LSTM for VAV System Multivariate Anomaly Detection

Shanghao Chen

January 29, 2019

1 Introduction

In this project, an anomaly detector is implemented based on LSTM (Long Short Term Memory Networks), which consists of a two-stage strategy of time-series prediction and anomaly score calculation. Two rule insights (daily and weekly) from 'VAV-755' system in TASCII building have been tested for this anomaly detection model.

2 Model Structure

We first train this model with a training set which contains no anomalies, then we use the trained model to detect anomalies based on anomaly scores in a test set, where anomalies are included.

2.1 Time-series Prediction Structure

A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit temporal dynamic behavior for a time sequence. An RNN composed of LSTM units is often called an LSTM network.

For each LSTM unit, its input are the $\hat{x}_t$ at that time-instant and the last time-instant output hidden state h_{t-1}. And its output are the next time prediction $\hat{x}_{t+1}$ and the time hidden state h_t. Hidden state is the internal operation process of LSTM, which can be briefly considered to store the information of the previous time series.

In order to ensure that the networks capture the normal structure of the sequence, 1-step predictions and multi-step predictions have been used. The following Figure 1 is an example for time series prediction. [4]

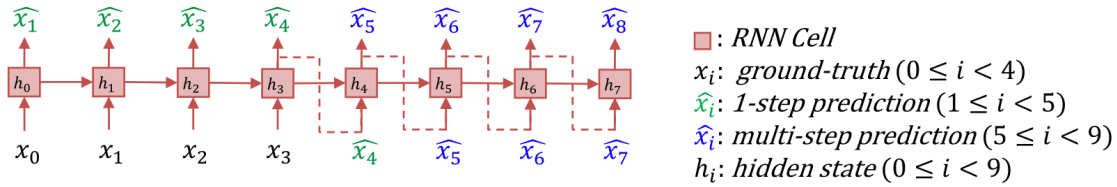

Figure 1: 1-step and multi-step predictions.

x_i is a vector contains multivariate value at time instant i. When the value of x_i is known from $i = 0$ to $i = t$, the model recursively predicts the value of x_i from $i = t + 1$ to $i = T$. In this figure, $t = 3$, $T = 8$.

2.2 Composition of the Loss Function

The loss function is a central part of any machine learning model. The training algorithm calculates how changes to any model parameter (e.g. any network weight) causes a corresponding change in loss. This results in gradient vectors of model parameters that are used to gradually improve the model.

The calculation of the loss function for LSTMs consists of three parts, teaching forcing loss, free running loss and simplified professor forcing loss. All three of them use the mean squared error(MSE).

$$\text{Loss} = \text{Teacher Forcing Loss} + \text{Free Running Loss} + \text{Professor Forcing Loss}$$

2.2.1 Teacher Forcing Loss

Teacher forcing works by using the actual or expected output from the training dataset at the current time step $x(t)$ as input in the next time step $x(t+1)$, rather than the output generated by the network. Teacher forcing is a strategy for training recurrent neural networks that uses model output from a prior time step as an input. The following Figure 2 is an example for teacher forcing loss. [4]

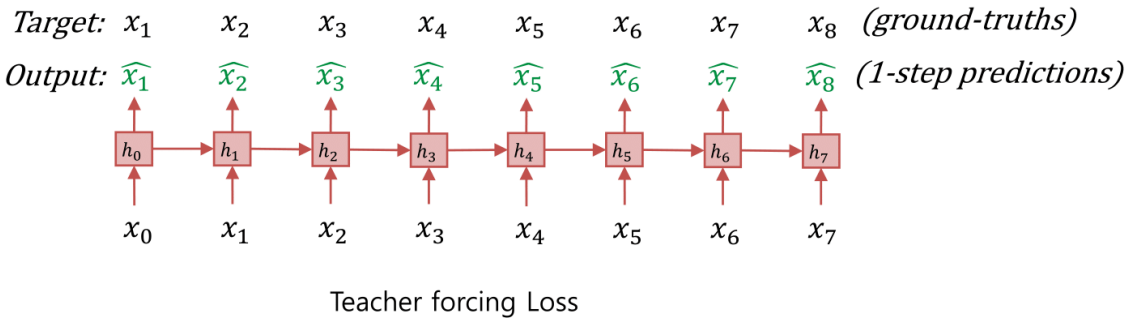

Figure 2: Teacher forcing loss.

2.2.2 Free Running Loss

Free running loss works by using one time point value to predict n-step(prediction window size) value to compare with n-step real value. The following Figure 3 is an example for free running loss. [4]

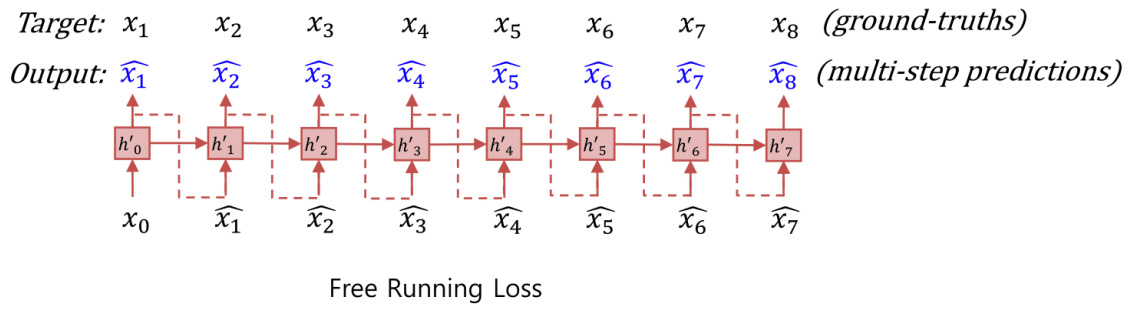

Figure 3: Free Running Loss.

2.2.3 Simplified Professor Forcing Loss

The basic idea of professor forcing is simple: while we do want the generative RNN to match the training data, we also want the behavior of the network (both in its outputs and in the dynamics of its hidden states) to be indistinguishable whether the network is trained with its inputs clamped to a training sequence (teacher forcing mode) or whether its inputs are self-generated (free-running generative mode).

Therefore, after training the neural network once, we can calculate the difference between the teacher forcing and free running hidden states. The following Figure 4 is an example for Simplified Professor Forcing Loss. [1]

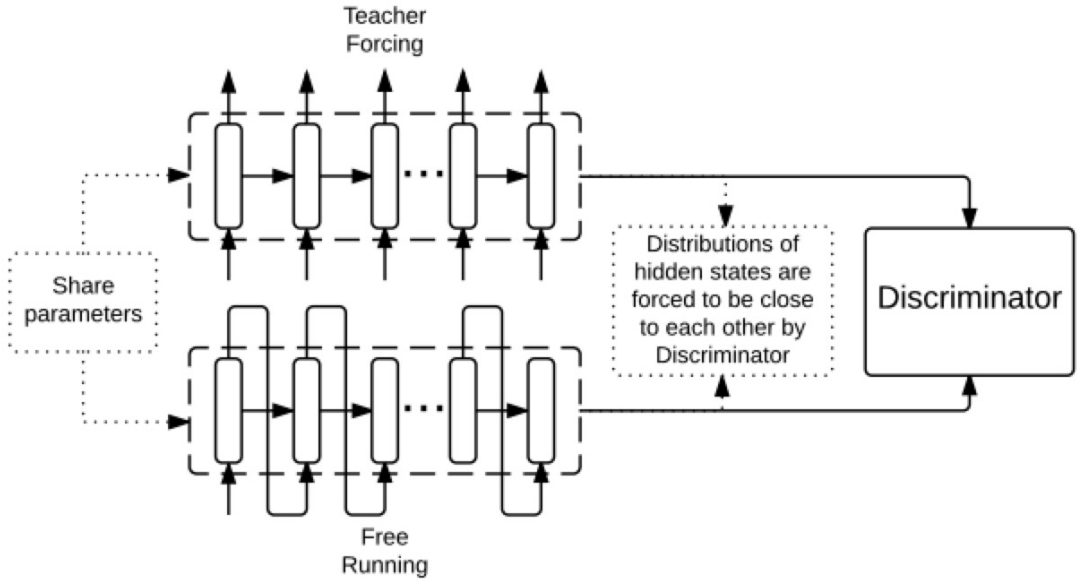

Figure 4: Simplified professor forcing loss structure.

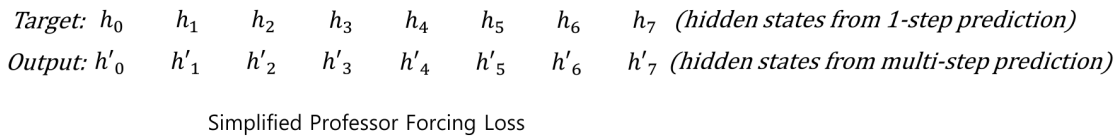

Figure 5: Simplified professor forcing loss.

2.3 Anomaly Scores Calculation

Consider a time series $X = \left\{x^{(1)}, x^{(2)}, ..., x^{(n)}\right\}$, where each point $x^{(t)} \in R^m$ is an m-dimensional vectors $\left\{x_1^{(t)}, x_2^{(t)}, ..., x_m^{(t)}\right\}$, whose elements correspond to the input variants. The prediction model learns to predict the next l time steps for 1 of the input variants.

The anomaly score described in the following is essentially a z-score of prediction differences, normalized over an interval of predictions with window length l, done separately for each of the m varieties.

To simplify, we consider 1-step prediction for a single variate only, $m = 1$, which means the prediction model learns to predict the next 1 time step for one variate, as in Figure 2. With a window length of l, one variate $x^{(t)} \in X$ at time-instant t $(1 \leq t < l)$ has a prediction $\widehat{x^{(t)}}$. We compute an error vector $e^{(t)}$ for variate $x^{(1)}$ to $x^{(l)}$ as $[e_1^{(1)}, e_1^{(2)}, ...e_1^{(l)}]$, where $e_i^{(t)}$ is the difference between its value $x^{(t)}$ and predicted value $\widehat{x^{(t)}}$ at time t.

The error vector for one variate $(m = 1)$ (e.g. $E = [e_1^{(1)}, e_1^{(2)}, ...e_1^{(l)}]$) is modelled to fit a Normal distribution $E \sim N(\mu, \sum)$ with the parameters μ and $\sum$ determined using maximum likelihood estimation. Then, for any point $x_i^{(t)}$, the anomaly score at time t for this 1-dimensional variate is as follows:

$$\text{anomaly\_score}^{(t)} = (e_1^{(t)} - \mu)^T \Sigma^{-1} (e_1^{(t)} - \mu)$$

When we use multi-step prediction, the error vectors for one variate at time t is $e_1^{(t)}$ is a vector with size $[1 \cdot \text{prediction\_size}]$. The error vectors are modeled to fit a multivariate Gaussian distribution and use the same way to calculate the anomaly scores.

3 VAV-755 System

3.1 Trend Logs

Table 1 describes all the VAV-755 system's trend logs information.

Info / Id	TL Object	Units	Type(Sample Time)
3351.200.TL3	AHU3_SAT_TL	$°C$	Polled(15min)
3351.200.TL19	AHU3_SFC_TL	NA	Cov
3351.274.TL1	VAV-755_RT_TL	$°C$	Polled(15min)
3351.274.TL11	VAV-755_HCV_OPEN_POLL_TL	NA	Polled(15min)
3351.274.TL12	VAV-755_CCV_OPEN_POLL_TL	NA	Polled(15min)
3351.274.TL2	VAV-755_RT_SP_TL	$°C$	Polled(15min)
3351.274.TL3	VAV-755_FLW_TL	L/S	Polled(5min)
3351.274.TL4	VAV-755_FLW_SP_TL	L/S	Polled(5min)
3351.274.TL6	VAV-755_DMP_TL	%	Polled(5min)
3351.274.TL7	VAV-755_SAT_TL	$°C$	Polled(15min)
3351.274.TL8	VAV-755_SAT_SP_TL	$°C$	Polled(15min)

Table 1: VAV-755 system's trend logs

3.2 Rule Insights

In the final test phases, we tested the following two rules insights for the anomaly detection model.

1. FDD-10007D(Daily): Determine if damper (Analog) requires tuning - VAV-755

   Description: Determine if a damper output is changing direction (Default 1 cycle = 5% change) too many times.Triggers an insight if the number of changes is more than 18 (Default) a day.

   Diagnosis:

   1.Input sensor(s) faulty or miscalibration. Verify input sensor(s) operation.

   2. Error in control sequence. Review and modify programmed sequence of operation.

   3. Issue with control loop. Review and modify controller tuning parameters.

2. FDD-20011D(Weekly): Determine if zone damper at 100% during occupied period for too long - VAV-755

   Description: Determine if zone damper is at 100% for more than 22 hours (Default) per week.

   Diagnosis:

   1. Input sensor(s) faulty or miscalibration. Verify input sensor(s) operation.

   2. Equipment faulty. Verify equipment operation.

   3. Primary air not available. Verify Air Handling Unit operation.

The following Table 2 explains why other rule insights were not tested.

Name	Description	Reason
FDD-10007HTG	Determine if HCV (Analog) requires tuning	no record
FDD-10007CLG	Determine if CCV (Analog) requires tuning	no record
FDD-20011D	Determine if zone damper at 100% for too long	no record
KPI-10001	Performance KPI - Zone RT vs RT_SP	KPI and no record
KPI-10003	Performance KPI - Zone AFLW vs AFLW_SP	KPI

Table 2: Not Test Rule Ingsihts

3.3 Data Interpolation and Resampeling Before Testing

In order to get the data for the same data sample interval, we re-sampled all the trend logs to 15 minutes and used linear interpolation to eliminate missing values.

4 FDD-10007D Anomaly Detection(Daily)

4.1 Training and Testing Data

The training data (3725 samples) is from 2016-11-07 to 2016-11-20 and all the data from training period is normal. The testing data (5732 samples) is from 2016-10-17 to 2016-11-06 and the normally and anomaly behavior generated history is as the following Table.

Time Period	State
2016-10-17 00:00:00 to 2016-10-19 00:00:00	anomaly
2016-10-19 00:00:00 to 2016-10-24 00:00:00	normally
2016-10-24 00:00:00 to 2016-10-25 00:00:00	anomaly
2016-10-25 00:00:00 to 2016-10-31 00:00:00	normally
2016-10-31 00:00:00 to 2016-11-01 00:00:00	anomaly
2016-11-01 00:00:00 to 2016-11-02 00:00:00	normally
2016-11-02 00:00:00 to 2016-11-04 00:00:00	anomaly

Table 3: Testing data normally and anomally behavior history

4.2 Testing Result with Filter Label

In this subsection, we add a new feature filter, which indicates whether data for this time stamp is missing or not.

4.2.1 Model Inputs

The following Table 4 indicates the input multivariate for training and testing. The second Table explains why some trend logs are deleted before training and testing.

ID	TL Object
0: 3351.200.TL3	AHU3_SAT_TL
1: 3351.274.TL1	VAV-755_RT_TL
2: 3351.274.TL3	VAV-755_FLW_TL
3: 3351.274.TL6	VAV-755_DMP_TL
4: 3351.274.TL7	VAV-755_SAT_TL
5: filter	missing data or not
6: label	normal or abnormal data

Table 4: Model input multivariate

ID	TL Object	Reason
3351.274.TL11	VAV-755_HCV_OPEN_POLL_TL	constant value
3351.274.TL12	VAV-755_CCV_OPEN_POLL_TL	almost constant(6)
3351.274.TL4	VAV-755_FLW_SP_TL	almost constant(2)
3351.274.TL2	VAV-755_RT_SP_TL	almost constant(2)
3351.274.TL8	VAV-755_SAT_SP_TL	almost constant(2)

Table 5: Deleted trend logs From input

4.2.2 Anomaly Detection Results

From the Figures 6-11, the anomaly detection details are shown as follows:

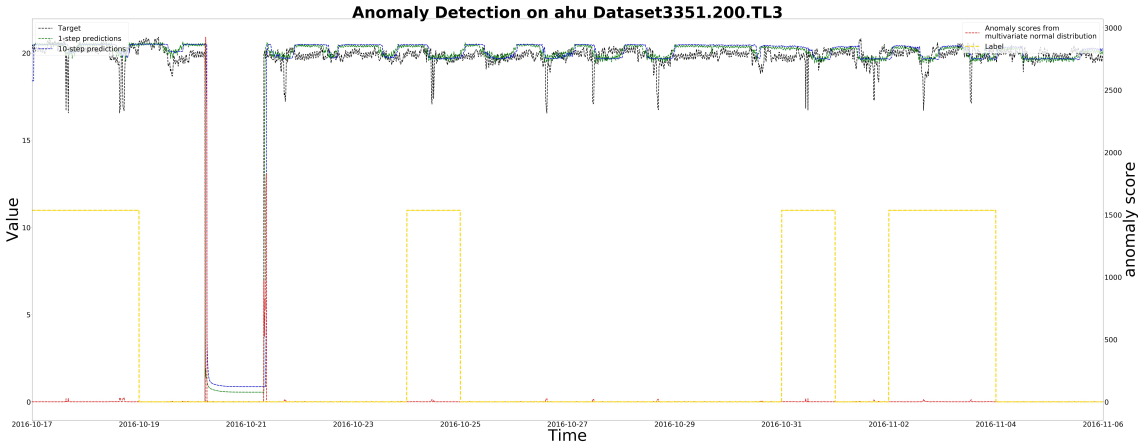

Figure 6: 3351.200.TL3 anomaly detection

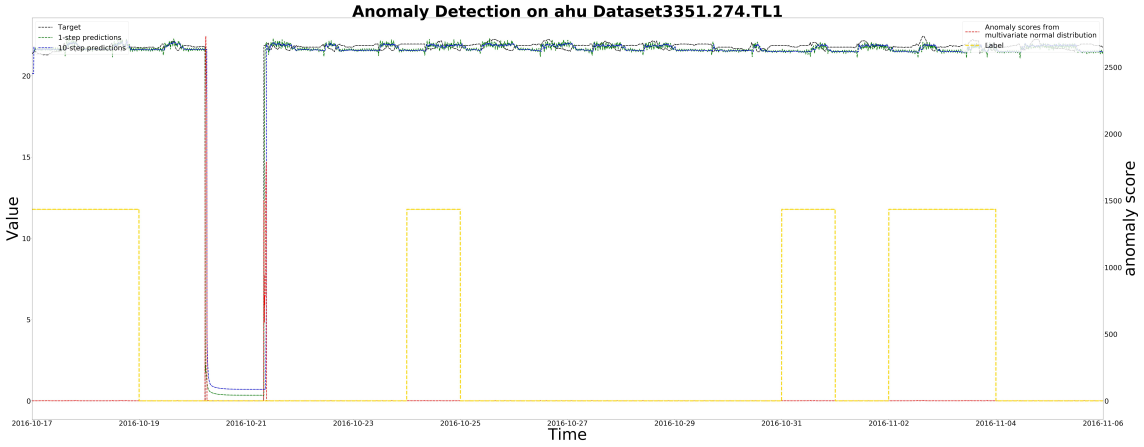

Figure 7: 3351.274.TL1 anomaly detection

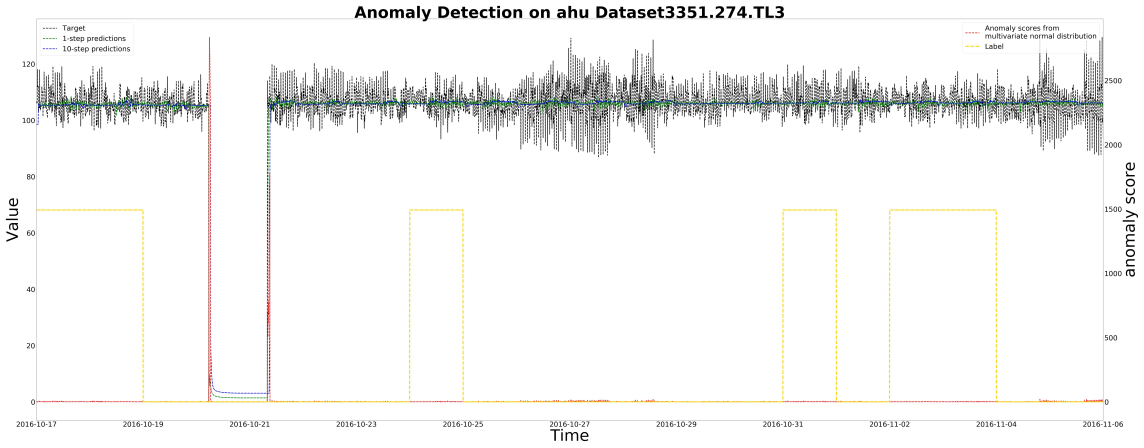

Figure 8: 3351.274.TL4 anomaly detection

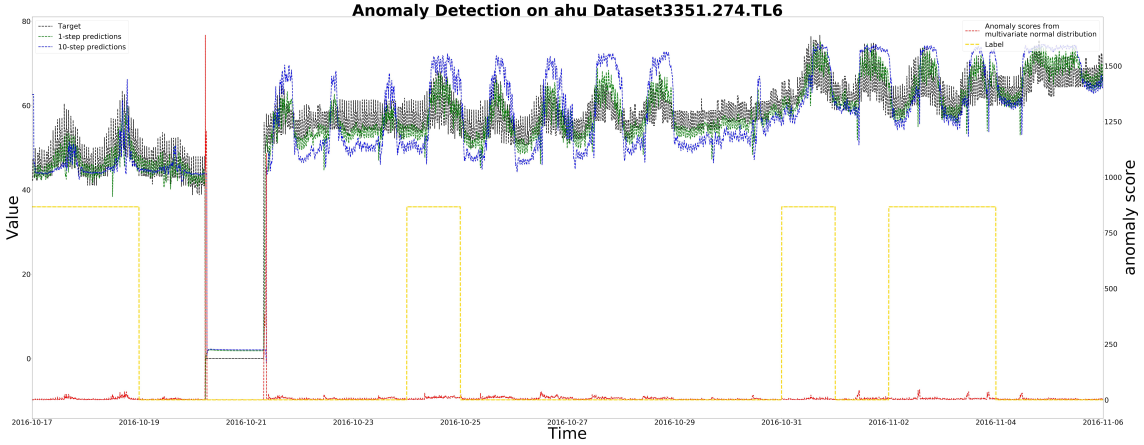

Figure 9: 3351.274.TL2 anomaly detection

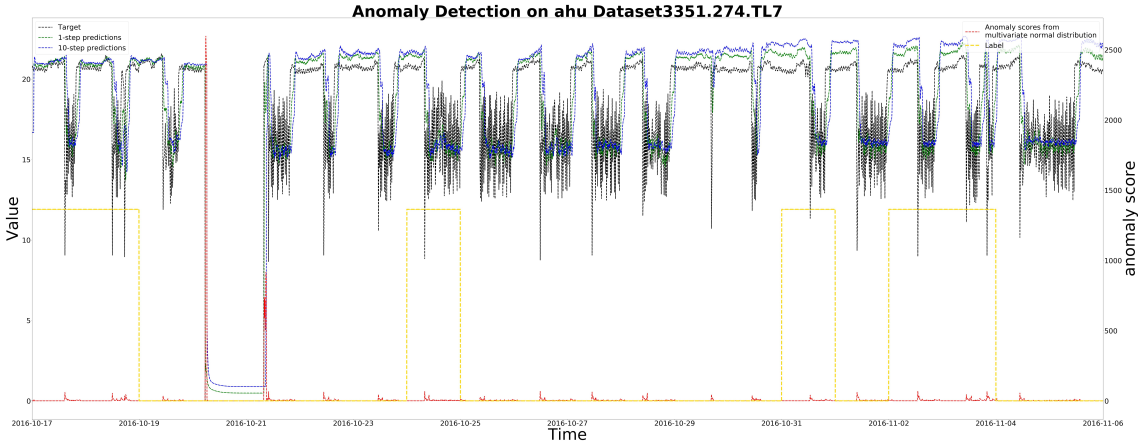

Figure 10: 3351.274.TL8 anomaly detection

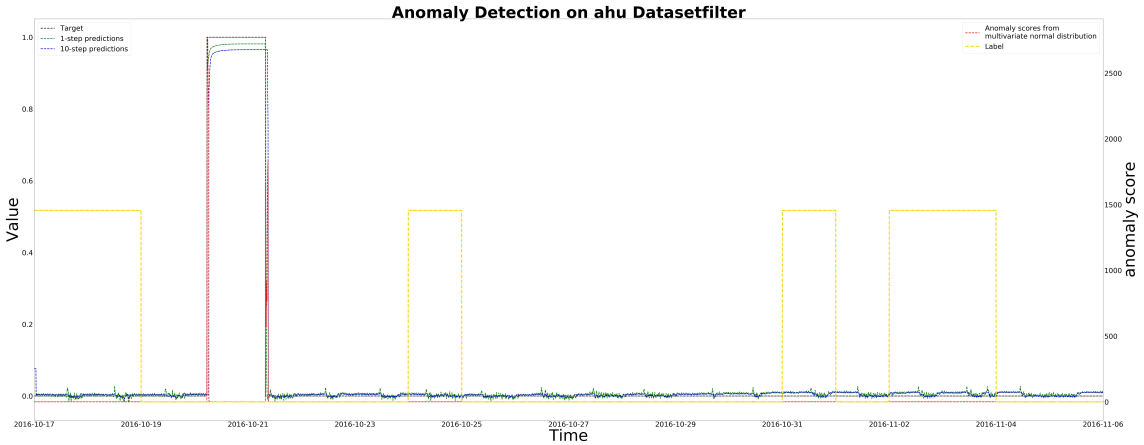

Figure 11: Filter

4.3 Testing Result with Sin and Cos

In this subsection, we add four features as input, which deriving a sine transform and cosine transform of the day and week.

4.3.1 Model Inputs

The following Table 6 indicates the input multivariate for training and testing. The second Table explains why some trend logs are deleted before training and testing. Also the deleted trend logs for model input are same as Table 5.

ID	TL Object
0: sin_seondOfday	Sin second for one day
1: cos_seondOfday	Cos second for one day
2: sin_dayOfweek	Sin day for one week
3: cos_dayOfweek	Cos day for one week
4: 3351.200.TL3	AHU3_SAT_TL
5: 3351.274.TL1	VAV-755_RT_TL
6: 3351.274.TL3	VAV-755_FLW_TL
7: 3351.274.TL6	VAV-755_DMP_TL
8: 3351.274.TL7	VAV-755_SAT_TL
9: label	normal or abnormal data

Table 6: Model input multivariate

4.3.2 Anomaly Detection Results

From the Figures 12-20, the anomaly detection details are shown as follows:

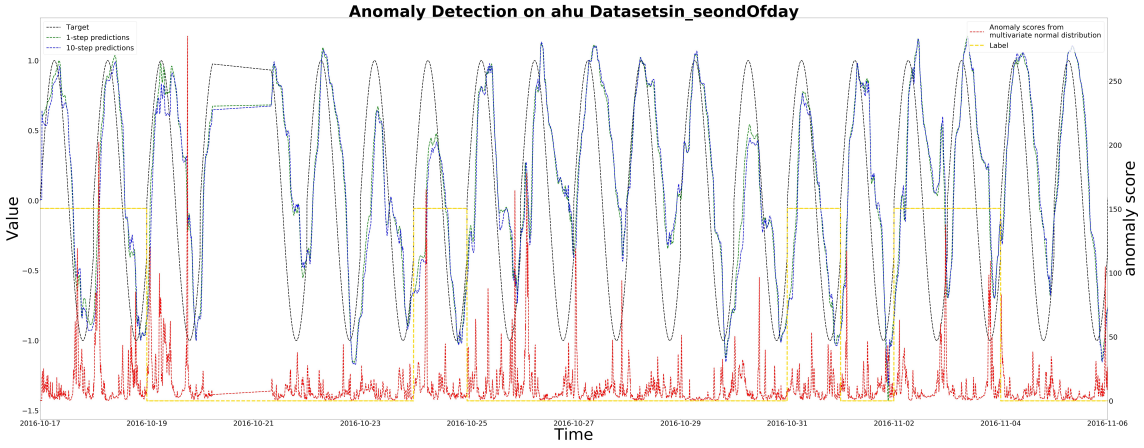

Figure 12: sin_seondOfday

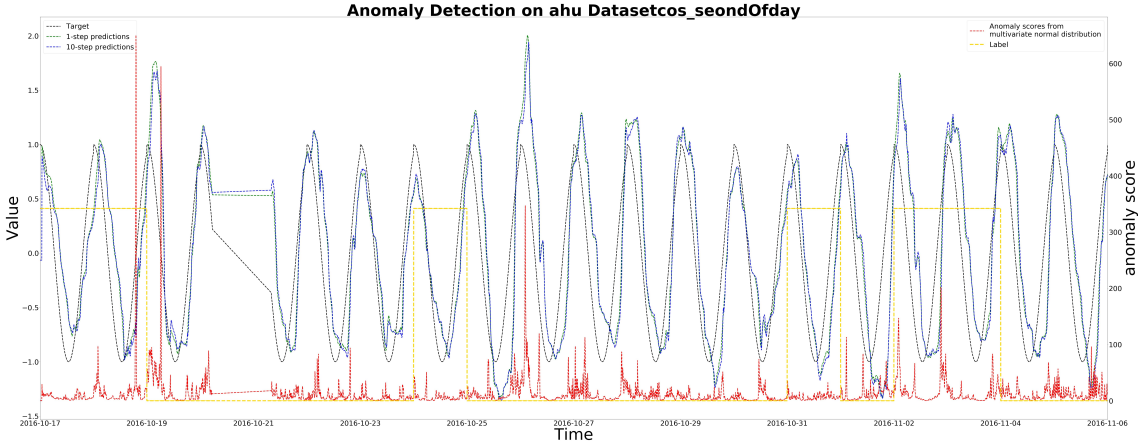

Figure 13: cos_seondOfday

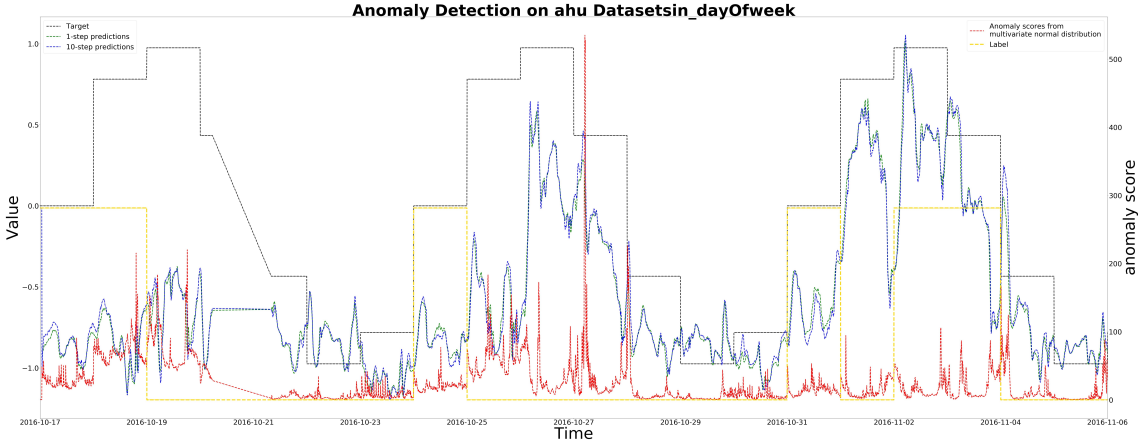

Figure 14: sin_dayOfweek

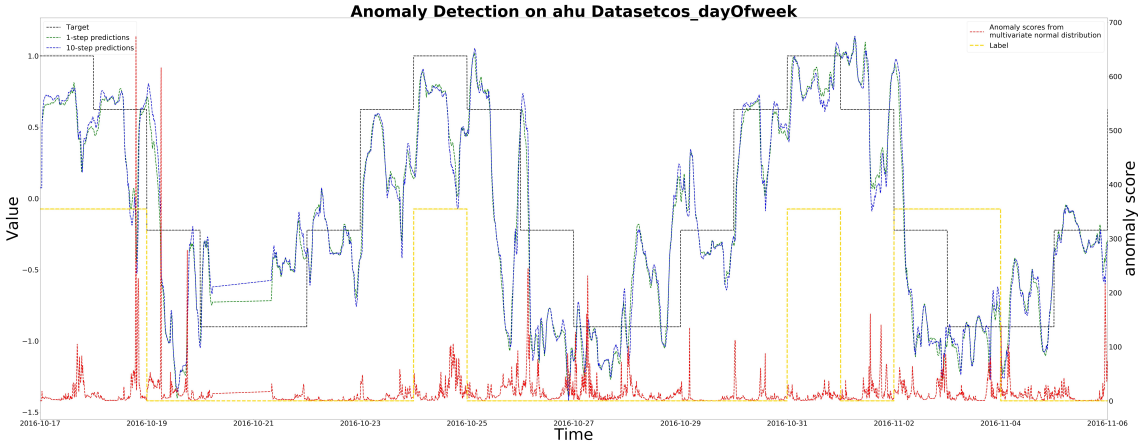

Figure 15: cos_dayOfweek

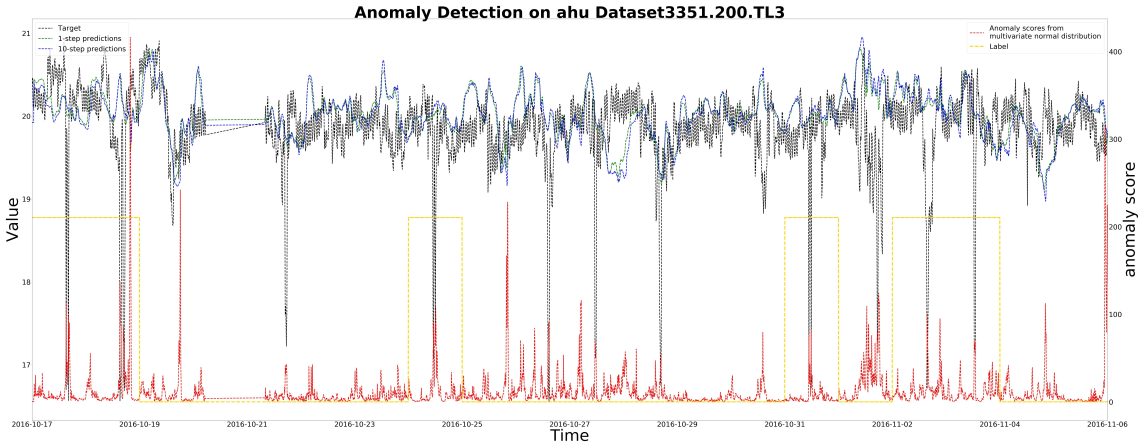

Figure 16: AHU3_SAT_TL

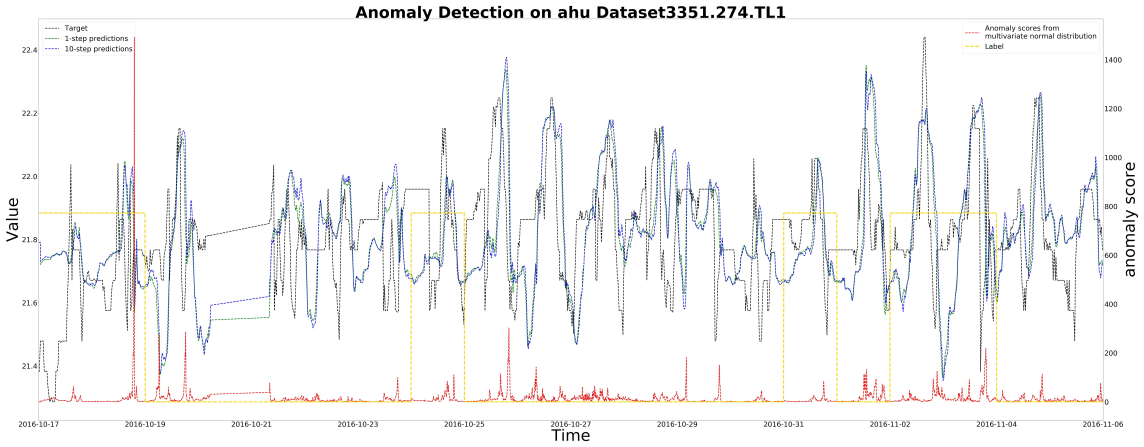

Figure 17: VAV-755_RT_TL

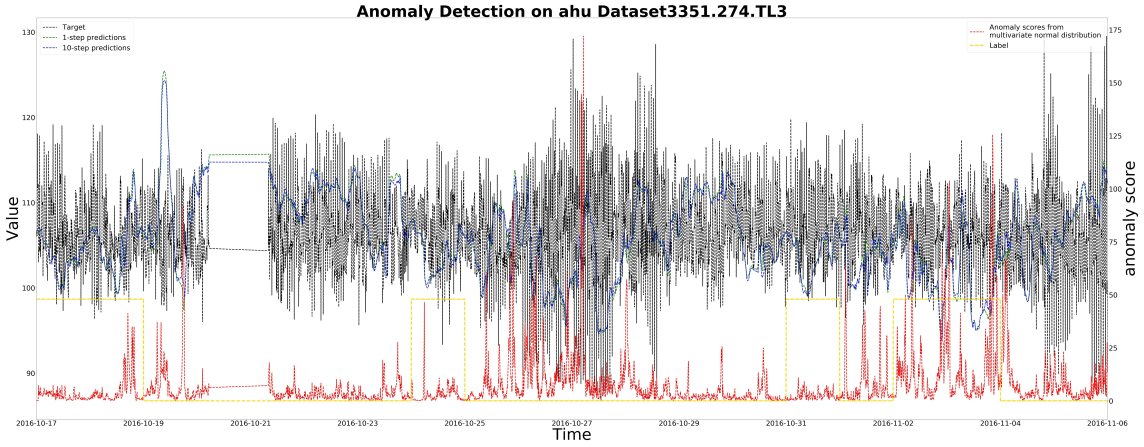

Figure 18: VAV-755_FLW_TL

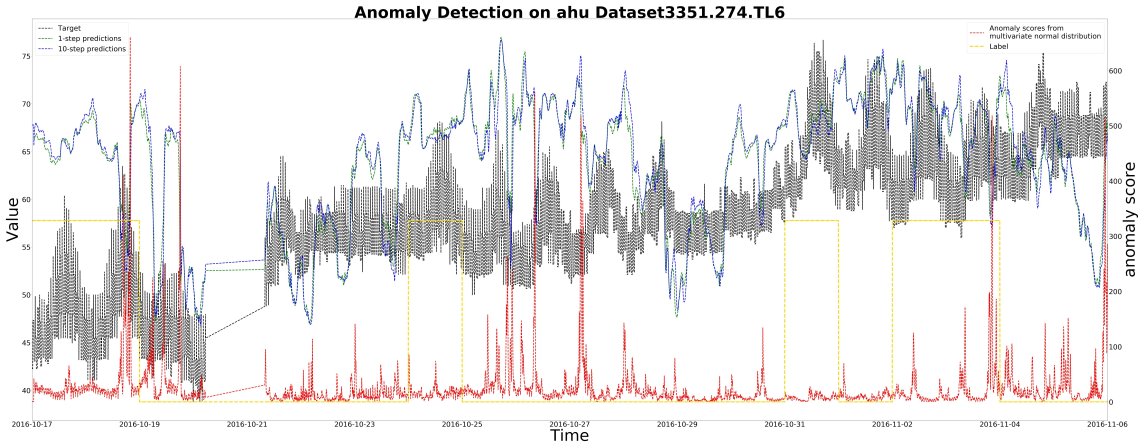

Figure 19: VAV-755_DMP_TL

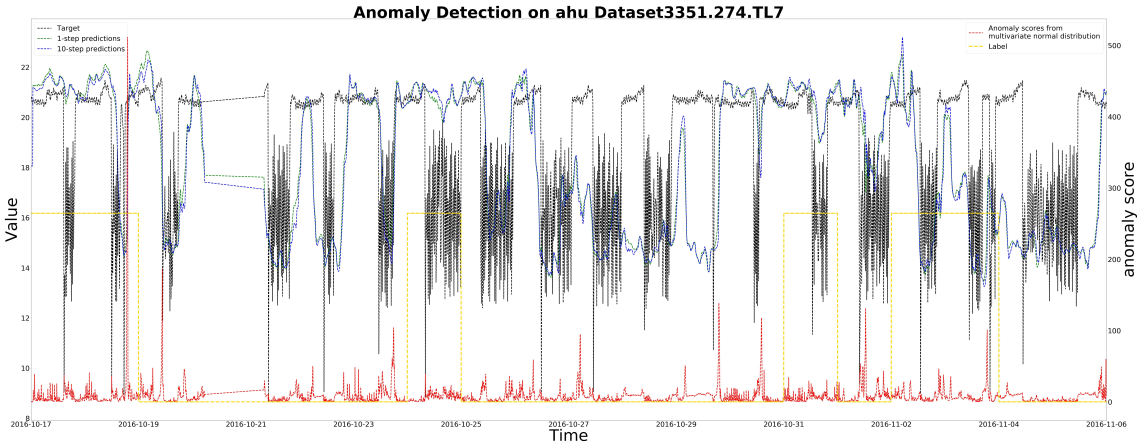

Figure 20: VAV-755_SAT_TL

4.3.3 Conclusions for Two Approaches

The test results from the previous subsections led us to the following conclusions:

1. From Figure 16, we found that the AHU3_SAT_TL anomaly score have a high correlation with rule insight FDD-10007D although this rule insight is not generated based on this trend log.
2. Compared to add day features (sin and cos) to deal with missing data, using a filter to mark the data is not a good solution.
3. This model cannot predict well when the trend log is abrupt upper or down for VAV system. We need to put all the interrelated trend logs to the model.

4.4 FDD-20011D(Weekly)

If a week is considered anomaly, there will be consecutive 2016 series data points are considered anomaly which led difficult to anomaly detection. For this reason, we didn't get good results for this rule insight.

5 Future Work

5.1 Anomaly Detection Model Optimization

5.1.1 LSTM-based Encoder-Decoder for Multivariate Anomaly Detection

The LSTM-based Encoder-Decoder model learns to reconstruct 'normal' time-series behavior, and thereafter uses reconstruction error to detect anomalies. Compared to the current model, it has two separate LSTM layers (encoder and decoder).
From the relevant literature tests, LSTM-based Encoder-Decoder performs better than the LSTM prediction model since 'reconstruction' is easier than 'prediction'. [2] [3]
The following Figure 21 illustrate the LSTM-based Encoder-Decoder for multivariate anomaly detection.

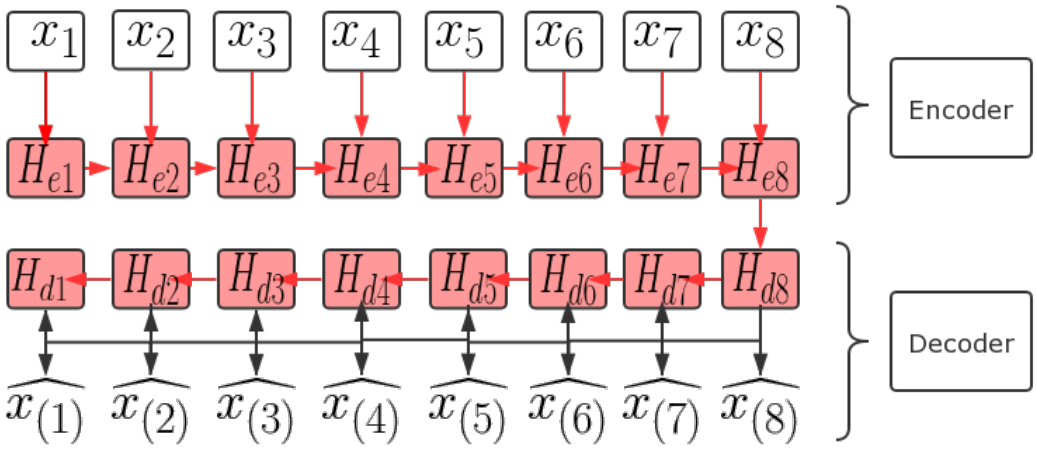

Figure 21: LSTM-based Encoder-Decoder Model

5.2 Implement Anomaly Detection for Other Systems

So far, we only tried the LSTM anomaly detection model for basic VAV systems. It would be interesting to also apply this model to more complex systems. To identify other suitable systems, we could start by reviewing the System Description Document for the TASC-II building.

5.3 Online Anomaly Detection Framework

Once a system's LSTM anomaly detection model is initialized based on the historical data stream, it can be used for the online assisted anomaly detection. For each accumulated batch of new streaming data, the model will predict it as normal or anomaly. If the data is predicted as anomaly behavior, someone from facilities management will get a warning of a potential anomaly. If this warning is not confirmed, the model will be updated.

Compared to the rule-based anomaly detection, this method could output anomaly warnings earlier and predict anomaly behavior which has never occurred. The following Figure 22 is a brief overview of the online anomaly detection framework.

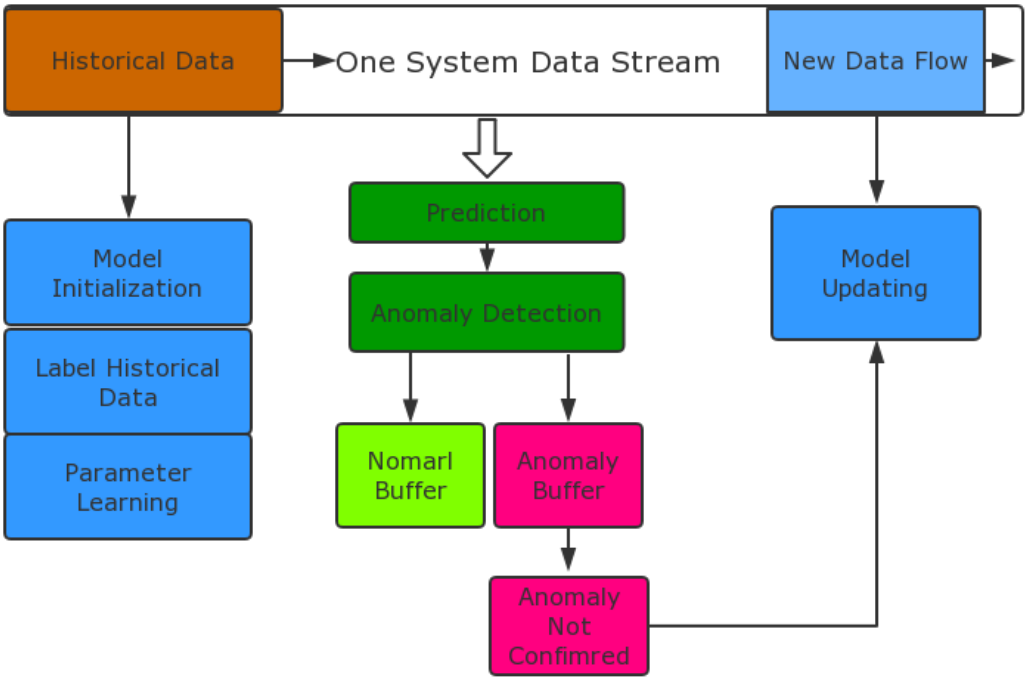

Figure 22: Online anomaly detection framework

Reference

[1] Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pages 4601–4609, 2016.

[2] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

[3] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. In *Proceedings*, page 89. Presses universitaires de Louvain, 2015.

[4] Jinman Park. RNN based Time-series Anomaly Detector Model Implemented in Pytorch. https://github.com/chickenbestlover/RNN-Time-series-Anomaly-Detection, 2018.