J Supercomput (2018) 74:1215–1226
https://doi.org/10.1007/s11227-017-2087-6

A load balancing scheme with Loadbot in IoT networks

Hye-Young Kim[1]

Published online: 12 July 2017

Abstract The Internet of Things' contribution lies in the increased value of information created by the number of interconnections among things and the subsequent transformation of processed information into knowledge for the benefit of society. The Internet of Things' sensors are deployed to monitor one or more events in an unattended environment. A large number of event data will be generated over a period of time in the Internet of Things. In the future, hundreds of billions of smart sensors and devices will interact with one another, without human intervention. Also, they will generate a large amount of data and resolutions, providing humans with information and the control of events and objects, even in remote physical environments. However, the demands of the Internet of Things cause heavy traffic or bottlenecks on particular nodes or on the paths of Internet of Things networks. Therefore, to resolve this issue, we propose an agent, Loadbot, that measures network loads and processes structural configurations by analyzing a large amount of user data and network loads. Additionally, in order to achieve efficient load balancing in the Internet of Things, we propose applying Deep Learning's Deep Belief Network method. Finally, using mathematical analysis, we address the key functions of our proposed scheme and simulate the efficiency of our proposed scheme.

Keywords Internet of Things (IoT) · Load balancing · Deep learning · Deep belief network · Smart sensors

✉ Hye-Young Kim
hykim@hongik.ac.kr

[1] School of Games/Game Software, Hongik University, Building B, Room #211, 2639 Sejong-ro, Jochiwon-eup, Chungcheongnam-do 339-701, Korea

1 Introduction

The very first definition of IoT derives from a "Things oriented" perspective: the things considered were very simple items with radio-frequency identification (RFID) tags. The term "Internet of Things" is, in fact, attributed to the Auto-ID Labs [1], a world-wide network of academic research laboratories in the field of networked RFID and emerging sensing technologies. These institutions, since their establishment, have been tasked to design the IoT, together with EPCglobal [2]. Their focus has primarily been on the development of the Electronic Product Code$^{\text{TM}}$ (EPC) to support the widespread use of RFID in world-wide modern trading networks, and to create the industry-driven global standards for the EPCglobal Network$^{\text{TM}}$.

The term "Internet of Things" (IoT) was first coined by Kevin Ashton in 1999 in the context of supply-chain management [3]. However, in the past decade, the definition has been more inclusive, covering a wide range of applications such as healthcare, utilities, transportation [4]. Although the definition of "things" has changed as technology has evolved, the main goal of allowing a computer to sense information without the aid of human intervention remains the same. The IoT now can allow people and things to be connected anytime, anyplace, with anything and anyone, and with any services. Our proposed standards are mainly designed to improve object visibility (i.e., the traceability of an object and the awareness of its status, current location). These standards, while not the only ones, can serve as a key component of the path to full deployment of the IoT vision.

Thus the International Telecommunication Union (ITU) vision of the IoT states: "From anytime, anyplace connectivity for anyone, we will now have connectivity for anything [5]." The vision of an Internet based on standard communication protocols connects the computer networks of IoT, Internet of People (IoP), Internet of Energy (IoE), Internet of Media (IoM), and Internet of Service (IoS) into a global IoT platform of seamless networks. This will create new opportunities in such fields as smart health, retail, green energy, manufacturing, home, city, and personal applications.

The quantity of data on the Internet is growing to approximately 2.5 quintillion bytes of data. Also, it is estimated that 90% of the data has been generated in the past 2 years [6]. IoT sensors are deployed to monitor one or more events in an unattended environment. A large number of event data will be generated over a period of time in the IoT. The IoT's contribution lies in the increased value of information created by the number of interconnections among things and by the transformation of processed information into knowledge for the benefit of society.

The benefits arising from the services controlled by communication between objects are now being increased by people who use these services in real life [7]. Hence, a load balancing protocol is critical to considerations in the design of the IoT.

Therefore, we propose an agent, Loadbot, that measures network load and processes structural configurations by analyzing a large amount of user data and network load. Moreover, in order to achieve efficient load balancing in the IoT, we propose applying Deep Learning's Deep Belief Network (DBN) method. Additionally, we address the key functions for our proposed scheme and simulate the efficiency of our proposed scheme using mathematical analysis.

The rest of the paper is structured as follows. Section 2 includes the background introduction about open-load balancing and deep learning issues. In Sect. 3, we describe our proposed architecture and provide a detailed load balancing scheme that uses an agent, Loadbot, that processes and measures network load and that processes structural configurations in order to achieve efficient load balancing in the IoT. In the final section, conclusion, we present a summary of our proposal and suggest a direction for further study.

2 Related works

The Internet of Things (IoT) is a new paradigm that is rapidly gaining ground in the field of modern wireless telecommunications. The basic foundation of the IoT is the pervasive presence around us of a variety of things or objects—such as radio-frequency identification (RFID) tags, sensors, actuators, mobile phones—which, through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals [8]. "Anytime, anywhere, any-devices" for a long time has been the vision pushing forward advances in communication technologies. The actualization of the IoT concept into the real world has been made possible through the integration of several enabling technologies. Furthermore, because of the widespread deployment of networked, intelligent sensor technologies, the IoT is steadily evolving, much as did the Internet decades ago. One example of a recently introduced IoT technology is Google's Nest Thermostat, the famous IoT gadget launched in 2014. Its designers maintain that it is carbon-neutral, in that the resources used in manufacturing and distributing the device are offset by the energy savings one obtains from using it [9].

A crucial role in the IoT will be played by sensor networks. In fact, sensor networks can cooperate with RFID systems to better track the status of things, their location, temperature, movements, etc. Thus, they can augment the awareness of a certain environment and act as a bridge between the physical and digital world. The usage of sensor net-1 New RFID tags, named chip-less tags, is under study. These tags do not use microchips, and have been proposed in several applications, such as environmental monitoring, e-health, intelligent transportation systems, military uses, and industrial plant monitoring [10].

With the growth of the IoT market, the number of autonomous devices with wireless connection capabilities is expected to grow significantly. As opposed to human-generated traffic, IoT traffic is mostly based on reporting sensor measurements on the uplink. Although such traffic is not expected to create congestion at the backbone due to its small message sizes, it may result in congestion during the channel access phase if a large number of Machine-to-Machine (M2M) devices access the channel in a short time frame. Such congestion is viewed as a serious problem, because it may result in significant access delays [11].

In the literature we have found several IoT architectures, such as LinkSmart Project [12], Vega-Barbas [13], Gao [14], Wang [15], Weiss [16] and López-de-Armentia [17].

Although numerous proposals exist for load control of the random access channel, these proposals are mostly based on announcing an access probability where the nodes are barred from channel access with some probability [18–23]. The service differentiation provided by these methods is limited; they either treat all nodes equally by announcing the same access probability, or they group nodes into a few predetermined service classes as in extended access barring [24] and treat them according to their access class. Such a limited-service differentiation is not enough for satisfying the wide range of service requirements of IoT devices. Without load balancing, a coordinator with an excessive number of wireless sensor nodes may lead to more energy consumption and experience data-reception delay, as compared to the other coordinators.

IoT sensors are deployed to monitor one or more events in an unattended environment. A large number of event data will be generated over a period of time in the IoT. Hence, a critical consideration in the design of the IoT is the load balancing protocol.

Recent breakthroughs in computer vision and speech recognition have relied on efficiently training deep neural networks on very large training sets. The most successful approaches are gleaned directly from raw inputs, using lightweight updates based on a stochastic gradient descent. By feeding sufficient data into deep neural networks, it is often possible to develop better representations than handcrafted features [25]. Perhaps the best-known success story of reinforcement learning is TD-gammon, which is learned entirely by reinforcing learning and self-play, and by achieving an exceptional level of play [26].

However, follow-up on TD-gammon, including applications of the same method to chess, Go, and checkers, were less successful. This led to a widespread belief that the TD-gammon approach was a special case that only worked with backgammon, perhaps because the stochastics in the dice rolls help explore the state space and also make the value function particularly smooth [27].

However, the principal idea behind deep learning is the use of a generative model that results in a joint distribution over the data and the labels. It has the ability to not only predict the label from the data, but also to predict the data given the label. These Deep Belief Networks (DBNs) have a general structure as shown on Fig. 1 [28]. Later, the introduction of convolutional belief networks expanded the editability of DBNs.

One really interesting application takes place where a deep neuronal network is trained on a cluster of 1000 machines (16,000 cores) to be “Selective for high-level concepts using entirely unlabeled data [29].” Using multiple concepts and ideas all through artificial intelligence (AI), the authors state that the main reason for deep networks just being able to find shallow hierarchical representations was the extent of the training time on these. So using the resources at hand they implemented a 1-billion parameter system and trained it with a large amount of data. As a result, human faces, bodies, and faces of cats were recognized with a 70% improvement over state-of-art classifications over 22k categories, where random guesses achieved less than 0.005% accuracy [30].

With this as a starting point, we propose an agent, Loadbot, that measures network load and processes structural configuration by analyzing a large amount of user data and network load, and applying Deep Learning’s Deep Belief Network method in order to achieve efficient load balancing in the IoT.

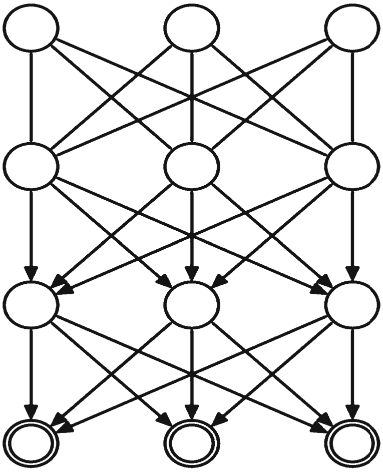

Fig. 1 Deep Belief Network [28]

3 Proposed scheme

3.1 Showing network load as a map

In order to model a complicated nonlinear relationship such as in an IoT network, we drew out a structure map of network load by using the DBN, which is a graph-generating model and which is connected to each layer by an in-depth connection between multiple layers that created a nerve network [31–35].

In other words, in a DBN used for modeling distinguished objects, each object can be represented in a hierarchical organization of basic image elements, and extra layers can muster features from low-ranked layers. The Restricted Boltzmann Machine (RBM) is a neural network that contains two layers. It has one layer of hidden units h that are not connected with each other. And the hidden units have undirected, symmetrical connections w to another layer of visible units v.

The DBN based on the RBM can represent the distribution of input samples. By using RBM's functionality, we designed a Multimodal Deep Belief Network (MDBN), as shown in Fig. 2.

By applying Multimodal Deep Belief Network, we drew out the network structure map in the IoT environment as follows.

$$P\left(x, h^1, \ldots, h^\ell\right) = \left(\prod_{k=0}^{\ell-2} P(h^k | h^{k+1})\right) P(h^{\ell-1}, h^\ell) \tag{1}$$

① First layer is an input $x = h^{(0)}$ of and is taught by Restricted Boltzmann Machine (RBM). (h means layer.)

② We make second layer by using the first layer

③ We also train the second layer by RBM.

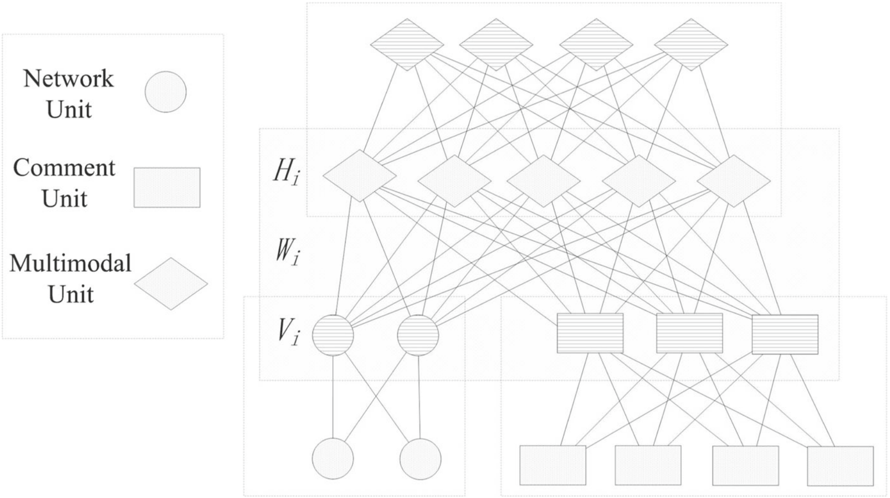

Fig. 2 Multimodal Deep Belief Network [34]

$$\Delta w_{ij}(t+1) = w_{ij}(t) + \eta \frac{\partial \log(p(\upsilon))}{\partial w_{ij}} \tag{2}$$

④ Repeat second and third procedure in order to meet the requirement. Below is the explanation of simple algorithm by using pseudo code

```
numpy_rng = numpy.random.RandomState(123)
print'...    building    the    model'    #    construct    the    Deep    Belief
Network
dbn = DBN (numpy_rng=numpy_rng,
                    n_ins=28*28, hidden_layers_sizes=[1000,1000,1000], n_outs=10)
```

3.2 Network load learning

Reinforcement learning of deep learning is acquired through actions that can be taken while carrying out network load learning in a given environment and reinforcing value in a scalar form, which we take from an assessment on a chosen action. This method can be effectively applied in network load learning because it can be performed in a dynamic environment effectively. This learning requires a dynamic-learning process of filtering variables depending on network circumstances and applying it in order to weigh the input value.

A Q-learning approach does not calculate optimum action from a current state but learns through optimum action on each condition, based on an experience of trial and error [34].

Therefore, our approach can make decisions on each action based on learned policy without complicated calculations that cost time. It can also decide an optimum action based on compensation value from each environment, even though it may not be aware

Table 1 Parameters of modeling

Parameters	Descriptions
S	Number of server in IoT environment
U	Number of users
p_u	Users' transmit and receive rate
p_s	Servers' transmit and receive rate
C_s	Service load value of server
V_u	Users' weighted value
w_x	User-centered load value
w_y	Suggested load value by Loadbot
L_b	User 'b's load value
E_s	Effectiveness of load balancing of server S

of the compensation value or probability value of every environment beforehand. Using an existing input and output data set, it learns to output wanted material when input is inserted. A network load-decision algorithm, using Q-learning, which is one of the model-free reinforcement learning techniques, is able to calculate and store current network load and learn its outcome value. Also, it has the ability to predict the outcome of network load by long-term analysis on massive amounts of data. Q value is set to 0 or initialized to a random number. The arrows in Fig. 1 show possible actions in each state.

Assuming that 100 weight value is given only when an action pointed toward G is performed and the other gets no action, the update rule about Q value causes the weight value of the target to slowly spread to all possible states, and thus allow us to make a Q-table about optimum network load. Model-free reinforcement learning is a technique to let an autonomous agent, which senses and calculates network load, to select optimum value to fulfill its purpose.

This approach can effectively be applied to a myopically optimal load-decision strategy that does not have forecasting ability. Therefore, this proposal plans to use the following algorithm in the research. Variables used in modeling the algorithm are as follows in Table 1.

In an IoT environment, there exist S number of servers and B number of users. It is assumed that server, at random time, adjust p_s by existing dynamic load balancing so when user B requests to transmit and receive service, the load value of user B, L_b is as follows.

$$L_b(p_s) = \begin{cases} \upsilon_b^{-} p_s & if\ p_a \leq \upsilon_6 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

We assume w_x when applying user-centered load value and w_y when applying the load value suggested by Loadbot. Then the variable showing the effectiveness of load balancing, E_s is, $E_s(p) = (p_s - c_s)D_s(p)$ and $D_s(p)$ is the user request rate provided by the server. Therefore, $D(p) = \rho U h_s(p) g(p_s)$ and $h_s(p)$ signify the percentage of server to be chosen by user; $g(p_s)$ is $\upsilon_u \geq p_s$ which means the user ratio.

Therefore, Eq. (4) calculates user-centered load value

$$h_s(p) = w_x \int_s^x (p) + w_u \int_s^u (p) \int_s^x (p) \tag{4}$$

$w_u \int_s^u (p)$ means calculation of load value by Loadbot.

$$E_s(p) = (p_s - c_s)h_s(p)g(p_s) \tag{5}$$

In an IoT network, for network load balancing, learn *E: S-> A*. The cumulative reward accumulated by random *E* is

$$V(s, \pi) = \sum_{t=0}^{\infty} \gamma' E(r_t|\pi, s_0 = s),$$
$$V(s, \pi) = r(s, a_\pi) + \gamma \sum V(s', \pi) \tag{6}$$

Here, s_0 is initial state, r_t is a weight value ins time t. a_π is decided by Loadbot, and s' shows the next state. By fundamental concept of Q-learning [34],

$$Q(s, a) = r(s, a) + r \sum_s V(s', u^*) \tag{7}$$

Next shows Q-learning approach and in this research, we apply this algorithm to apply it on network load learning.

```
For each state s and action a, randomly initialize table element; Perceive current states
Do forever {
    Select action a and carry it out;
    Get immediate reward r;
    Perceive new state s'; renew table element about
        Q(s,a) (Q: estimate value about real Q);
    Q (s, a)← r + γmaxa' Q(s', a' )
    s←s';
}
```

We model value of $w_u \int_s^u (p)$ and calculate $Q(s, a)$ to materialize Loadbot that measure load in IoT.

3.3 Experimental results description

In this section, we introduce the perform simulation in IoT to evaluate the effectiveness and performance of our approach. Then we compare and analyze the result of our scheme and others. Our implementation is based on the MATLAB10 for simulation and performance analysis.

We note that the most data consumption takes place proportionally when the number of sensor nodes is very large, as are network regions in the IoT.

The load problem is defined as a nonlinear problem with linear constraints, and we also present the results obtained by the simulation. We obtained the proportions

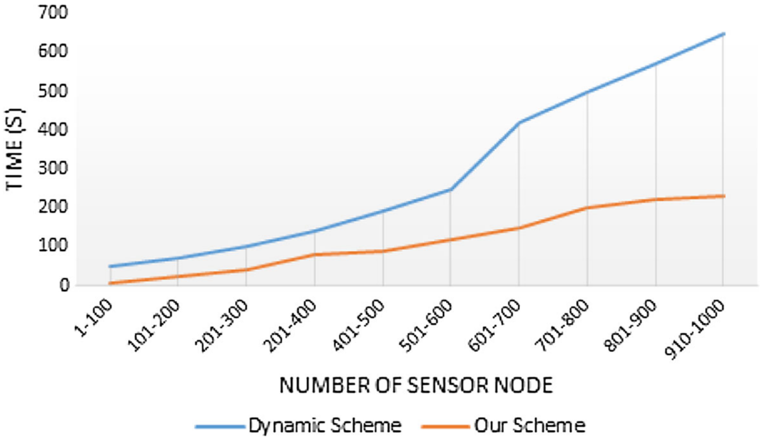

Fig. 3 Showing time of network load

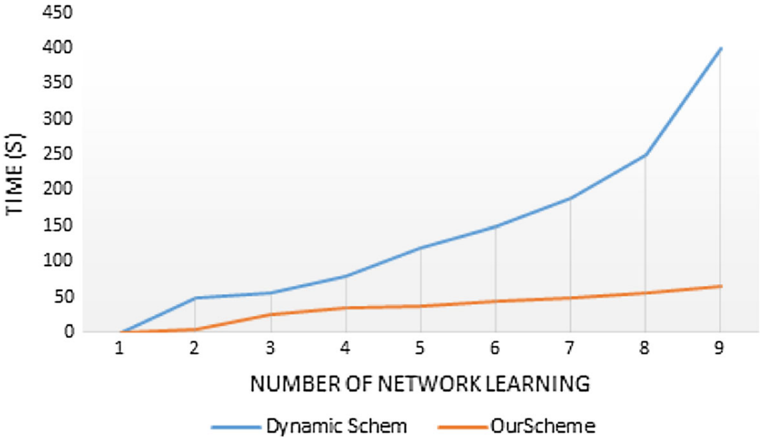

Fig. 4 Processing time of number of network load learning

of traffic very close to the optimal case. Our scheme works more efficiently than the dynamic scheme, since, while the number of sensor nodes has increased, the occurrence of migration for load balancing did not increase as much in the IoT. As shown in Fig. 3, when the number of mobile nodes exceeds 700, our scheme provides a substantial difference compared to the dynamic schemes.

Our scheme also has an advantage on large-scale networks of sensor nodes. As seen in Fig. 4, the time for processing a network structure does not increase much, even when the network itself increases in size. The processing time of our scheme shows almost no increase. Therefore, our scheme is more efficient than the dynamic scheme when comparing the time needed to process a structure for network load.

In addition, we triggered messages between mobile nodes in order to measure the migration of regions in the IoT. We note that such migration occurs less frequently than in the dynamic scheme, as Fig. 5 shows. As seen in Fig. 5, applying the Loadbot

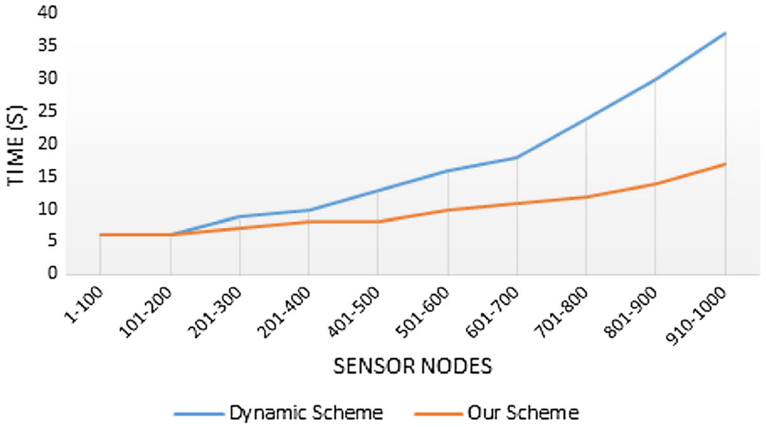

Fig. 5 Migration by number of nodes

that learns network load enables the number of times the network load is re-allocated to decrease and therefore maintain accuracy, even when the network size increases. By re-allocating network load even when the network experiences growth, the number of migrations do not increase much, therefore increasing accuracy.

4 Conclusion

In this paper, we have proposed a load-balance scheme through Loadbot that measures network load and processes structural configuration by analyzing a large amount of user data and network load, and applying Deep Learning's Deep Belief Network method. Our scheme processes a neural load prediction algorithm based on Deep Learning's Q-learning method and a neural prior ensemble in order to achieve efficient load balancing in the IoT. We address the key functions for our proposed scheme and simulate the efficiency of our proposed scheme using mathematical modeling. Then, we compare the results with previous research. Our results show the effectiveness of our scheme, and we look to further works for additional comparisons with previous research. Our applying Loadbot, which allows for network learning load and thus allows the number of times network load is re-allocated to decrease and therefore maintain accuracy, succeeds even when network size increases. By re-allocating network load even when the network experiences growth, the number of migrations do not increase significantly in our proposed scheme.

Acknowledgements This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No. 2016RIA2B4012386).

References

1. Auto-Id Labs. http://www.autoidlabs.org/
2. The EPCglobal Architecture Framework, EPCglobal Final Version 1.3. www.epcglobalinc.org. Approved 19 March 2009
3. Ashton K (2009) The “Internet of Things“. RFID J 22:97
4. Sundmaeker H, Guillemin P, Friess P, Woelfflé S (2010) Vision and challenge for realising the Internet of Things. Clust Eur Res Proj Internet Things-CERP IoT 3:34
5. ITU Internet Reports, the Internet of Things (November 2005)
6. Atzoti L, Iera A, Morabito G (2010) The Internt of Things: a survey. Comput Netw 54:2787–2805
7. IBM, Bringing Big Data to the Enterprise. http://www-01.ibm.com/software/data/bigdata/. Accessed 10 July 2012
8. Giusto D, Iera A, Morabito G, Atzori L (eds) (2010) The Internet of Things. Springer, New York. ISBN 978-1-4419-1673-0
9. http://net.com/blog/2014/04/22todays-earth-day-tomorrow-shold-be-too
10. Zhu Y, Jin Q (2012) An adaptively emerging mechanism for context-aware service selections regulated by feedback distributions. Hum-centric Comput Inf Sci 2:15
11. Kirubasri G, Maheswari U, Venkatesh R (2014) A survey on hierarchical cluster based routing protocols for wireless multimedia sensor networks. J Converg Inf Technol 9(6):19–30
12. Qin W et al (2011) RestThing: a restful web service infrastructure for mash-up physical and web resources. In: Proceedings of EUC 2011, pp 197–204
13. Vega-Barbas M, Casado-Mansilla D, Valero MA, López-de-Ipiña D, Bravo J, Flórez F (2012) Smart spaces and smart objects interoperability architecture (S3OiA). In: Proceedings of IMIS 2012, pp 725–730
14. Gao L, Zhang C, Sun L (2011) RESTful Web of Things API in sharing sensor data. In: Proceedings of ICITST 2011, pp 1–44
15. Wang H-I (2014) Constructing the green campus within the Internet of Things architecture. Int J Distrib Sensor Netw 10(3). doi:10.1155/2014/804627
16. Weiss M, Guinard D (2010) Increasing energy awareness through Web-enabled power outlets. In: MUM 2010, pp 20–30
17. López-de-Armentia J, Casado-Mansilla D, López-Pérez S, López-de-Ipiña D (2014) Reducing energy waste through eco-aware every-day things. J Mob Inf Syst 10(1):79–103
18. Hasan M, Hossain E, Niyato D (2013) Random access for machine-to-machine communication in LTE-advanced networks: issues and approaches. IEEE Commun Mag 51(6):86–93
19. Diachina J, Schliwa-Bertling P, Bergstrom A, Claes-Göran P (2010) Access control for machine-type communication devices. Patent US 8462722 B2
20. Phuyal U, Koc A, Fong M-H, Vannithamby R (2012) Controlling access overload and signaling congestion in M2M networks. In: Asilomar Conference on Signals, Systems and Computers, pp 591–595
21. Galinina O, Turlikov A, Andreev S, Koucheryavy Y (2013) Stabilizing multi-channel slotted aloha for machine-type communications. In: IEEE ISIT, pp 2119–2123
22. Duan S, Shah-Mansouri V, Wong V (2013) Dynamic access class barring for M2M communications in LTE networks. In: Proceedings of IEEE Global Communications Conference (GLOBECOM), pp 4747–4752
23. Wu H, Zhu C, La R, Liu X, Zhang Y (2013) FASA: accelerated S-ALOHA using access history for event-driven M2M communications. IEEE/ACM Trans Netw 21(6):1904–1917
24. Huidong W, Xiaoyun Z, Jianqiang Y (2014) Reconfigurable flight control based on nonlinear dynamic inversion and dynamic control allocation. Int J Syst Control Inf Process 1(3):274
25. Mostafa A, Hassanien A, Fahmy A (2010) Deep belief network for clustering and classification of a continuous data. In: Proceedings of the 10th IEEE International Symposium on Signal Processing and Information Technology, pp 472-477
26. Quinonez A (May 2016) A Model for general video game learning. Dissertation for Master Degree of Universidad Nacional De Colombia
27. Hamid M, Szepesvári C, Bhatnagar S, Richard S (2010) Toward off-policy learning control with function approximation. In: Proceedings of the 27th International Conference on Machine Learning (ICML 2010), pp 719–726
28. Barto A, Konidaris G (2010) Applying deep belief networks to the game of go. MIT Press, Massachusetts

29. O'Connor P, Daniel N, Shih-Chii L, Tobi Delbruck T, Pfeiffer M (2013) Real-time classification and sensor fusion with a spiking. J Front Neurosci 178(17):61–73
30. Krizhevsky A, Sutskever I, Geoff H (2012) Imagenet classification with deep convolutional neural networks. J Adv Neural Inf Process Syst 25:1106–1114
31. Yu L, Yang Z, Tang L (2016) A novel mutistage deep belief network based extreme learning machine ensemble learning paradigm for credit risk assessment. Flex Serv Manuf J 28:576
32. Hasselt H, Guez A, David S (2015) Deep reinforcement learning with double Q-learning In: Proceedings of the Thirteeth AAAI Conference on Artificial Intelligence 2015
33. Xiao W, Yang C, Zhao H, Bao L (2014) Parallelized extreme learning machine ensemble based on min-max modular network. J Neurocomput 128:31–41
34. Feng L, Bingquan L, Chengjie S, Ming L, Xiaolong W (2015) Mutimodal deep belief network based link prediction and user comment generation. ICONIP 2015:20–28
35. Srivastava N, Salakhutdinov R (2014) Multimodal learning with deep Boltzmann machines. J Mach Learn Res 15:2949–2980