2016 IEEE Computer Society Annual Symposium on VLSI

Online Unusual Behavior Detection for Temperature Sensor Networks

Hengyang Zhao*, Sheldon X.-D. Tan*, Hai Wang†, and Hai-Bao Chen‡
* Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521
† School of Microelectronic and Solid-State Electronics
University of Electronic Science and Technology of China, Chengdu, China 610054
‡ Department of Micro/Nano-electronics, Shanghai Jiao Tong University, Shanghai, China 200240

***Abstract*—In modern smart building climate control systems, accurate detection of unusual behavior in temperature sensors (outliers) can help reduce or prevent waste of energy consumption in a Heating, Ventilation and Air Conditioning (HVAC) system. In this work, we propose online learning-distance based outlier detection method. In the new method, we train and tune a multi-layer neural network to learn a nonlinear distance function from historical building operation data and detect outliers according to the calculated distance. The online detection method is less computational expensive than the offline version. By gradually including new and drop old building operation record, the new method is capable to adjust the underlying distance function on-the-fly. The converging speed of the learned distance function and tuning difficulty of network training are also discussed. The proposed online outlier detection method can work in an unsupervised manner except requiring only one data-specific parameter. In the experiments of two simulated buildings, the data-specific parameter can be chosen from a relatively wide range, which allows less tuning effort, to achieve good online detection precision and recall.**

I. Introduction

In the United States, it is estimated that commercial and residential sectors accounted for about 37% of energy consumption in 2012 and 39% in 2013. The building section is also responsible for 70% of electricity use. About 50% of the energy consumed in buildings are directly related to space heating, cooling and ventilation (HVAC) [1]. As a result, energy efficient management of the HVAC system is vital and also contingent on having highly reliable and trustworthy thermal sensors.

To improve credibility of sensor read values, reliable detection methods of unusual behavior or outlier/offset is required. Reliable detection here means the chance of reporting correct outliers is maximized and the chance for reporting false alarms is minimized (more precise definitions and measurements are given later). However, an existing fault detectors provided by real building management system can report more than 10000 alarms per day asserting that the system is not operating under correct status, which have to be mostly ignored by building managers because of poor correctness [2]. Being aware that these faults could be caused by temperature sensor outliers, it is important to improve the precision of automatic outlier detection.

Many existing works have been proposed to detect outliers from observed data. Classification and clustering techniques have been used in [3] to identify recurring patterns. The method proposed in [4] uses a partition algorithm and distance evaluation method within a partition, with which outliers are recognized by having an outstanding distance. Outlier detection methods can be divided into three categories [5]: Type 1 detectors determine outliers without prior knowledge of data; type 2 detectors work with knowledge of both normality and abnormality; type 3 detectors work with knowledge of normality and very little knowledge of abnormality. Type 2 methods are limited in real applications because the requirement of pre-labeled outliers are sometimes impractical. On the other hand, a type 2 detector can detect outliers more accurately because of the richer knowledge. With practical considerations, a type 3, or better, a type 1 detector is preferred if good detection performance can be achieved.

In this work, we target on developing type 1 outlier detection technique. We first study the nature of temperature outlier behaviors in smart building operations. We propose a learning-distance based outlier detection method. We train and tune a multi-layer perceptron neural network to learn a nonlinear distance function from historical building operation data and detect outliers according to the calculated distance. The historical operation data used as training set can contain small amount of unlabeled, or even no operation outliers, allowing the proposed method to have less practical restrictions. The resulting learning-distance detection method can work in an unsupervised manner except only one data-specific parameter being required. The proposed online outlier detection method can work in an unsupervised manner except requiring only one data-specific parameter. In the experiments of two simulated buildings, the data-specific parameter can be chosen from a relatively wide range, which allows less tuning effort, to achieve good online detection precision and recall.

II. Review of EnergyPlus Software

In this section, we review the EnergyPlus software program, which provides accurate input and output traces from buildings in our experiments on the outlier detection methods. The EnergyPlus software package [6] is a suite of algorithms that calculate the energy required to operate a building and its resulting thermal behavior based on numerous considerations ranging from the specifics of the structure, to heat sources and sinks within the building, in addition to weather effects. EnergyPlus consists of an integrated solution manager which manages the calculation of the heat balance of various surfaces in the building, the heat balance of the air, and the heat balance on the mechanical systems. The solution to each of these three elements are calculated separately and communicated to each other using the manager at each time-step.

An input data file (IDF) and a weather file are needed for the EnergyPlus simulation. An IDF includes all the information of a simulation such as building size, structure, position and configuration of HVAC systems, etc. We reconfigure two example building simulations by respectively altering the settings in the IDFs. The two buildings are respectively a 4-zone general purpose building and a 5-zone small office, as shown in Fig. 1 and Fig. 2.

III. The study of basic outlier detection methods and their performance analysis

In real world smart building applications, deployed sensors may work abnormally. In a running close-loop climate control system, sensors such as thermostat or humidistat may read values deviated from true room temperature or humidity. These

This work is supported in part by NSF grants under No. CCF-1255899 and No. CCF-1527324.

978-1-4673-9039-2/16 $31.00 © 2016 IEEE
DOI 10.1109/ISVLSI.2016.120

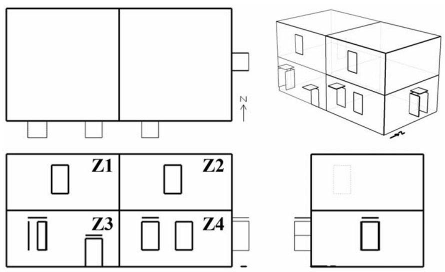


Fig. 1: 4-Zone building

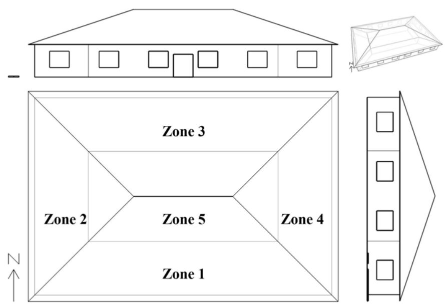


Fig. 2: 5-Zone small office building

faults can be caused by hardware aging, hardware flaw, or interference by other factors.

Smart building applications make the outlier detection different from other open-loop systems: if a certain sensor offset occurs in relatively less severity, the offset will be complimented by the close-loop control. To better demonstrate this situation, we randomly put temperature sensor outliers in a 5-zone office building operation. The simulated sensor read value and power consumption curves are shown in Fig. 3. The zone temperature is set to be no higher than 22°C. Because of the close-loop temperature control, read value of a certain room is expected to follow the temperature setpoint. The artificially assigned offsets can be almost fully compensated and therefore indistinguishable from other read values under normal operation. We should also notice that in many applications the setpoint of room temperature varies over time, so when some relatively low offset comes into the sensor, it is difficult to be detected by outlier analysis only considering the read value series. In fact, if the above problem occurs, the system should have operated differently in terms of power usage, etc. Although the faulty sensor may stay in normal read value because of close-loop control, the underlying energy cost will be different from normal cases. That is, with faulty sensor read values, the sensor read values may remain in similar statistical status, but the control system state may shift from normal operations.

Based on the above reasoning, to detect such outliers in smart building sensors, a viable way is to build a model between *system states* (power consumptions, outdoor conditions, etc.) and the true values read from fault-free sensors. A suspicious read value will be reported as faulty if it cannot fit in the pre-built model along with the underlying system states. The sensors, as well as the system, are considered as outlier-free as long as the system states and sensor read values can fit the pre-built model with certain tolerance.

Let $\boldsymbol{u}^{(m)} = [u_1^{(m)}, u_2^{(m)}, \cdots, u_n^{(m)}]^T$ denote the n-dimensional vector of observed system states and sensor read values sampled at m-th time step, our goal is to classify whether a given $\boldsymbol{u}^{(k)}$ a normal value or outlier, based on the knowledge of existing record $\boldsymbol{u}^{(1)}$, $\boldsymbol{u}^{(2)}$, $\cdots$, $\boldsymbol{u}^{(k-1)}$.

To describe outlier detection accuracy, we borrow terminologies *recall* and *precision* from machine learning area. These two metrics are used to describe the performance of a given outlier detector. Recall and precision are two ratios range from 0 to 1, indicating a good detector if both are close to 1. Recall and precision are respectively calculated by

$$\text{Recall} = \frac{D}{O}, \ \text{Precision} = \frac{T}{D} \tag{1}$$

where D denotes the number of data points detected as outliers; O denotes the total number of outliers; T denotes the number of outliers which are correctly detected. Intuitively, higher recall implies the outliers have higher chance to be reported and higher precision implies less false alarm. We use these two terms to quantify how good a detector is in the experiments.

A. Feed-forward neural network

Multilayer perceptron network can be applied in modeling non-linear functions with theoretically arbitrarily high accuracy. Meanwhile it needs some tuning-effort and carefully selected training parameters to achieve good modeling results. Given enough input vectors correctly labeled as normal or faulty, the detection problem can be reduced to a classification problem.

We build the two networks using TensorFlow [7] and train them using RMSProp [8] algorithm stopping at 10000th epoch. The feed-forward networks are constructed with sigmoid hidden layers. Results show that the two networks described above both have 100% detection precision and recall. In other words, for each given vector $\boldsymbol{u}^{(k)}$, the networks can correctly detect the existence of outlier as labeled. The above observation shows that a feed forward neural network is a good type 2 detector.

B. Mahalanobis distance based detection method

Experiments on feed-forward neural network reveal the fact that the current system status has major contribution in classifying abnormal sensor read values. Inspired by this fact, we mainly focus on detecting abnormal sensor values by extracting the characteristics of current system status, without considering the operation history.

In real world applications, it is difficult or even impractical for a supervised learning algorithm to be trained based on rich knowledge of faulty situations. Therefore it is necessary to seek type 1 or type 3 techniques that are built on only normal data (or including very small amount of outliers information, if any). In the area of anomaly detection, distance based detection is a widely used technique, which calculates center and covariance matrix based on normal data (small amount of outliers is allowed) and classifies a data point as outlier if it's *distance* is greater than a preset threshold. The distance can be calculated in sense of Euclidean, Mahalanobis, etc. We evaluate the performance of Mahalanobis distance based method in the following.

Given vectors $\boldsymbol{u}^{(1)}$, $\boldsymbol{u}^{(2)}$, $\cdots$, $\boldsymbol{u}^{(k)}$ recorded in daily building operation data, denoted by matrix $\boldsymbol{U}_{n\times m}$, the Mahalanobis distance is calculated by

$$D_{\mathrm{M}}(\boldsymbol{u}) = (\boldsymbol{u}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{u}-\boldsymbol{\mu}) \tag{2}$$

where $\boldsymbol{\mu} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{u}^{(i)}$, $\boldsymbol{\Sigma} = \frac{1}{m}\boldsymbol{U}\boldsymbol{U}^T$.

Based on the observation of the experiment results, plain Mahalanobis distance based method (as described in equation 2) cannot provide a clear distance threshold to separate outliers from normal operation, which implies that detection precision

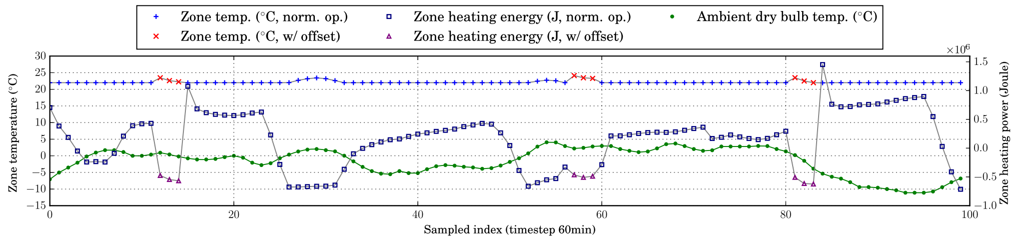


Fig. 3: Sampled temperature sensor read values (with offset randomly assigned) and power consumptions

and recall cannot be both good using one threshold. The reason is that system states may be clustered or correlated in non-linear manner. Therefore it is desirable to generalize the distance based method to adapt irregular distributions.

IV. Learning-distance based method and offline detection performance

Inspired by the distance based technique, we can enhance the distance function f_{D} to calculate distance from center of irregular distribution. For a given vector $\boldsymbol{u}$, greater distance $f_{\mathrm{D}}(\boldsymbol{u})$ should imply higher possibility to be an outlier, and $|f_{\mathrm{D}}(\boldsymbol{u})| = 0$ should imply the smallest possibility. To construct such a distance function, we use learning based method to extract the distribution information. It would be problematic if we setup a learning process using $f_{\mathrm{D}}(\boldsymbol{u})$ and 0 respectively as training input and output, because a trivial constant function $f_{\mathrm{D}}(\boldsymbol{u}) = 0$, will be learned, detecting anything as normal.

With those factors in mind, we need to carefully construct the training dataset to make the training process to learn a non-trivial distance function. We introduce an extra stage, parameter selection, to help setting up better training dataset. Parameter selector takes vector $\boldsymbol{u}$ as inputs and divides it's elements into two parts: $\boldsymbol{v}_{\text{NN\_IN}}$ and $\boldsymbol{v}_{\text{NN\_OUT}}$, which will later serve as training input and output respectively. Let function φ denote the learned function from training set $\{\boldsymbol{v}_{\text{NN\_IN}}, \boldsymbol{v}_{\text{NN\_OUT}}\}$, a non-trivial distance function can be expressed as

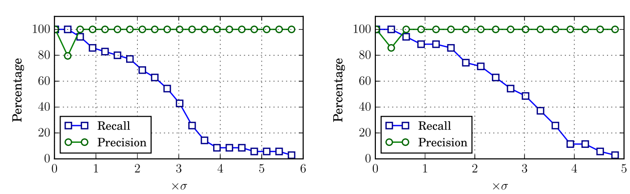
$$f_{\text{NN\_DIST}}(\boldsymbol{u}) = \|\varphi(\textsc{Param}_{\text{IN}}(\boldsymbol{u})) - \textsc{Param}_{\text{OUT}}(\boldsymbol{u})\| \quad (3)$$

The resulting new learning-distance detector architecture is shown in Fig. 4. In this method, we use a σ-limited selector to find any $\boldsymbol{u}$ having distance $f_{\text{NN\_DIST}}(\boldsymbol{u})$ deviating more than $\lambda_C\sigma$ from center of distribution, where σ denotes the standard deviation of distances $\{f_{\text{NN\_DIST}}(\boldsymbol{u}^{(n)})\}$, and λ_C denotes the cut-off coefficient. In addition, we apply a saturating counter to reduce the σ-limited detector's sensitivity, and use a normalizer stage before parameter selection to accelerate training and mitigate numerical stability issues. The normalizer shifts and scales each input respectively using the mean and standard deviation. The saturating counter yields a boolean value with reduced sensitivity compared to it's input. Output of the saturating counter output is regarded as the final decision whether to report the system to be operating with outliers. We set the saturating limit to 3 timesteps, correspondingly 30 minutes into the simulation. It should be noted that the saturating limit is not a data-specific parameter. Instead, it should be specified by a system administrator with consideration of the detection sensitivity and delay. Window size n_W is set to infinity to use all observed data to perform offline network training and detection.

We use TensorFlow [7] to construct the feed forward neural network using a linear input layer, three sigmoid hidden layers and one linear output layer. For the 4-zone building, the layer sizes are respectively 10 (input), 20 (hidden), 20 (hidden), 20 (hidden), and 1 (output). We select ambient dry/wet bulb temperatures, wind speed/direction, solar factors, server power, HVAC system cooling power, and east zone temperature sensor read value of last-time step as network input $\boldsymbol{v}_{\text{NN\_IN}}$, with living zone temperature sensor read value of current time step as network output $\boldsymbol{v}_{\text{NN\_OUT}}$. For the 4-zone building, layer sizes are respectively 12 (input), 25 (hidden), 25 (hidden), 25 (hidden), and 1 (output). To train the neural network, we select ambient dry/wet bulb temperatures, solar factors, zone #1 temperatures sensor read value and adjacent zone temperatures as network input $\boldsymbol{v}_{\text{NN\_IN}}$, with zone #1 temperature sensor read value and HVAC system cooling/heating power as network output $\boldsymbol{v}_{\text{NN\_OUT}}$.

Figure 5 demonstrates the detection result. Red and blue dots attached to vertical bars respectively denote operations with and without outlier. Sections reported as with outliers are marked with shadows with constant height. Saturating counter potentially introduces delayed reporting of outliers, while better identifying sections with outliers and reducing the number of false alarms.

To evaluate the effect of choosing different limiting coefficient λ_C, we sweep the value of λ_C from 0 to 6 (4-zone) and 0 to 5 (5-zone) respectively. We calculate recall and precision as defined in equation 1. Fig. 6 shows the sweeping results. As λ_C increases, recall curves decrease and precision curves increase as expected. Precision appears to be very high with small λ_C because the whole time period is detected as with outliers, which is the only one detected and correctly detected section. The curves imply that the selection of λ_C can be between 0.5 to 1, which is not too narrow, achieving 85% recall and 100% precision. Specifically, $\lambda_C = 0.5$ yields perfect (100%) recall and precision in both experiments of the two buildings.



(a) 4-Zone (b) 5-Zone

Fig. 6: Offline learning-distance method: precision and recall v.s. cut-off coefficient λ_C

V. Online learning-distance based detection

One of the critical requirements of outlier detection is the ability to adapt the underlying system as it slowly changes. Therefore it is important to allow the learned distance function to change in order to consistently yield small values for regular operations and large values for suspected outliers. The learning-based technique as proposed in section IV requires batch training using a fixed time span of operation data.

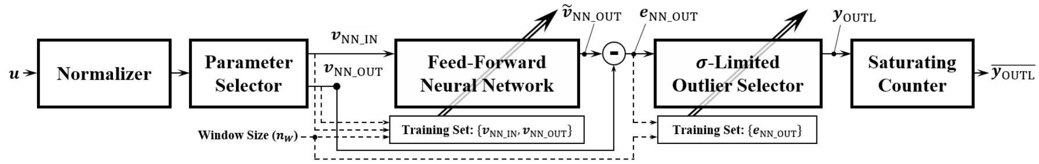


Fig. 4: Block diagram of the proposed learning-distance outlier detector

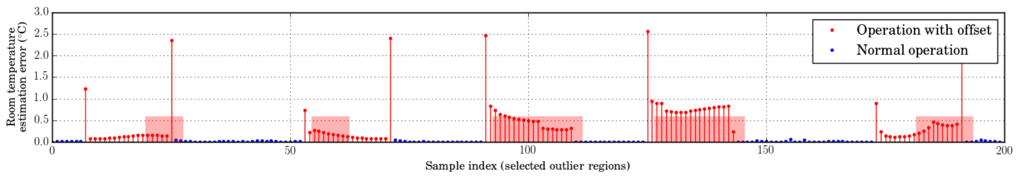
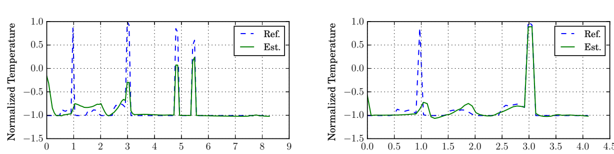


Fig. 5: Examples of detected outliers (Shaded sections are detected as operating with outliers)

Batch training method, although delivering good detection performance, is computational expensive and normally takes long time (from several hours to days) to converge with moderate computation devices.

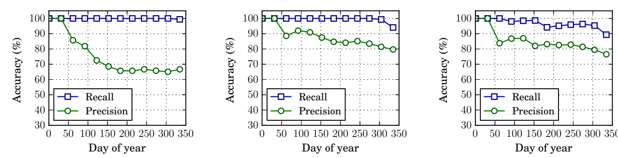
On the other hand, to help saving power consumption or identifying system faults, outlier must be reported in real-time or with acceptable short delay for system administrators to respond in time. To update the model as the weather changes over the year, or other reasons like demand changing, batch training method cannot meet the real-time requirement.

To make the learning-distance based method to be quickly adaptive and able to do real-time detection, we reduce the size of training set. Instead of considering operation data of a whole year, we use a fixed-length time window of recent operation record. The window shifts forward as time goes on, including new operation records and correspondingly dropping old records. In the following experiments, we set the time window size (in Fig. 4) to be 1 month. The window shifts on every hour. As shown in Fig. 4, during the online outlier detection, only the operation records within the shifting window are considered, which allows the learned distance function to be adaptive using relatively less training efforts. The training efforts are reduced to less than 100 epochs per hour. The online detection suffers accuracy problem at the beginning because of not enough training data. The underlying distance function can be learned in less than 5 days (as shown in Fig. 7), which is acceptable in practical outlier deployment. We evaluate the impact of cut-off coefficient λ_C on detection accuracy three different λ_C values. As shown in Fig. 8, we evaluate detection precision and accuracy of the 4-zone building over the simulated one-year period. As λ_C changes from 0.5 to 1, precision and recall both remain higher than 80%, which implies the relatively low tuning effort of the data-specific parameter.



(a) 10 epochs on shifting (b) 50 epochs on shifting

Fig. 7: Training converging speed using different number of epochs on window shifting



(a) $\lambda_C = 0.5$ (b) $\lambda_C = 0.8$ (c) $\lambda_C = 1.0$

Fig. 8: Detection precision and recall using different λ_C values

VI. Conclusion

In this paper, we have proposed a type 1 online outlier detection, which combines neural network and the distance based method, to achieve both high accuracy and unsupervised learning capability. The online outlier detector has theoretically same detection performance as the offline version, which has good detection performance (100% recall and 85%+ precision). Comparing to the offline version, the online detector has an adaptive distance function, lower computational cost, and low accuracy loss. Experimental results showed that the online detection method also can achieve good detection recall (95%+) and precision (80%+).

References

[1] Building Energy Data Book of DOE. [Online]. Available: http://buildingsdatabook.eren.doe.gov

[2] B. Narayanaswamy, B. Balaji, R. Gupta, and Y. Agarwal, "Data driven investigation of faults in hvac systems with model, cluster and compare (mcc)," in Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings. ACM, 2014, pp. 50–59.

[3] A. Capozzoli, F. Lauro, and I. Khan, "Fault detection analysis using data mining techniques for a cluster of smart office buildings," Expert Systems with Applications, vol. 42, no. 9, pp. 4324–4338, 2015.

[4] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in ACM SIGMOD Record, vol. 29, no. 2. ACM, 2000, pp. 427–438.

[5] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," Artificial Intelligence Review, vol. 22, no. 2, pp. 85–126, 2004.

[6] D. B. Crawley, L. K. Lawrie, F. C. Winkelmann, W. F. Buhl, Y. J. Huang, C. O. Pedersen, R. K. Strand, R. J. Liesen, D. E. Fisher, M. J. Witte, and J. Glazer, "EnergyPlus: creating a new-generation building energy simulation program," Energy and Buildings, vol. 33, no. 4, pp. 319–331, Apr. 2001.

[7] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin et al., "Tensorflow: Large-scale machine learning on heterogeneous systems, 2015," Software available from tensorflow. org.

[8] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural Networks for Machine Learning, vol. 4, p. 2, 2012.