Machine learning applied to building performance metrics

Tobias Maile

Introduction

The performance of air conditioning systems in buildings is hard to evaluate and typically includes an expert analyzing large time-series data sets to find operational patterns that are anomalous. Due to the size of the data sets, this is a time consuming process and thus very expensive. This leads to a small subset of buildings for which such an analysis has been done and its energy performance evaluated. This projects aims to apply machine learning algorithms to support experts in analyzing these huge time-series datasets. As a first step necessary preprocessing of the data was identified to eliminate irregularities in the raw data set. Furthermore, an anomaly detecting framework has been implemented to determine anomalies in the time series. Lastly some experimentation on clustering the datasets has been conducted.

The data set

The basic dataset consist of time-series from both observations and predictions. The predictions were created with a Building Energy Performance Simulation engine (namely EnergyPlus [1]). The measured data was collected at a naturally ventilated office building and consists of 30 different temperature points, a pressure difference measure, 10 velocity measures and window opening measurements. The building in question uses a natural ventilation strategy and these measurements have been installed to verify the performance of this innovative ventilation strategy. The initial design simulations have been updated with actual measurements resulting in fairly accurate predictions of the actual behavior of the building. In this particular case the raw data comparison did not lead to reliable results for the anomaly detection, since the differences between the observations and predictions did only have similar patterns, which had some differences in absolute values. Thus the data needed an additional preprocessing step besides the outlier detection and elimination and interval averaging. Thus it has been corrected by its mean and scaled on a daily subset basis. The predictions have been updated through the use of updated input. For example the measured outside air temperature has been used to input actual external conditions. The measured data is collected in one minute time steps and has been divided into daily sets. Since the predicted data is based on 10 minute time intervals the observed data also needed to be averaged over 10 minutes. There are two reasons why a division into daily sets is useful: Due to the dependency on the outside air temperature, which usually follows a daily pattern (low temperature during nights and high temperature in the afternoon), all time series follow this pattern. Furthermore, a day is a reasonable time frame to look at for further manual analysis in case anomalies have been detected.

Preprocessing

Three techniques have been used to preprocess the data set. To remove outliers a simple quartile algorithm has been implemented, furthermore, to minimize the differences between the observed and predicted data set, both have been normalized on a daily basis to time series with equal mean and scale. In addition, the different time intervals have been accounted for by averaging the one minute intervals over 10 minutes.

Outlier detection and elimination

Preprocessing of the data is necessary, since the measured data is raw data collected from sensors. The data contains some outliers due to false readings of the sensors. The following formulas have been used to identify the outliers:

$$X^{(i)} < Q_1 - 3 * IQR$$

$$X^{(i)} > Q_3 + 3 * IQR$$

Where Q_1 and Q_3 are the first and third quartiles of the data and IQR is the difference between Q_3 and Q_1. This simple procedure [2] could successfully detect the outliers in the data set; see Figure 1 for an example of a detected outlier.

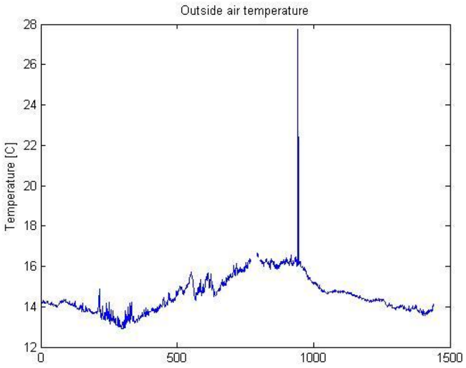

Figure 1: Outlier in time-series

Detected outliers where replaced through interpolated values of neighbor points. Since the outliers in this dataset are only single points this simple technique was able to detect all seven outliers in the dataset that were identified during the data analysis.

Anomaly detection

One major task during the data analysis of professional experts is to find days or part of days where the performance differs from the expected performance. This corresponds to anomaly detection in multivariate time series. While it is not very interesting to highlight all differences between the predicted and observed dataset, the algorithm needs to detect significant different patterns occurring in the dataset.

Related work

Recently there have been various efforts in anomaly detection and analysis of time-series. A technique described by Salvador et al. detects anomalies in single time-series based on states and rules [3]. This technique as a whole is not applicable for this data set, since the thermodynamic processes can not easily be described with different states and regular transitions between them. However, the referred data clustering technique Gecko and the L-method seem applicable to automatically estimate the needed number of clusters, in case clustering of resulting anomalies would have been pursued. In addition automated analyses for multivariate time-series are available. Bay et al. [4] use local models to describe subsets of a reference time-series and test the training data on the determined local models. Another technique (Hotelling’s T) is described by Ye at al. [5] is used to identify cyber attacks based on network traffic time-series. It uses norm profiles from training data to detect anomalies in the test dataset. Initial implementations of the Hotelling’s T method did not lead to reliable results. All related implementations have in common that a combination of algorithms has been used to fully manage anomaly detection.
Furthermore, various techniques to describe time-series with some kind of parameters are available. E.g. a technique to index multivariate time-series seems [6] to be a useful technique to represent time series in a lower dimensional space, but has not been further evaluated in this project due to time constrains.

Modified DARTS algorithm

One existing framework to detect anomalies in multivariate time series is the so-called DARTS algorithm [4]. The framework consists of 6 major steps illustrated in Figure 2.

1) Preprocessing
2) Create local models for the train data (predictions)
3) Create local models for the test data (observations)
4) Compare the local models in the parameter space and calculate the anomaly score
5) Calculate the anomaly cutoff
6) Identify the anomaly regions

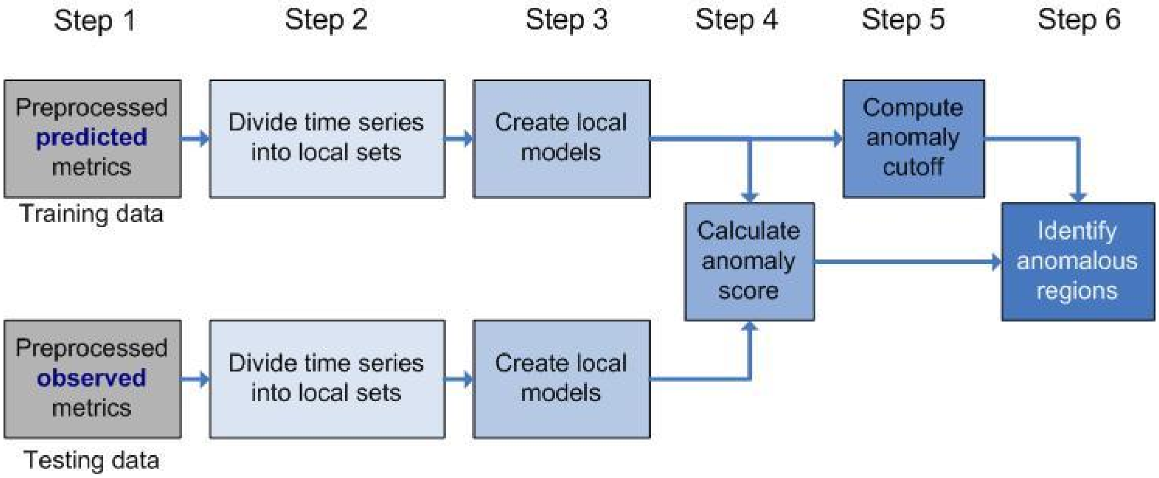
Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
Preprocessed predicted metrics
Training data
Divide time series into local sets
Create local models
Compute anomaly cutoff
Calculate anomaly score
Identify anomalous regions
Preprocessed observed metrics
Testing data
Divide time series into local sets
Create local models

Figure 2: Process of the modified DARTS framework

The first step has already been described earlier in this report.
The second and third step entail the estimation of local AR (autoregressive) model for the datasets. Two parameters characterize this process: the order of the AR-models and the length of the timeframe of the local models. While the order of the AR models should be relatively small to minimize data loss, the duration of the local models has major implications on the performance of the algorithm. If the size of the local models is small, the data can be fitted very well; however, this leads to overfitting and produces high anomaly scores for most of the testing data. On the other hand local models for a large time period underfit the data and produce fewer anomalies. For this dataset a timeframe of 4 hours which corresponds to 24 data points worked well.
For the forth step to compare the local models in the parameter space the probability density function has not produced reliable results. Bay et al. have identified the same problem in their work. The referred problem is that the multiplication of very small probabilities within the Kernel method tends to produce zero values in areas where anomalies occur. However, the k-th nearest neighbor Euclidian distance worked well for this dataset. From my experiments the number of k has a small influence to the actual detection of anomalies, if it stays within midrange values. Any k-value between 10 and 40 identifies the same set of anomalies.
The fifth step calculates a so-called anomaly cutoff that declares anomaly values that are above the cutoff for significant. It is determined by testing the local models of the train data with the train data itself. This significant anomaly values identify the detected anomalies, which is the last and sixth step.

This framework has produced reliable results for the dataset in question. While the actual manual data analysis is still in progress, there was no final identification of anomalies available. However, regions where this implementation produces anomalies show significant variations and or differences in the patterns and many of them have been part of the discussion during the data analysis. Two examples for detected anomalies of temperature time series are illustrated in Figure 3 and 4.

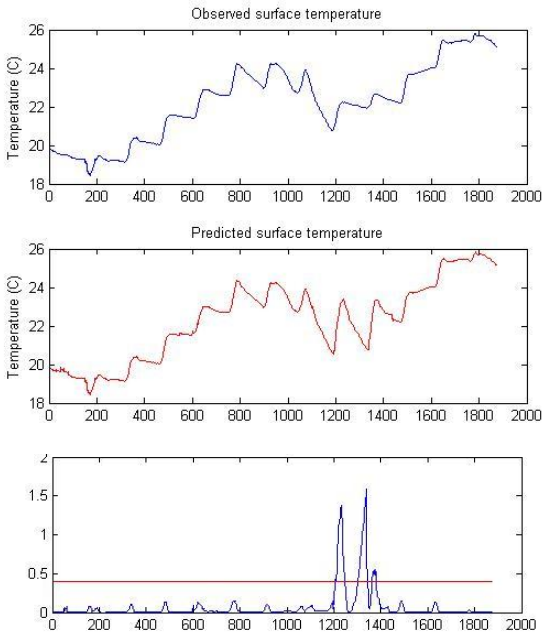

Figure 3: Ceiling surface temperature

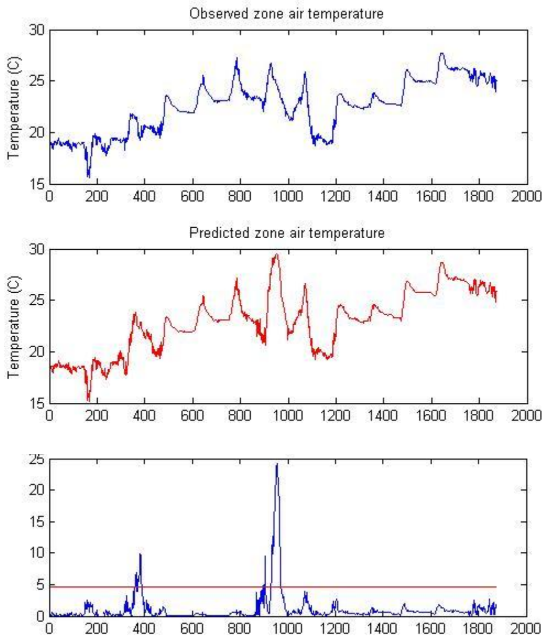

Figure 4: Zone air temperature

The two anomalies detected in Figure 3 have decreased slops and therefore smaller variations of the observed temperature (between 1200 and 1400). This is an anomaly which has been identified as one during the manual analysis process, but its cause was somewhat unresolved. One possible theory is that the air is moving on different paths than usual on these two days. Additional measurements have been installed in the meantime to further investigate this phenomenon.

The first identified anomaly in Figure 4 (just before 400) has not been identified as such during the manual process. However, one can see that there is significant difference between the two time-series. The second anomaly (between 900 and 1000) can be described as a higher peak in temperature during the day. Interestingly, the solar shield has fallen of the sensor allowing for a greater solar influence and thus a higher temperature reading occurs during the day.

Clustering (k-means)

Initial k-means clustering implementations could successfully categorize the outside air temperature into three different sets. One where there is only a small temperature difference between day and night, one where this difference is medium and one with a high difference. While this implementation has not been focused on further, it could be useful to create three sets of data based on the clustering and perform analysis for each of the subsets to better understand the differences that occur due to outside air conditions.

In addition, clustering of the anomalies after their detection may be useful to identify anomaly categories and support the data analysis further. Due to the limited number of anomalies for each characteristic time series there has no been any useful results for clustering of the anomalies. This may be done, if the dataset has grown bigger.

Conclusions

For the time consuming and expensive procedure of evaluating building energy performance based on manual data analysis of observed and predicted time series the modified DARTS framework has proven to be very useful in identifying anomalies of the available dataset. Outlier detection and normalization were needed to get the data in a form that was suitable for the anomaly detection. Within the framework the representation of the data within local models and the parameter space was used as a technique to highlight only the significant differences and/or anomalies in the dataset. Additional procedures could be implemented that determine some of the parameters found during testing, such as the length of the time window of local models, their AR-model order and others. While the described procedure worked well for this data set, it has to be tested for further data sets of this building and data set of additional buildings to validate its usefulness in general for this data analysis problem. Further methods such as clustering the detected anomalies or identifying possible sources of the anomalies could extend this work.

References

[1] EnergyPlus: www.energyplus.gov

[2] Mathematical definition of outliers at Wikipedia
http://en.wikipedia.org/wiki/Outlier

[3] Salvador, S.; Chan, P.; Brodie, J.: *Learning States and Rules for Time Series Anomaly Detection*, American Association of Artificial Intelligence, 2004

[4] Bay, S.; Saito, K.; Ueda, N.; Langley, P.: *A Framework for Discovering Anomalous Regimes in Multivariate Time-Series Data with Local Models*, 2004.

[5] Ye, N.; Chen, Q.; Emran, S.M.; Vilbert, S: *Hotelling's T^2 Multivariate Profiling for Anomaly Detection*, Proceedings of the 2000 IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY, 6-7 June, 2000.

[6] Vlachos, M.; M. Hadjieleftheriou, M.; Gunopulos, D.; Heogh, E.: *Indexing multi-dimensional time-series with support for multiple distance measures*. In Proc. ACM SIGKDD, pages 216-225, 2003.