International Journal of Applied Engineering Research, ISSN 0973-4562 Vol. 10 No.90 (2015)

Dynamic Sensor Fusion and Control Framework of IoT-based Device using Classification Restricted Boltzmann Machine

Hyunsoo Lee[1], Heedong Hong[2*]
[1]School. of Industrial Engineering, Kumoh National Institute of Technology, Gumi, Korea
[2*]Dept. of Electronics Commerce, Semyung University, Jecheon, Korea
myclub114@semyung.ac.kr

Abstract - As more advanced sensor technologies and the related embedded architectures have been developing, Internet of Things (IoT) devices have been ubiquitous in many industrial fields as well as a housing life. In order to increase the objectives of an IoT device, the contemporary trends is that multiple sensors are embedded on one device. As these sensors shares similar functionalities, it is important that the sensor fusion technologies is required for their effective controls. While many sensor fusion techniques have been suggested, most of the methods tend to ignore the dynamic environmental changes surrounding an IoT device. In order to overcome this limitation, a more effective senor fusion framework is proposed. The suggested framework is based on Classification Restricted Boltzmann Machine (Class RBM) as its basic learning model. The Class RBM based sensor fusion framework contributes to the more accurate sensing and the integration of embedded multiple sensors. In order to verify the effectiveness of the proposed control, an IoT device with multiple sensors is implemented with an effective embedded architecture and is experimented for its controls and analyses.

Keywords - Sensor fusion; Embedded System; IoT; Dynamic control; Class RBM

I. INTRODUCTION

The development of embedded system architectures and the related sensor platforms has led the introduction of many IoT devices to various fields. These devices have been used as household devices for a convenient and safe life generally, in general. For instance, an intruder detection system uses several IoT devices for checking an intruder and for notifying it to the house owners or polices. Compared with previous information systems with similar functions, the system using IoT devices makes it possible to classify the given target more accurately and to control environments efficiently. Many application using IoT technologies have been expanded from home appliances to various industrial processes and social infrastructures.

The definition and architectures of “IoT” is summarized well in Gubbi, et al.’s research study [1]. According to their study, the IoT devices are implemented with the integration of sensors, knowledge and communications. The knowledge in many objects is measured, classified and controlled using its embedding sensors and those are transmitted to other platforms using communication protocols.

While the device depends on the embedding sensors heavily for drawing several decisions and controls, there are many issues in the sensor platform. Lee and Hong [2] introduced several issues in IoT devices using multiple sensors. In order to draw more accurate classifications and decisions, there has been a trend that the IoT devices use multiple sensors more. Even though some of these sensors’ detection functionalities are coincided, the development of efficient sensors with the low energy consumption ability makes the usages of multiple sensors in one platform. These trends are named as the sensor fusion. The sensor fusion is defined as the software techniques for integrating several sensors with special purposes by Crowley [3]. While there are many research studies for the efficient sensor fusion, the general methodology for selecting the more suitable sensors in a special application has been studied less.

This paper focuses on more effective methodology handling sensor fusion. When the assumption that an IoT device’s multiple sensors share similar functionalities is supposed, the effective sensor function control method and its learning mechanism are suggested. The provided mechanism is based on a deep learning method – Class RBM. The Class RBM is an evolved machine from RBM for classifying targets. While classical RBMs are used for identifying hidden layers, the Class RBM can be used for solving general pattern recognition problems. This paper uses a Class RBM based approach as a basic learning model for resolving sensor fusion issues.

The general backgrounds and the literature reviews are provided in the following section. The embedded architecture of an IoT device with multiple sensors is provided in Section 3 and its control framework using Class RBM and the learning mechanism are explained in Section 4, respectively. Then, the implementations and experimental analyses are provided in Section 5 for verifying the effectives of the proposed framework.

II. BACKGROUND AND LITERATURE REVIEWS

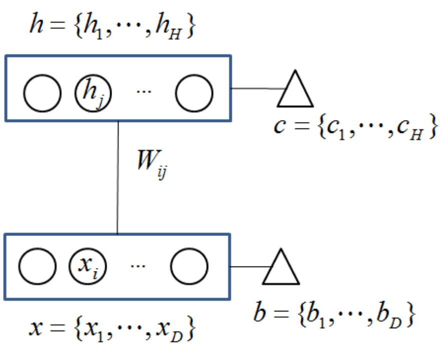
The concepts and the brief definitions of “IoT” and the sensor fusion are provided in the previous section. The most representing reason of the IoT development is explained with the development of sensor technologies and more accessible communication protocols. Lee and Banerjee [4] develops an IoT-based household vacuum cleaning robot system. The device is connected with a home ecosystem. In order to detect dusts and dirty regions, several camera sensors and dust sensors are embedded on the suggested system. Yi and Cho [5] suggested a moving platform with a sensor fusion for deciding more accurate six degrees of freedom for its ego-motion. The device has two cameras (a mono camera and a stereo camera), a laser scanner, a GPS unit and an embedded unit. While these installed sensors shares common functionalities such as viewing and detecting, the final decision is generated using computer vision techniques. It means that the usage and controls of those sensors are predefined and those are unchanged despite of the changes of environmental surroundings.

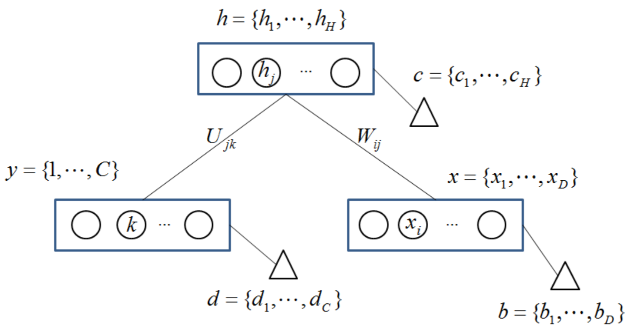
Most of the related research studies use embedded sensors with the predefined strategies and the fixed priorities. This tendency may give an inefficient control of an IoT device and may draw wrong decisions. Furthermore, the dynamic change of the environment surrounding the device may occur more serious results in its controls. This paper dissolve these issues and suggest an effective control sensor fusion framework.

Lee and Hong [2] suggested the related detailed issues and considerations for the sensor fusion. Table 1 is expanded from their study with the more extended issues.

Table 1. Detailed sensor fusion issues and considerations

Type	Considerations
Functional issues	- Number of sensing targets and objective functions - Time periods for executions - Synchronous / asynchronous execution - Embedded sensor fusion techniques - Location of function module (e.g. Server side module Vs. Device side module)
Physical issues	- Power consumption mechanism and efficiency - Database connection - Circuit types and Integration types - Distributed modules and environments
Communication issues	- Types of communication protocols - Communication periods
Structural issues	- Reconfiguration modules Vs. Fixed modules
Dynamic adaptation issues	- Dynamic adaption Vs. predefined strategies - Types of adaption methodologies





(a) RBM (b) Class RBM [6]

Figure 1. RBM and Class RBM.

While several physical, communication and structural issues shown in Table 1 have been studied comparatively frequently in embedded systems including IoT devices, the dynamic adaptation issues have been studied comparatively less. This paper focuses on the more effective sensor fusion control handling dynamic adaptation issues using a deep learning methodology – Class RBM. As mentioned in the previous section, Class RBM is based on RBM. While several usual back propagation based neural networks are trained for minimizing training errors between the estimated output and the observed output, RBM is trained for minimizing the lowest Boltzmann energy level. In order to obtain the lowest energy level, several RBM parameters are learned.

As shown in Figure 1 (a), a RBM has a parameter set consisting of W_{ij} , b and c. Then, the Boltzmann energy in the RBM is represented with (1).

$$E(x,h) = -h^T Wx - b^T x - c^T h \qquad (1)$$

Even though a RBM is useful in identifying the characteristics among the given x set using the identified hidden layers and nodes, the machine is restricted in the direct classification method. This fact makes an usual RBM as a preprocessed machine for many classification methods.

In order to overcome the limitation, Hinton, et al. [7] introduced Class RBM. As shown in Figure 1(b), Class RBM has an additional layer - the output layer for classifying given patterns. As the layer is linked with the hidden layer and it has a bias set, the Boltzmann energy is calculated using (2).

$$E(x,y,h) = -h^T Wx - b^T x - c^T h - d^T e_y - h^T U e_y \quad (2)$$

where, $e_y = (1_{i=y})_{i=1}^{C}$

Then, the conditional probabilities for lowering the Boltzmann energy are calculated and the parameters are tuned up using these probabilities. Choo and Lee [6] applies the Class RBM method for monitoring human postures such as walking, sitting, standing and other postures. This paper uses the class RBM based method for control multiple sensors against dynamic environmental changes surrounding an IoT device. The detailed mechanisms and implementations are explained in the following sections.

III. IOT EMBEDDED FRAMEWORK AND SENSOR FUSION

This section provides an embedded architecture of an IoT device with multiple sensors. As explained in Section 1 and 2, the embedded sensors may share similar functions and detections domains. While the usage of multiple sensors helps more accurate decision makings and more wide range detections, the change of surrounding environment surrounding the device may decrease its detection ability. This issue is given from the fixed sensors' control strategy, mainly.

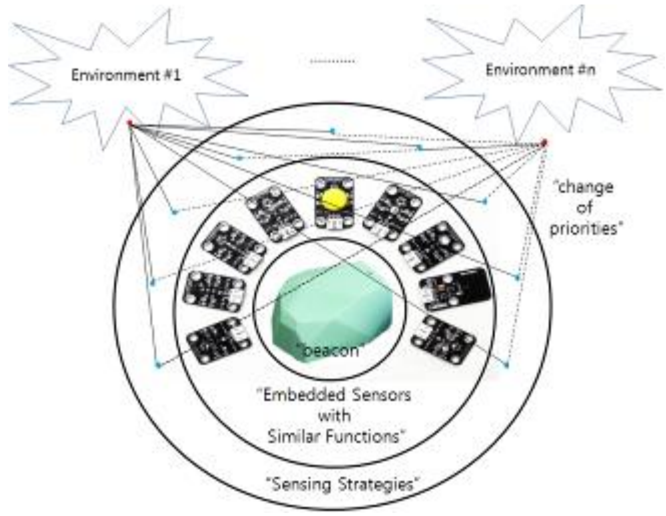


Figure 2. Conceptual sensor fusion control.

Figure 2 shows the conceptual dynamic sensor fusion control with different priorities against dynamic environment changes. As a possible example, an IoT device with three visual detection sensors is supposed. The first sensor is based on the light (e.g. a photo register) and the second visual sensor is supposed to use infrared. The latter is assumed to use different light source. In addition, it is supposed that these sensors have different detection ranges, respectively. When the device's objectives are to detect moving objects and to classify harmful animals, there are different criteria considering the size of moving objects, the light levels of the detected environments, the sensor's location and others. As it is difficult to fix with one control algorithm, it may be led to acquire a bad detection result using the fixed sensing strategy.

In order to challenge this issue, a self-propelled IoT device is modeled with its embedded architecture shown in Figure 3. The detail implementation is explained in Section 5. The device has four servo motors for its self moving control. The main role of the device is to detect moving objects and to classifying its identity (e.g. human, pets and/or obstacles such as wall and drop). In order to achieve this objective, it has several visual sensors using different light sources. This paper is based on the scenario that the priorities of these sensors are changed with the change of sensing environments (e.g. different light level in the room, objects' moving and so on). This device has a radio frequency based communication module. This module is used for transmitting the sensing results using multiple sensors to an analyzing server.

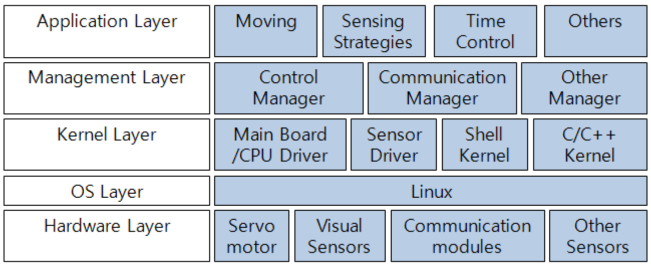


Figure 3. Embedded architecture and detailed layers of IoT device.

As shown in Figure 3, the device has the five-level embedded architecture. The hardware layer has the basic embedded system cores such as CPU, memory chips, servo motors, sensors and communication modules. The Linux is used as a basic operation system of the device, then the hardware related driver and control kernels are installed in the Kernel layer. The detailed control and learning strategies are implemented in Application layer and the required managing modules are implemented in Management Layer. Section 5 provides the detail implementations realizing the provided embedded architecture with its control mechanisms. The modeled device is used for a representing a standard IoT model equipped the proposed control framework as well as for the verification of the suggested method. The following section explains the detail mechanism of the sensor fusion control using Class RBM.

Table 2. Class RBM's layers and the mapped domain

Class RBM layer	Considerations
Input Layer (Sensor Fusion Layer)	- Multiple Sensors (e.g. Visual Sensor #1, …, Visual Sensor #n)
Output Layer (Target Layer)	- Identified Target (e.g. Human, Dog, Cat, Wall, Obstacle and others)
Hidden Layer (Control Layer)	- Relationships among installed sensors and objects - Sensor Priority

IV. SENSOR FUSION CONTROL USING CLASS RBM

As shown in Figure 2, contemporary IoT devices tend to have multiple sensors sharing similar functions. The related issues are introduced in Section 2. This section resolves the issues using the Class RBM, one of basic deep learning mechanism. As a general Class RBM has the three layers – input, output and hidden layer, each layer is mapped in Table 2.

As represented in Table 2, each embedded sensor has a binary value (0 or 1) in the input layer. The "1" means the usage of the sensor in an IoT device and the other value indicates that the embedded sensor is not used for detecting an object in a particular environment. The detecting target is represented in the output layer. Similarly, each target has a binary value. The value "1" indicates that the target is identified successfully, and "0" means that the target is not detected. In this manner, the input layer, output layer and hidden layer are considered as the sensor fusion layer, the target layer and the control layer, respectively.

In general, the number of hidden layers and the number of nodes in each hidden layer is called meta-parameters. It is assumed that the meta-parameters are predetermined in this paper. The terminologies and notations for the suggested Class RBM follow those shown in Figure 1. The general Boltzmann energy is calculated using (2). The final objective of Class RBM is to calculate parameters (W, U, b, c, d sets) obtaining the lowest Boltzmann energy.

As a basic learning algorithm for the suggested machine, Hinton [8] and Larochelle, et al. [9]'s contrastive divergence method is applied. These parameter values are decided using three stages.

In the first stage, the h set is estimated with the given conditions ($x_0, y_0, W, b, c, and\ d$) using (3).

$$\tilde{h}_0 = \frac{1}{1+e^{-c-Wx_0-Ue_{y_0}}} \qquad (3)$$

The estimated $\tilde{h}_0$ is used for calculating a direction to the lower Boltzmann energy. The main objective in the second stage is to calculated the desired values of x, y and h. $\tilde{h}_1$ is calculated with the objective function (the energy minimization). Before calculating $\tilde{h}_1$, the likelihood of h, h' is estimated using (4).

$$h' = \prod_j \frac{1}{1+e^{-c-Wx_0-Ue_{y_0}}} \qquad (4)$$

Then, y_1 and x_1 are driven using (5) and (6) Using h', respectively.

$$y_1 = \frac{e^{(d+h'U)}}{\sum_k e^{(d+h'U)}} \qquad (5)$$

$$x_1 = \frac{1}{1+e^{-b-Wh'}} \qquad (6)$$

These equations are driven using the related conditional probabilities. Finally, the desirable $\tilde{h}_1$ is calculated from (7) using the updated y_1 and x_1 in the second stage.

$$\tilde{h}_1 = \frac{1}{1+e^{-c-Wx_1-Ue_{y_1}}} \qquad (7)$$

Then, the weight between the sensor fusion layer and the control layer is determined using (8).

$$W' = W - \lambda \cdot (x_0 \cdot \tilde{h}_0 - x_1 \cdot \tilde{h}_1) \cdot (-1) \qquad (8)$$

where, λ is the learning rate

Similarly, the weight between the target layer and the control layer is determined using (9).

$$U' = U - \lambda \cdot (y_0 \cdot \tilde{h}_0 - y_1 \cdot \tilde{h}_1) \cdot (-1) \qquad (9)$$

Then, the bias set b, d and c (the bias sets in the sensor fusion layer, the target layer and the control layer) are updated using (10), (11) and (12), respectively in the final stage.

$$b' = b - \lambda \cdot (x_0 - x_1) \cdot (-1) \qquad (10)$$

$$d' = d - \lambda \cdot (y_0 - y_1) \cdot (-1) \qquad (11)$$

$$c' = c - \lambda \cdot (\tilde{h}_0 - \tilde{h}_1) \cdot (-1) \qquad (12)$$

The suggested Class RBM and the learning procedures are used for identifying the relationship between multiple sensors and the identifying patterns in an IoT device considering dynamic environment changes. While a general IoT device is controlled with the fixed strategy, the provided method contributes to the dynamic environmental adaptation of IoT device using multiple sensors. The following section shows the implementation of an IoT device with the proposed effective senor fusion framework.

V. IOT IMPLEMENTATION AND CASE ANALYSIS

This section verifies the effectiveness of the provided Class RBM based framework with the implementation of an IoT device with the sensor fusion technique. The implemented device is equipped with the architecture explained in Section 3. The device has four servo motors for moving itself. The objective of the IoT device is to identify human, pets, wall, drop and others. In order to achieve the objective, several sensors are embedded on the device. In addition, the radio frequency based communication module is installed for transmitting the identifying results. Figure 4 (a) shows a sever motor and Figure 4 (b) shows a sensor platform with Zigbee module.

As shown in Figure 4 (b), *Arduino* [10, 11, 12] is used as a basic control unit and multiple sensors (Infrared sensor, photo register, ambient light sensor and camera). The detailed embedded architecture is represented in Table 3.

The embedded architecture follows the sensor fusion architecture shown in Figure 3, Section. The overall system consists of an IoT device and an analyzing server. The server's role is to acquire detection results and to check the related Boltzmann energy during the detection processes. In addition, the additional remote controller is implemented in the server side.

(a) A servo geared motor (b) Arduino platform with three visual sensors and Zigbee module

Figure 4. Several components of the implemented IoT device.

The each sensor's target identification mechanism is implemented and embedded in the Linux system. In order to achieve the objectives, two *Arduino* are integrated into one platform and Linux (Ubuntu) is embedded as an operation system. The installed sensor has each target detection algorithm.

In order to train the Class RBM and to test its sensor fusion ability, the detection algorithms are implemented using the simple shape and color based matching techniques. Figure 5 shows the implemented IoT device with multiple sensors.

Table 3. Embedded system architecture and components of the implemented device

Layers in system architecture	**Contents and Components in Device Side**	**Contents and Components in Server Side**
Application Layer	- Drop detection function - Human detection function - Animal detection function - Line tracer detector	- Detection result analyzer - Boltzmann energy detector - Moving controller
Management Layer	- Sensor interface module - Communication interface module - Motor control module	- *Matlab* – *Arduino* Interface
Kernel Layer	- Sensor driver - Zigbee driver	-
OS layer	- Linux / *Ubuntu* Platform (embedded on *Arduino* board)	- MS-Windows Platform
Hardware Layer	- 2 Arduino devices - 4 geared servo motors - 1 camera - 1 ambient light sensor - 1 infrared sensor - 1 line inductive module - 1 Zigbee communication module - 1 ultrasonic sensor	- 1 Zigbee communication module

Figure 5. The implemented IoT device.

While the control for the sensor fusion can be fulfilled using various methods including mathematical programming based methods [13], the implemented device is controlled using the proposed Class RBM. Table 4 represents the used parameters for training the Class RBM.

Table 4. Meta-parameters in the Class RBM

Meta-parameters	**Value**
Number of control layer	1
Number of nodes in one control layer	10
Number of Target classes	5 (Human, Pets, Wall, Drop and Other obstacles)
Training data size	100
Number of training epoch	10,000

In particular, the training data considering the various dynamic environments is used. The changes of an IoT's measuring environment include the color changes of the room, the moving of living objects and the changes of the light level.

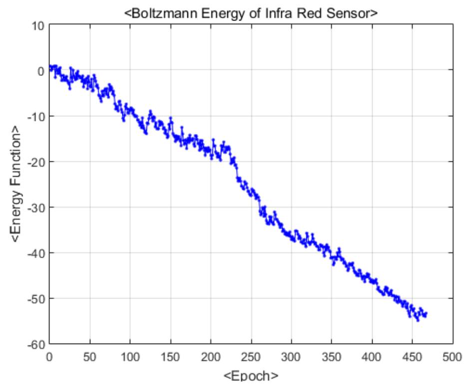


Figure 6. Adaptation of embedded infrared sensor.

Figure 6 shows the training process of one embedded sensor (Infrared sensor) in terms of its Boltzmann energy. Table 5 shows the detection rate in various situations. The same identification algorithms are applied for the experiment.

Table 5. Embedded system architecture and components of the implemented device

Situation	**Detection Success Rate (%)**	
	IoT device with fixed sensor fusion technique	**IoT device with Class RBM based dynamic sensor fusion technique**
Environment #1 (Fixed Objects, Bright Room)	93.27	91.52
Environment #2 (Fixed Objects, Dark Room)	75.64	85.69
Environment #3 (Moving Objects, Bright Room)	68.47	83.39
Environment #4 (Moving Objects, Dark Room)	59.61	80.59

As shown in Table 5, the Class RBM based dynamic sensor fusion method is considered more effective in the dynamic changes of the environment.

VI. CONCLUSION AND FURTHER STUDIES

The development of sensor fusion methods contributes to more exact measurements and controls of IoT devices. As a lot of IoT devices are introduced in industrial fields as well as a housing life, the more effective sensor fusion techniques are required. However, contemporary research studies tend to ignore the importance of dynamic sensor fusion control for adapting the environmental changes surrounding an IoT device, itself. This paper focuses on the effective dynamic sensor fusion control framework using a Class RBM technique. The data obtained considering the changes in various environments is used for training a Class RBM.

The proposed Class RBM has three layers – the sensor fusion layer, the control layer and the target layer. Each layer has a role of the input layer, the hidden layer and the output layer, respectively. The reasoned connection weights and the bias set indicate the priorities of the embedded sensors considering various environmental changes. The proposed framework contributes to increase the classification abilities and helps to achieve the objectives of an IoT device.

As further studies, this paper has the basic assumption that each sensor has the same detection algorithm in the case of dynamic environment changes. When embedded sensors' detection algorithm is changed with a particular environmental change, the proposed sensor fusion framework can be merged with the various detection algorithms for more accurate sensing. In addition, the suggested control framework is based on Class RBM. As the machine is used for a general deep learning framework, this framework is extended to a control method with deep learning techniques.

ACKNOWLEDGMENT

This paper was supported by Research Fund, Kumoh National Institute of Technology.

REFERENCES

[1] Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M., 2013, "Internet of Things (IoT): a Vision, Architectural Elements, and Future Direction," Future Generation Computer Systems, 29, pp. 1645-1660.

[2] Lee, H., and Hong, H., 2015, "Development and Control of Integrated Sensor System using Multiple Sensor Platform," Proc. 5th International Conference on Convergence Technology, Hokkaido, Japan, pp.824-825.

[3] Crowley, J. L., 1993, Principles and Techniques for Sensor Data Fusion, Springer, Heidelberg.

[4] Lee, H., and Banerjee, A., 2015, "Intelligent Scheduling and Motion Control for Household Vacuum Cleaning Robot System using Simulation based Optimization," Proc. 2015 Winter Simulation Conference, Los Angeles, USA, pp. 1-6.

[5] Yi, C., and Cho, J., 2015, "Sensor Fusion for Accurate Ego-motion Estimation in a Moving Platform," International Journal of Distributed Sensor Networks, Article in Press, pp.1-6.

[6] Choo, S., and Lee. H., 2015, "Framework and Analysis of Human Activity Monitoring System using Class RBM," Proc. KIIS Autumn Conference, Korea, pp. 1-2.

[7] Hinton, G. E., Osindero, S., and The Y. W., 2006, "A Fast Learning Algorithm for Deep Belief Nets," Neural Computation, 18, pp. 1527-2554.

[8] Hinton, G. E., 2002, "Training Products of Experts by Minimizing Contrastive Divergence," Neural Computing, 14, PP. 1771-1800.

[9] Larochelle, H., Mandel, M., Pascanu, R., and Bengio, Y., 2012, "Learning Algorithm for the Classification Restricted Boltzmann Machine," Journal of Machine Learning Research, 13, 643-669.

[10] Schmidit, M., 2011, Arduino: a Quick Start Guide, Pragmatic Bookshelf.

[11] Pearce, J. M., 2012, "Building Research Equipment with Free, Open-source Hardware," Science, 337(6100), pp. 1303-1304.

[12] Monk, S., 2011, Programming Arduino Getting Started with Sketches, McGraw-Hill.

[13] Hong, H., and Lee, H., 2015, "Prediction and Control IoT-based Device using Multi-sensory Data and Stochastic Programming," Proc. 5th International Conference on Convergence Technology, Hokkaido, Japan, pp. 826-827.