Journal of Korea Multimedia Society Vol. 23, No. 2, February 2020(pp. 155-165)
https://doi.org/10.9717/kmms.2020.23.2.155

# Smart Thermostat based on Machine Learning and Rule Engine

Tran Quoc Bao Huy†, Sun-Tae Chung††

## ABSTRACT

In this paper, we propose a smart thermostat temperature set-point control method based on machine learning and rule engine, which controls thermostat's temperature set-point so that it can achieve energy savings as much as possible without sacrifice of occupants' comfort while users' preference usage pattern is respected. First, the proposed method periodically mines data about how user likes for heating (winter)/cooling (summer) his or her home by learning his or her usage pattern of setting temperature set-point of the thermostat during the past several weeks. Then, from this learning, the proposed method establishes a weekly schedule about temperature setting. Next, by referring to thermal comfort chart by ASHRAE, it makes rules about how to adjust temperature set-points as much as low (winter) or high (summer) while the newly adjusted temperature set-point satisfies thermal comfort zone for predicted humidity. In order to make rules work on time or events, we adopt rule engine so that it can achieve energy savings properly without sacrifice of occupants' comfort. Through experiments, it is shown that the proposed smart thermostat temperature set-point control method can achieve better energy savings while keeping human comfort compared to other conventional thermostat.

**Key words:** Thermostat, Machine Learning, Rule Engine, Data Mining, LSTM, K-means Clustering

## 1. INTRODUCTION

The energy savings in home automation are primarily gained by changing the thermostat's set-point temperature according to situations; for example, while occupants are away from the home or when they are asleep. A thermostat is an electronic device which senses the temperature of a physical system like house or building, compares it with the set-point temperature and performs actions according to the comparison results so that the system's temperature is maintained near a desired set-point temperature [1,2]. Appropriate scheduling of temperature set-points is important for human comfort as well as for energy savings.

According to ways how to configure temperature set-point, thermostats can be categorized as follows: manual, programmable, responsive, and smart thermostat. Programmable thermostats support programming set-point schedule [3]. It is supposed that programmable thermostats can save more energy than manually setting thermostats, but is reported [4] that since the static thermostat schedule may not reflect dynamic situation of indoor occupants, they may be less effective in energy savings unless users can configure set-point schedule appropriately and on proper times. A responsive thermostat deals with dynamic situations of changing occupants by adopting occupancy sensors [5]. Current technologies about detection of the number of occupants are not accurate enough for successful commercialization. Furthermore, responsive thermostat cannot provide immediate comfortable indoor environments since indoor

※ Corresponding Author : Sun-Tae Chung, Address: (06978) 369 Sangdo-ro 680, Dongjak-gu, Seoul, Korea, TEL : +82-2-820-0638, E-mail : cst@ssu.ac.kr
Receipt date : Jan. 13, 2020, Approval date : Jan. 20, 2020
† Dept. of Information and Telecommunication Engineering, Graduate School, Soongsil niversity
(E-mail : huy.tsusak@gmail.com)
†† Dept. of Smart System Software, Soongsil University

temperature takes time to reach the set-point temperature; it means that responsive thermostat alone cannot achieve proper predictive temperature control. Some problems of programmable and responsive thermostats can be handled by adopting rule-based scheduling, where dynamic situation of changing indoor occupants, varying seasonal heating/cooling appropriation level and predictive control can be integrated on the elaborate rules [6]. However, simple rules in [6] usually specify the static conditions so that they may not handle dynamically changing situations.

The energy savings are realized simply by generating a setback schedule that suits their daily routines. Further energy savings can be gained through (1) Auto-Away; or reducing heating/cooling and air conditioning (HVAC) usage during extended absences from the home (e.g. while on vacation), and (2) making very small changes in the set-point temperature (e.g. by changing the set-point by 1℃), perhaps in response to learnings from past usage pattern. However, if an energy saving method may cause occupants' discomfort, then it is not preferable even though it can save lots of energy.

Recently, smart thermostats [5], which automatically adjusts heating and cooling temperature settings at homes/offices so as to achieve both occupants' comfort and energy savings, have been deploying on the market since Google Nest [7]. Google Nest generates auto weakly schedule of set-point temperature values by applying machine learning algorithms to collection of past user's setting set-point data. In this case, the learned weekly set-point schedule reflects the user's preference pattern about how he/she wants to keep his/her house's temperature. Thus, the weekly schedule can be properly utilized for management of automatic thermostat setting. However, since most users have very little knowledge of the HVAC systems and the thermal inertia of the house and office they live and work in [8], consequently they tend to overcompensate when increasing or decreasing the set-points in hoping to accelerate the heating/cooling process according to two thermostat use surveys [9,10]. This results in suboptimal control of the indoor temperature from perspective of energy usage and occupants' comfort. In addition, many occupants tend not to fine-tune their thermostat settings actively to exploit the seasonal variations in their clothing levels. Rather, they set their thermostats such that they feel thermally neutral in both heating and cooling seasons with minimum number of control actions—with little intention to save energy. Thus, with respect to energy savings, the schedule has some rooms for further adjustments while keeping human comfort by considering seasonal comfort zone margin and changing occupants. Then, a rule-based adjustment can handle such dynamic situations more properly.

In this paper, we propose a machine learning and rule engine based smart thermostat set-point control method, which controls thermostat's temperature set-point intelligently by machine learning and rule engine so that it can achieve energy savings without sacrifice of occupants' comfort. The proposed method first periodically learns user's preference about thermostat's set-point temperature value from user's past set-point setting pattern data, and updates a weekly schedule for the next week. Fine tuning rules, which are elaborately designed by referring to thermal comfort zone guideline due to ASHRAE, adjust the temperature value of the constructed schedule table, and determine the thermostat's final set-point temperature value more properly.

We implement the proposed method on a single board computer [11] and on a commercial wallpad [12], and test how it works. Through extensive experiments on the single board computer and the commercial wallpad, it is shown that the proposed thermostat temperature set-point control method can achieve energy savings much while guaranteeing occupants' comfort.

The organization of the paper is as follows. Section 2 briefly introduces the background knowledges for the paper, and the proposed smart thermostat set-point control method is described in Section 3. Experimental results are explained in Section 4 and Finally Section 5 concludes.

## 2. BACKGROUNDS

### 2.1 Humidity

Relative humidity represents a percentage of water vapor in the air that changes when the air temperature changes[13]. Higher percentage of relative humidity means that the air-water mixture is more humid. As air temperature increases, air can hold more water molecules, and its relative humidity decreases. When temperatures drop, relative humidity increases. Temperature therefore directly relates to relative humidity. In hot summer weather, a rise in relative humidity increases the apparent temperature to humans by hindering the evaporation of perspiration from the skin. For example, according to the Heat Index, a relative humidity of 75% at air temperature of 80.0°F (26.7°C) would feel like 83.6°F±1.3°F (28.7°C±0.7°C) [13]. Thus, room temperature and relative humidity directly affects human comfort.

### 2.2 Thermal comfort (Human comfort)

Thermal comfort is the condition of mind that expresses satisfaction with the thermal environment and comfort level is subject to how each person feels [14]. However, for weather conditions on each season, there are rough comfort level range for temperature and humidity.

There is no direct relationship between temperature and humidity. However, humidity can change a person's comfort at the same temperature as humans are sensitive to the water vapor content in the atmosphere because the human body uses evaporative cooling as the primary mechanism to regulate temperature. Thus, a rise in relative humidity increases the apparent temperature to humans (and other animals) by hindering the evaporation of perspiration from the skin. ASHRAE Standard 55-2010 uses the PMV (Predicted Mean Vote) model to set the requirements for indoor thermal conditions, which requires that at least 80% of the occupants be satisfied [15]. The results are displayed on a temperature-relative humidity chart (Fig. 1) which indicates the ranges of temperature and relative humidity for human comfort.

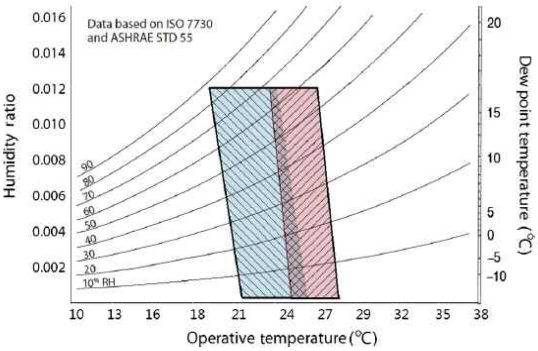



Fig. 1. ASHRAE Comfort Chart ; acceptable range of operative temperature and humidity for the thermal comfort zones of ASHRAE [15].

### 2.3 Knowledge-base system and rule engine

Knowledges are represented as true/partially true facts and rules, which consist of condition part and action part such as 'If (facts are satisfactory), Then (do proper actions)'. Rule engine checks the existing or new coming facts and if they are satisfactory, then do proper actions, which may produce new true/partially true facts. Newly produced facts fire rule engine's operation again [16].

In this paper, rules are constructed to adjust set-point temperature of the reference schedule table appropriately depending on dynamic situations and considering human comfort level. Drools [17], a popular open source rule engine, is utilized for implementation of rules constructed for the proposed smart thermostat.

## 3. PROPOSED SMART THERMOSTAT

### 3.1 Overview

The proposed smart thermostat operates in three modes: Manual mode, Auto-Away mode, and Auto-Operation mode. Manual mode keeps the manual setting of users until the next setting. Auto-away mode operates during extended absences from the home and reduces heating, cooling and air conditioning (HVAC) usage according to preassigned setting values. Auto-Operation mode, whose algorithm is the main subject of this paper, operates based on a learned weekly schedule with adjustment by rules. A weekly temperature set-point schedule is periodically learned from the past history of user's thermostat setting pattern. The learned reference weekly temperature set-point schedule reflects the user's preference pattern about how he/she wants to keep his/her house's temperature. As stated in Section 1, Introduction, by the reference weekly table, user's oversetting or undersetting cannot be compensated on thermostat's temperature set-point control properly. Also, such a smart thermostat cannot handle dynamically changing situations. To compensate the learned set-point schedule's inflexibility, elaborately constructed rules check conditions of modes, status, comfort zone level, and etc., and appropriately adjust the weekly schedule table's set-point values to produce the final thermostat's temperature set-point values.

Fig. 2 illustrates the overall working architecture of the proposed smart thermostat's operation. Smart thermostat has several modules. Among them. main module for smart control are Data acquisition and Saving module, Weekly scheduling module, Humidity prediction module, and operation control module. Operation control  module works based on rule engine. If mode setting or status changes or events happen, then it executes the appropriate controls. In Auto-Operation mode, operation control module adjusts set-point temperature

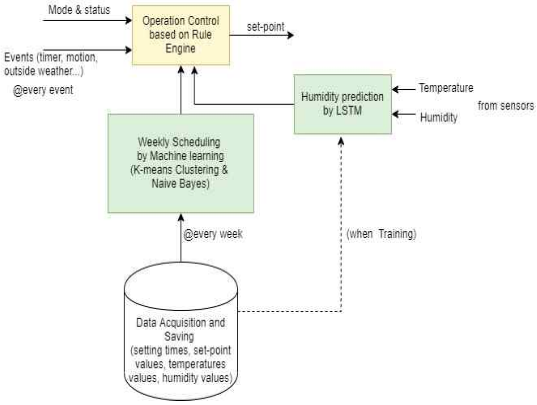



Fig. 2. Working architecture of the proposed smart thermostat's Operation.

value of weekly schedule table according to rule design. Proper execution of rules is done by operation of rule engine. The rules designed in this paper need humidity values for checking conditions. Since it takes a while to let the room temperature follow the setting temperature value, the designed rules use predicted humidity value in the future (5 minutes layer in the experiments in this paper). Thus, the proposed smart system needs humidity prediction module where prediction is processed based on LSTM[18]. Weekly scheduling is achieved by applying Naïve Bayes algorithm [19] into collection of past user's usage pattern data.

### 3.2 Scheduling Weekly Temperature Set-point Table for Auto-Operation mode

At the end of each week, the proposed schedule algorithm automatically learns the user's setting pattern of the thermostat set-point temperatures from the past collected data and will produce a weekly schedule for the thermostat setting. The weekly set-point schedule designates when and what set-point value should be set.

Thermostats like Nest[6] have temperature sensor, humidity sensor, and human detection sensor inside together with WiFi connection to outside, and this paper assumes that the considered thermostat has also same sensors and WiFi connection.

Thus, such thermostats can collect temperature data, humidity data, calendar time, setting time, setting temperature set-point value, human detection events, and outside information such as weather information. In this paper, we assume the thermostat can collect similarly.

The proposed schedule algorithm first learns the set-point setting times from the thermostat's past collected data by applying K-means clustering algorithm [20] for the past set-point setting time data. $K$ in the $K$-means clustering algorithm is determined by elbow point method [21] explained in Section 2, Backgrounds. The proposed algorithm learns clustering on all weekday data, and all weekend data, respectively. Centroid value (time) of each cluster is rounded up to 15-minute interval like 8:00, 8:15, 8:30, etc.

As for prediction of set-point value on each learned set-point time interval, the schedule algorithm developed in the proposed smart thermostat utilizes a Naïve Bayes algorithm [19] as follows.

For each set-point setting time interval group, set-point data in the group are clustered into 3 clusters, which is determined by applying $K$-means algorithm, and each cluster is represented by its center value: $ST_i^1, ST_i^2, ST_i^3$ where $i$ represents $i$ th interval. For the temperature data belonging to each cluster $ST_i^k (k = 1,2,3)$ in the interval $i$, we also cluster them into 3 groups: $T_i^1, T_i^2, T_i^3$. We calculate the priori probability as follows:

$$P(ST_i^k) \cong \frac{|set(ST_i^k)|}{|set\ of\ the\ whole\ set\ point\ data\ in\ the\ interval\ i|} \quad (2)$$

with $k$=1,2,3 where $set(ST_i^k)$ is the set of set-point value data clustered as $ST_i^k$ in the interval $i$ and $|A|$ means the number of the elements of the set $A$. Then, the posterior probability $P(ST^k|T^\alpha)$ is calculated by Bayes theorem as:

$$P(ST_i^k|T_i^\alpha) = P(T_i^\alpha|ST_i^k)P(ST_i^k)/P(T_i^\alpha) \quad (3)$$

Since $\sum_{k=1}^{3} P(ST_i^k|T_i^\alpha) = 1$

$$P(ST_i^k|T_i^\alpha) = P(T_i^\alpha|ST_i^k)P(ST_i^k)/(\textstyle\sum_{k=1}^{3} P(T_i^\alpha|ST_i^k)P(ST_i^k)) \quad (4)$$

here, we take an approximate calculation of:

$$P(T_i^\alpha|ST_i^k)\ (\alpha = 1,2,3) \quad (5)$$

as

$$P(T_i^\alpha|ST_i^k) \cong \frac{|set(T_{i,k}^\alpha)|}{|set(ST_i^k)|} \quad (6)$$

where $set(T_{i,k}^\alpha)$ means the set of temperature data belonging to the cluster $T_k^\alpha (\alpha = 1,2,3)$ among $set(ST_i^k)$. Then, we determine the new set-point for each interval $i$:

$$ST_i = \sum_{\alpha=1}^{3}(\sum_{k=1}^{3} ST_i^k * P(ST_i^K | T_i)) * P(T_i^\alpha) \quad (7)$$

### 3.3 Proposed Rules for Operation Control Module

The weekly schedule obtained from the proposed schedule algorithm needs to be adjusted for further energy savings while keeping comfort by reflecting the dynamic situations.

The proposed thermostat method applies a new set-point, which is adjusted from the weekly schedule table by rules, when events happen. Events include arrived new schedule times in the schedule table, occupant detection, users' mode setting, a server system notifies a new setting, outside urgent weather situation, and etc. Under Away mode or Auto-schedule mode, the thermostat's rule engine starts to work, checks conditions and executes actions associated with the satisfactory conditions whenever an event happens. Those actions adjust the value from schedule table by reflecting dynamic situations such as season, occupant detection, outside weather condition, comfort table, and etc.

The proposed rules, which we call comfort rules, consider Mode and Status setting. When status is either cooling or heating, then a set-point value from the schedule table will be adjusted by consideration of comfort zone in the temperature-relative humidity chart (Fig. 1.). So, set-point temperature can be further increased on cooling (Summer) status or decreased on heating (Winter) status as long as the newly adjusted set-point temperature and relative humidity together match comfort zone.

Some of the comport rules in the proposed method are described in the below Table 1.

### 3.4 Prediction of Relative Humidity

Room temperature are affected by indoor energy status controlled by thermostat setting and outside weather. Room humidity are affected by room temperature, outdoor weather including humidity. We consider relative humidity as a function of past temperatures and relative humidity values as

$$\hat{h}_t = f(h_{t-1}, \cdots, h_0, T_{t-1}, \cdots, T_0) \quad (1)$$

where $h_i$ means relative humidity at time $t$, and $T_t$ means temperature at time $t$.

From our simulations before, it was found that prediction of the humidity at time t, $\hat{h}_t$ for (1) based on a linear ARMA model does not perform well. Instead, we predict the relative humidity based on LSTM [22].

We construct a neural network architecture to implement prediction task. The constructed RNN consists of two layers of 10 LSTM cells where 2nd layer of 10 LSTM cells is stacked on the 1st layer of 10 LSTM cells. The outputs of each LSTM cell of the 2nd layer are linearly combined to generate the final output. The architecture of the constructed network is shown in Fig. 3. For experiments in this paper, we choose 5 minutes interval since it does not take long for temperature setting to be effective in Testbed box.

The input and output of our neural network:
- Input: 10 pairs of temperature and humidity during past 50 minutes
- Output: Relative humid value at the 5 minutes later.

## 4. EXPERIMENTS

### 4.1 Experiment Environments

Through experiments, we want to test that the

Table 1. Comport Rules Design based on Thermal Comfort Chart of Fig. 1 (Winter Case)

| No. | Rules | |
|---|---|---|
| | Condition (When or If) | Action (Then) |
| 1 | When the mode setting happens, | Then, modify the Mode as selected |
| 2 | When no occupant detection event happens. | Then, switch the thermostat Mode into Away (no occupant) mode or user's setting mode (Auto-stay or manual mode) (occupation). |
| 3 | When the otherwise events happen and Mode is Auto-schedule and it is Summer (Winter) time. | Then, set the Status as "Cooling"("Heating") and set the 'Thermostat's set-point temperature' by 'Set-point temperature' reading from the weekly schedule table. |
| 4 | When it is "Heating" status and 'Scheduled Thermostat set-point temperature > 19℃' and ' 50% <= 'Predicted relative humidity of 5 minutes later < 88%' . | Then, adjust the Thermostat's set-point temperature as 19℃. |
| 5 | When it is "Heating" status and 'Scheduled Thermostat set-point temperature > 20℃' and '22% <= predicted relative humidity of 5 minutes later < 50% '. | Then, adjust the Thermostat's set-point temperature as 20℃. |
| 6 | When it is "Heating" status and 'Scheduled Thermostat set-point temperature > 21℃' and 'predicted relative humidity of 5 minutes later < 20%'. | Then, adjust the Thermostat's set-point temperature as 21℃. |

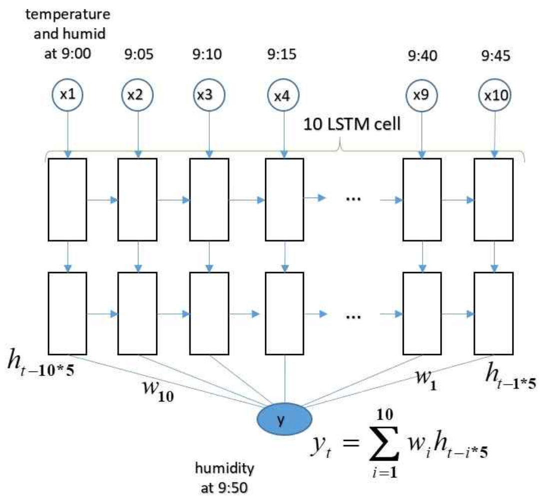



Fig. 3. RNN model for predicting humidity from past humidity and temperature.

proposed rule-based thermostat temperature set-point control method utilizing the thermal comfort chart with humidity prediction can save more energy without sacrifice of human comfort than the method based on user usage pattern learning like google smart thermostat Nest [7]. For the experiment purpose, we set up a testbed in our lab (Fig. 4), which consists of two sets.

Each set was built with the following devices and elements:

- 60W electric cooling fan ; for circulating outside air into a TestBox
- Electric heating pad of 14cm x 5cm ; for supplying heating energy into a textbox
- Raspberry Pi 3 with Linux : for collecting temperature and humidity sensor data and sending data to a wallpad or a single board computer and for controlling relay and cooling fan
- DHT22 ; Temperature and humidity sensor
- Relay with 4 channels; for power control of heating pads
- A TestBox ; for emulating an indoor room
- A wallpad and a single board computer (Odroid-C2 [11]) with Android 7.0
- A mining server; Intel(R) Core(TM) i5-8400 CPU(2.80GHz), 16GB, Windows 10.

On Wallpad[12] and a single board computer [11], Operation control module including a rule engine(Drools [17]) is running. Intelligent modules such as Weekly Scheduling(which processes K-means clustering and naïve-Bayes algorithm) and Humidity prediction module need more computational power for fast processing, and thus run on the mining server.

We utilize two TestBoxes as emulating 2 rooms. Each room has electric cooler, electric heating pad and a DHT22 sensor. In cooling mode, if the temperature of inside TestBox is higher than set-point temperature, then Raspberry PI system will turn on the electric cooling fan until the TestBox temperature reaches the setting point. The raspberry

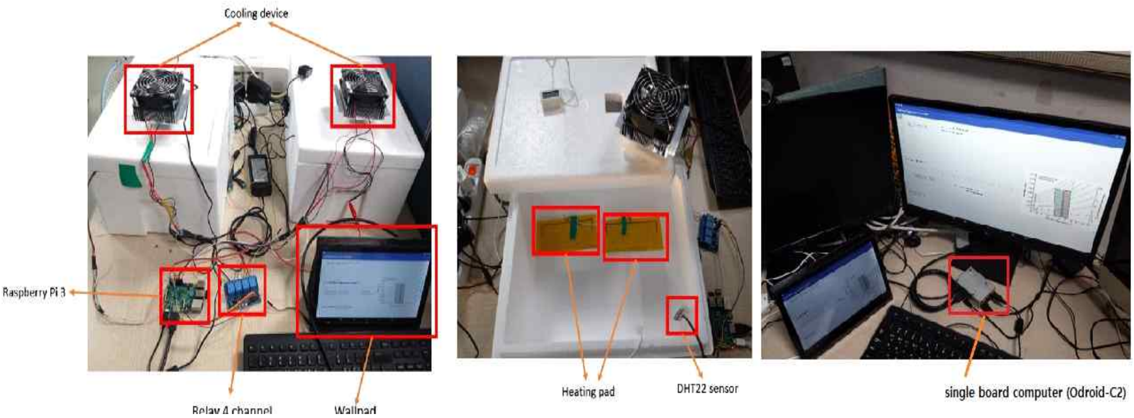



Fig. 4. Experiment testbed and components of the testbed.

will also control a relay for heating pad similarly in heating mode. In order to compare between set-point control of weekly schedule and that of the proposed comfort rule, we set up operations of two test boxes as follows:

TestBox1 (Comfort Rule-based Control): temperature and relative humidity value from the sensor are collected (every 5 minutes) by Raspberry and send to the wallpad. After that, the wallpad send them to the server and receive the RH prediction value. Then, based on comfort rule, a new temperature set-point is calculated and it is sent to Raspberry Pi system. Finally, the Raspberry PI system controls a cooling fan or a relay for heating pad device based on that setting-point.

TestBox2 (Weekly Schedule-based Control ): The Raspberry PI system controls a cooling fan or a relay for heating pad based on the setting point value in weekly schedule.

### 4.2 Dataset

After we finished experiment setup as in Fig. 4, We had collected data for training from, from the testbed for a month (Oct. 30th~December 1st). For temperature and humidity, we collected them from DHT22 sensor at every 5 minutes and we set temperature set-point between 22~27℃, 3~5 times in a day from morning till night. Fig. 5 shows UI for our application for controlling the temperature set-point.

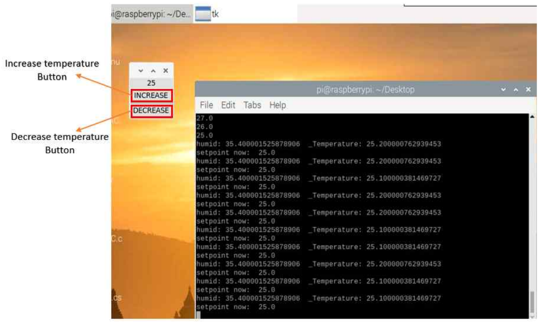



Fig. 5. Application for controlling temperature set-point.

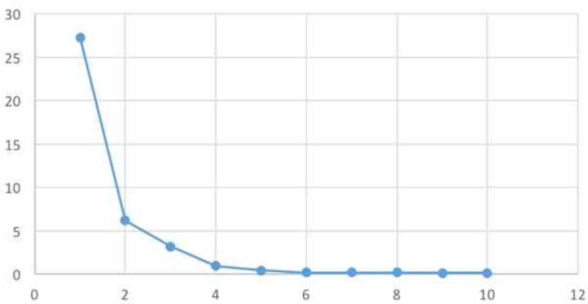
### 4.3 Scheduling weekly thermostat set-point Table

First, in order to learn setting times for weekday, and weekend (Saturday, Sunday) from experiment data set, we apply $K$-means clustering algorithm explained in Section 3. Secondly, in order to find out estimation of set-point temperature value for each clustered setting-time interval, we applied the Naïve Bayes algorithm [19] as described on Section 3.1. Eventually, we can generate Weekly Schedule Table as shown in Fig. 6(b).

### 4.4 Humidity Prediction

#### 4.4.1 Training

Before training, the collected data were divided into 3 parts: training set, validation set and testing set. During training process, we estimate accuracy of model on validation set. The model which produced best accuracy on validation set was chosen




(a) Sum of Squared Error for weekday data

(b) Weekly Schedule Set-point Table

Fig. 6. (a) Graph of Sum of Squared Error for weekday data (K=4 by 'elbow' method, (b) Weekly Schedule Set-point Table learned from User's usage pattern.

Table 2. Accuracy Evaluation for model construction

| Validation | | Test | |
|---|---|---|---|
| MSE | MAE | MSE | MAE |
| 0.115 | 0.032 | 0.114 | 0.030 |

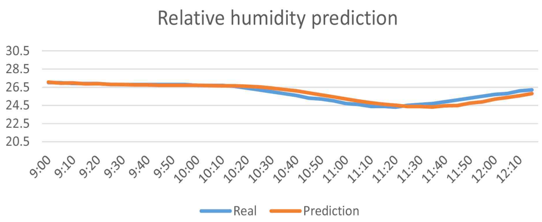



Fig. 7. Prediction results on some sample data in test dataset. X-axis shows 5 minutes time interval.

to run on the test set. Mean Absolute Error(MAE) and Mean Square Error (MSE) were adopted as metrics for evaluating the model. The result in Table 2 shows that the accuracy is not much different between validation set and test set.

Prediction results on some sample data from the test dataset in Fig. 7 shows superiority of the LSTM-based humidity prediction method developed in this paper.

#### 4.4.2 Humidity Prediction

X-axis in Fig. 8 has 5 minutes interval. Thus, Fig. 8 shows graphs of the predicted relative humidity (Real RH) and the real measured relative humidity (Real measured RH) at TestBox1 during 60 minutes. The 60 minutes'experiment results in Fig. 8 shows pretty good prediction accuracy; MAE of 1.016 (%) and and MSE of 2.778(%). Throughout many other extensive experiments, almost same accuracy of the LSTM-based humidity prediction proposed in this paper is observed.

### 4.5 Temperature Set-point Control

Experiment results in Fig. 9 illustrates comparison between two temperature control methods of Comfort Rule and Weekly Schedule in summer mode. From Fig. 9, one can know that the real measured temperature in TestBox1(where comfort rule is appled) is higher than that of TestBox2 (where weekly schedule is applied). It means that in the experiments, the cooling fan in TestBox2 was heavierly used than that in TestBox1, which implies that the proposed comfort rule-based temperature set-point control works superior to the weekly schedule-based one.

## 5. CONCLUSIONS

In this paper, we proposed a new rule-based thermostat temperature set-point control method, which utilizes rule engine in addition to machine

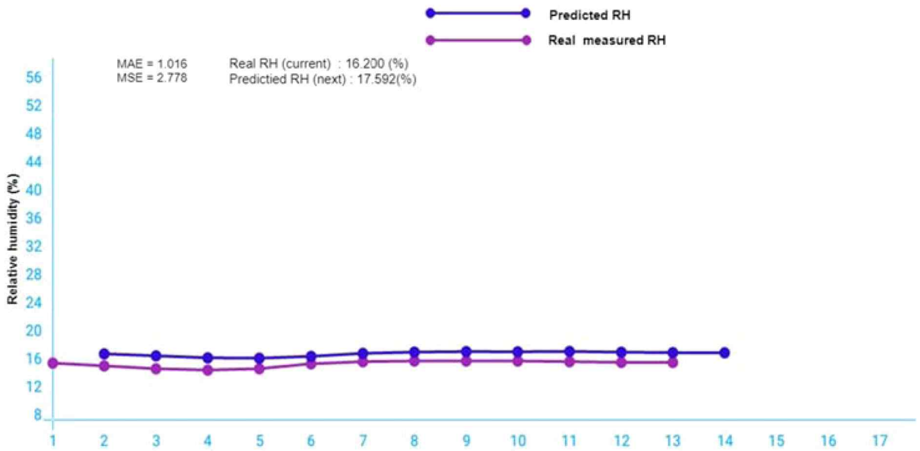



Fig. 8. Real-time Relative Humidity Prediction and Measured Relative Humidity (at TestBox1).

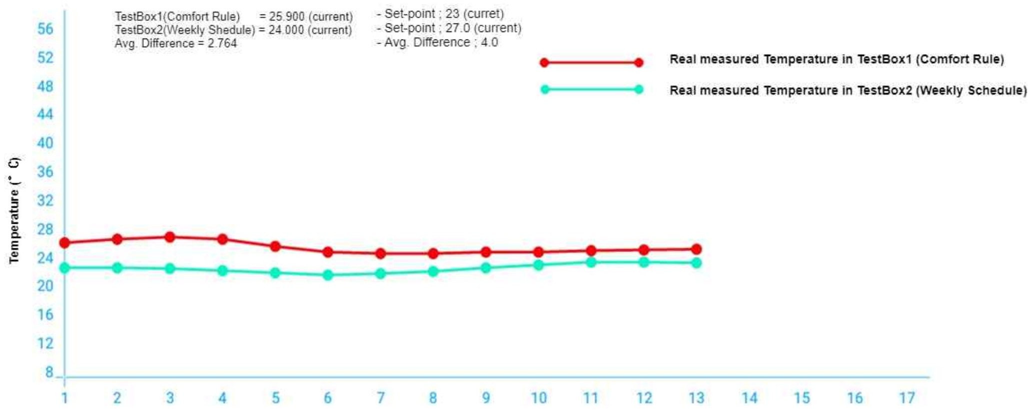



Fig. 9. Comparison of two temperature control methods (Comfort Rule (TestBox1) vs Weekly Schedule (TestBox2)) in summer mode.

learning techniques so that it can achieve energy savings without sacrifice of occupant comfort. The proposed smart thermostat set-point control method first periodically learns user's preference about thermostat's set-point temperature from user's past set-point setting pattern data, and updates a weekly schedule for the next week. The rules elaborately developed from considering many things including ASHRAE's recommended comfort chart adjust the temperature value of the constructed weekly schedule table appropriately for seasons and some dynamic situations, and determines the thermostat's set-point temperature value. Extensive experiments on a real wallpad or an embedded board system showed that the proposed comfort rule-based thermostat temperature control method can achieve energy savings better without sacrifice of human comfort than that of weekly schedule learning from user's preference usage pattern like google Nest thermostat [7].

## REFERENCES

[1] Thermostat, https://en.wikipedia.org/wiki/Thermostat (accessed January 11, 2020).

[2] J. W. Park, J. Ko, J. H. Park, M. H. Hong, Y. H. Lee, and J. Shim, "A Wireless Temperature Control System based on FPGA," *Journal of Korea Multimedia Society*, Vol. 16, No. 7, pp. 920-930, 2012.

[3] Programmable Thermostat, https://en.wikipedia.org/wiki/Programmable_thermostat (accessed January 11, 2020).

[4] Environmental Protection Agency, *Summary of Research Findings from the Programmable Thermostat Market,* Office of Headquarters, 2004.

[5] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, et al., "The Smart Thermostat: Using Occupancy Sensors to Save Energy in Homes," *Proceedings of the 8th Association for Computing Machinery Conference on Embedded Networked Sensor Systems,* pp. 211 - 224, 2010.

[6] A. Keshtkar and S. Arzanpour, "Design and Implementation of a Rule-based Learning Algorithm Using Zigbee Wireless Sensors for Energy Management," *Proceeding of IEEE 27th Canadian Conference on Electrical and Computer Engineering*, pp. 1-6, 2014.

[7] Google Nest Thermostat, https://nest.com/thermostat/meet-nest-thermostat (accessed January 11, 2020).

[8] S. Karjalainen, "Usability Guidelines for Room Temperature Controls," *Intelligent Buildings International*, Vol. 2, No. 2, pp. 85-97, 2010.

[9] S. Karjalainen, "Thermal Comfort and Use of Thermostats in Finnish Homes and Offices," *Building and Environment,* Vol. 44, No. 6, pp. 1237-1245, 2009.

[10] W. Kempton, D. Feuermann, and A.E. Mc Garity, "I Always Turn It on Super: User Decisions about When and How to Operate Room Air Conditioners," *Energy and Buildings*, Vo. 18, No. 3, pp. 177-191. 1992.

[11] Odroid-C2, https://www.hardkernel.com/main/products/prdt_info.php?g_code=G145457216438 (accessed January 11, 2020).

[12] Commax Wallpad, https://www.commax.com (accessed January 11, 2020).

[13] Humidity, https://en.wikipedia.org/wiki/Humidity (accessed January 11, 2020).

[14] L. Yang, J. Yan, and J.C. Lam, "Thermal Comfort and Building Energy Consumption Implications, a Review," *Applied Energy*, Vol. 115, No. 2, pp. 164-173, 2014.

[15] American Society of Heating Refrigerating and Air Conditioning Engineers(ASHRAE), *Thermal Environmental Conditions for Human Occupancy*, Standard 55-2010, 2010.

[16] Knowledge-based Systems, https://en.wikipedia.org/wiki/Knowledge-based_systems (accessed January 11, 2020).

[17] Drools, https://www.drools.org (accessed January 11, 2020).

[18] H.S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural Computation*, Vol. 9, No. 8, pp. 1735-1780, 1997.

[19] Naive Bayes for Machine Learning, https://machinelearningmastery.com/naive-bayes-for-machine-learning, (accessed January 11, 2020).

[20] *K*-means Clustering, https://en.wikipedia.org/wiki/K-means_clustering (accessed January 11, 2020).

[21] Using the Elbow Method to Determine the Optimal Number of Clusters for K-means Clustering, https://bl.ocks.org/rpgove/0060ff3b656618e9136b (accessed January 11, 2020).

[22] I. Park, J. Lee, H.S. Kim, C.H. Song, and H.K. Kim, "Relative Humidity Prediction Model with Missing Data Refinement Using a Long Short-term Memory Neural Network," *Proceeding of the 3rd World Congress on Civil, Structural, and Environmental Engineering*, pp. 141-1~141-2, 2018.

Quoc Bao Huy Tran

He received B.S. degree in Computer Science from University of Information Technology, Ho Chi Minh, Vietnam in 2017. He is currently a research assistant at Embedded Real-time Computing Laboratory, Soongsil University, Seoul, South Korea. His main areas of research interests include machine learning, deep learning, expert engine, and computer vision.

Sun-Tae Chung

He received B.E. degree from Seoul National University, and M.S. degree and Ph.D. degree in Electrical Eng. and Computer Science from the University of Michigan, Ann Arbor, USA, in 1986 and 1990, respectively. Since 1991, he had been with the School of Electronic Eng. at the Soongsil university, Seoul, Korea where he is now a full professor. Now, he has been with the Dept. of Smart Systems Software, at the Soongsil Univ. since 2015. His research interests include: machine learning, computer vision, and digital signage systems.