

# PUE optimization problem with small server room

## SI 670: Applied Machine Learning Final Project

**Amber Wu**  
wuca@umich.edu

**Yung-Chun Lee**  
yungclee@umich.edu

### Abstract

In this report, we used a simple LSTM model to predict the ambient temperature of the room and try to trigger the AC control room switch ahead if the predicted temperature exceed the default threshold. Our proposed method still need to be tested on the real environment to show if can reduce the PUE by triggering the power before the event happens.

## 1 Motivation

The number of Data center has grown exponentially ever since we entered the data era, Google has published its results in successfully reducing the total power usage by increasing the PUE (Power usage effectiveness) by using Machine Learning. In this study, we are interested in if we can use the similar approach and apply their method onto a smaller and more common setup - server room. Server room are common for medium even a small size company to maintain their data or provide online services. We aim to find out given the environment parameters such as temperature, humidity, and  $CO_2$  level in the server room with proper power usage monitor the overall power usage can be optimized through machine learning algorithm.

## 2 Literature Review

There are several popular control system such as Proportioned Integral Derivative (PID) feedback control, Scheduled-based setpoint control systems that has been on the market for energy preservation. And after google published its machine learning application on the Data center power usage effectiveness (PUE) optimization, more novel techniques has been proposed, Li et al (2018), proposed a cooling control algorithm with offline

trace that utilizes reinforcement learning to optimize the efficiency. Related studies such as Heating, Ventilation, and Air Conditioning (HVAC) Control using Deep Reinforcement learning proposed by Wei et al(2017), Zhang et al(2018), and Building Control proposed by Jia et al (2018) are all application of deep learning algorithm to preserve the energy use. However, the biggest challenge is that all the papers mentioned above are all implemented on a large scale Data center and none of the algorithms were tested on the *real environment*. Even though our environment is relatively simple, we still could not obtain the the real-time reward from the agent (AC, in our case) from historical records.

## 3 Project Goal

With the development of IoT industry, cloud computing is not the only the only way to compute large amount of data, we wish to develop a schema that can be computed locally, or edge computing. Such that the risk of internet and connection failure can be avoid and real-time decision making can be available such as self-driving car. Our goal is to optimize the energy use like in the Google paper, or minimizes 'PUE', which is defined as:

$$PUE = \frac{\text{Total power usage}}{\text{IT power usage}} \approx 1 + \frac{\text{cooling power usage}}{\text{IT power usage}}$$

Cooling is inevitable for all electronic devices. Overheating may cause system failure, malfunction, or worst case — shutdown. We propose to use a simple LSTM model to predict the ambient temperature of the server room so that the AC system can be turn-on if the prediction reaches predefined threshold before it actually reaches it, this could potentially save some cooling power usage and hence lower the PUE. The proposed method is aimed to be simple and fast so that even the local machine can handle the amount of computing for real-time decision making.

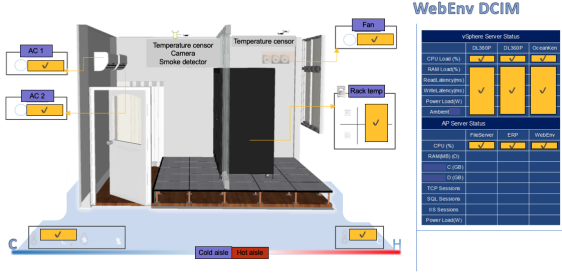


Figure 1: Server room layout<sup>1</sup>

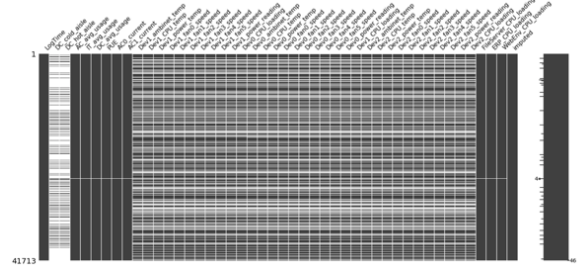


Figure 2: Data missingness<sup>2</sup>

## 4 Data Source



Our data is provided by the courtesy of [WebEnv IoT](#), a company founded in 2000 in Tainan, Taiwan that first started as a manufacturer to design and produce network equipment for server room environmental monitoring, and over the last decade, it dedicated to IT systems integration services and network devices solutions under the information security environment.

The data is collected in each of the controller installed in the server room (depending on the customer demand and budgets different model and spec will be used) and reported to the server every 5 minutes or 1 minute as long as the connection is not interrupted or whenever a special event is trigger such as authorized personnel entered the server room, abnormal temperature detected in the server room. However, the war data is only available for one-month long (currently set to be 6 months to obtain more data for the future work of this case study) and the previous data were integrated and only summary statistics are preserved.

The data is a time series that consists of the following key variables:

Variable Description	Data Type
Device Model	categorical
timestamp	datetime
Device ambient temperature	float
Device CPU loading	float
Device power temperature	float
AC current #1-2	float
fan status #1-10	float

Total of 45 variables were retrieved in this study.

<sup>1</sup>The figure shows the real layout of the server room where our data is retrieved, yellow boxes with the check indicate the information that were obtained.

## 5 Data Preprocess

The data were distributed into 3 rack units and 12 devices, 45 ports in total. Since the record is not recorded on the minute and the record time is affected by the internet stability and connection time for each device. To integrate the data, we first rounded the timestamp to its closest minute and the use it as key to merge all the data into one matrix. By doing so, there will be an entry for every minute over the span of one month (the current data storage capacity setup).

We use imputation to impute the missing values for this case. Since the record was rounded to the closest minute, we systematically fill forward one entry and backward for one entry. Then for each columns, we select the longest non-missing sequence available and generate missing mask and test various imputation technique and select the method that produces the minimal MSE. By using this method, the imputation is less subjective and robust. The algorithm should be able to select the best algorithm to capture the time-series structure for each variable.

### Algorithm 1 Column imputation

**Data:** Missing data columns

**Result:** complete column with imputed values  
initialization

**if** Data has missing values **then**

    forward fill 1 timestamp

    backward fill 1 timestamp

    select longest non-missing sequence

    create 10 % random missing to select imputation method using MSE

    impute column with the best method from previous step

**end**

**return**

<sup>2</sup>The figure shows the pattern of how the data are miss-

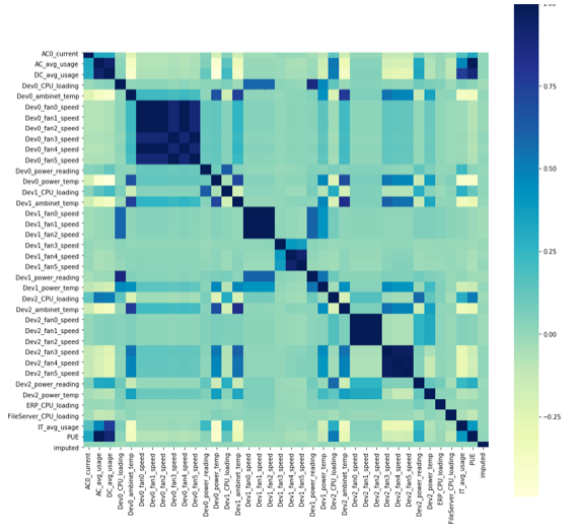


Figure 3: Data correlation

## 6 Exploratory Data Analysis

From the exploratory data analysis, we confirmed our initial assumption from both correlation plot and the heatmap presentation of the daily data that the air conditioning power usage is highly correlated to the ambient temperature reading of the rack units. Figures below shows the AC power usage (green/blue palette) and the ambient temperature reading from one of the rack unit (red/brown palette), the x-axis of the heatmap is 1440 minutes per day and y-axis represents the days over the span of our data set. Some pattern can be observed from the heatmaps, first the AC current heatmap shows that between 0-8 am the power usage is fairly low and consequently the ambient temperature is higher(brighter color) than the other time of the day, and around 8am-2pm the pattern is reversed, temperature is higher while the AC power usage is higher.

From the correlation plot, we can observe that there are three blocks with the dark blue color along the diagonal, which three devices are highly correlated within themselves, especially the fan speed setting within each rack unit are almost identical.

ing, since the data was not recroded at the same time, some preprocessing need to be performed to obtain the most information, the figure is the data before the imputation algorithm is applied.

<sup>3</sup>the heatmap is smoothed for visual effect

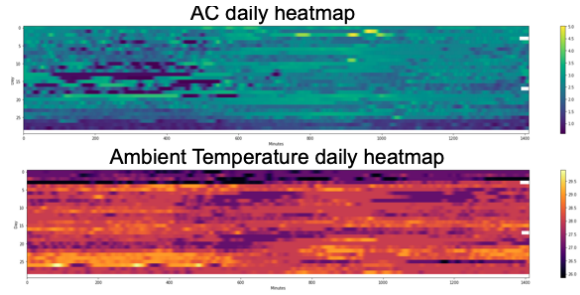


Figure 4: AC and Ambient Temperature heatmap<sup>3</sup>

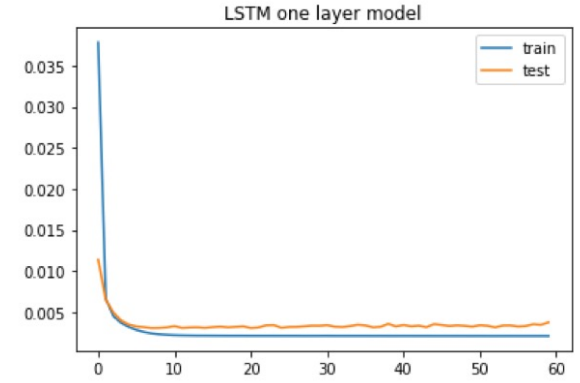



Figure 5: Loss-epochs for proposed model<sup>4</sup>

## 7 Methods

### 7.1 Limitation and constraints

In order for the algorithm to be executable within the local machine(as in the project goal section, the local computing power is restricted to a raspe-

berry pi chip ) Reinforcement learning will be better suitable for IoT platform to handle the amount of computation needed.

### 7.2 proposed algorithm

As proposed, we used a simple one layer LSTM for fast computing time and the algorithm is modularized for the goal that could be deploy locally for a real-time prediction and update. The model is simple and converge really fast after couple iteration. The proposed algorithm achieved about 0.005 mae which corresponding to 0.849% of the mean value for the response. However, our goal is to reduce the PUE usage, at the current phase we are able and successfully predict the temperature using this relatively simple model, will need to deploy our proposed model in the devices and confirm if this will indeed reduce the PUE.

## 8 Conclusion

The current model is developed for the goal that it can be performed and installed locally in the case where internet connection is disabled or not available. The actual performance of whether the proposed approach using prediction to trigger AC switch ahead of time will still need to be test after deployment. Some future work including deploying the model and potentially use EnergyPlus or Building Virtual to create a simulation environment to make *reinforcement learning* possible and can be used as model performed in the IoT platform when the internet connection is stable.

## 9 Reference

- [1] Li et al. "Transforming Cooling Optimization for Green Data Center via Deep Reinforcement Learning." 2018.
- [2] Wei et al. "Deep Reinforcement Learning for Building HVAC Control." 2018.
- [3] Jia et al. "Advanced Building Control via Deep Reinforcement Learning." 10th International Conference on Applied Energy (ICAE2018), 22-25 August 2018, Hong Kong, China.
- [4] Jim Gao. "Machine Learning Applications for Data Center Optimization." Google, 2014.