J Supercomput (2018) 74:6859–6875
https://doi.org/10.1007/s11227-018-2288-7

IoT service classification and clustering for integration of IoT service platforms

Daewon Lee[1] · **HwaMin Lee**[2]

Published online: 4 May 2018

Abstract With the rapid development of sensors, wireless communication, and cloud computing, information technology today focuses on service environments created by the Internet of Things (IoT). IoT technologies have become widely used in various contexts including smart homes, building management, surveillance services, and smart farms. Some IoT applications such as Siri are popular in everyday life. IoT requires communication and interaction between various devices and services. To solve the various complex problems associated with IoT services, earlier research focused on IoT service platforms such as gateways and mobile edge computing services. However, the similarities and reusabilities of IoT services have received little attention. In this paper, we develop an IoT service classification and clustering system. We classify the operation of an IoT service into four steps that differ in their characteristics. Based on this classification, we extend the classic EM (expectation–maximization) algorithm to cluster IoT services in terms of their similarities. To validate our proposed classification and clustering system, we divide over 100 commercial IoT services into five clusters, showing that such services are well clustered by similarity and purpose.

Keywords Internet of Things (IoT) · Service similarity · Module reusability · Integrated IoT service platform · IoT classification · IoT clustering

✉ HwaMin Lee
leehm@sch.ac.kr

Daewon Lee
daelee@skuniv.ac.kr

[1] Seokyeong University, Seoul, Korea

[2] Soonchunhyang University, Asan, Korea

1 Introduction

In recent years, Internet of Things (IoT) services such as Siri have become progressively more widespread. With rapid developments in wireless networks and data processing, the earlier Internet environment has become enhanced with various “things.” Wireless communication and near-field communication networks, and cloud computing, have rendered personal mobile devices increasingly intelligent. In the IoT, existing mobile devices and embedded platforms communicate to provide useful services [1–4].

Most IoT devices are mobile devices with both inbuilt and external sensors monitoring ambient conditions, gravity, orientation, and acceleration; they also serve as GPS receivers to provide spatial and temporal measurements of the local environment. If IoT services are to be stable, two critical problems must be overcome, both of which depend on device performance. The first is that both power and data storage must be adequate for IoT computations; storage causes battery drain. To overcome this problem, IoT computation and storage are fully offloaded to remote computing resources such as a grid and/or the cloud. Second, device mobility may render network connectivity and device availability unstable. Such uncertainty is attributable to unpredictable node mobility, varying rates of battery drain, hardware failures, and lack of a priori knowledge on the performances of various mobile hardware and software platforms [5–8].

Most current IoT services provide their own platform. However, IoT services need to be combined with other services to form a single integrated service sharing sensed data and integrating management, a development that is compromised by the lack of power and random mobility of personal devices [9–13]. It is essential to develop integrated platforms for IoT services that exchange and manage data from heterogeneous sources. Several problems are encountered when integrating and managing the explosion in IoT services; these involve the types of sensor devices used, the form of data transmission, and the types of computation employed.

Although each IoT service has its own distinctive features, many share similar characteristics. For example, healthcare IoT services fulfill a variety of needs, but they all use similar sensors and modes of analysis. Despite these commonalities, the systems are independent, and it is difficult to integrate them. When IoT service platforms are integrated, accessibility is reduced, data complexity increases, and computation modules are duplicated [14, 15]. When integrating IoT services into a single platform, problems caused by the need to have heterogeneous devices and services collaborate and combine in a manner allowing uniform management must be solved. Reuse of computational modules in integrated IoT platforms reduces the complexity of grid/cloud manipulations and the network offloading overhead on the system side, and renders the development of new IoT services easier on the developer side. However, to ensure reusability, IoT services must be classified and clustered on the basis of similarities.

To solve these issues, we present a clustering system derived using the EM algorithm. Before clustering existing IoT services, we define IoT services by their operative characteristics: sensing, data management, processing, and execution.

- *Sensing step* We classify devices by their characteristics;

- *Data management step* We classify how data are preprocessed in the end device of the local manager or server;
- *Processing step* We classify the computational models of data analysis, data manipulation, and decision-making;
- *Execution step* We classify how services are executed.

We then use the EM algorithm to calculate similarities among IoT services. Our system is scalable and flexible, and can easily accept new IoT services. To validate the efficacy of the system, we implemented it using the baseline dataset developed in our recent work [16]. The principal contributions of this paper are:

1. We analyze the existing approaches toward IoT service classification. We define the detailed operative steps of IoT services in terms of their characteristics;
2. We propose a clustering system for IoT services, derived using the EM algorithm. This algorithm is the most effective technique available for appropriate probabilistic clustering. Additionally, the algorithm easily recognizes categorical and continuous attributes without requiring distance specifications;
3. To validate the efficacy of the system, we implemented it using the baseline dataset developed in our recent work [16]. This dataset consists of 37 IoT services that perform their own computations. Although the dataset is small, we derived it by surveying over 100 commercial IoT services in current use.

The rest of the paper is organized as follows: In Sect. 2, we discuss several related works on IoT architecture, the uncertainty of the IoT environment, and previous IoT service classification schemes. In Sect. 3, we present the system architecture that we use to develop an integrated platform on the public cloud. In Sect. 4, we describe our IoT service classification and clustering system. In Sect. 5, we evaluate the proposed system by surveying over 100 commercial IoT services in current use. Finally, Sect. 6 contains our conclusions.

2 Related works

2.1 IoT architecture

Recently, the IoT service environment has focused on communication and interaction among different devices. The architecture of an IoT service basically consists of three layers: a sensing layer, a network layer, and an application layer.

- *Sensing layer* Sensing devices such as RFID tags or smartphones;
- *Network layer* The collected data are transmitted, communicated, and processed;
- *Application layer* The IoT engages in processing and/or execution.

Our service-oriented platform provides various IoT services involving large numbers of service operations such as monitoring, discovery, and service classification [10, 12, 14, 15]. Figure 1 shows the three-layered architecture of the IoT environment.

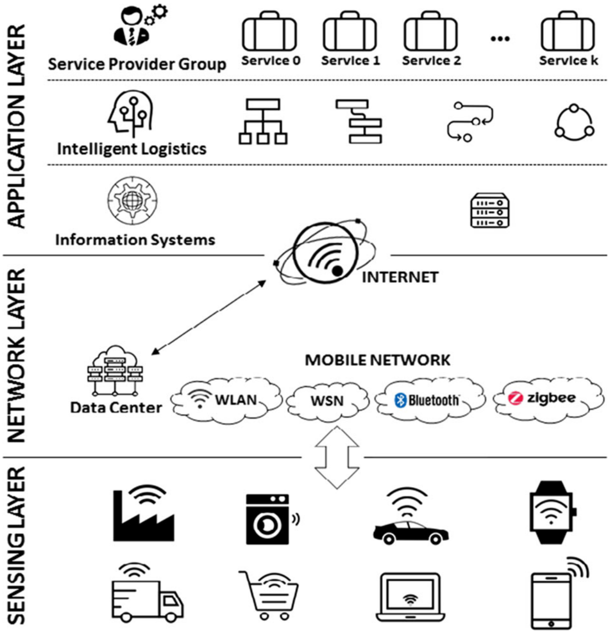


Fig. 1 Basic IoT architecture

2.2 Uncertainty

Developments in wireless networks and devices are associated with changes in IoT services from fixed to mobile nodes. In addition, many high-level, real-time applications are being developed. Therefore, mobile edge computing (MEC)-based IoT services are required to reduce the response time of the central cloud server. Figure 2 shows an MEC environment consisting of various mobile networks and mobile devices [21–23].

Although an MEC reduces the time required, the changed IoT environment features several uncertainties when it is sought to provide stable IoT services [5–8].

- The mobility of the mobile node cannot be predicted;
- The battery of the mobile device can become suddenly exhausted or discharged;
- Network failure may occur because of the low communication bandwidth of the wireless network;
- Communication and computation must deal with the heterogeneous platforms of mobile nodes.

Thus, self-provisioning and self-recovery mechanisms are essential to ensure rapid responses and the high-level service quality of real-time services such as autonomous vehicles, emergency aid, and object recognition and tracking.

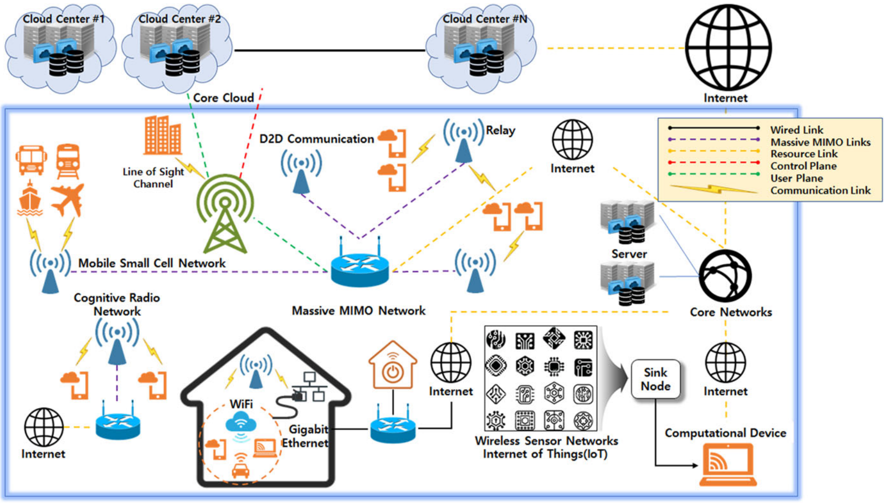


Fig. 2 Overview of an MEC environment

2.3 IoT service classification

Previous classifications of IoT services used several criteria such as the components employed, the services provided, device power, and sensor information. Sammarco et al. [17] classified IoT devices in three ways. Level 1 was based on storage and power availability, and included identification and sensing devices such as simple sensors, and passive and semi-passive RFID tags. Level 2 was based on connection methods, including ad hoc connections between sensors and wireless devices (examples: active tags and Zigbee full-function devices). Level 3 was based on the communication method used (connections between wireless devices and the Ethernet; IP/non-IP-based and Bluetooth-based devices).

In Thoma et al. [18], an IoT service allowed or blocked user access and managed sensing, action, and identification. The sensing layer was simple (for example, light, wind, or humidity). The action layer was either simple (such as on/off) or complex. The identification layer was a combination of vision, service description, and service ID.

Ning and Hu [19] classified IoT services into four types. The low-level type was a set of sensors (access devices or resources, end-mobile devices). A resource service was a set of devices managing sensors such as regional supervisors and IoT gateways. An entity service was a single service consisting of sensors and management devices. This was the core IoT service blending low-level services with resource services such as Amazon Echo. An integrated service consisted of several entity services organized in groups within an IoT service environment (smart home, smart building).

Zhu et al. [15] suggested that an integrated IoT platform should be used to share both collected and analyzed data. The platform was based on the common cloud and combined several single services.

Figure 3 shows sensor-based biomedical IoT services, indicating why cloud-based computations are needed.

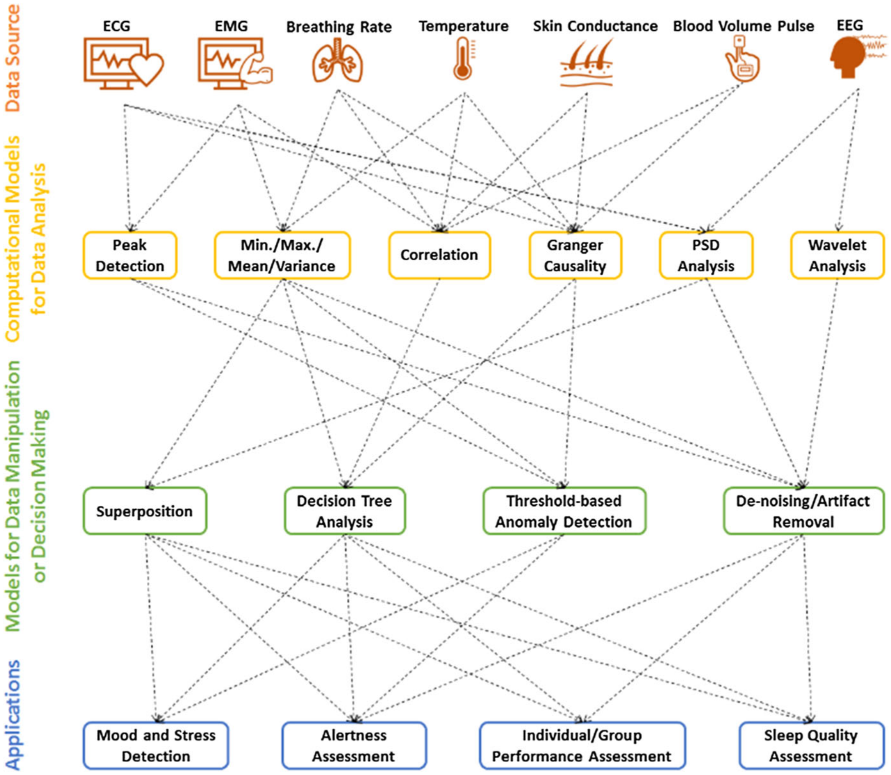


Fig. 3 Example of cloud-based computation of biomedical IoT services

Kelly et al. [20] classified IoT services in terms of power processing and proposed the use of an IoT gateway to solve the battery drain problem of low-power non-IP sensor devices that continuously communicate with others.

3 System architecture

To develop a common platform for various IoT services, we wish to place an integrated platform on the public cloud. The system environment of this study is shown in Fig. 4.

The system environment consists of four layers:

- *Sensor* The device that collects data;
- *Service environment* The environment of sensors and coordination devices;
- *Service cluster* The cluster of services within the same service environment;
- *Platform* The integrated system managing various service clusters.

In the sensor layer, sensors collect valuable data on temperature, pressure, heat, light, and sound. Sensors may provide low-level services such as alerts. If only battery power is available, high-level service (complex analysis) may be difficult; sensors must offload data to the cloud. The service environment is the local area in which

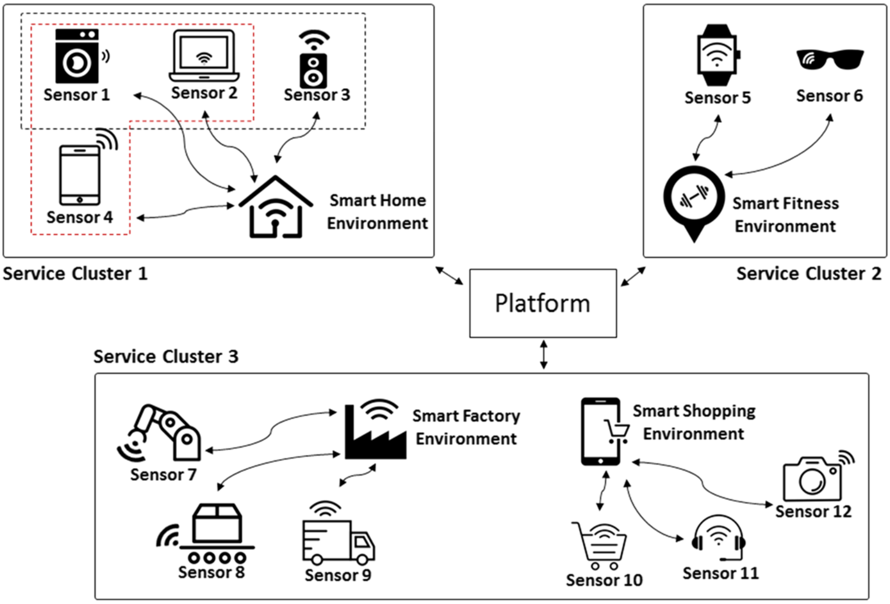


Fig. 4 System environment

a complex service is provided. For example, Amazon Echo uses local sensors and sends the appropriate response/action back to the user. A service cluster is a cluster of services providing similar services. For example, a smart home service cluster could consist of Amazon Echo, Apple HomePod, and Google Home. Our integrated platform includes several clusters (smart home, smart fitness, and smart factory service clusters). Our platform facilitates reusability of the configuration modules of each service.

4 IoT service classification and clustering

In this section, we present the classification criteria and our clustering algorithm for integrated management and enhanced reusability of the configuration modules of IoT services. Based on the service layers mentioned in Sect. 3, we divide the operation of each IoT service into four steps: sensing, data management, processing, and execution (Table 1).

4.1 Sensing step

Various sensors are used to collect input data for IoT services. These sensors perform either simple or complex actions. Sensors may perform identifications, communicate with other sensors, and communicate with servers. Here, we focus on sensor power, transmission, and operation. Table 2 classifies sensing devices in this manner.

Table 1 Classification criteria for sensing devices

CRITERION	TYPE	EXPLANATION
Power	Self-powered	Activated by battery power
	AC/DC	Activated by AC/DC power or recharged by user
	Self_recharge	Recharged independently
	Auto_recharge	Has an inbuilt charging ability
Transmission	IP	Transmit based on IP
	Non IP	Transmit based on non-IP
Operation	Event	Stay dormant until woken up by event
	Frequency	Collect data continuously
	Event2Frequency	Wake up on the event and collect data temporarily
	Timer	Collect data on a scheduled cycle

Table 2 Classification criteria for data management

CRITERION	TYPE	EXPLANATION
Pre-processing	Client side	Unify data format by client
	Server side	Unify data format by server
	Gateway side	Unify data format by gateway
Data store	Volatile	Keep data temporarily
	Non-volatile	Keep data permanently
Transmission	Distributed	Receive data from each sensor
	Centralized	Receive data from IoT gateway
Trust formation	Single	Data integrity inspection by supervisor sensor
	Multiple	Data integrity inspection by sensor

The first criteria assess the sensor power. A self-powered sensor activates its own battery. This can be a stand-alone sensor. An AC/DC sensor becomes activated using AC/DC power or its own battery and is then recharged by the user (smart lights, smart phones). Self-recharging sensors include smart cleaners. Auto-recharge sensors have inbuilt charging systems such as the ability to use solar power or movement-derived power. The second criteria explore sensor transmission (IP-based or non-IP-based such as GPS and Bluetooth). The last criteria focus on operation. An event sensor remains dormant until it is woken by an event (sound or movement). A frequency sensor (such as a temperature sensor) is always on. An event-to-frequency sensor remains dormant until woken up and then operates for a certain period. A timer-controlled sensor operates during scheduled cycles.

Table 3 Classification criteria for data processing

CRITERIA	TYPE	EXPLANATION	
Parallel	Data parallel	Compute by data parallel	
	Task parallel	Compute by task parallel	
CRITERIA	TYPE	TYPE	TYPE
Computational models for data analysis	Peak detection	Min	Max
	Mean	Variance	PSD analysis
	Wave analysis	Correlation	Granger causality
Models for data manipulation or decision-making	Belief theory	Bayesian	Fuzzy
	Weighted sum	Regression analysis	Superposition
	Decision-tree analysis	Threshold-based anomaly	De-noising/artifact removal

4.2 Data management step

No standard integrated IoT service platform has yet been clearly defined. To provide integrated management, the format of transmitted data should be unified. However, most IoT services use individualized data formats and filing systems. Thus, various data formats should be converted to the same format, allowing sharing. In the data management step, we focus on data format in transmission, the maintainability of stored data, data transmission, and trust formation. Table 2 classifies data management.

First, data are preprocessed into various sides. The data are then stored. Depending on the IoT service policy, the data can be stored temporarily or permanently (as in smart home IoT services). The data are then transmitted; data are received from sensors or IoT gateways. The last criteria involve trust. The integrity of incoming data is inspected by a sensor supervisor or by consensus among sensors.

4.3 Processing step

Most IoT services have their own models of data manipulation or decision-making. However, these models are combinations of basic operations and computational models for data analysis. Table 3 classifies the data processing modes.

4.4 Execution step

IoT services execute actions depending on their purpose. For example, a smart air-conditioning service may decide to initiate cooling by processing ambient data and sending a “cool” command to an air conditioner; temperature variations are saved and

Table 4 Classification criteria for data management

Criterion	Type	Explanation
Preprocessing	Client side	Unify data format by client
	Server side	Unify data format by server
	Gateway side	Unify data format by gateway

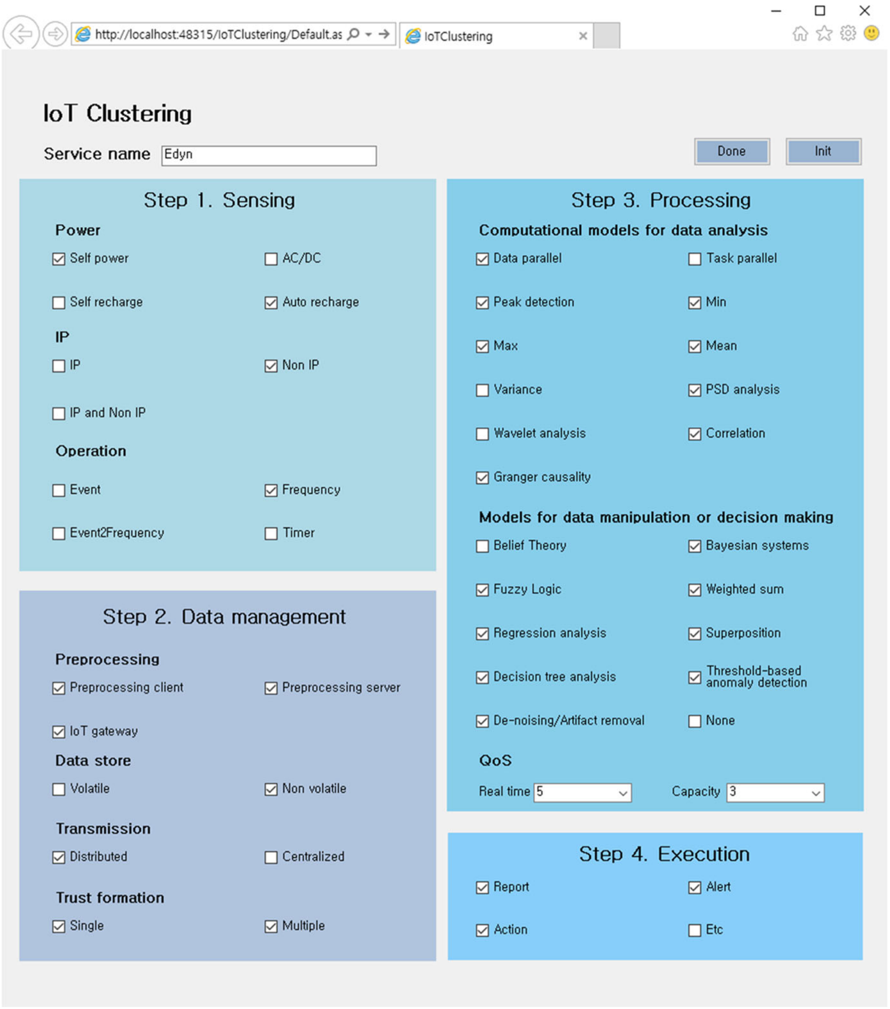
Table 5 EM-based IoT service clustering algorithm

```
Clustering Algorithm is
Observed data: $X$
Unobserved latent data: $Z$                    // dataset by classification
Initial: Estimate of the parameter $\theta$
for until convergence
  E-Step
    Calculate the expected value
    $Q(\theta|\theta^{(t)}) = \sum_z p(Z|X,\theta^{(t)}) \log L(\theta;X,Z)$
  M-Step
    Find the parameter that Maximizes quantity $\theta^{(t+1)} = argmaxQ(\theta|\theta^{(t)})$
  Additional-Step                              // preventing single instance in a cluster
    if cur_count == pre_count && cur_num == pre_num
      break
    end if
    if the std. dev. of cluster == 0
      increase cur_count and store pre_count
      increase cur_num_cluster and store pre_ num_cluster
    end if
end for
```

reported monthly. Most IoT services report their analyses, create alerts, and request action. Table 4 classifies the execution criteria.

4.5 IoT service clustering

In this section, we present our EM (expectation–maximization)-based IoT service clustering algorithm. The EM algorithm is the most effective technique available for probabilistic clustering. EM does not require distance measures and readily admits categorical and continuous attributes [24–28]. As mentioned above, our method focuses of the details of each step. We added an additional step (removal of one-member "clusters"). When the standard deviation is zero, we compare the number of clusters in the current iteration and the number of clusters in the previous iteration. This does not pose



IoT Clustering

Service name: Edyn

Done | Init

Step 1. Sensing

Power

☑ Self power ☐ AC/DC

☐ Self recharge ☑ Auto recharge

IP

☐ IP ☑ Non IP

☐ IP and Non IP

Operation

☐ Event ☑ Frequency

☐ Event2Frequency ☐ Timer

Step 2. Data management

Preprocessing

☑ Preprocessing client ☑ Preprocessing server

☑ IoT gateway

Data store

☐ Volatile ☑ Non volatile

Transmission

☑ Distributed ☐ Centralized

Trust formation

☑ Single ☑ Multiple

Step 3. Processing

Computational models for data analysis

☑ Data parallel ☐ Task parallel

☑ Peak detection ☑ Min

☑ Max ☑ Mean

☐ Variance ☑ PSD analysis

☐ Wavelet analysis ☑ Correlation

☑ Granger causality

Models for data manipulation or decision making

☐ Belief Theory ☑ Bayesian systems

☑ Fuzzy Logic ☑ Weighted sum

☑ Regression analysis ☑ Superposition

☑ Decision tree analysis ☑ Threshold-based anomaly detection

☑ De-noising/Artifact removal ☐ None

QoS

Real time: 5 Capacity: 3

Step 4. Execution

☑ Report ☑ Alert

☑ Action ☐ Etc

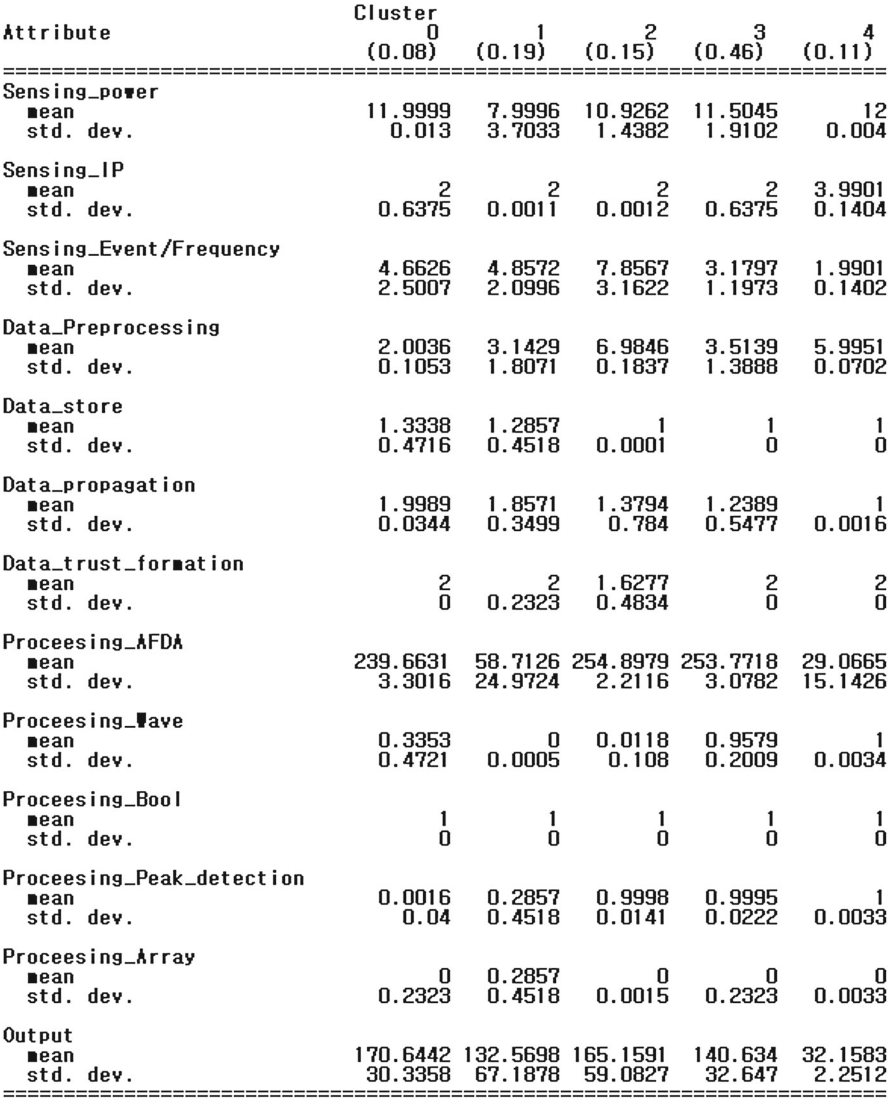
Fig. 5 Clustering platform

a clustering problem, but the dataset of commercial IoT services is currently small and focused on specific IoT service types. We mention this problem below. Table 5 shows the classification algorithm.

5 Experimental analysis

In this section, we use our EM-based IoT service clustering algorithm to evaluate over 100 commercial IoT services. The experimental environment featured a single cluster running on eight heterogeneous desktops. The experimental cluster consisted of two parts: One was Intel i7-based [8 cores (including 16 hyperthread cores), a 3.2 GHz processor, 32 GB of memory, and 256 GB of SSD]; and the other Intel i3-based [4

Table 6 Five clusters with the means and SD of each attribute



```
                             Cluster
Attribute                          0         1         2         3         4
                              (0.08)    (0.19)    (0.15)    (0.46)    (0.11)
==============================================================================
Sensing_power
  mean                       11.9999    7.9996   10.9262   11.5045        12
  std. dev.                    0.013    3.7033    1.4382    1.9102     0.004

Sensing_IP
  mean                             2         2         2         2    3.9901
  std. dev.                   0.6375    0.0011    0.0012    0.6375    0.1404

Sensing_Event/Frequency
  mean                        4.6626    4.8572    7.8567    3.1797    1.9901
  std. dev.                   2.5007    2.0996    3.1622    1.1973    0.1402

Data_Preprocessing
  mean                        2.0036    3.1429    6.9846    3.5139    5.9951
  std. dev.                   0.1053    1.8071    0.1837    1.3888    0.0702

Data_store
  mean                        1.3338    1.2857         1         1         1
  std. dev.                   0.4716    0.4518    0.0001         0         0

Data_propagation
  mean                        1.9989    1.8571    1.3794    1.2389         1
  std. dev.                   0.0344    0.3499     0.784    0.5477    0.0016

Data_trust_formation
  mean                             2         2    1.6277         2         2
  std. dev.                        0    0.2323    0.4834         0         0

Proceesing_AFDA
  mean                      239.6631   58.7126  254.8979  253.7718   29.0665
  std. dev.                   3.3016   24.9724    2.2116    3.0782   15.1426

Proceesing_Wave
  mean                        0.3353         0    0.0118    0.9579         1
  std. dev.                   0.4721    0.0005     0.108    0.2009    0.0034

Proceesing_Bool
  mean                             1         1         1         1         1
  std. dev.                        0         0         0         0         0

Proceesing_Peak_detection
  mean                        0.0016    0.2857    0.9998    0.9995         1
  std. dev.                     0.04    0.4518    0.0141    0.0222    0.0033

Proceesing_Array
  mean                             0    0.2857         0         0         0
  std. dev.                   0.2323    0.4518    0.0015    0.2323    0.0033

Output
  mean                      170.6442  132.5698  165.1591   140.634   32.1583
  std. dev.                  30.3358   67.1878   59.0827    32.647    2.2512
==============================================================================
```

cores (including 4 hyperthread cores), a 4.0 GHz processor, 16 GB of memory, and 128 GB of SSD]. We used Ubuntu 14.04 as the operating system (C# 7.0).

We clustered current commercial IoT services. The baseline dataset featured over 100 such services. However, many low-level services performed only sensing and alerting. We removed 80 such services and selected 37 as the experimental dataset; all perform their own processing. Figure 5 shows our clustering platform.

We entered the dataset into our EM-based, IoT service clustering system; 37 clusters were created by reference to purpose. However, 22 clusters contained a single service

Table 7 Characteristics of Cluster 0

Cluster ID	Cluster 0	
Basic sensor type	Self-powered, non-IP	
Data_management type	Pre-processing on server, distributed transmission	
Common operation modules	Min, Max, Mean, Compare, Bool	
Service_name	Explanation	Sensor specifications
IT bra	Healthcare_wearable	Novel sensor (monitors changes in cell temperature)
8cups	Water habits monitoring	Weight, Wireless, Bluetooth
Pillow Talk	Monitoring of sleeping	Heart rate, Temperature, Bluetooth

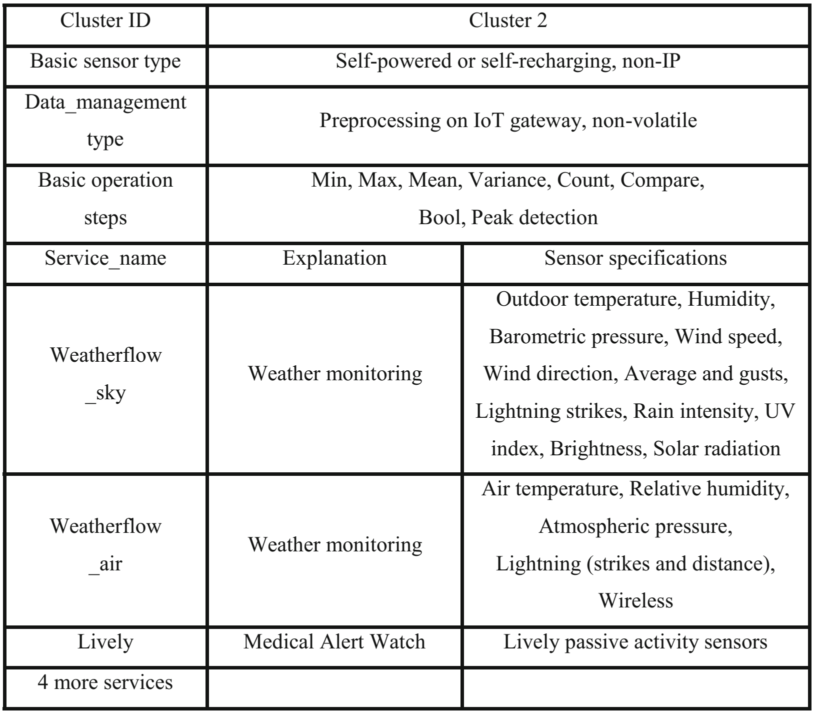
Table 8 Characteristics of Cluster 1

Cluster ID	Cluster 1	
Basic sensor type	Self-powered or AC/DC, non-IP	
Data_management type	Pre-processing on server, distributed transmission, non-volatile	
Common operation modules	Max, Compare, Bool, Peak detection, Array	
Service_name	Explanation	Sensor specifications
Bistro	Smart Cat Feeder	Heart rate, Temperature, Cat face recognition
Petfit	Pet monitoring	Heart rate, Temperature, Accelerometer
Dog Caller	Pet tracking	Heart rate, Temperature, Accelerometer, GPS
4 more services		

because of the current absence of similar services. This is why we added an additional step to the EM algorithm. After such addition, five clusters were formed. Table 6 shows the clustering results with the means and SD of each attribute of the 37 commercial IoT services.

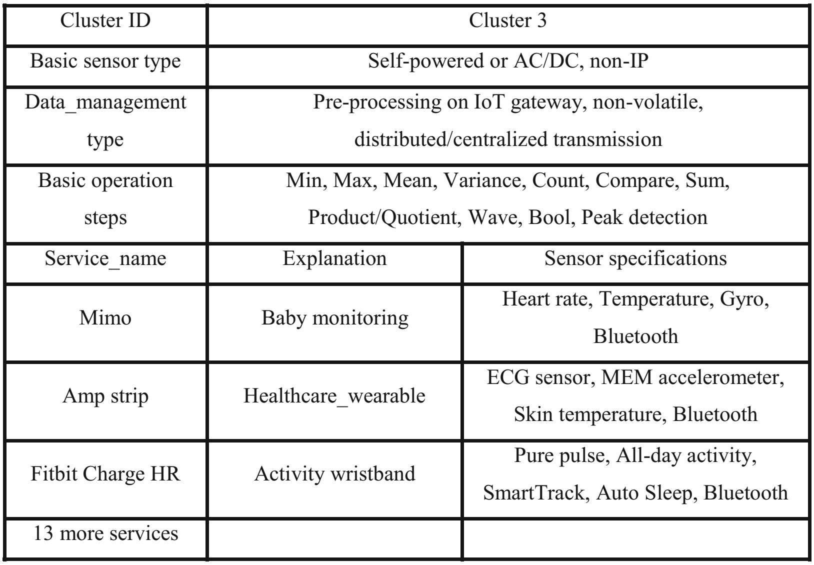
Tables 7, 8, 9, 10 and 11 show the clusters formed by our EM-based IoT service clustering algorithm.

Table 9 Characteristics of Cluster 2



Cluster ID	Cluster 2	
Basic sensor type	Self-powered or self-recharging, non-IP	
Data_management type	Preprocessing on IoT gateway, non-volatile	
Basic operation steps	Min, Max, Mean, Variance, Count, Compare, Bool, Peak detection	
Service_name	Explanation	Sensor specifications
Weatherflow _sky	Weather monitoring	Outdoor temperature, Humidity, Barometric pressure, Wind speed, Wind direction, Average and gusts, Lightning strikes, Rain intensity, UV index, Brightness, Solar radiation
Weatherflow _air	Weather monitoring	Air temperature, Relative humidity, Atmospheric pressure, Lightning (strikes and distance), Wireless
Lively	Medical Alert Watch	Lively passive activity sensors
4 more services		

Table 10 Characteristics of Cluster 3



Cluster ID	Cluster 3	
Basic sensor type	Self-powered or AC/DC, non-IP	
Data_management type	Pre-processing on IoT gateway, non-volatile, distributed/centralized transmission	
Basic operation steps	Min, Max, Mean, Variance, Count, Compare, Sum, Product/Quotient, Wave, Bool, Peak detection	
Service_name	Explanation	Sensor specifications
Mimo	Baby monitoring	Heart rate, Temperature, Gyro, Bluetooth
Amp strip	Healthcare_wearable	ECG sensor, MEM accelerometer, Skin temperature, Bluetooth
Fitbit Charge HR	Activity wristband	Pure pulse, All-day activity, SmartTrack, Auto Sleep, Bluetooth
13 more services		

Table 11 Characteristics of Cluster 4

Cluster ID	Cluster 4	
Basic sensor type	Self-powered and AC/DC, IP, Event2Frequency	
Data_management type	Pre-processing on device and server; non-volatile centralized transmission	
Basic operation steps	Variance, Count, Compare, Wave, Bool, Peak detection	
Service_name	Explanation	Sensor specifications
Amazon Echodot	Voice recognition/home secretary	High-sensitivity microphone, Sensors in area
Amazon Echo	Voice recognition/home secretary	High-sensitivity microphone, Sensors in area
Apple Homepod	Voice recognition/home secretary	High-sensitivity microphone, Sensors in area
Google Home	Voice recognition/home secretary	High-sensitivity microphone, Sensors in area

6 Conclusions

Over the past 5 years, IT has trended to form the IoT service environment, which is of major academic and industrial interest. IoT services have given us smart homes, building management tools, surveillance services, and smart farms. IoT services such as Siri are popular. We integrated these services into a cloud-based IoT service platform optimizing communication and interaction among heterogeneous devices and services. Here, to improve accessibility, reduce data complexity, and reuse computation modules in a single IoT service platform, we present a means of IoT service classification and clustering. We classify IoT services into four steps. The first step is sensing using various sensor devices. The second step is data management focusing on the data format transmitted, the maintainability of stored data, further transmission, and security. The third step is processing, divided into computational models for data analysis and models for data manipulation or decision-making. The last step classifies IoT services by the form of their execution. We extended the classic EM algorithm to cluster the services by similarity.

To validate our classification and clustering system, we surveyed over 100 commercial IoT services, eliminated over 80 low-level services, and entered 37 commercial IoT services into an experimental dataset; all perform their own processing. Experimentally, the IoT services fell into five clusters that were similar in terms of purpose. In future, we will implement our system in public clouds such as amazon EC2 and Azure.

Acknowledgements This research was supported by Next-Generation Information Computing Development Program and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (NRF-2017M3C4A7083417 & NRF-2016R1C1B1008330).

References

1. Nambi SAU et al (2014) A unified semantic knowledge base for IoT. In: 2014 IEEE World Forum on Internet of Things (WF-IoT). IEEE, pp 575–580
2. Fan YJ et al (2014) IoT-based smart rehabilitation system. IEEE Trans Ind Inf 10(2):1568–1577
3. Dorsemaine B et al (2015) Internet of things: a definition & taxonomy. In: 2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies. IEEE, 2015, pp 72–77
4. Truong H-L, Dustdar S (2015) Principles for engineering IoT cloud systems. IEEE Cloud Comput 2(2):68–76
5. Viswanathan H et al (2015) Uncertainty-aware autonomic resource provisioning for mobile cloud computing. IEEE Trans Parallel Distrib Syst 26(8):2363–2372
6. Chen H et al (2016) Uncertainty-aware real-time workflow scheduling in the cloud. In: 2016 IEEE 9th International Conference on Cloud Computing (CLOUD). IEEE, pp 577–584
7. Jamshidi P, Pahl C, Mendonça NC (2016) Managing uncertainty in autonomic cloud elasticity controllers. IEEE Cloud Comput 3(3):50–60
8. Nastic S et al (2015) Governing elastic IoT cloud systems under uncertainty. In: 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom). IEEE, pp 131–138
9. Soldatos J, Serrano M, Hauswirth M (2012) Convergence of utility computing with the internet-of-things. In: 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS). IEEE, pp 874–879
10. Botta A et al (2014) On the integration of cloud computing and internet of things. In: 2014 International Conference on Future Internet of Things and Cloud (FiCloud). IEEE, pp 23–30
11. Perera C et al (2014) Context aware computing for the internet of things: a survey. IEEE Commun Surv Tutor 16(1):414–454
12. Castro M, Jara AJ, Skarmeta AF (2012) An analysis of M2M platforms: challenges and opportunities for the Internet of Things. In: 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS). IEEE, pp 757–762
13. Nastic S et al (2014) Provisioning software-defined IoT cloud systems. In: 2014 International Conference on Future Internet of Things and Cloud (FiCloud). IEEE, pp 288–295
14. Lea R, Blackstock M (2014) City hub: a cloud-based IoT platform for smart cities. In: 2014 IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom). IEEE, pp 799–804
15. Zhu Q et al (2010) IoT gateway: bridging wireless sensor networks into internet of things. In: 2010 IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC). IEEE, pp 347–352
16. Lee D, Park K (2017) Development of IoT service classification algorithm for integrated service platform. Int J Adv Sci Eng Inf Technol 7(4):1206–1212
17. Sammarco EC, Iera EA, De Capua EC. Energy efficient procedures for an autonomous Internet of Things. http://www.chiarasammarco.com/pub/phdthesis2012-sammarco.pdf
18. Thoma M et al (2012) On IoT-services: survey, classification and enterprise integration. In: 2012 IEEE International Conference on Green Computing and Communications (GreenCom). IEEE, pp 257–260
19. Ning H, Hu S (2012) Technology classification, industry, and education for future Internet of Things. Int J Commun Syst 25(9):1230–1241
20. Kelly SDT, Suryadevara NK, Mukhopadhyay SC (2013) Towards the implementation of IoT for environmental condition monitoring in homes. IEEE Sens J 13(10):3846–3853
21. Beck MT et al (2014) Mobile edge computing: a taxonomy. In: Proceedings of the Sixth International Conference on Advances in Future Internet. Citeseer, pp 48–55
22. Mao Y et al (2017) A survey on mobile edge computing: the communication perspective. IEEE Commun Surv Tutor 19(4):2322–2358
23. Mao Y, Zhang J, Letaief KB (2016) Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE J Sel Areas Commun 34(12):3590–3605

24. Kumar Y, Sahoo G (2017) An improved cat swarm optimization algorithm based on opposition-based learning and Cauchy operator for clustering. JIPS (J Inf Process Syst) 13(4):1000–1013
25. Iam-On N, Boongoen T (2017) Generating a descriptive model for student dropout: a review of clustering approach. Hum Centric Comput Inf Sci 7(1):1
26. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B (Methodological) 1977:1–38
27. Masson P, Pieczynski W (1993) SEM algorithm and unsupervised statistical segmentation of satellite images. IEEE Trans Geosci Remote Sens 31(3):618–633
28. Bradley PS et al (1998) Scaling EM (expectation–maximization) clustering to large databases. Redmond, technical report MSR-TR-98-35, Microsoft Research