th IEEE Signal Processing Society Workshop*, 133–142. IEEE.

Osborne, M. A.; Garnett, R.; Swersky, K.; and Freitas, N. D. 2012. Prediction and fault detection of environmental signals with uncharacterised faults. In *Proceedings of Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12)*.

Pearl, J.; Verma, T.; et al. 1991. *A theory of inferred causation*. Morgan Kaufmann San Mateo, CA.

Pearl, J. 2000. *Causality: models, reasoning and inference*, volume 29. Cambridge Univ Press.

Rattigan, M. J.; Maier, M. E.; and Jensen, D. 2011. Relational blocking for causal discovery. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 145–151.

Scutari, M. 2010. Learning bayesian networks with the bnlearn r package. *Journal of Statistical Software;35:122.*

Spirtes, P.; Glymour, C. N.; and Scheines, R. 2000. *Causation, prediction, and search*, volume 81. MIT press.

Tsamardinos, I.; Brown, L. E.; and Aliferis, C. F. 2006. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning* 65(1):31–78.

Wang, Z., and Chan, L. 2011. Using bayesian network learning algorithm to discover causal relations in multivariate time series. In *Data Mining (ICDM), IEEE 11th International Conference on*, 814–823.