Future Generation Computer Systems 102 (2020) 965–977

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

IoT network slicing on virtual layers of homogeneous data for improved algorithm operation in smart buildings

Roberto Casado-Vara [a,*], Angel Martin-del Rey [e,f], Soffiene Affes [g], Javier Prieto [a], Juan M. Corchado [a,b,c,d,*]

[a] BISITE Research Group, University of Salamanca, Calle Espejo 2, 37007, Salamanca, Spain
[b] Air Institute, IoT Digital Innovation Hub (Spain), Calle Segunda 4, 37188, Salamanca, Spain
[c] Department of Electronics, Information and Communication, Faculty of Engineering, Osaka Institute of Technology, 535-8585 Osaka, Japan
[d] Pusat Komputeran dan Informatik, Universiti Malaysia Kelantan, Karung Berkunci 36, Pengkaan Chepa, 16100 Kota Bharu, Kelantan, Malaysia
[e] University of Salamanca, Department of Applied Mathematics, Calle del Parque 2, 37008 Salamanca, Spain
[f] University of Salamanca, Institute of Fundamental Physics and Mathematics, Department of Applied Mathematics, Calle del Parque 2, 37008 Salamanca, Spain
[g] INRS-EMT, 800, de la Gauchetière Ouest, Bureau 6900, Montreal, Qc, H5A 1K6, Canada

ARTICLE INFO

Article history:
Received 20 February 2019
Received in revised form 29 May 2019
Accepted 21 September 2019
Available online 30 September 2019

Keywords:
IoT
Complex networks
Clustering
Layer slicing
Algorithm design
Data quality

ABSTRACT

With its strong coverage, low energy consumption, low cost and great connectivity, the Internet of Things technology has become the key technology in smart cities. However, faced with a large number of terminals, the rational allocation of limited resources, the topology and non-uniformity of smart buildings, the fusion of heterogeneous data become important trends in Internet of Things research. As a result, this paper proposes a novel technique for processing heterogeneous temperature data collected by an IoT network in a smart building and transforms them into homogeneous data that can be used as an input for monitoring and control algorithms in smart buildings, optimizing their performance. The proposed technique, called IoT slicing, combines complex networks and clusters in order to reduce algorithm input errors and improve the monitoring and control of a smart building. For validating the efficiency of the algorithm, it is proposed as a case study using the IoT slicing technique to improve the operation of an algorithm to self-correct outliers in data collected by IoT networks. The results of the case study confirm, irrefutably, the effectiveness of the proposed method.

1. Introduction

Several reports indicate that commercial and residential buildings account for about 35% of total use of energy in the United States and Europe [1,2]. Consequently, buildings are noted for being the biggest contributor of final power consumption, followed by industry and transport. Due to the potentially large energy savings that can be achieved through optimized use of energy in buildings, they have become one of the main targets for reducing global consumption. Despite being great consumers, private and public smart buildings have not exploited fully the full range of the energy efficiency chances presented to them. On the other hand, they are suffering a quite significant waste of energy which is in part caused by ineffective power systems such as cooling, lighting, heating and others (devices) [3], as well as the consumption behavior of inhabitants (behavior) [4] and poor insulation efficiency. Although in the first and third categories the implementation of heterogeneous energy measures is rather expensive, soft measures, that focus on changing the inhabitants' behavior, are cheap and at the same time very effective in reducing energy use [5].

In smart building the IoT devices are spread all over the building in order to monitor it. This research focused on temperature IoT devices. Notice that temperature IoT devices depend on the physical topology of the building and the non-uniformity of the building temperature. Thus, we will assume that the data collected by IoT devices are heterogeneous due to the topology of the building (offices that are closed and with their own heating, corridors, large common areas, etc.) and the non-uniformity of the temperature (i.e., the temperature vary between different zones because users can choose their comfort temperature, open or close doors and windows, etc.). Due the above mentioned reasons, heterogeneous data are not optimal inputs for temperature control algorithms; as a result the precision of control algorithms is lower than if homogeneous data were used. Addressing the above mentioned inefficiencies as a result of lack of control algorithm

* Corresponding authors.
E-mail addresses: rober@usal.es (R. Casado-Vara), corchado@usal.es (J.M. Corchado).

https://doi.org/10.1016/j.future.2019.09.042

which use heterogeneous data one could consider using homogeneous data to improve the accuracy of them and, in particular, homogeneous data collected by IoT networks. This research have the following challenges in the field of monitoring and control in smart buildings with IoT networks.

1. Indoor temperature collected by IoT devices can be affected by the **topology** of a smart building. For example, the topology of a physical space can capture the layout of a smart building, including its structural relationships, such as containment (e.g., a building contains rooms, corridors, common spaces) and connectivity (e.g., two rooms are connected through a door or corridor).
2. Indoor temperature collected by IoT devices can be affected by the **non-uniformity** of a smart building. Traditionally, indoor air environments have been considered uniform, and therefore, monitoring and control algorithms may be having less accurate results than expected. However, IoT devices are located in different rooms with heating systems with displacement ventilation systems, underfloor distribution systems or customized heating systems (i.e., the user can choose their comfort temperature). In the case of a mixed heating system, temperature and discrepancy may also vary by location.

Motivated by the above observations and challenges, this article presents a novel technique for transforming heterogeneous data collected by IoT networks into homogeneous data to improve the IoT network efficiency in control and monitoring. In this work the effectiveness of this technique will be tested with a data quality algorithm, which increases data confidence and detects wrong data. Compared with the existing results, the advantages of the proposed approach are summarized as follows.

1. Combining graph theory and clustering techniques together with the heterogeneous data collected by IoT networks, this paper proposes a novel model to transform heterogeneous data in homogeneous data separated by layers. The proposed technique has the advantages of improving the effectiveness of the algorithm which have these data as input since our new model keep the clusters.
2. Using the IoT network slicing on virtual layer technique, algorithms are applied on homogeneous data. This way, these algorithms are not malfunction due to the use of inaccurate data collected by IoT devices.

To prove the efficiency of the innovative IoT network slicing technique a case study is proposed. This technique presented in this research work serves to transform heterogeneous data into homogeneous data. Therefore, in order to demonstrate the efficiency of this new technique, an algorithm will be applied to the data collected by the IoT network, and the same algorithm will be applied to the data collected by the same IoT network but using the IoT network slicing technique. The algorithm that will be used to demonstrate the efficiency of this technique is the algorithm developed by *Casado-Vara et al.* [6]. This algorithm increases data quality collected by the IoT network by making coalitions of neighboring IoT devices. In this way data quality algorithm can self-correct the wrong data (i.e., outliers data values). Therefore, this algorithm smoothes the temperatures of the whole smart building without regard to the topology or uniformity of the temperature of the building. Notice that the novel technique presented in this article is IoT network slicing, which transforms heterogeneous data (regarding topology and non-uniformity of data) into homogeneous data. While the case study used to test the efficiency of the IoT network slicing technique is an algorithm that has as input the data processed by the new technique presented in this paper.

The structure of this paper is described as follows: In Section 2 overviews related works to the main aspects of this paper. The details of our proposal are presented in Section 3. Section 4 shows the case study details proposed to test the efficiency of our new method. Section 5 presents two simulations, the results of which validate the competency of the proposal. Finally, Section 6 draws conclusions from the conducted research and describes future lines of work.

2. Related work

IoT networks usually have heterogeneous data, but the most of algorithms do their best with homogeneous data. Therefore, there is a need to be able to make the algorithms we apply for monitoring and control of IoT networks can have as input homogeneous data. This requires models, theories, methodologies, tools and mechanisms to develop a system that is able to adapt and organize itself to possible upcoming changes in its surroundings. Therefore, this paper suggests a fusion of technologies, such as graph and complex network theory, clustering techniques and game theory with IoT, as the technological context key to addressing the existing requirements in smart building environments.

2.1. Graph theory

A graph is a mathematical presentation of a network and shows the connection between vertices and edges. Graph theory is used to represent real-life phenomena however, sometimes they encounter difficulties when representing certain phenomena due to the uncertainty introduced by the different attributes of a system. The definition of fuzzy graphs has been in many cases motivated by real-world phenomena. Kauffman [7] introduced fuzzy graphs using Zadeh's fuzzy relation [8]. Fuzzy-graph theory is growing rapidly, with numerous applications in many domains, including networking, communication, data mining, clustering, image capturing, image segmentation, planning, and scheduling. Graphics are extensively used for modeling the structured information, molecular structures [9,10], including routings [11], and social networks [12]. Due to the widespread use of graph information, significant work has been carried out to improve methods for the efficient analysis and management of graph information, such as graph similarity search [13,14], graph matching [15], and graph mining [16,17]. Sampathkumar [18] presented the notion of graph structures. Graph structures are the generalization of graphs and are widely used in the study of some structures, like signed graphs, edge-colored graphs, semi-graphs, and edge-labeled graphs. Graphs structures are extremely useful in researching several domains of computational intelligence and computer science.

2.2. Clustering techniques

During the last few years, the automatization of data recording and acquisition has involved an avalanche of information on different types of systems [19,20]. As a consequence, a number of methods for modeling and organizing data have been designed. These methods have been inspired by their generalized applicability in education, forecasting , diagnostics and many other fields. These methodologies are defined, evaluated and applied in the field of automatic learning, which has developed into one of the main subareas of statistics, mathematics and computer science due to its central role in the contemporary society. Automatic learning covers a number of different fields, such as classification [21], feature selection methods [22] and regression analysis [23]. The last-mentioned implies the assignment of classes to the dataset objects. There are three main focuses for grouping: unsupervised grouping, semi-supervised and supervised. Clustering approaches are usually more demanding than supervised methods, but they also require a better comprehension of complex data. Such classifiers are the main focus of the present paper.

In [24], a comparison of the clustering approaches was carried out in the context of the verification task of language-independent speakers, utilizing three datasets of documents. There were considered two methods: clustering approaches in order to minimize an objective distance-based function and an approach based on Gaussian models. Algorithms compared were: random swap, k-means, hierarchical clustering, expectation maximization, diffuse c-means and self-organized maps (SOM). In [25], five clustering methods were studied: spectral, hierarchical clustering, multivariate Gaussian mixture, nearest neighbor methods and k-means. Four measures of proximity were used in the experiments: the Euclidean distance, Cosine similarity, and Pearson and Spearman correlation coefficient. These algorithms have been tested within the scope of 35 gene expression from cDNA or Affymetrix chip platforms, utilizing the standardized rand index for the evaluation of performance. In [26], experimental work was carried out to cross-check five different clustering algorithm approaches: self-organized mapping-based method, CLICK, dynamical clustering, hierarchical and k-means. Gene expression time series datasets for Saccharomyces cerevisiae were used. In the literature several different kinds of clustering approaches have been suggested [27,28]. Some methods are used more often than other approaches, although this diversity exists [29]. Many taxonomies have been suggested to classify the different kinds of family clustering algorithms. As some taxonomies classify algorithms according to their target functions [30], some are focusing on reaching the desired cluster-specific patterns (e.g. hierarchical) [31].

Over the past few years, it has become extremely important to process high-dimensional data efficiently, and because of this, it is important to, anyone looking for a method of obtaining accurate partitions should make sure that the chosen method include this feature. We have discussed the use of clustering algorithms for static data. However, while performing data analysis, it is important to consider whether the data is static or dynamic. Dynamic data, unlike static data, changes with the time. Some types of data, such as network packages hosted by routing, IoT temperature flow and extended credit card transaction flows, are temporary in nature and are referred to as data flows.

2.3. Game theory and IoT

A Networked Control System (NCS) is a control system wherein the control loops are closed through a communication network [32]. Advantages include a reduced maintenance and deployment fee, reduced wiring and flexibility [33]. NCSs are suitable for a broad variety of fields [34]. The weaknesses of NCS are a result of random communication time delays and package send failure. As regards existing industrial application, centralized end-to-end conventional control is not appropriate since it does not comply with such new features as modularity, decentralized/distributed control, easy and fast maintenance as well as reduced cost. As a result, in the past few years NCS has attracted both academic research and industrial implementations, which has contributed to significant progress in this field [35]. IoT networks often include a large number of IoT nodes as well as control nodes and actuators. Each of these distributed IoT nodes are competing to transmit their information to the network. It is necessary to implement a control system that supervises and controls the sending of packets from the IoT nodes to the network. Also, IoT networks continuously produce lots of information. The amount of data complicates the supervision and control of IoT networks. In order to control the IoT network a search of the repositories is required, which results in a delay in the operation of the control system within the IoT network. This also involves high energy and resource consumption.

While the issue of communications bottlenecks in networks has been well researched, there has been comparatively few work in optimizing queues analysis. In addition, all of these studies do not include network features in the control system [36–38]. The issue is still not solve. The characteristics of IoT networks are still having a major impact on system efficiency. Therefore, it is necessary to address these characteristics in the control algorithm applied by the networks [39]. The majority of the investigation is focused on the improvement of the energy use to improve the performance of the NCSs. The use of NCS with IoT network features, which includes queuing analysis, packet dropout and service rate, needs to be the subject of greater research. In an effort to improve the efficiency of IoT network supervision and control, some researchers suggested new technologies that improve the searching rate in large databases. Xu et al. propose another technologie to improve data quality and their reliability. In this case, the authors propose the RFID technologie. RFID technology is widely used in the Internet environment of things (IoT) for tracking objects. With the expansion of its application areas, the demand for reliability of business data is becoming increasingly important. To meet the needs of high-level applications, data cleansing is essential and directly affects the accuracy and integrity of business data, so RFID data needs to be filtered and processed. In this paper, the authors propose an SMURF scheme that relies on dynamic tags and considers the influence of data redundancy [40]. Zhou et al. developed a comprehensive approach to meeting the needs of multi-objective consultation between users and the data platform, as well as query accuracy, promptness and confidentiality restrictions [41]. Other investigation suggested using a binary hash for higher searching speed; Cao et al. compared these hashing techniques from different tests [42].

The approaches introduced in the revised bibliography address different topics, such as the distributive IoT platform, privacy, data authentication, and security issues, among others. However, all research findings in the state of the art propose as part of their approach, the use of heterogeneous data in IoT networks. However, we discovered a gap in literature reviews since there are no algorithms that can effectively manage heterogeneous data. In this work, we propose a methodology that allows to transform heterogeneous data into homogeneous data using clustering techniques and theory of complex networks.

3. IoT slicing method

In this section, the solution that we propose to address the above problem is described. The inaccuracy problem arises from the application of temperature control algorithms to IoT nodes with heterogeneous data. To counteract this problem we propose the combination of several mathematical and artificial intelligence techniques with which we have developed an intelligent and self-adaptive model that allows for the use of temperature control algorithms in all types of IoT networks. The operation of this model begins with the collection of data by the IoT nodes. These data are usually heterogeneous (i.e., the temperatures collected by the IoT network in a smart building are very different depending on the area of the smart building where they are collected). The proposed method encompasses the following techniques or algorithms:

(1) First, a graph is constructed in which the nodes will be the IoT nodes and the edges of the graph will be the doors or corridors that join those IoT nodes. Since a graph has been built with the IoT network, a complex network can then be built with this graph.
(2) Now we apply clustering algorithms using the temperature data collected by the nodes of the complex network (i.e., IoT nodes). In this way, heterogeneous data are separated into homogeneous clusters.
(3) Next, a multiplex is built in which each of the layers represents one cluster.
(4) Multiplex layers that have unconnected networks will use virtual nodes to build a related network. In this way the algorithms can be applied correctly in the following stages.
(5) The control algorithms required to be used will be applied depending on the purpose on homogeneous data in each of the layers.
(6) Each layer is projected on to the complex network and the control signal is sent to each one of the actuators assigned to the IoT nodes.

In this section, we describe all the techniques that are required for the development of this model. A flowchart summarizing this model is presented in Fig. 1.

3.1. Graph design module

The graph is constructed from the topology of the IoT nodes in the intelligent building. In this way, a graph is formed where the vertices of the graph are the IoT nodes and the edges of the graph are the physical connections between the rooms where the IoT nodes are located (i.e. there is no obstacle on the way from one node to another). That is, if between the rooms in which the IoT nodes are located there is a door and a corridor, then there is an edge in the graph. The graph represents the heat transfer between the physically connected rooms. An illustrative example can be found in Fig. 2. This is how we build a graph from the IoT network. Now let us consider this graph as a complex network, also we consider non-directed graphs. Then, we create the *Laplacian matrix* of the graph as follows:

$$A = \begin{cases} 1 & \text{if } (i,j) \in E \\ t_i & \text{if node } j = i \ (t_i = \text{temperature of the } i\text{th node}) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where E is the set of edge in the graph. For the example proposed in the figure, the *Laplacian matrix* is:

$$A = \begin{pmatrix} t_1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & t_2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & t_3 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & t_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & t_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & t_6 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & t_7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & t_8 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & t_9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & t_{10} \end{pmatrix} \quad (2)$$

3.2. Clustering module

Temperature data collected by the IoT network of the smart building are usually heterogeneous data (taking into account the topology and the non-uniformity of the smart building). The main purpose of applying the clustering technique is to find IoT devices with similar data (regarding topology and non-uniformity of the smart building). Thus, we can create "similar group of IoT devices" according their collected data based in the clusters. So, the Iot slicing method transform heterogeneous data into groups of similar data based in the cluster technique output. We call these groups of similar data, homogeneous temperature data groups. These IoT devices with homogeneous data are formed by IoT devices with similar smart building topology (devices in corridors, devices in common areas, etc.) and the same smart building uniformity (similar heating/cooling systems, similar temperature conform choices, etc.). Then, these groups of homogeneous temperatures data are a different layer in the multiplex, this way we can apply algorithm in each layer, and the algorithm's input are the homogeneous temperature data.

In order to determine the number of clusters that the cluster algorithms have to find, an expert in energy efficiency was consulted. The distribution of zones with topological and uniformity characteristics in smart buildings are usually: (1) Rooms or offices. In these rooms the users can choose their comfort temperature and therefore, they will have slightly different temperatures to the rest of the building. (2) Open-plan zones. These common areas such as the reception, the rooms where the offices are, etc. have different temperatures due to the fact that the doors or windows can be open and this can modify the temperature of the smart building in relation to what is outdoors. (3) Corridors. The corridors are areas that are open and communicate the different areas of the building that may be at different temperatures. In this way, using the same monitoring and control system to control the temperature of the three zones we have defined is not optimal, as each of the zones will have very different temperature data from the previous ones. In some cases, this choice of clusters may vary and have a larger number, depending on the topology and uniformity of the smart building. An illustrative example is shown in Fig. 3. We apply this clustering technique to our 1-dimensional temperature array and the output is the graph with the clusters.

The chosen clustering technique is Gaussian Mixture Models (GMMs) that give us more flexibility than K-Means. With GMMs we assume that the data points are Gaussian distributed; this is a less restrictive assumption than the k-means algorithm uses the mean to form circular clusters. That way, we have two parameters to describe the shape of the clusters: the mean and the standard deviation. With an appropriate amount of component mixture, it is also possible to estimate almost all continuous probability density functions. Gaussian mixture density is defined as

$$p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k) \quad (3)$$

where x is a d-dimensional random variable, $N(x|\mu_k, \Sigma_k)$ is a multivariate normal distribution with mean μ_k and covariance matrix Σ_k and π_k are the so-called mixing coefficients for the k components of the distribution $p(x)$ which have to satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$ to form a convex combination of the mixture components [43].

3.3. Multiplex module and virtual network module

Many real world systems are characterized by multiple types of interactions that are happening simultaneously. The need for networks that allow different types of interaction has long been recognized in sociology. Human beings are involved in many social interactions at the same time [44]. The quality of a social bond can be completely different, whether two individuals are siblings, married, colleagues, or have only met once on a bus. Transport networks have a multi-relational structure, where different types of links correspond to different modes of transport. Multiplex networks allow us to represent different types of interaction between nodes [45].

An example that is very easy to understand is the transport network that includes, for example, the airport network, the

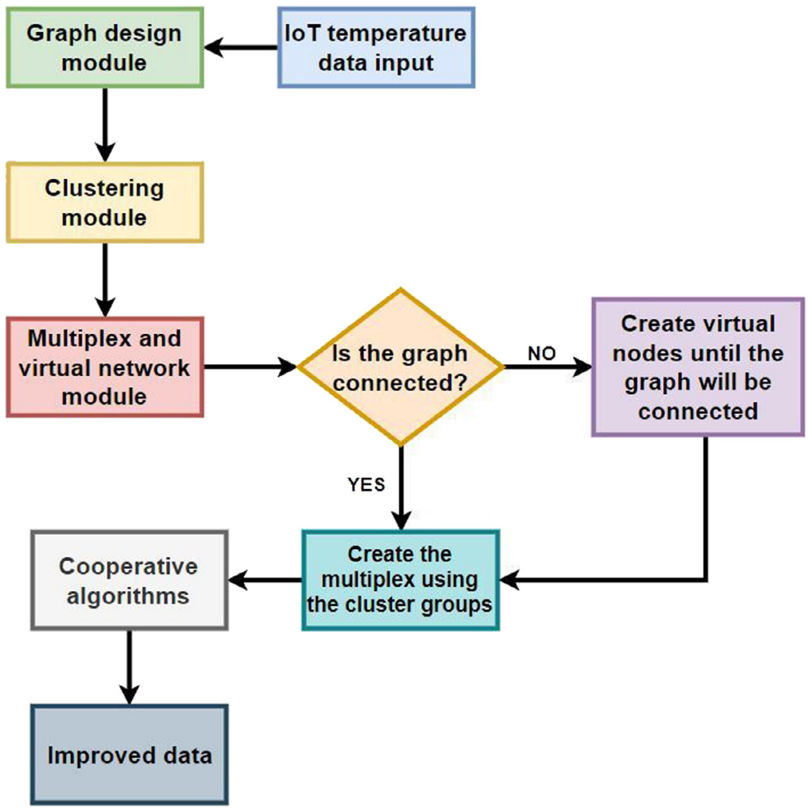

Fig. 1. Proposed model flowchart.

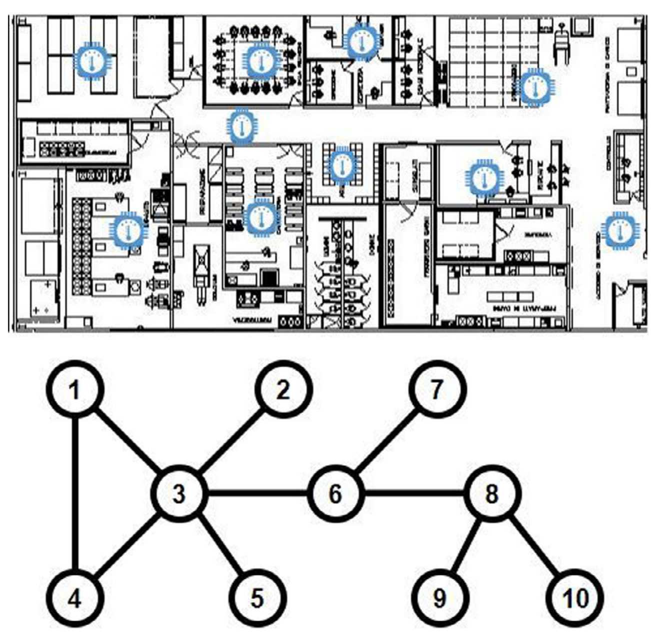

Fig. 2. Illustrative example of the construction of a graph based on the position of the IoT nodes on a map.

road network and the metro network. In the case of the airport network it is usually modeled as a single network where the nodes are the airports and the edges are the direct flights between airports. However, a better representation would be to consider a multiplex network made up of many networks each of which would correspond to a particular airline, where each node would again be the airports and each corner would correspond to the direct flights between airports performed by that airline. Each of these networks may have the same set of nodes but different edges, as not all airlines have flights between the same airports. Multi-layer or multiplex networks can be defined as those that incorporate different connectivity channels, and describe systems that are interconnected with different categories of connection: each channel is represented by a sub-network or layer and the

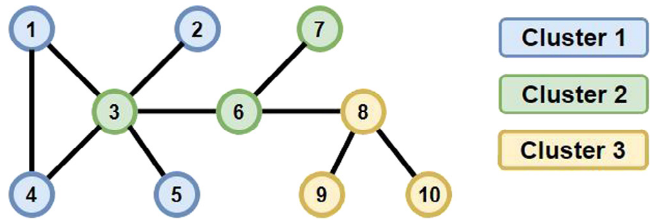

Fig. 3. Illustrative example of the application of the GMMs cluster technique to the graph with the temperatures of the IoT network.

same node can have different types of links and a different neighborhood in each layer. Let us formalize the concept of multiplex networks or multilayer networks.

Multiplex network is a couple $M = (G, E)$ where $G = \{G_\alpha, \alpha = 1, \dots, M\}$ is a family of graphs $G_\alpha = (N_\alpha, L_\alpha)$ (i.e., layers of the multi-layer network) and $E = \{E_{\alpha\beta} \subset N_\alpha x N_\beta, \quad \alpha \neq \beta\}$ is the set of links between the nodes of the different layers. The E elements are called cross-layers, the L_α elements are called interlayer connections. The set of nodes of each layer is denoted by $N_\alpha = \{n_1^\alpha, \dots, n_{N\alpha}^\alpha\}$ and the *Laplacian matrix* of each layer is denoted by $A^{[\alpha]}$. Each node can be present on each of the layers. Links always connect nodes on the same layer.

The technical problem we have in this research is that we have a graph with the temperatures of a smart building that are interrelated between them and cannot be analyzed in isolation. Therefore, the solution is that an approach is made using the multiplex technique. In this case it can be modeled as a multiplex network where the nodes (i.e., IoT devices) are the sensors and the edges are the physical connections between the different IoT devices. Each layer of this multiplex network will correspond to a temperature cluster, where each node of each layer has a temperature similar to the other nodes of the same layer, and the edges are the physical connections between them. The importance of using the multiplex approach to solve this problem is that each of the layers of the multiplex will have homogeneous temperatures, which will improve the functioning of the algorithms that are applied on the data. As an example, the graph in Fig. 3 has 10 nodes and 3 clusters. In this graph you can find the temperatures collected by IoT devices, their physical relationship and the areas with similar temperatures (rooms, corridors and open-plan sites). The steps to build the multiplex from this graph are as follows:

1. Each of the clusters corresponds to a layer in the multiplex.
2. Each one of the nodes belonging to a cluster is placed in its corresponding layer of the multiplex with its edges.
3. If the network of each layer is not connected, the minimum number of nodes necessary for it to be related is virtualised. In case it is related, nothing is done.

The multiplex built from Fig. 3 are presented in Fig. 4. Also, the process of the IoT devices collecting the data until the creation of the multiplex is presented in Fig. 1.

Once the multiplex is constructed, the connection between each graph and its layer is verified. Is essential that in case they are not connected, as many virtual nodes are created as necessary to link the graph with the layer *ith*. The virtual nodes are vertices of the graph of other layers, however we act as if it were in that layer for practical purposes. In this way, the graphs will be connected and the control algorithms can be applied subsequently. If there is only one element in a layer, as many vertices of other layers will be virtualized as the amount of edges of a node in the initial graph. In the *Laplacian matrix* we will represent a virtualized node as 1^*.

Applying this definition now to the graph that we have constructed in the previous point, we are going to create a multiplex. Each one of the layers of the multiplex is going to be each one of the nodes of the graph that are in the same cluster. In order to assist in creating the multiplex we are going to create each of the adjacency matrices for each of the layers ($A^{[\alpha_1]}$, $A^{[\alpha_2]}$ and $A^{[\alpha_3]}$) as follows:

Layer 1 of the multiplex contains only the IoT nodes 1, 2, 4 and 5. Also, we have to create a virtual node to virtualize the IoT node 3.

$$A^{[\alpha_1]} = \left(\begin{array}{ccccc|ccccc} t_1 & 0 & 1^* & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & t_2 & 1^* & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1^* & 1^* & t_3^* & 1^* & 1^* & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1^* & t_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1^* & 0 & t_5 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) \quad (4)$$

Layer 2 of the multiplex contains only the IoT nodes 3, 6 and 7.

$$A^{[\alpha_2]} = \left(\begin{array}{cc|ccccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & t_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & t_6 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & t_7 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) \quad (5)$$

Layer 3 of the multiplex contains only the IoT nodes 8, 9 and 10.

$$A^{[\alpha_3]} = \left(\begin{array}{ccccccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_8 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & t_9 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & t_{10} \end{array}\right) \quad (6)$$

In the *Laplacian matrix*, each layer of the multiplex has been marked with lines. This way, the next step in the algorithm is to build each layer of the multiplex. Fig. 4 shows a multiplex which illustrates the above techniques.

3.4. Protected data output

Once the control algorithms have been applied to the homogeneous multiplex data, the results are projected on to the layer $\alpha = 0$ (i.e., initial layer). In this way, we have a system similar to a Multi-input Multi-output (MIMO) system. This method improves the performance of the algorithms by the data processing done with the IoT slicing method.

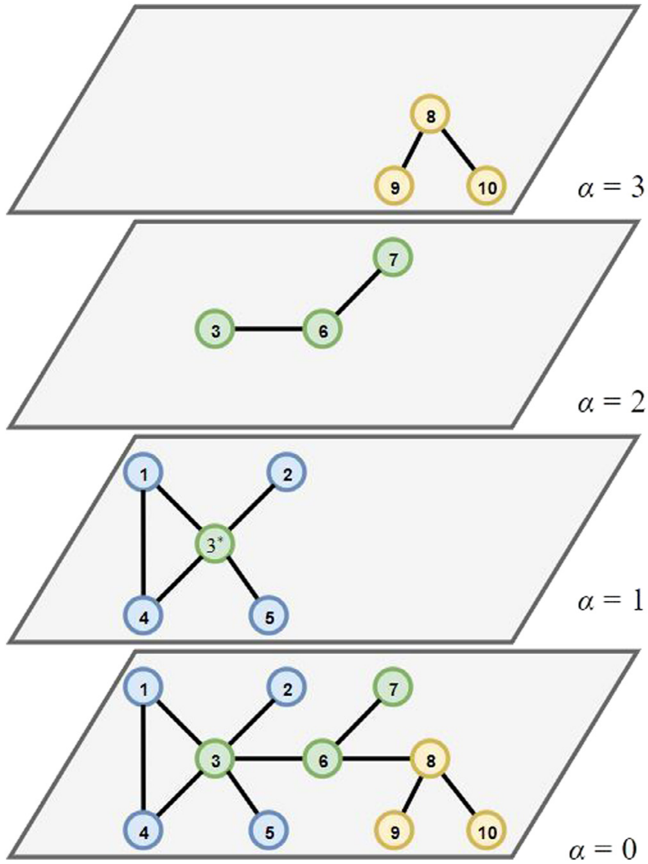

Fig. 4. Multiplex with 3 layers. In each of the layers are the IoT nodes that have similar temperatures according to the clustering algorithm used.

4. Performance evaluation

In this section, we evaluate through a experimental case study the proposed method, which is about providing optimal data with the IoT slicing method. To demonstrate the efficiency of the IoT slicing technique the data quality algorithm will be used in this case study. In order to test the efficiency of this technique, the results obtained by an algorithm in the two scenarios will be compared. In one scenario, the IoT slicing technique will be used to treat the data collected by the IoT devices and in the other, the data collected by the IoT devices will be used directly. In this case study, temperature data collected by the IoT devices in a smart building will be used.

4.1. Data quality algorithm case study

The algorithm chosen to test the efficiency of the IoT slicing method is the data quality algorithm designed by Casado et al. [6]. IoT devices collect temperature data in a smart building. But we have no way of knowing if that data can be trusted to monitor and control the temperature of the smart building (e.g., an open window of a room may be in summer). To enhance reliability of temperature data collected for smart building monitoring and control, Casado et al. proposes a consensus algorithm based on game theory, so that neighboring IoT devices form coalitions based on their temperature and self-correct temperatures that are not similar to the temperature of the coalition. Therefore, one can define as ‘‘faulty data", data collected by the Iot devices that is adjusted by the data quality algorithm (e.g., Assume we have 4 IoT devices in a smart building that have the following temperatures: 20,20,20 and 23. If we apply the data quality algorithm, a consensus will be formed between the IoT devices that collect 20 whose temperature is 20, and correct the temperature collected by the IoT device whose temperature is 23. In this way, the IoT devices whose temperature is 23 are collecting faulty data). In order to illustrate how the algorithm works, a brief summary of this algorithm is presented in the next paragraphs.

This algorithm compares the neighborhood temperature of the sensors using a cooperative game based on game theory to detect errors in data and increase data quality gathered by the IoT nodes. In this algorithm, we want the neighborhood coalitions to democratically decide the temperature of the main sensor. To do this, they will form coalitions that will decide on the final temperature of the IoT node, which will be determined by whether they can vote or not in the process. From the characteristic function, if the value is 1(0), the coalition can vote (not vote) respectively. s_i is the main sensor with its associated temperature t_{s_i}, the characteristic function is built in the following way:

1. First, the average temperature of all the sensors is calculated:

$$T^k_{s_i} = \frac{1}{V}\sum_{i}^{V} t_{s_i} \tag{7}$$

here $T^1_{s_i}$ represents the average temperature of the sensors' neighborhood s_i (including it) in the first iteration of the game and V is the number of neighbors in the coalition.

2. The next step is to compute an absolute value for the temperature difference between the temperatures of each sensor and the average temperature:

$$\overline{T}^k_{s_i} = \left(\frac{1}{V}\sum_{i}^{V} \mid t_{s_i} - T^k_{s_i} \mid^2\right)^{\frac{1}{2}} \tag{8}$$

3. Using the differences in temperature values and the average temperature $\overline{T}_{s_i}^k$ (see Eq. (8)) a confidence interval is created and defined as follows:

$$I_{s_i}^k = \left(T_{s_i}^k \pm t_{(V-1,\frac{\alpha}{2})} \frac{\overline{T}_{s_i}^k}{\sqrt{V}} \right) \tag{9}$$

in Eq. (9) we use the Student's-t distribution with an error of 1%.

4. In this step we use a hypothesis test. If the temperature of the sensor lies within the interval $I_{s_i}^k$, it belongs to the voting coalition, otherwise, it is not in the voting coalition:

$$u^k(s_1, \dots, s_n) = \begin{cases} 1 & \text{if } t_{s_i} \in I_{s_i}^k \\ 0 & \text{if } t_{s_i} \notin I_{s_i}^k \end{cases} \tag{10}$$

5. The characteristic function will repeat this process iteratively (k is the number of iterations) until all the sensors in that iteration belong to the voting coalition. In each iteration k, the following payoff vector of the coalition S_j (with $1 \leq j \leq n$ where n is the number of sensors in the coalition) is available in step k ($PV(S_j^k)$):

$$PV(S_j^k) = (u^k(s_1), \dots, u^k(s_n)) \text{ where } \sum_i^n u^k(s_i) \leq n \tag{11}$$

The stop condition of the game iterations is $PV(S_j^k) = PV(S_j^{k+1})$ the process end. That is, let $PV(S_j^k) = (u^k(s_1), \dots, u^k(s_n))$ and let $PV(S_j^{k+1}) = (u^{k+1}(s_1), \dots, u^{k+1}(s_n))$. The iteration process ends when both payoff vectors contain the same elements. This process is represented by the following equation:

$$\begin{cases} u^k(s_1) = u^{k+1}(s_1) \\ \vdots \\ u^k(s_n) = u^{k+1}(s_n) \end{cases} \tag{12}$$

A detailed description of this algorithm can be found in [6].

5. Results

In this section we are going to compare the result of two case studies in which the temperature data quality algorithm was used in smart buildings: (1) Using the data quality algorithm for comparison purposes and (2) will use the model described in this paper and then the data quality algorithm. The data quality algorithm gets data collected by the IoT nodes and self-corrects the faulty data. Furthermore, in case the algorithm finds that an IoT node malfunctions, it will create a virtual temperature sensor in order to keep the reliability of the IoT network. In this way, the monitoring and control efficiency of the IoT network is improved.

5.1. General description of the case study

To test the proposed model, we have chosen a smart building. The chosen smart building is University of Salamanca's R&D building, located on Espejo Street, a reference center for the region of Castilla y Leon as well as nationally (Spain) and even internationally. Built between 2010 and 2014 with the latest materials and techniques for energy efficiency, and an investment of 25 million euros (including equipment), the building has more than 13,000 square meters distributed over six floors, three of them in the basement and another three in height and in which more than 250 people work. As the smart building is such a large building it is difficult to monitor and control the temperature of its interior throughout the year. For this reason, we would like to make the case for this research in this building. To monitor and control the temperature in the smart building, a mesh was placed into the IoT devices on the smart building with the help of laser levels, the IoT devices were placed vertically one in every section of the building. A total of 25 IoT nodes were deployed. The smart building where the case study was deployed is shown in Fig. 5. For this experimental case study it was decided that the number of clusters to be considered would be 4. This was motivated by the characteristics of smart building. The building has large open-plan zones, long corridors and two different types of rooms: (1) Rooms for development and computer research. (2) Rooms for biomedical research. These two types of rooms have different thermal needs, and we have considered for this case study, that each of these 4 types of groups to be considered are uniform and have the same topology for the purpose of applying the clustering technique to find IoT devices with similar temperatures and form homogeneous groups of IoT devices.

The type of sensor deployed in the building was a combination of the ESP8266 microcontroller in its commercial version "ESP-01" and a DHT22 temperature and humidity IoT node. The sum of both allows us for greater flexibility (e.g., by comparing the characteristics of this sensor with its previous model, the temperature measurement range of the DHT11 sensor is from 0° to 50° Celsius and the accuracy is +/-1° Celsius.) when collecting data and adaptability to the case study, since the DHT22 sensor is designed for indoor spaces (it has an operating range of 0 ° C to 50 ° C) according to its datasheet. The microcontroller obtains data from this sensor through the onewire protocol and communicates it to the environment via Wi-Fi using HTTP standards and GET/POST requests. The ESP-IDF programming environment provided by the manufacturer of the microcontroller, was used to program the device.

The temperature IoT device had been collecting data at 5 min intervals, for 6 h in the same day. For the analysis we selected the data collected by the sensors in the following time interval 2018-12-10T08:30:00Z and ended on 2018-12-10T14:30:00Z. To test the efficiency of the temperature algorithm a disturbance has been introduced in the temperature of smart building (our process) at 1 h intervals to simulate the random behavior of people's thermostat use (i.e., a group of people could select different temperatures in their office thermostats). These disturbances have been introduced by in the members of our research group who had no consensus on which temperatures to introduce, so these temperatures can be considered pseudo-random. Below, a statistical summary of the measurements collected by the IoT devices is presented in Table 1.

5.2. Case study results and discussions

The first thing we do is build the graph of the IoT network and apply the clustering algorithm. In Fig. 6 one can find both graphs. After applying the clustering algorithm, it can be observed how 4 clusters are formed. Therefore, the multiplex in Fig. 7 has got 4 layers. Thus, data is homogeneous. In this case study, non-connected graphs are formed on all layers of the multiplex using the data collected from the smart building. This is a common situation, and the algorithm self-corrects this by virtualizing all the required nodes for the graphs to be connected.

The results obtained from the comparison of the use of the control algorithm are presented in Fig. 8. In this figure you can find the result of the application of the data quality algorithm on heterogeneous data. In this case the data quality algorithm smoothes the data collected by the IoT network so that the data quality algorithm understood that the heterogeneous data were homogeneous. For this reason, the algorithm detects some

Table 1
Statistical table of measurements of the IoT nodes.

Timestamp start	Total timestamp	Min temp	Max temp	Mean	Standard deviation
2019-28-01T09:00	06:00:00Z	20.4 °C	24.7 °C	22.8 °C	0.87 °C

Fig. 5. Map of the case study building.

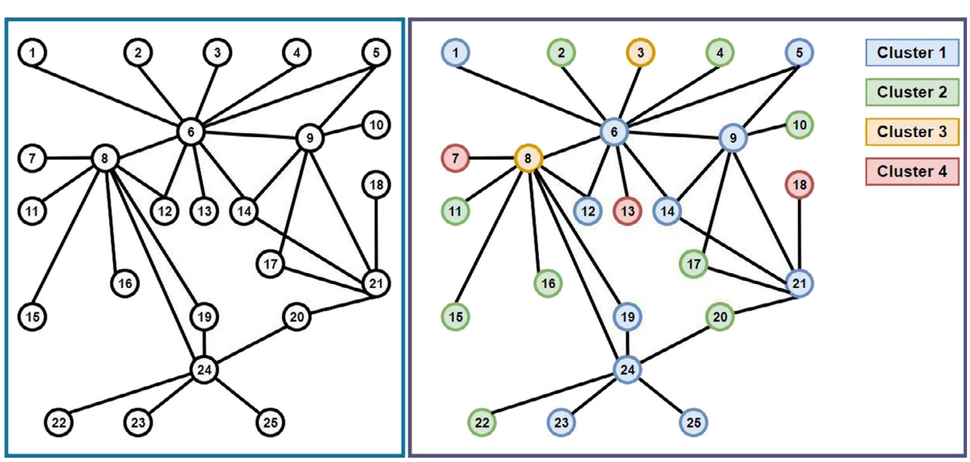

Fig. 6. Graph of the IoT nodes in the smart building and colored graph by clusters.

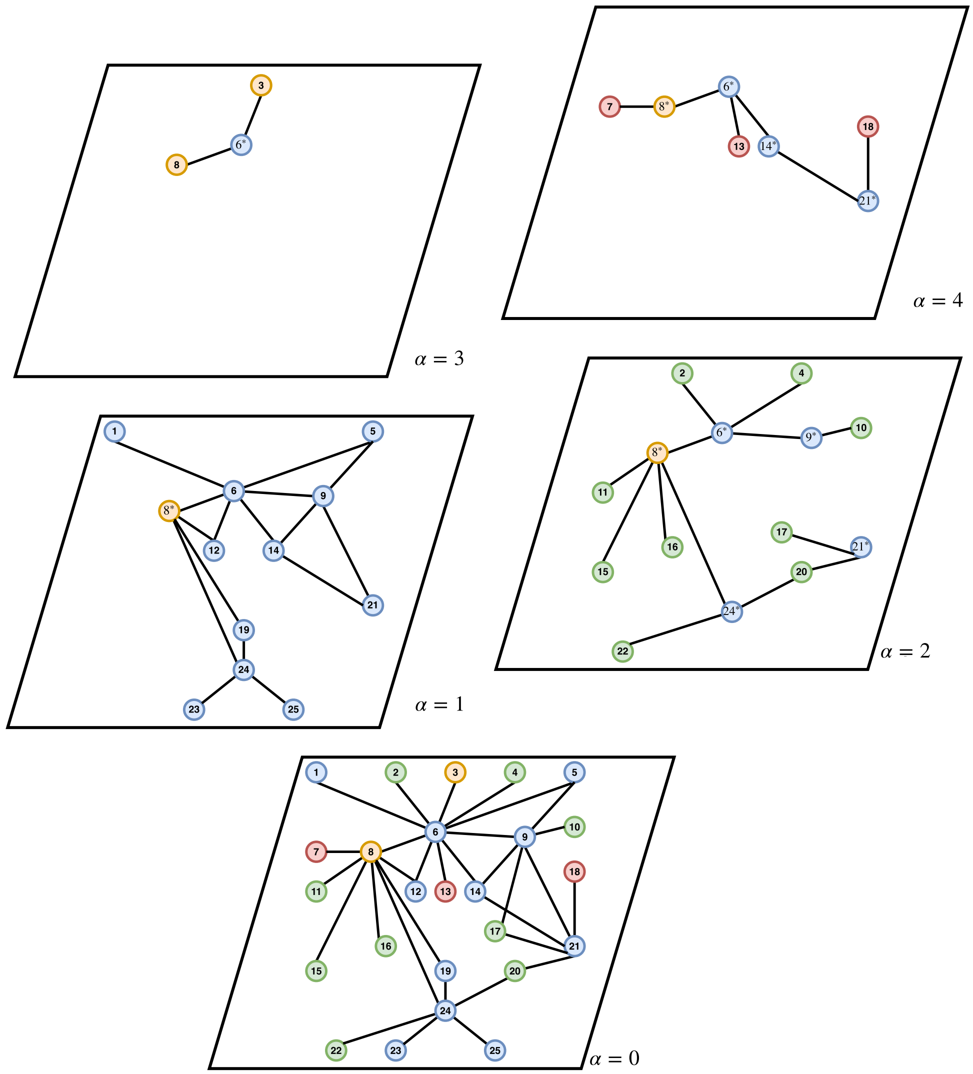

Fig. 7. Multiplex with the 4 layers. In every layer we have to virtualize some IoT nodes to the graph will be connected.

Table 2
Final temperature in the case study in both experiments.

Final temperature without slicing	Final temperature with slicing
$\begin{bmatrix} 22.98 & 23.07 & 23.19 & 23.22 & 23.17 \\ 22.90 & 23.08 & 23.21 & 23.19 & 23.13 \\ 22.67 & 22.84 & 23.01 & 23.04 & 23.03 \\ 22.69 & 22.77 & 22.76 & 22.70 & 22.77 \\ 22.78 & 22.81 & 22.78 & 22.70 & 22.67 \end{bmatrix}$	$\begin{bmatrix} 22.92 & 23.05 & 23.21 & 23.24 & 23.14 \\ 22.93 & 23.11 & 23.21 & 23.29 & 23.14 \\ 22.48 & 22.42 & 22.98 & 23.21 & 23.03 \\ 22.64 & 22.74 & 22.71 & 22.61 & 22.69 \\ 22.81 & 22.88 & 22.86 & 22.71 & 22.62 \end{bmatrix}$

temperatures collected by the IoT network and considers them outliers. Then the self-correcting algorithm corrects these values and the output of the data quality algorithm are homogeneous temperature data but this would not increase the energy efficiency as it does not use the information from the clusters to distinguish this information. On the other hand, using the technique proposed in this paper, through which heterogeneous data are transformed into homogeneous data with clustering techniques and complex networks, the energy efficiency of the IoT network is increased since the data collected by the IoT network by applying the control algorithm only acts for the homogeneous areas and thus maintains the temperature clusters as output of the data quality algorithm.

In Fig. 9, temperature isoterm maps are presented from a temperature point of view. In these isotherm maps one can see the differences between the two experimental results carried out using the IoT slicing technique and not using it. This figure shows a comparison between the two results and the temperatures collected by the IoT devices. In both experiments, the algorithm we have used to test the efficiency of our new technique removes the outliers, but in the experiment that the IoT slicing smooth technique is not used the temperatures without considering the topology and uniformity of the smart building. Meanwhile, using the IoT slicing technique the smart building characteristics are considered, and when one uses data quality algorithm one obtains an output similar to data collected by the IoT devices but without wrong data (i.e., outliers). On a quantitative point of view, the final temperatures of the experiments done in the case study to validate the efficiency of the IoT slicing technique are presented in Table 2.

The evaluation results have shown that the proposed system can transform heterogeneous data into homogeneous data taking

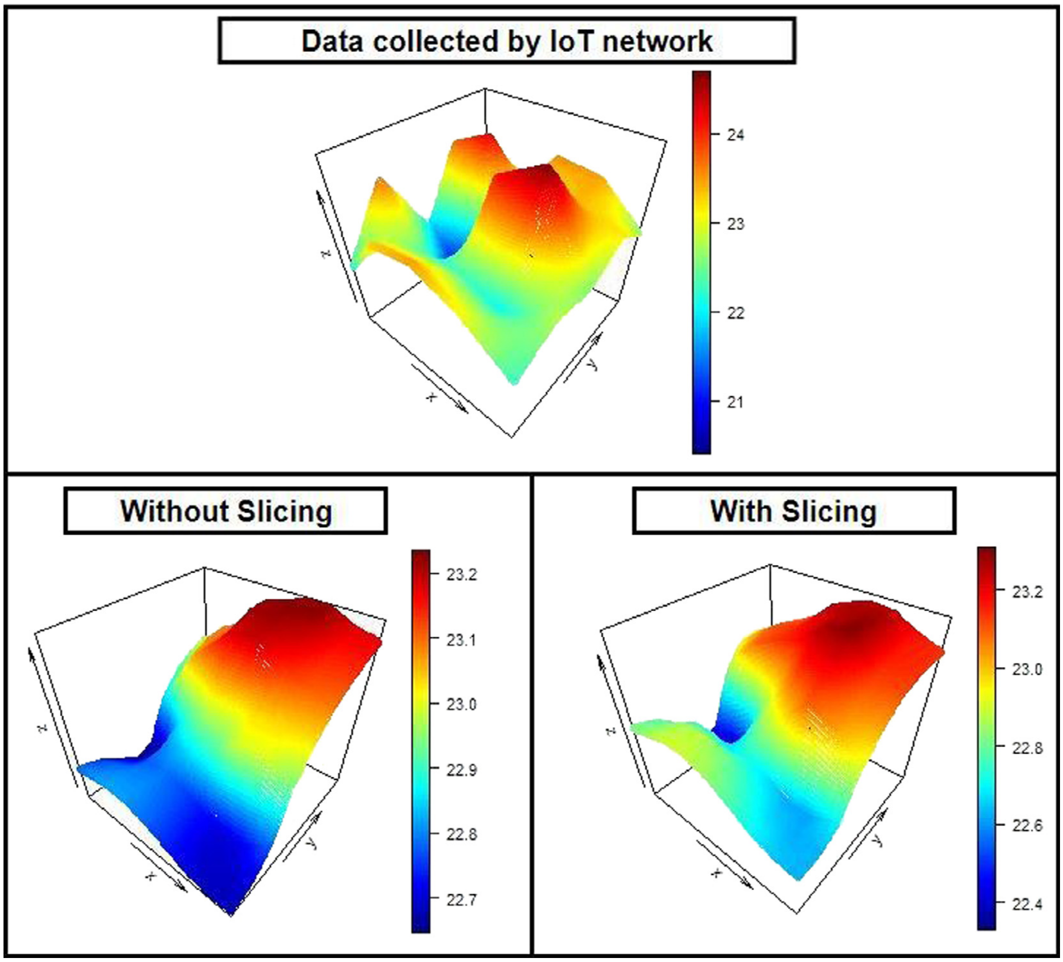

Fig. 8. The results obtained by the data quality algorithm are compared using the technique proposed in this article and without using the technique.

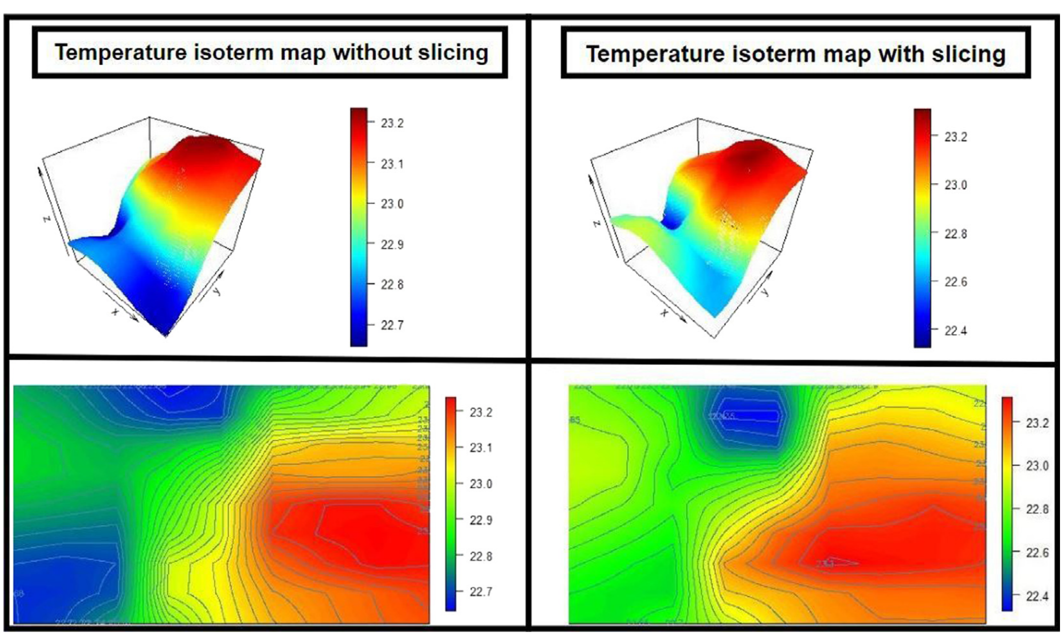

Fig. 9. The results obtained by the data quality algorithm are compared using the technique proposed in this paper and without using the technique.

into account the topology and lack of uniformity of the intelligent building. The problem with the data collected by the IoT network is that the topology and non-uniformity of smart buildings are not considered. In this way, temperature control algorithms can be applied to these smart buildings, and the algorithm will smooth the entire temperature of the building considering only the comfort temperatures selected by the users (with thermostats). With the new IoT slicing technique proposed in this paper, the topology and non-uniformity of smart buildings are taken into account in order to transform heterogeneous data into homogeneous data. This implies a considerable increase in the functioning of the algorithms that use the data collected by IoT devices. In this paper, the effectiveness of this new technique is tested with a data quality algorithm developed by our research group with better results than expected in the research project.

6. Conclusions

This paper has investigated the inaccuracy problem of IoT network algorithms using heterogeneous input data. Through the introduction of a complex network and clustering techniques, these heterogeneous data can be virtualized into segmented virtual layers considering the clusters in order to transform heterogeneous data into homogeneous data that optimizes the operation of the algorithms. Using the virtual segmentation technique provided by our new method, the algorithms guaranteeing an optimized performance considering the different areas of the topology of the IoT network. Finally, a case study result is given to demonstrate the efficacy of the proposed IoT slicing method. Future work will be concentrated on application of this novel method to several IoT control algorithms with heterogeneous data input and the increase of IoT devices that are used. Also, we will test the efficiency of our technique with other algorithms, for example, temperature control algorithms.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.future.2019.09.042.

Acknowledgments

This paper has been partially supported by the Salamanca Ciudad de Cultura y Saberes Foundation under the Talent Attraction Programme (CHROMOSOME project).

References

[1] B. Lapillonne, C. Sebi, K. Pollier, N. Mairet, Energy efficiency trends in buildings in the EU—lessons from the ODYSSEE MURE project, ADEME. Retrieved December 30 (2012), 2014.

[2] E. Efficiency, Buildings energy data book, US Department of Energy. http://buildingsdatabook.eere.energy.gov/. John Dieckmann is a director and Alissa Cooperman is a technologist in the Mechanical Systems Group of TIAX, Cambridge, Mass. James Brodrick, Ph. D., is a project manager with the Building Technologies Program, US Department of Energy, Washington, DC.

[3] F. Terroso-Saenz, A. González-Vidal, A.P. Ramallo-González, A.F. Skarmeta, An open IoT platform for the management and analysis of energy data, Future Gener. Comput. Syst. 92 (2019) 1066–1079.

[4] D. Coley, T. Kershaw, M. Eames, A comparison of structural and behavioural adaptations to future proofing buildings against higher temperatures, Build. Environ. 55 (2012) 159–166.

[5] M. Vellei, S. Natarajan, B. Biri, J. Padget, I. Walker, The effect of real-time context-aware feedback on occupants' heating behaviour and thermal adaptation, Energy Build. 123 (2016) 179–191.

[6] R. Casado-Vara, F. Prieto-Castrillo, J.M. Corchado, A game theory approach for cooperative control to improve data quality and false data detection in WSN, Int. J. Robust Nonlinear Control (2018) http://dx.doi.org/10.1002/rnc.4306.

[7] A. Kaufmann, Introduction à la théorie des sous-ensembles flous à l'usage des ingénieurs: Éléments théoriques de base, vol. 1, Masson, 1973.

[8] L.A. Zadeh, Similarity relations and fuzzy orderings, Inform. Sci. 3 (2) (1971) 177–200.

[9] P. Mahé, N. Ueda, T. Akutsu, J.-L. Perret, J.-P. Vert, Graph kernels for molecular structure- activity relationship analysis with support vector machines, J. Chem. Inf. Model. 45 (4) (2005) 939–951.

[10] R.M. Marin, N.F. Aguirre, E.E. Daza, Graph theoretical similarity approach to compare molecular electrostatic potentials, J. Chem. Inf. Model. 48 (1) (2008) 109–118.

[11] J. Beasley, N. Christofides, Vehicle routing with a sparse feasibility graph, European J. Oper. Res. 98 (3) (1997) 499–511.

[12] D.J. Watts, P.S. Dodds, M.E. Newman, Identity and search in social networks, Science 296 (5571) (2002) 1302–1305.

[13] X. Chen, H. Huo, J. Huan, J.S. Vitter, MSQ-Index: a succinct index for fast graph similarity search, 2016, arXiv preprint arXiv:1612.09155.

[14] X. Zhao, C. Xiao, X. Lin, W. Wang, Efficient graph similarity joins with edit distance constraints, in: Data Engineering, ICDE, 2012 IEEE 28th International Conference on, IEEE, 2012, pp. 834–845.

[15] D. Conte, P. Foggia, C. Sansone, M. Vento, Thirty years of graph matching in pattern recognition, Int. J. Pattern Recognit. Artif. Intell. 18 (03) (2004) 265–298.

[16] X. Yan, J. Han, Gspan: graph-based substructure pattern mining, in: Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on, IEEE, 2002, pp. 721–724.

[17] J. Xuan, J. Lu, G. Zhang, X. Luo, Topic model for graph mining., IEEE Trans. Cybern. 45 (12) (2015) 2792–2803.

[18] E. Sampathkumar, Generalized graph structures, Bull. Kerala Math. Assoc. 3 (2) (2006) 65–123.

[19] J. Bollen, H. Van de Sompel, A. Hagberg, R. Chute, A principal component analysis of 39 scientific impact measures, PLoS One 4 (6) (2009) e6022.

[20] S.A. Golder, M.W. Macy, Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures, Science 333 (6051) (2011) 1878–1881.

[21] I.H. Witten, E. Frank, M.A. Hall, C.J. Pal, Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann, 2016.

[22] A.L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artif. Intell. 97 (1–2) (1997) 245–271.

[23] Y. Wang, Y. Fan, P. Bhatt, C. Davatzikos, High-dimensional pattern regression using machine learning: from medical images to continuous clinical variables, Neuroimage 50 (4) (2010) 1519–1535.

[24] T. Kinnunen, I. Sidoroff, M. Tuononen, P. Fränti, Comparison of clustering methods: a case study of text-independent speaker modeling, Pattern Recognit. Lett. 32 (13) (2011) 1604–1617.

[25] M.C. de Souto, I.G. Costa, D.S. de Araujo, T.B. Ludermir, A. Schliep, Clustering cancer gene expression data: a comparative study, BMC Bioinf. 9 (1) (2008) 497.

[26] I.G. Costa, F.d.A. de Carvalho, M.C. de Souto, Comparative analysis of clustering methods for gene expression time course data, Genet. Mol. Biol. 27 (4) (2004) 623–631.

[27] S. Guha, R. Rastogi, K. Shim, CURE: an efficient clustering algorithm for large databases, in: ACM Sigmod Record, vol. 27, ACM, 1998, pp. 73–84.

[28] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications, vol. 27, ACM, 1998.

[29] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, S.Y. Philip, et al., Top 10 algorithms in data mining, Knowl. Inform. Syst. 14 (1) (2008) 1–37.

[30] A.K. Jain, A. Topchy, M.H. Law, J.M. Buhmann, Landscape of clustering algorithms, in: Null, IEEE, 2004, pp. 260–263.

[31] C. Fraley, A.E. Raftery, How many clusters? which clustering method? answers via model-based cluster analysis, Comput. J. 41 (8) (1998) 578–588.

[32] X.-M. Zhang, Q.-L. Han, X. Yu, Survey on recent advances in networked control systems, IEEE Trans. Ind. Inf. 12 (5) (2016) 1740–1752.

[33] X. Ge, F. Yang, Q.-L. Han, Distributed networked control systems: a brief overview, Inform. Sci. 380 (2017) 117–131.

[34] J. Qiu, H. Gao, M.-Y. Chow, Networked control and industrial applications [special section introduction], IEEE Trans. Ind. Electron. 63 (2) (2016) 1203–1206.

[35] J.P. Hespanha, P. Naghshtabrizi, Y. Xu, A survey of recent results in networked control systems, Proc. IEEE 95 (1) (2007) 138–162.

[36] W. Sun, Z. Zeng, C. Luo, S.-K. Nguang, Robust-gain codesign of networked control systems, Internat. J. Robust Nonlinear Control (2018) 1–15.

[37] K. Liu, A. Seuret, E. Fridman, Y. Xia, Improved stability conditions for discrete-time systems under dynamic network protocols, Internat. J. Robust Nonlinear Control (2018) 1–21.

[38] R. Casado-Vara, Z. Vale, J. Prieto, J. Corchado, Fault-tolerant temperature control algorithm for IoT networks in smart buildings, Energies 11 (12) (2018) 3430.

[39] C. Tan, L. Li, H. Zhang, Stabilization of networked control systems with both network-induced delay and packet dropout, Automatica 59 (2015) 194–199.

[40] H. Xu, J. Ding, P. Li, D. Sgandurra, R. Wang, An improved SMURF scheme for cleaning RFID data, Int. J. Grid Util. Comput. 9 (2) (2018) 170–178.

[41] Z. Zhou, H. Zhang, S. Li, X. Du, Hermes: a privacy-preserving approximate search framework for big data, IEEE Access 6 (2018) 20009–20020.
[42] Y. Cao, H. Qi, W. Zhou, J. Kato, K. Li, X. Liu, J. Gui, Binary hashing for approximate nearest neighbor search on big data: a survey, IEEE Access 6 (2018) 2039–2054.
[43] C.M. Bishop, Pattern Recognition and Machine Learning, in: Information Science and Statistics, vol. 60 (1), 2006,
[44] F. Downton, The reliability of multiplex systems with repair, J. R. Stat. Soc. Ser. B Stat. Methodol. 28 (3) (1966) 459–476.
[45] L. Solá, M. Romance, R. Criado, J. Flores, A. García del Amo, S. Boccaletti, Eigenvector centrality of nodes in multiplex networks, Chaos 23 (3) (2013) 033131.

Roberto Casado Vara earned a degree in Mathematics from the University of Salamanca, a Master in Big Data & Visual Analytics at the International University of la Rioja. He has worked at Viewnext as a data scientist and Powercenter consultant for important clients in the pharmaceutical and public administration sectors.

He is currently working on his Ph.D. in Computer Engineering as a member of the BISITE research group. As a researcher, his interests are focused on Deep Learning, advanced mathematical models for intelligent robust and non-linear control and monitoring, blockchain and knowledge discovery data as well as other fields.

Angel Martin del Rey is an applied mathematician at the Institute of Fundamental Physics and Mathematics (Spain). He has nearly of 20 years of experience designing mathematical models to simulate different phenomena. Angel holds a B.S. in Mathematics from University of Salamanca (Spain) and a Ph.D. in Mathematics from the UNED, Spain. Nowadays he is associated professor at the department of Applied Mathematics (University of Salamanca).

Soffiene Affes received the Diplôme d'Ingénieur in telecommunications and the Ph.D. degree (Hons.) in signal processing from the École Nationale Supérieure des Télécommunications, Paris, France, in 1992 and 1995, respectively. He was a Research Associate with INRS, Montreal, QC, Canada, until 1997, an Assistant Professor until 2000, and an Associate Professor until 2009. He is a currently a Full Professor and the Director of PERWADE, a unique U.S. $4 million research training program on wireless in Canada involving 27 faculty from 8 universities and 10 industrial partners. He has been twice a recipient of a Discovery Accelerator Supplement Award from NSERC, from 2008 to 2011 and from 2013 to 2016. In 2008 and 2015, he received the IEEE VTC Chair Recognition Award from the IEEE VTS and the IEEE ICUWB Chair Recognition Certificate from the IEEE MTT-S for exemplary contributions to the success of the IEEE VTC and the IEEE ICUWB, respectively. In 2006 and 2015, he served as a General Co-Chair of the IEEE VTC'2006-Fall and the IEEE ICUWB'2015, respectively, both held in Montreal, QC, Canada. From 2003 to 2013, he was a Canada Research Chair in wireless communications. He will act as a General Chair of the IEEE PIMRC'2017 to be held in Montreal, QC, Canada. He was an Associate Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2007 to 2013 and the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2010 to 2014. He is currently an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS and the Wireless Communications and Mobile Computing journal (Wiley).

Javier Prieto earned the Ph.D. in Information and Communication Technologies and the Extraordinary Performance Award for Doctorate Studies from the University of Valladolid in 2012. At this University, he also received the Telecommunication Engineer degree (2008) and the Marketing Research and Techniques degree (2010). Since 2007, he has worked in different public and private research centers, such as the University of Valladolid (Spain) and the Massachusetts Institute of Technology. Currently, he is a member and an assistant lecturer of the University of Salamanca.

He has published more than 50 papers in international journals, books and conferences, participated in more than 35 research projects, and been author of 2 Spanish patents. Moreover, he serves as an associate editor for the IEEE Communications Letters and for the Wireless Communications and Mobile Computing journal, as a guest editor for the International Journal of Distributed Sensor Networks and as a reviewer for numerous international journals. He is member of the scientific committee of the Advances in Distributed Computing and Artificial Intelligence Journal, member of the technical program committee of prestigious international conferences (such as IEEE GLOBECOM or IEEE ICC), and reviewer in a large number of them. In 2013, he received the award "Young entrepreneurs of Castilla y Le n" from the Youth Institute of Castilla y Le n (Spain).

His research interests include social computing and artificial intelligence, for developing smarter and safer cities, localization and navigation technologies, for indoor and outdoor environments, and Bayesian inference techniques, for improving social welfare and environmental sustainability.

Juan Manuel Corchado (born May 15, 1971 in Salamanca, Spain). He is Full Professor with Chair at the University of Salamanca. He was **Vice President for Research and Technology Transfer** from December 2013 to December 2017 and the **Director of the Science Park** of the University of Salamanca, Director of the Doctoral School of the University until December 2107 and also, he has been elected twice as the Dean of the Faculty of Science at the University of Salamanca. In addition to a PhD in Computer Sciences from the University of Salamanca, he holds a Ph.D. in Artificial Intelligence from the University of the West of Scotland. Juan Manuel Corchado is **Visiting Professor** at Osaka Institute of Technology since January 2015, Visiting Professor at the University Teknologi Malaysia since January 2017 and a Member of the Advisory group on Online Terrorist Propaganda of the European Counter Terrorism Centre (EUROPOL).

Corchado is the Director of the **BISITE (Bioinformatics, Intelligent Systems and Educational Technology)** Research Group, which he created in the year 2000, President of the IEEE Systems, Man and Cybernetics Spanish Chapter, Academic Director of the Institute of Digital Art and Animation of the University of Salamanca. He also oversees the Master's programs in Digital Animation, Security, Mobile Technology, Community Management and Management for TIC enterprises at the University of Salamanca.

Corchado is also editor and Editor-in-Chief of Specialized Journals like ADCAIJ (Advances in Distributed Computing and Artificial Intelligence Journal), IJDCA (International Journal of Digital Contents and Applications) and OJCST (Oriental Journal of Computer Science and Technology).