Estimating Data Center Thermal Correlation Indices from Historical Data

Manish Marwah, Cullen Bash, Rongliang Zhou, Carlos Felix, Rocky Shih, Tom Christian
HP Labs
Palo Alto, CA 94304
Email: firstname.lastname@hp.com

ABSTRACT

In order to better manage the cooling infrastructure in a data center with multiple computer room air conditioning (CRAC) units, the relationship between CRAC settings and temperature at various locations in the data center needs to be accurately and reliably determined. Usually this is done via a commissioning process which is both time consuming and disruptive. In this paper, we describe a machine learning based technique to model rack inlet temperature sensors in a data center as a function of CRAC settings. These models can then be used to automatically estimate thermal correlation indices (TCI) at any particular CRAC settings. We have implemented a prototype of our methodology in a real data center with eight CRACs and several hundred sensors. The temperature sensor models developed have high accuracy (mean RMSE error is 0.2 ˚C). The results are validated using manual commissioning, demonstrating the effectiveness of our techniques in estimating TCI and in determining thermal zones or regions of influence of the CRACs.

KEY WORDS: data center, temperature sensors, CRAC, thermal zones, thermal correlation index, regression trees, random forest, machine learning

INTRODUCTION

Data centers consume a significant amount of energy, e.g., 61 billion kilowatt-hour (kWh) in the US in 2006. This has led to an increased focus on making data centers more energy efficient, particularly the cooling infrastructure, which can account for up to 50% of a data center's total energy consumption. State-of-the-art data centers deploy temperature sensors so that cooling resources such as computer room air conditioning (CRAC) units can be managed to provide cooling only to the extent needed and at the location and time needed. This requires knowledge of the relationship between a CRAC unit settings and its cooling impact at a sensor location in a data center. In the past, this relationship for a particular location in a data center and a CRAC unit has been quantified by a thermal correlation index (TCI) [1], which can be defined as the change in temperature at the location, given a unit change in the supply air temperature of the CRAC. The TCI values can also be used to determine the regions of influence or thermal zones associated with each CRAC unit. However, computing TCI values for all the CRAC units in a data center is usually a manual, time-consuming and disruptive process where each CRAC unit is independently perturbed and its response at each sensor location measured.

In this paper, we propose AutoTCI, a machine learning based technique to estimating TCI values using historical CRAC settings and temperature sensor data e.g. such data from the past few weeks. This non-intrusive mechanism simplifies computing TCI value for a particular sensor location with respect to a particular CRAC unit. Furthermore, unlike prior work, a linear relationship between a sensor temperature and CRAC settings is not assumed. TCI values can be computed for *any* CRAC reference settings (but within the range of CRAC setting values present in the historical data). Note that typically the TCI for a CRAC with respect to a sensor will depend on its VFD speed and settings of other CRAC units.

The technique involves two main steps. First, a machine learning model is developed between CRAC settings and sensor temperature. A separate model is developed for each sensor. In particular, an ensemble method, called random forests [3], is used to model the sensor temperatures. The models are quite flexible and the inputs can include CRAC settings such as supply air temperature, CRAC blower variable frequency drive (VFD) speed, and potentially other influencing variables such as vent tile openings, etc. Next, to estimate the TCI at a given CRAC state, the local slope of the model is computed. The thermal zone or region of influence of a CRAC is determined by clustering the TCI values.

While the accuracy of the estimate is dependent on the CRAC setting variations seen in the historical data, this method requires no user intervention. We have applied AutoTCI on data from a real data center and comparison of results with TCI values obtained through a manual commissioning process show a good match (correlation coefficient of 0.91).

The rest of the paper is organized as follows. We briefly discuss related work in the next section. Then we describe our modeling methodology, followed by experimental results from a real data center. Finally we conclude and discuss some future work.

RELATED WORK

Bash et al. [1] describe the concept of thermal zones, or regions of influence associated with each CRAC unit, for more effectively controlling the CRAC units in a data center. Typically, to compute TCI's a commissioning process is used where all CRACs are kept at a reference state and then each CRAC is individually perturbed and its response at each sensor is measured. Since commissioning is time consuming and disruptive, a less intrusive method for computing TCI's is proposed in this paper.

Li and Hamann [2] propose a statistical approach where CRAC and sensor data is used to determine thermal zones and TCIs without requiring explicit manipulation of the CRAC settings. They linearly model the correlations between temperatures observed at the discharge of a CRAC and sensors in the data center. The objective of this paper is similar, however, we do not assume that the relationship between the temperature at a CRAC discharge (CRAC supply air temperature (SAT)) and a sensor temperature is linear. Furthermore, our technique is generic enough to incorporate additional variables, e.g. rack outlet temperatures, vent tile settings, etc.

MODELING METHODOLOGY

We use a data-centric methodology to estimating TCIs by training machine learning models to predict a rack sensor temperature based on CRAC settings, and potentially other variables that could influence the temperature, such as, vent tile openings. Specifically, we use an ensemble learning technique called random forests [3], described below, that uses a large number of regression trees as predictors, which while individually not very accurate, can be combined into an accurate predictor.

Regression Trees

Tree based models for classification and regression are widely used. In our case, the task is to predict a continuous valued output, Y (rack sensor temperature), based on inputs, X_1, X_2, … X_n, such as CRAC settings, vent tile settings, and other actuator settings. Regression trees are non-parametric models consisting of a binary tree with a variable test at each node as shown in Figure 1. Given an input or feature vector, making a prediction involves starting at the root of the tree and applying a test, based on a particular input variable, at each node. The test determines whether to take the right or the left branch of the tree. Finally, the prediction is made at a leaf node, which contains a local model for all input vectors that reach that particular leaf. Thus, in essence, regression trees are a means of splitting the problem input space into multiple mutually exclusive regions, which are then locally modeled.

In order to train a regression tree, the input space is recursively partitioned based on a single feature variable until a small number of training data points remain in a partition. The local model at each leaf is typically a constant value that is the average of the training points at that leaf. A detailed description of regression trees can be found in Hastie et al. [4]. While regression trees are quite flexible, e.g. they can handle different kinds of inputs (continuous, discrete, categorical, etc) and perform well with missing data, their prediction accuracy is not always very high, and they can overfit the data. Therefore, instead of using a single regression tree in the model, we use a regression tree ensemble technique called random forests that is described next.

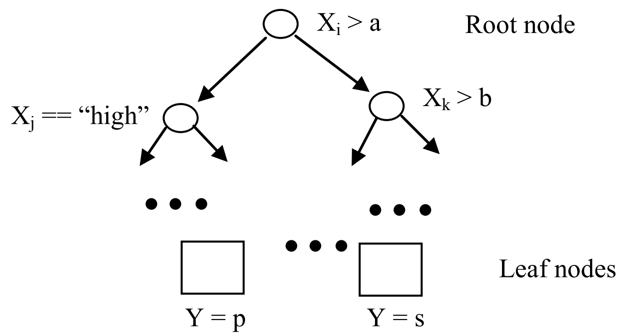

Figure 1: A regression tree model

Random Forests

Ensemble learning methods use many models for the same task and aggregate the results [4, 5]. There are many methods to combine the different predictors. Breiman [3] introduced the random forests ensemble learning technique where a large number of independent trees are created from bootstrap samples of the training data, and the final prediction is the average of all the models. Such ensemble methods perform better if the individual models are not correlated. In random forests, two different mechanisms are at work to ensure that different ensemble components are uncorrelated at least to some extent. First, the bootstrap samples results in a slightly different training set for each model. Second, another element of randomness is added – only a randomly chosen subset of variables is available at each split in the tree during training phase. The random forests training algorithm is described next.

Random Forest Algorithm

The random forests model training involves the following steps:

- Create a bootstrap sample from the original data set. The data that is not selected in the bootstrap sample is called out-of-bag data and is used for testing (it is usually about 1/3 of the total data).
- The bootstrap sample is used to fit a regression tree. A random subset of features is considered at each split.
- For each tree, error is computed on the out-of-bag data set.
- Once all the trees are trained, the final prediction, given an input feature vector, is given by the average of the prediction of all the trees.

As the number of features increases the chances of overfitting also increase and thus it is advisable to use one of the many feature selection algorithms [4, 5] to select only the most important features, reducing the complexity of the model, and hence the chances of overfitting the data. In this work since the test bed data center only has 8 CRAC units, the number of features is not large (16 in all – SAT and VFD values for each CRAC). However, if we were to also include vent tiles opening settings, rack outlet temperature, or other additional features, it would be important to include a simple feature selection method.

EXPERIMENTAL RESULTS AND DISCUSSION

Data Center Infrastructure

Most data centers are air-cooled with a raised floor plenum to distribute cool air from CRAC units, which extract heat from the hot air expelled from the computer racks. The blowers in the CRAC units pressurize the plenum with cool air which enters the data center through vent tiles located on the raised floor near the inlet of racks. Typically the racks are laid out in rows separated by hot and cold aisles as shown in Figure 2. This separation is done for thermal efficiency considerations. Air inlets for all racks face cold aisles while hot air is expelled to hot aisles.

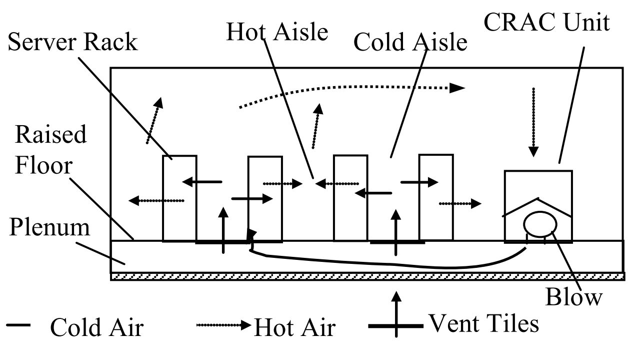

Figure 2: A typical raised-floor data center

Data Center Layout

Figure 3 shows the layout of the data center where we demonstrated our methods to estimate TCI. It is about 3000 sq. ft. facility with IT power of about 350 KW. There are ~80 racks of computing equipment, arranged in 10 rows named A, Aext, B, Bext, C, Cext, D, E, F, and G. Cooling is provided by eight CRAC units, marked CRAC1-8 in the figure. Curtains in the data center minimize hot air from recirculating to the cold aisles.

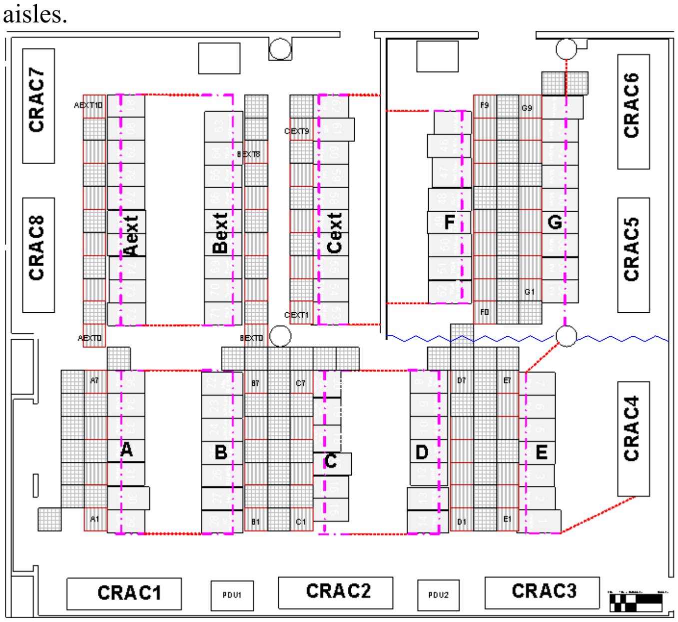

Figure 3: Layout of racks and CRACs in the data center

Sensor Network

Temperature data is collected from sensors, mounted at the inlet and outlet of racks as shown in Figure 4. They provide temperature data at both air inlet and outlet of the racks. The digital output from each sensor has an accuracy of ± 0.5 ˚C. Ten temperature sensors are attached to each rack, with five at the inlet and the other five at the outlet. The inlet sensors are labeled T1 through T5 – with T1 being closest to the floor and T5 the top most sensor in a rack. In this study only sensors at the rack inlet were considered. Each rack also contains a base station to which all sensors on a rack are connected. The base station has an Ethernet interface and multicasts the temperature data collected on the data center LAN. In addition to rack temperature sensors, other data collected includes CRAC unit supply air temperature and variable frequency drive (VFD) speeds.

Data set

The data is collected for seven days from 305 temperature sensors located at the inlet of 61 racks and the eight CRAC units. For each CRAC, both supply air temperature (SAT) and VFD speed data is collected. Although the data is collected every 25 seconds, we average over a window to compensate to some extent for the time lag between change in actuation (e.g. CRAC settings) and its impact at a sensor. (Note that the goal is to model the steady state behavior.) The averaging window used is 2.5 minutes. Figure 5 shows the CRAC SAT and VFD settings over the seven days. The variation in the 305 rack temperature values (in ˚C) over the same period is shown in Figure 6.

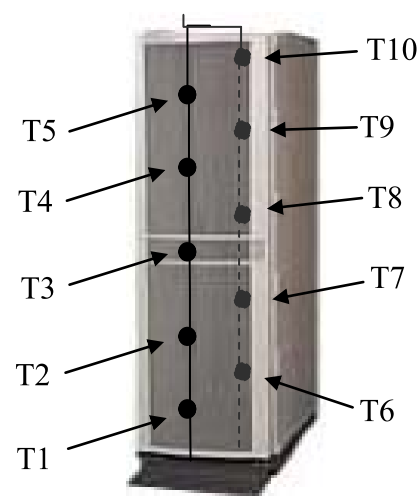

Figure 4: Ten temperature sensors (five at air inlet, T1 through T5, and five at air outlet, T6 through T10) are mounted on a data center rack.

Models

Random forests models for the 305 sensors were created using the algorithm described above. The input features consist of the eight CRAC SAT and the corresponding VFD values. The output of the model is a sensor temperature. The number of trees used per model was varied and the mean squared error (MSE) plotted to determine a suitable number. Finally, ninety trees were used in all models. Root mean square error for out-of-bag data (not used for training) for all the 305 models is shown in Figure 7. The average error across all models is around 0.2 ˚C. Note that these models are only good in the range of the training data, and perform poorly at extrapolating. Thus, they can be used to compute TCI only in the range of CRAC settings shown in Figure 5. In future, in addition to CRAC settings, we also plan to include vent tile openings, rack outlet temperatures, or any other measured attribute that might influence the temperature at a sensor.

AutoTCI: Estimating TCI

Once a model for a sensor is built, it can be used for estimating TCI, which is essentially the sensitivity of the model output (sensor temperature) with respect to its inputs (CRAC SAT and VFD speed). Since the models are not linear, the TCI with respect to an input will change with the input vector. In this paper, we only compute TCI with respect to CRAC SAT. Also, although in the past TCI is only defined between sensor temperature and CRAC SAT, our definition is broader and can include CRAC VFD, vent tile openings, or even a combination of variables (e.g. some change in both CRAC SAT and VFD). Note that our modeling technique is flexible enough to easily include additional features such as rack outlet temperature, vent tile openings, etc.

In order to compute TCI of a sensor with respect to a CRAC SAT at a particular point, we need to determine the local slope of the model at that point with respect to that SAT. We do this by varying the CRAC SAT around that point by small increments totaling ΔSAT, running those inputs through the model and fitting a linear function through the predicted outputs. A ΔSAT of 2˚C was used for each SAT. This procedure was used to estimate TCI for all the 305 sensors with respect to the SAT of the eight CRACs. The TCI values for each of the CRACs are plotted in Figure 8. These TCI values are for the median settings seen in the historical data. They are unlikely to match well with the TCI values obtained through the regular commissioning process described in [1] due to the different VFD settings and the interactions between CRACs.

Thermal Zones

In order to compute thermal zones or the region of influence of each CRAC we can set a threshold for the TCI values, and any sensor with a higher TCI value will then be part of that CRAC's thermal zone. However, this threshold may be different for different CRACs, and may even change with CRAC settings. To find natural gaps in the TCI values to separate sensors that belong to the thermal zone of a CRAC from those that do not, we use clustering. Figure 9 shows the results of partitioning the sensors into three regions: the green and blue sensors belong to a CRAC's thermal zone while the pink ones do not. For each CRAC, the sensors in its region of influence are divided into a low and high category, indicated by the green and blue color respectively.

Comparing the thermal zones in Figures 9 with the data center layout in Figure 3, it can be seen that the sensors included in a CRAC's thermal zone are typically in the proximity of that CRAC. Sensors from rows A and B are mainly part of CRAC 1's thermal zone; sensors from row B of CRAC 2's; sensors from rows D and E and some from G are part of the thermal zones of CRACs 3 and 4. Table 1 lists the sensors rows where most of the sensors in the thermal zone of a CRAC reside. Note that we did not have data for any sensors in row C in Figure 3. However, this does not impact the results for any of the other rows.

Table 1: CRACs and rows where most of the sensors in their region of influence reside

CRAC Number	Thermal zone sensor rows
1	A, B
2	B
3	D, E
4	D, E, some in G
5	F, G
6	F, G
7	Aext, Bext, some in Cext
8	Aext, Cext, some in B

AutoTCI Validation

In order to validate our solution, we compare it with a manual process of determining TCI's at a given setting. The manual process, similar to the commissioning process [1], consists of adjusting the CRAC settings to the settings at which the TCI's were computed using AutoTCI. Then each CRAC is selected in turn and "perturbed" manually. The perturbation consists of increasing a CRAC SAT by *delta*/2 and then decreasing it by *delta.* After each change in the SAT value, the data center is allowed to attain thermal equilibrium by waiting for about 1 hour. A delta of 2˚C was used, same as that used for AutoTCI. The change in all the sensor temperature values is recorded when thermal equilibrium is reached. The TCI of sensor *i* with respect to CRAC *j* is computed as, $TCI_{ij} = \Delta T_{i,\ sensor} / \Delta\ T_{j,\ CRAC}$. The TCI's computed are plotted in Figure 10, and look very similar to those computed using AutoTCI (shown in Figure 8). In order to quantify the difference between the two sets of TCI values, we computed their correlation coefficient which is 0.91, demonstrating that AutoTCI estimates the TCI values with good accuracy.

Comparison with Regular Commissioning

We also compare the TCI values computed through AutoTCI to those determined by the conventional commissioning process [1]. We did not expect these to match very well with those computed through AutoTCI, since the regular commissioning process assumes linear relationship between sensors are CRAC settings. The TCI's determined from the regular commissioning process for CRACs 4 through 8 are plotted in Figure 11. The correlation coefficient between the TCI computed using AutoTCI and regular commissioning is 0.81.

Summary & Conclusions

In this paper, we presented AutoTCI, a non-intrusive method for estimating TCI's and thermal zones for CRACs using machine learning techniques. In particular, we used random forests ensemble technique to accurately model rack inlet temperatures as a function of CRAC settings. These models were then used to estimate TCIs. Furthermore, TCI are not assumed to be constant throughout the range of the CRAC settings and can be computed for a particular CRAC setting. We used clustering to partition the TCI values to determine sensors that belong to a particular CRAC's thermal zone. The results of applying AutoTCI to a real data center are validated with a manual process and show a close match, demonstrating the effectiveness of our techniques. In the future, we plan to include additional features in our models such as rack outlet temperature and vent tile settings.

Acknowledgments

The authors would like to thank Zhikui Wang for providing the figure of the data center layout.

References

[1] C.E. Bash, C.D. Patel, R.K. Sharma, "Dynamic Thermal Management of Air Cooled Data Centers," Intersociety Conference on Thermal and Thermo-mechanical Phenomena in Electronic Systems (ITHERM 2006), San Diego, California, June 2006.

[2] H. Li, H.F. Hamann, "A Statistical Approach to Thermal Zone Mapping," Proceedings of IPACK 2011.

[3] L. Breiman. Random Forests. Machine Learning, 45(1): 5–32, 2001.

[4] Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Verlag, 2009.

[5] Christopher M. Bishop. Pattern Recognition and Machine Learning, Springer, 1st ed. 2006.

.

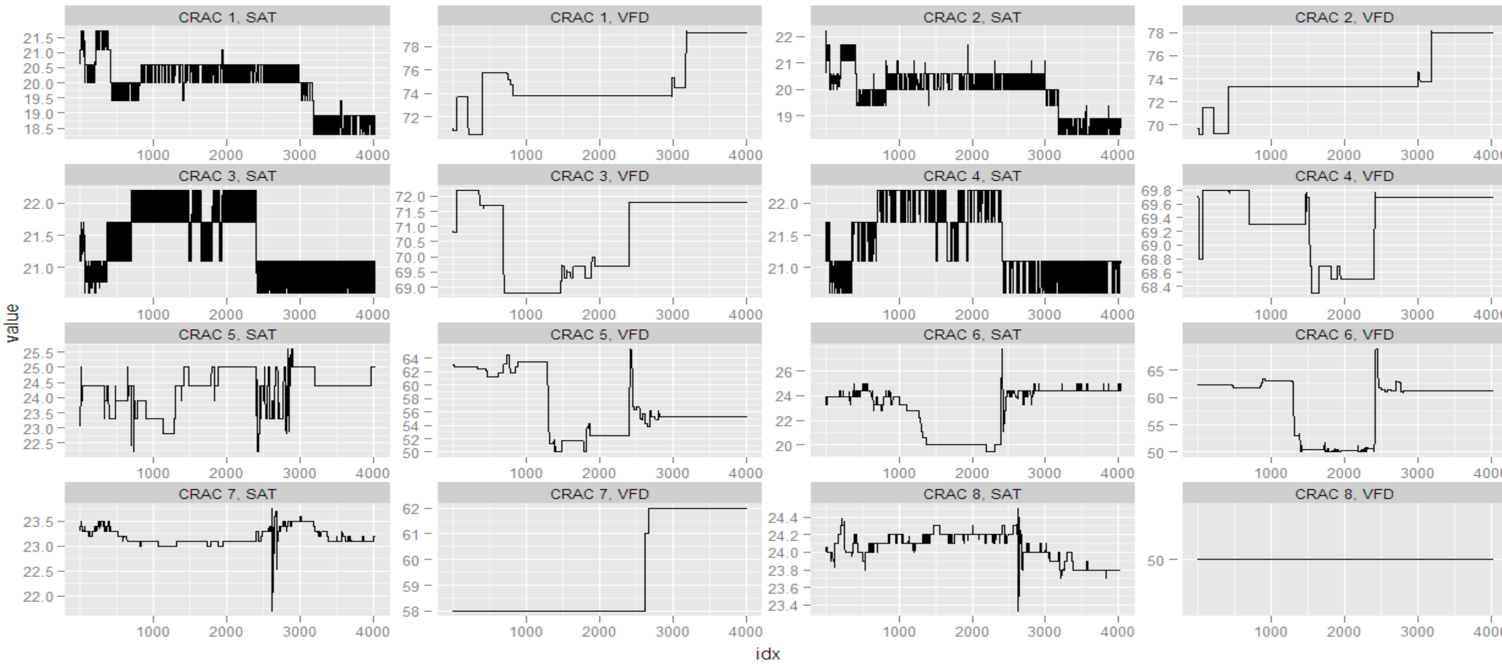

Figure 5: SAT and VFD data for eight CRACS over a period of seven days. This data was used to train all the sensor models.

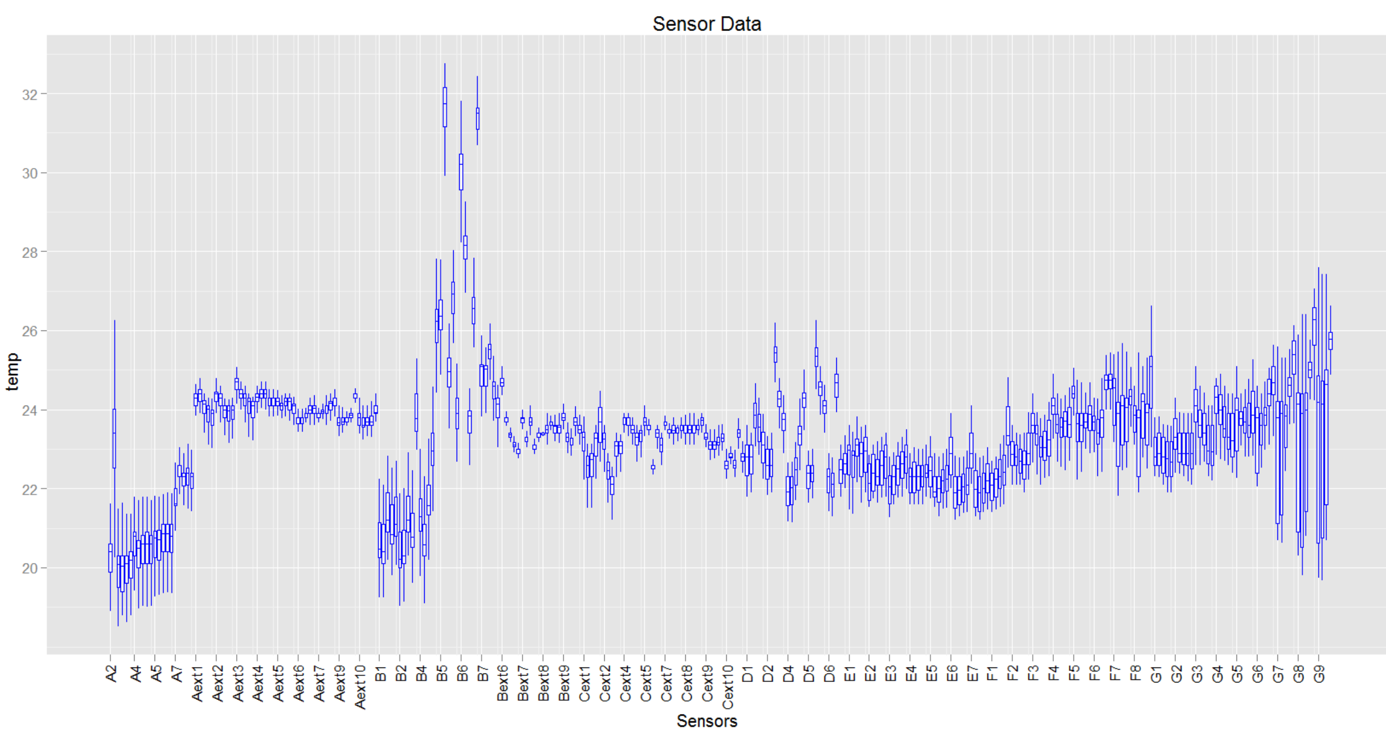

Figure 6: Variability in sensor temperatures in the seven day data. This figure shows 305 sensors located at inlets of 61 racks. The x-axis labels are names of racks (see Figure 3 for their locations in the data center). The rack label indicates the T1 sensor; it is followed by four other sensors (T2 through T5). The y-axis shows the sensor temperature in ˚C.

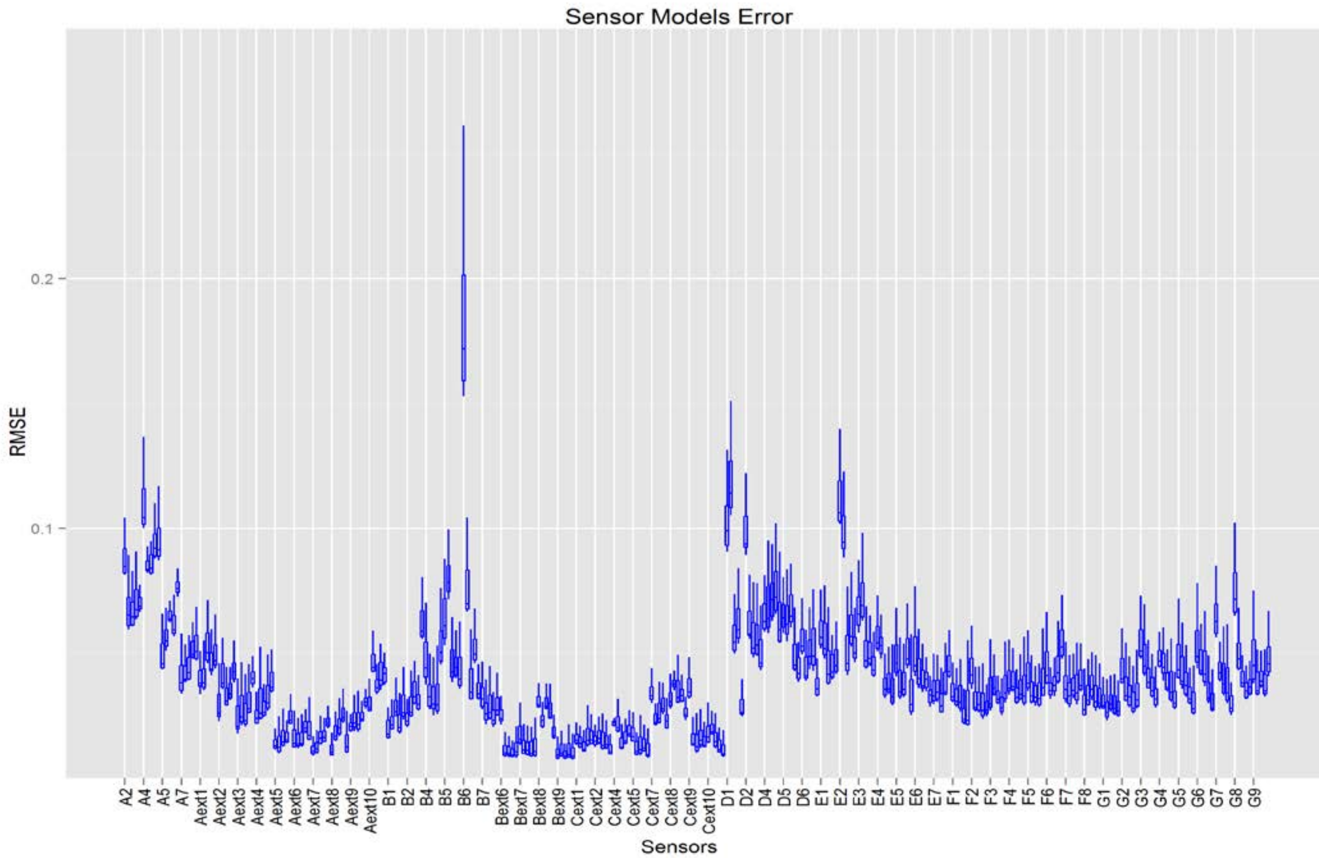

Figure 7: This shows the errors of the random forests sensor temperature models. The Y-axis is the root mean square error in ˚C.

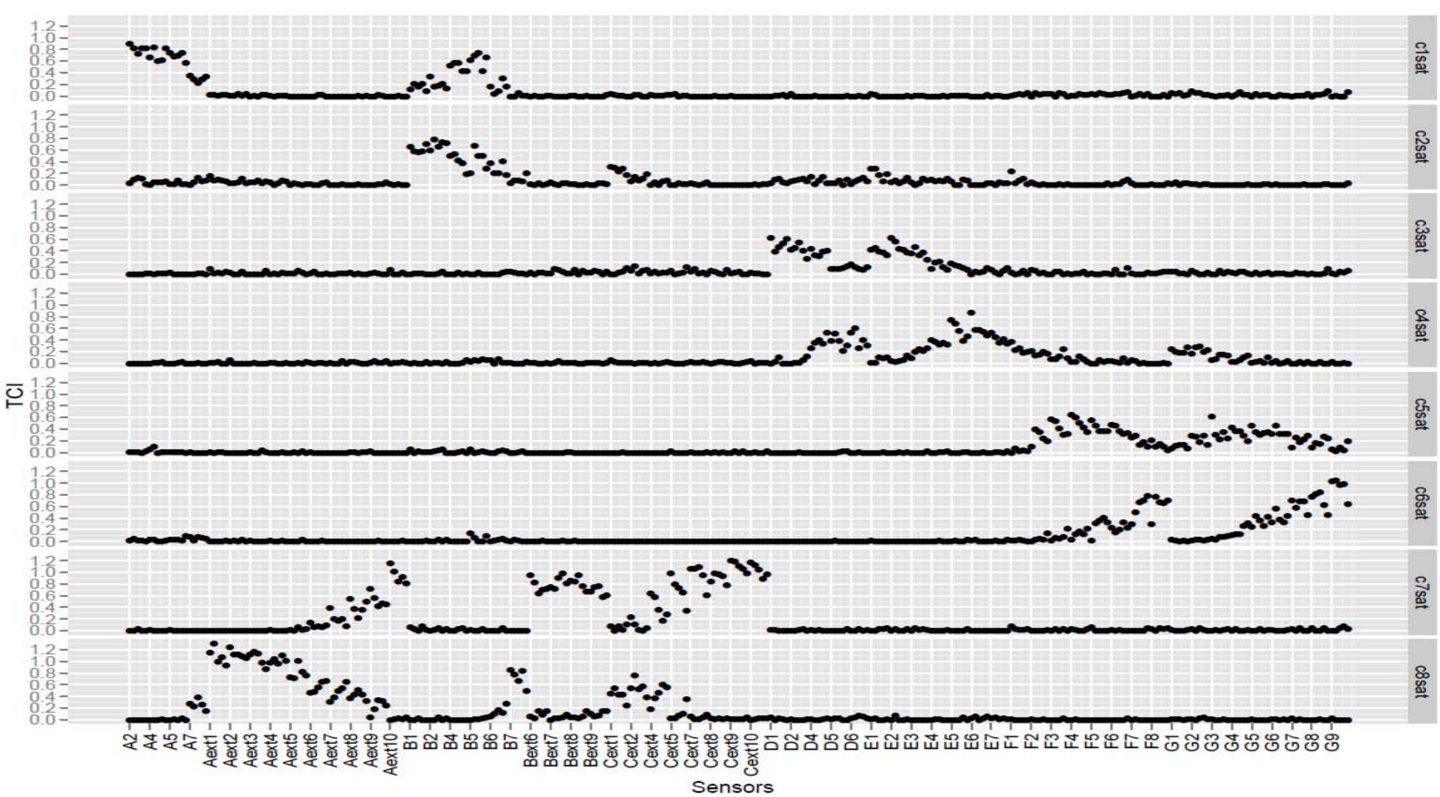

Figure 8: This shows the TCI values estimating using AutoTCI for all the rack sensors with respect to the eight CRAC SATs.

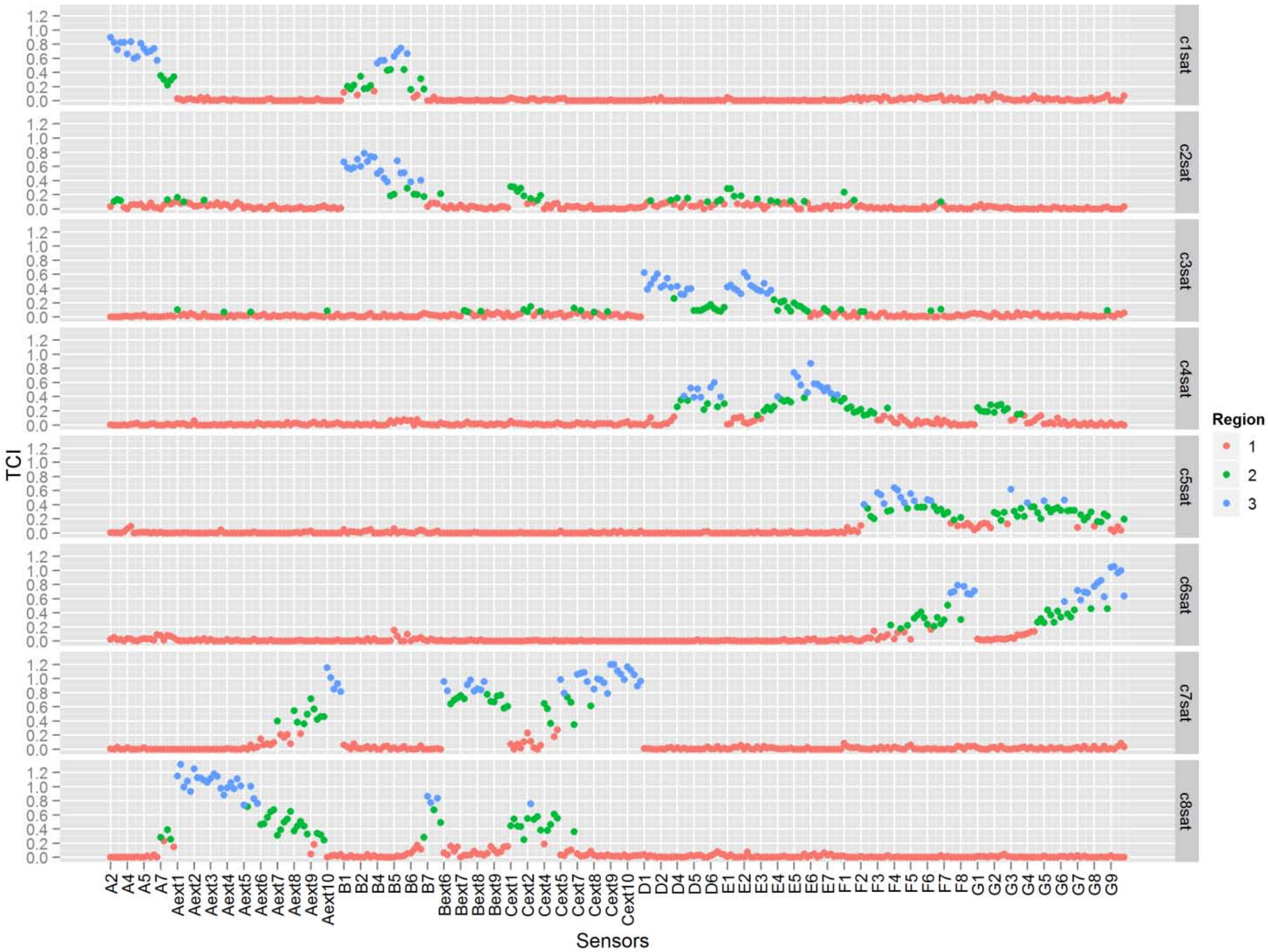

Figure 9: This shows the TCI values clustered into three different groups to allow differentiation into two levels within a thermal zone of a CRAC. Both the green and the blue sensors belong to a thermal zone, with the CRACs having a greater impact at the blue sensors than the green ones. The pink sensors do not belong to the thermal zone of the corresponding CRAC.

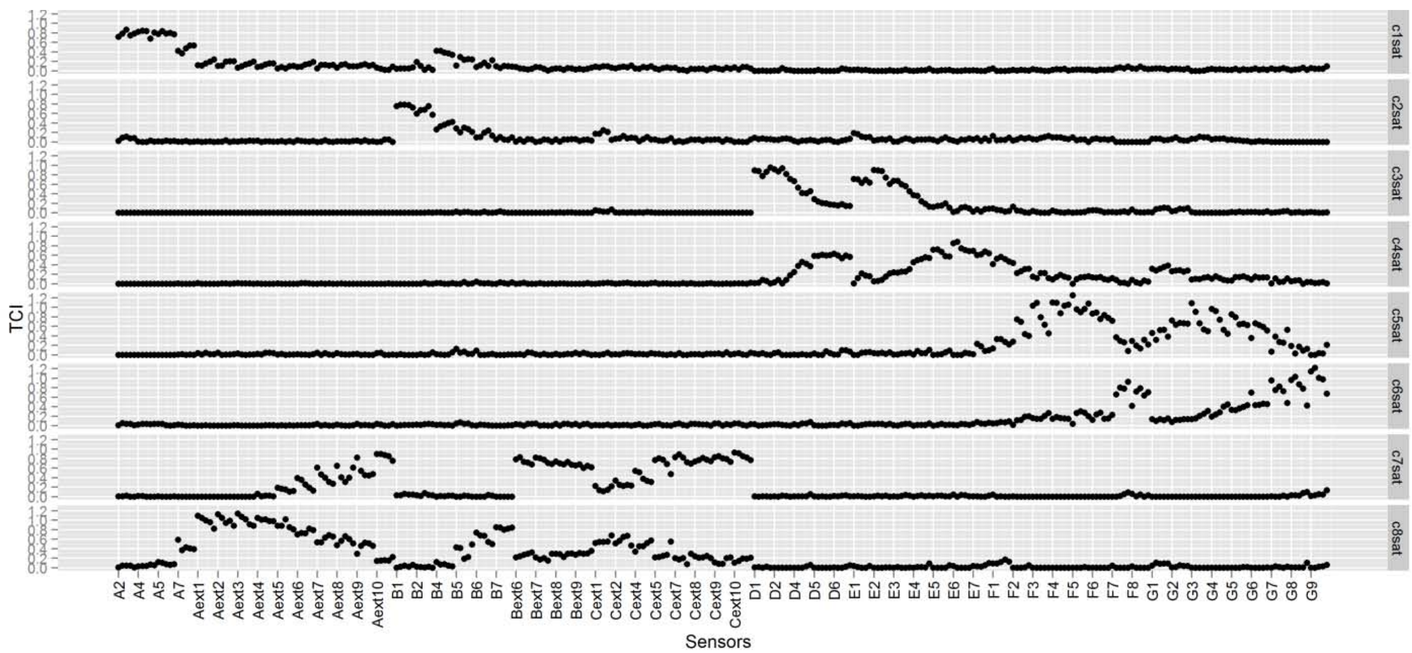

Figure 10: TCI values computed manually to validate the AutoTCI results.

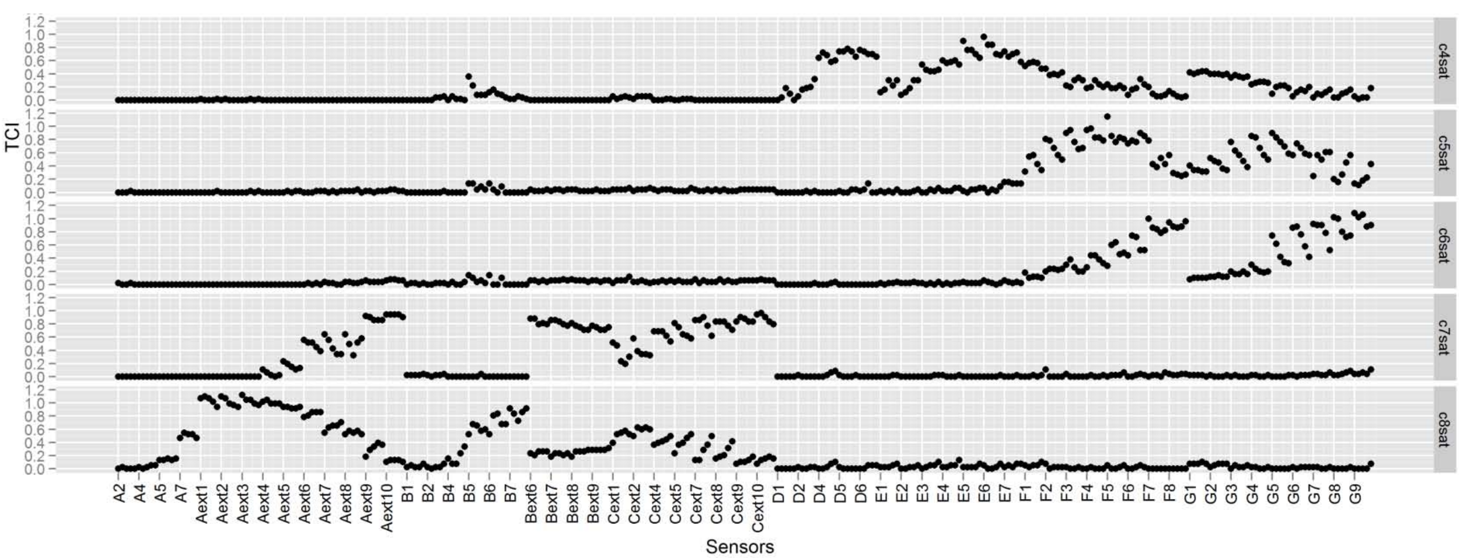

Figure 11: TCI values computed for CRACs 4 through 8 using the regular commissioning process.