Computer Communications 152 (2020) 323–332

Contents lists available at ScienceDirect

Computer Communications

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

Smart city as a distributed platform: Toward a system for citizen-oriented management☆

Pablo Chamoso [a,*], Alfonso González-Briones [a,b], Fernando De La Prieta [a], Ganesh Kumar Venyagamoorthy [c], Juan M. Corchado [a,b]

[a] BISITE Research Group, University of Salamanca. Calle Espejo s/n. Edificio Multiusos I+D+i, 37007, Salamanca (Salamanca), Spain
[b] Air Institute, IoT Digital Innovation Hub, Calle Segunda 4, 37188, Salamanca, Spain
[c] Real-Time Power and Intelligent Systems Laboratory, Holcombe Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, 29634, USA

ARTICLE INFO

Keywords:
Smart cities
Internet of things
Distributed systems

ABSTRACT

Today's society must prioritize the design and development of platforms for Big Data processing. Smart cities generate large volumes of valuable data which the government can use to manage cities more intelligently. Aware of the value of data, many researchers have proposed architectures for optimized data collection and use. This paper proposes a novel approach that enables smart cities to reuse the functionalities of old applications by adapting them to new architectures. To be able to process large amounts of data and solve a variety of problems, smart city platforms need extra computing power. Two case studies have been conducted to verify the performance of the proposed platform. Those case studies have demonstrated that the development process of a smart platform can be simplified by implementing functionalities and components taken from earlier platforms.

1. Introduction

Smart cities promote a sustainable lifestyle. Infrastructure, innovation and technology are the components that make smart cities efficient and self-sufficient. Examples of sustainability measures include the use of photovoltaic panels to provide electricity to buildings, street signs or electric vehicles. In addition, wind turbines are used to power street lights and the use of bicycles is encouraged as a means of moving around the city etc. The degree of sustainability of smart cities is measured according to a series of parameters, including: public management, social cohesion, governance, technology, urban planning, environment, mobility and transport, international projection, human capital and economy. Smart cities have got a huge variety of applications to monitor and control each of those parameters [1].

However, one of the barriers to efficiency is that different companies use different smart city applications to offer services to citizens. This can cause communication problems and can therefore hinder efficiency. Moreover, demand for new services is continuous in smart cities, so companies that wish to provide new services must make new developments instead of reusing and customizing the old ones. In addition, current smart city applications encounter major scalability problems due to the large amount of data they must process [2].

The problems described above evidence the importance of the solution proposed in this paper. The developed platform has a modular design which facilitates the efficient and transparent use of urban resources. Each module is responsible for the management and monitoring of a different element of a smart city. The platform interacts with both human beings and computers; it collects data autonomously and obtains data from users e.g. human interaction on social networks or third party data sources. Moreover, all the technology that has been used for its development is environment-friendly.

This platform has been designed to foster efficient habits among users through persuasive recommendations and collective challenges, leading to considerable changes in the behavior of citizens and fostering efficiency in cities. Thanks to exhaustive data analyses, the platform may identify the specific needs of the different areas of a city, recommending the implementation of specific measures.

The platform is Cloud-based and it has a Big Data infrastructure that makes it capable of processing heterogeneous smart-city data

☆ This research has been partially supported by the European Regional Development Fund (ERDF) within the framework of the Interreg program V-A Spain–Portugal 2014–2020 (PocTep) under the IOTEC project grant 0123_IOTEC_3_E and by the Spanish Ministry of Economy, Industry and Competitiveness and the European Social Fund under the ECOCASA project grant RTC-2016-5250-6.
* Corresponding author.
E-mail addresses: chamoso@usal.es (P. Chamoso), alfonsogb@usal.es (A. González-Briones), fer@usal.es (F. De La Prieta), gvenaya@clemson.edu (G.K. Venyagamoorthy), corchado@usal.es (J.M. Corchado).

https://doi.org/10.1016/j.comcom.2020.01.059
Received 30 December 2019; Received in revised form 23 January 2020; Accepted 27 January 2020
Available online 31 January 2020
0140-3664/© 2020 Elsevier B.V. All rights reserved.

coming from multiple sources. The data that flows into the platform is processed and analyzed in order to generate personalized recommendations and motivate citizens to develop efficient and environmentally friendly habits.

In addition, the platform must overcome a series of problems associated with the large amount of heterogeneous data it collects. Smart city monitoring and control platforms experience the following problems:

- Scalability; the needs and characteristics of a smart city change continually, existing platforms lack the ability to effectively adapt to those changes. Monitoring and control platform must operate in a dynamic environment, thus, the technology they use must be scalable.
- Reuse; the process of developing a new platform from scratch is very time consuming and complex. Greater development efficiency must be achieved through the reuse and customization of old functionalities in new platform.

This research presents a novel platform that reuses software in order to simplify the application development process. It has the capacity to process the large volume of data generated by smart cities. In addition, its features respond to the needs of today's smart cities; the platform is scalable which means that it can adapt to the Internet of Things (IoT) or Industry 4.0 paradigms.

The platform has been designed under the paradigm of social computing and is based on Kafka, a distributed streaming platform. Kafka Connect and Apache Arrow enable communication between applications, scalability and the reuse of software.

The purpose of the conducted case study has been to test the performance of the proposed platform. A smart city model has been employed to simulate the different situations and conditions that would normally occur in a real smart city. This scenario addressed the problem of energy optimization in the smart city. The platform has been used to manage and analyze the data of the city, with the aim of optimizing energy consumption.

The case study made it possible to assess whether there is a change in the citizens' energy consumption behavior and whether personalized, persuasive recommendations and collective challenges help maintain energy efficient habits. Automatic learning techniques have been applied to heterogeneous data (smart sensors, smart counters, social and public data) in order to generate recommendations and challenges. Moreover, recommendations are personalized through the segmentation of the target population according to gender, socio-economic, demographic or cultural differences and through a sociological analysis of the factors influencing consumer choices.

Furthermore, to test the platform in a real environment, the platform has been connected to an intelligent control and automation system deployed in homes/buildings. The platform obtained data from the system in order to monitor consumption in real time and establish more efficient schedules while maintaining a level of comfort.

A second case study has been conducted to show that the platform can integrate multiple data sources of different nature; the data do not necessarily have to come from sensors. In this case study, the platform has used information retrieved from the Internet (social networks or websites obtained in search results) to create profiles of real citizens. It has also been demonstrated that the system can accept requests that involve large volumes of information and take longer to process.

The main contribution can be summarized as follows:

1. The platform transforms the current understanding of client/server communications. Its database has been converted into an interface; this means that it is not necessary to have any knowledge about the operation or programming of network communications when establishing connections between various applications, and only a basic understanding of DB is required.
2. The abstraction layer allows to execute scripts for the reuse of modules.
3. Apache Arrow processes data and stores them in binary formats, facilitating faster data transfer.
4. The platform is symmetrical, this means that scalability can be achieved easily, making it suitable for any project.

The paper is organized as follows. Section 2 reviews related literature. The proposed architecture is presented in Section 3. Then, Section 4 describes the two case studies that have been conducted to validate the proposed architecture in both simulated and real scenarios. Finally, the conclusions drawn from the results are presented in Section 5.

2. Related work

This section focuses on the current status of information management platforms in smart cities. First, we review the current state of the smart city concept and the work that is being carried out in this area, then we describe the main smart city management platforms, and finally we identify the shortcomings of the platforms and propose possible solutions to be adopted in our proposal.

2.1. Smart city as the center of technological innovation

In recent years, smart homes have been a broadly discussed topic in politics. ICT (Information and Communication Technology) infrastructure has received considerable attention, although much of the research has also focused on the role of human capital/education, social and relational capital and the environment; they have all been viewed as important drivers of urban growth [3]. The European Union (EU) has made a considerable effort to develop a strategy for 'smart' urban growth in its metropolitan areas. The Intelligent Community Forum conducts research on the local effects of ICTs which are now available worldwide. The OECD (Oslo Manual and Eurostat, 2005) highlights the role of innovation in the ICT sector and provides a set of tools for determining the coherence of innovation policies to shape a sound framework of analysis for researchers on urban innovation [4]. At the regional level, there has been renewed interest in soft communication infrastructures and their effect on economic performance. There are several applications that researchers are developing for a wide range of fields, wireless sensors networks in smart cities [5,6] or in agriculture [7,8], energy optimization [9–12], optimal resource allocation [13], risks and challenges of adopting electric vehicles in smart cities [14], vehicular networks [15,16] and route optimization [17–19].

IoT is the very backbone of smart cities [20,21]. IoT uses sensors to collect and store data in the cloud for easy monitoring and control of smart cities [22]. The city's data can be used to effectively transform and manage the city, and promote urban planning and development. To the best of our knowledge, Big Data does not have a proper industrial definition, but a survey shows that the key features of big data include high volume, high velocity, and high variety.

A key feature of a Smart City architecture is its ability to acquire and process data from multiple data sources (databases, crawlers, third-party apps, sensors). Architectures require extra computing power to process large volumes of data. The world of embedded systems has changed dramatically over the last decade and its evolution is unlikely to slow down in the years to come. Multicore processing (in the form of symmetric multiprocessing (SMP) and asymmetric multiprocessing (AMP)) is becoming common, with embedded multicore CPUs expected to grow by a factor of x^6 in the next years (Venture Development Corporation). In addition, Field Programmable Gate Arrays (FPGAs) have grown in capacity and decreased in cost, providing the high-speed functionality that could only be achieved with Application Specific Integrated Circuits (ASICs) [23]. Finally, virtualization is blurring the connection between hardware and software by allowing multiple operating systems to run on a single processor.

Through the aggregation of extra computing power we obtain an architecture with high-performance computing, capable of solving complex problems. To this end, the designed architecture relies on a distributed computing model which employs technologies such as clusters or agent-based parallel computing. The majority of current distributed computing proposals involve high-performance computing.

2.2. Smart city platforms

Nowadays there are multiple platforms whose functionalities promote the IoT paradigm. These functionalities range from sensor networks to information analysis, which involves information transmission, information fusion, pre-processing, etc. The development of IoT platforms is a trend in the IT sector which has resulted in the creation of numerous private/open and general/specific platforms [24–26]. The most popular IoT platforms are described below.

- FIWARE is a platform provides a series of APIs for the development and deployment of Internet applications oriented toward multiple vertical sectors. Many of those sectors are responsible for offering numerous SC services.
- Carriots is a PaaS-type platform designated for IoT and M2M. It can be used to connect the information-providing infrastructure to a SC. However, the platform remains at this level, without offering user-oriented services, a layer that would have to be created independently of the platform.
- Kaa is an initiative that defines an open and efficient cloud platform to provide IoT solutions. Among the most common solutions is the connection of all types of sensors that can be found or deployed in a SC.
- Sofia2 is a middleware that enables interoperability between multiple systems and devices, offering a semantic platform that makes real-world information available to smart, mainly IoT-oriented applications.
- Webinos is a web application platform that allows developers to access native resources through APIs. Webinos makes it easy to connect any IoT device.
- ICOS system is an open repository of solutions for SCs, offering a set of existing applications and projects that can be reused for application creation.

2.3. Deficiencies detected in current smart city platforms

Although a lot of research is being done on Smart Cities, there is still a need for a compact system that would be more efficient and scalable. None of the existing Smart City platforms perform a comprehensive Big Data analysis that would consider the wide range of aspects of a city. Thus, we have found a research gap in the literature and propose a novel architecture capable of reusing the functionalities of existing applications. This enables different applications to offer the same functionalities within the same city or in other cities. As a result, the development process is simplified and less time is required to launch a new service and provide it to citizens.

As has been observed in the detailed study of the platforms for application development and implementation, not all platforms provide user-oriented services so that these applications can be used on different platforms. Similarly, the proposal stands out from the rest of the platforms because it allows developers to use the technologies they know best or those best adapted to the problem they want to solve. So that these technologies can be integrated transparently in the platform.

As has been observed in the detailed study of the platforms for application development and implementation, not all platforms are capable of providing user-oriented services. Similarly, the proposal stands out from the rest of the platforms because it allows developers to use the technologies they know best or those best adapted to the problem they want to solve. So that these technologies can be integrated transparently in the platform. The other platforms often do not allow to work in a distributed and modular way. Similarly, the scalability and calculation capacity are also deficiencies that the proposal covers.

In addition, one of the main characteristics of Smart Cities is the heterogeneity of all its components [27], both in the final applications and in the technology used in the deployed infrastructure. For example, within the IoT sector, there are many manufacturers, protocols and communication technologies [28,29]. As detailed below, one of the main advantages of the platform presented in this paper is that it is compatible with any technology or manufacturer.

To achieve this objective, high-performance computing (HPC) is supported by computing technologies such as clusters, supercomputers or parallel computing. The majority of current proposals on distributed computing are based on high-performance computing.

3. Proposed architecture

The main novelty of the article lies in the proposed platform (a scheme is shown in Fig. 1), which allows any city to offer a series of applications as services in a simple way. The citizens and the administration personnel of each city are the direct users of these applications.

The platform consists of two distinct parts, the communication architecture and the deployment architecture. These two parts are described in-depth in separate subsections. Firstly, we focus on the communication architecture, which makes it possible for simple nodes to process information in a distributed way. Furthermore, we describe how the central system enables communication between the different elements that form part of the smart city, ensuring adaptation to the size and computing needs of the elements that communicate. The second subsection focuses on the deployment architecture.

3.1. Communication architecture

The development of applications is simplified thanks to the possibility of reusing previously developed components without having to understand their implementation, technology or communication requirements (there are no APIs to be called). In order to reuse the functionality of any previously developed module or to create a new module with a previously developed functionality, it is only necessary to perform a data dump of the application database (of any type: MongoDB, MariaDB, etc.). Then, an answer is generated by the corresponding component and is delivered to the database. It is necessary to perform an initial configuration by introducing some simple parameters in the platform configuration scripts. The components that execute the requests of each application in the platform are based on a mechanism called Kafka, Kafka Connect and Apache Arrow. The architecture also integrates a rules engine (JESS). JESS facilitates the definition of rules which are executed once the conditions defined by the users are met. These tools allow to send formatted data in real-time.

The information is exchanged with the developer in a transparent manner by means of Kafka topics. More specifically, a topic is used to send the input information to each of the developed modules (Development input topics in Fig. 1). As many topics as needed are sent to communicate with the applications that require a certain functionality (Development-App output topics in Fig. 1). These topics are configured through two bash scripts (included in the modules “App connector” and “Development connector” of Fig. 1). Kafka connectors have been implemented to perform this exchange, which are in charge of transferring the information from databases to topics and from topics to databases. In addition, Apache Arrow is used as an auxiliary technology, which makes it possible to convert binary data, speeding up both the information transfer and the data parsing process. The platform runs in a fully distributed manner and can use different existing cloud environments, both externally (Azure, Amazon, etc.) and locally/privately managed.

Model of communication between an app and a development through Kafka and Kafka Connect (see Fig. 2). Two topics (Kafka) are used for this purpose: Application1 input topic: where input information for App1 is dumped. Application1 client1 output topic: where the content of the application1 output is dumped. There are four Kafka connectors, of two types: (1) source connectors: they return information from the database to the topic; (2) sink connectors: they dump information from the topic to the database. Finally, it should be

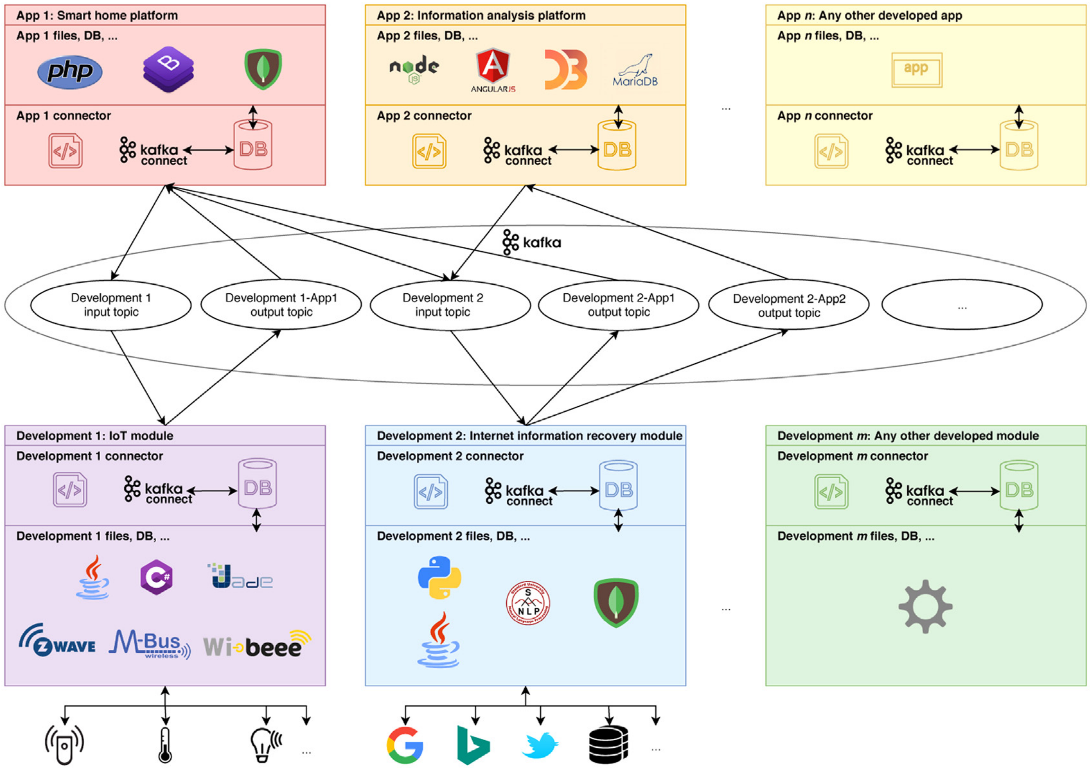

Fig. 1. Proposed architecture schema showing the three different levels: citizen and administration applications (top), information management block (middle) and developed modules providing the main functionalities.

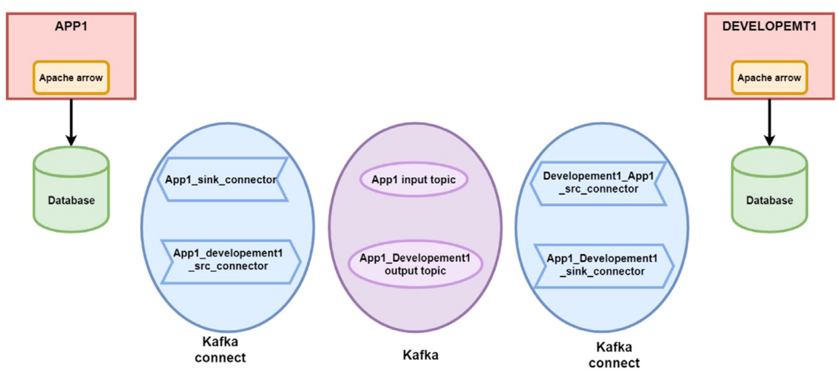

Fig. 2. Mechanism for the automatic exchange of information.

noted that in Application1input topic, the inputs (for Application1) of all the Developments are dumped. And a new output topic is created for each new development (Application1 Development1 output topic) so that the respective connectors can perform the data transfer.

The collection of data through the use of sensors has been a crucial aspect of the architecture design because it enables HPC. Since the architecture is modular and performs big data analysis, it must be capable of acquiring data from very diverse sources (crawlers, social networks, other applications and sensors). The sensors the architecture uses are KNX as middleware. KNX facilitates simple virtualization. In this way, intelligent embedded automatisms can be created. Virtualization blurs the connection between hardware and software by allowing multiple operating systems to run on a single processor. This enables the aggregation of computing power to solve complex energy optimization problems in Intelligent Buildings or Smart Cities. Agent-based architectures enable complex data analysis through distributed computing [30,31]. In addition, these architectures allow to incorporate calculation capacity in individual agents through the use of the GPU calculation power of the device in which it is running (an agent or a group of agents can be deployed in different devices).

3.2. Deployment architecture

To support the central system in which the Kafka environment is deployed in a distributed mode, it is necessary to calculate the number of nodes that is needed to correctly perform the processing. This allows to adapt the use of resources to the size of the city. In addition, the nodes deployed optimally, in accordance with the consumption predicted for any given moment, as described in the study in [32].

To determine the number of nodes that a city needs at each instant, it is necessary to compute a prediction for an instant of time t, that follows the current moment. The length of this instant depends on the booting period (bp) of the servers in which the system is deployed. To design a forecasting system it is important to consider both, the citizens' usual behavior and any changes that may occur (such as festivities in the city). Given that the prediction approach must be both static and dynamic, the solution is based on a hybrid prediction system.

The static part of the prediction system is in charge of estimating the number of future predictions on the basis of historical request patterns. According to the estimate, the required number of nodes is deployed. It responds to requests in less than s seconds (the restriction we have set in this case is two seconds). The prediction system implements the static approach, thanks to an SVR that detects patterns according to different inputs, such as: weekday, day time, month, holiday period, number of active users, number of deployed applications, current value, previous value.

The dynamic approach is in charge of re-adjusting the system in the event of an increased number of predictions. The static part of the prediction system is not capable of detecting the increase in requests/messages. Thus, the dynamic part of the prediction system is based on the Poisson distribution [33], which predicts the number of requests/messages according to the queuing theory. Thus, different Poisson distribution formulas are applied to predict the number of requests/messages within a confidence interval.

$$P(k,\lambda) = \frac{e^{-\lambda}\lambda^k}{k!} = P(X=x) \tag{1}$$

where λ ($\lambda \in \mathbb{R}, \lambda > 0$), is the mean number of requests/messages for every interval, k is the number of requests/messages and e represents the Euler's number. $P(k,\lambda)$ is the result of the equations, that is, the probability of receiving k ($k \in \mathbb{N} \cup \{0\}$) requests/messages.

The amount of nodes to be deployed in each period $n * bp$ (where $n = 1, 2, 3...$ and bp is the "booting period" of each node) must be estimated. For this reason, the prediction system is executed to calculate the number of nodes needed in the interval $n * bp + bp$ to $n * bp + 2 * bp$. The SVR and the Poisson predictor need input information to be able to estimate the required number of nodes. This input information is obtained by the system.

The two parts of the node number prediction system are detailed below:

3.2.1. Static approach

The static prediction system is based on an SVR (Support Vector Regression) [34], which predicts the number of requests/messages (that is used to calculate the number of nodes to deploy) considering the patterns found in previous precedents, which are mainly associated with the habits of the platform's users.

This SVR predictor considers the following inputs:

- Weekday: the day of the week (1–7).
- Daytime: the time of the day in seconds (0–86400).
- Month: the month of the year (1–12).
- Public holiday: this parameter indicates if "today" is a working day or not (0,1).
- Amount of active users on the city platform.
- Amount of the applications deployed in the system for that city.
- Amount of requests/messages in t.
- Amount of requests/messages in $t-1$.

This part of the system follows the static approach, it predicts the number of requests/messages that will be received at a selected time instant (req_R) and the predicted number is used as output. The two solutions given by the static and the dynamic approaches are then used to calculate the number of nodes to be deployed to respond to those requests/messages, even when they reach their highest number (this aspect will be described further on).

3.2.2. Dynamic approach

The following equations have been defined to describe the dynamic approach:

$$CDF = P(X \le k) = e^{-\lambda} \sum_{i=1}^{\lfloor k \rfloor} \frac{\lambda^i}{i!} = \sum_{i=0}^{\lfloor k \rfloor} P(k,\lambda) = P_{AC}(k,\lambda) = prob \tag{2}$$

where $prob \in [0, 1]$

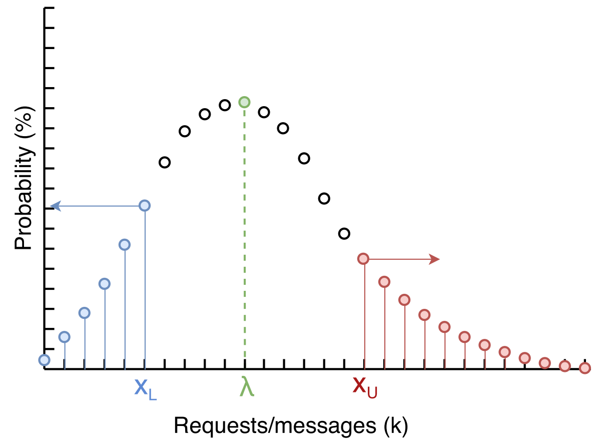

Fig. 3. Dynamic approach behavior — requests/messages vs. probability.

Eq. (2) is used to calculate the cumulative probability of the k number of requests/messages.

$$CDF^{-1} = max\{k \in \mathbb{N} \cup \{0\} : P_{AC}(k,\lambda) \le prob\} = P_{AC}^{-1}(prob,\lambda) \tag{3}$$

Eq. (3) represents the quantile function for the Poisson distribution (or inverse cumulative probability), that provides the number of requests/messages up to the cumulative probability as a percentage.

$$CDF^{-1} = k = X_U = P^{-1}(X \le k) \tag{4}$$

$$probError = 1 - prob \tag{5}$$

The value of $probError$, represented in Eq. (5), is based on the margin of confidence.

$$RPC = RPN - (X_U - (X_U \bmod RPN)RPN) \tag{6}$$

RPC is the "Remaining Processing Capacity" and it is calculated as defined in Eq. (6). It represents the number of requests/messages X_U that can still be attended by the deployed nodes. The variable RPN is the number of requests per node, which represents the highest number of requests/messages that every node (all of them have the same features) can manage to process in less than two seconds.

$$LIM_U = CDF^{-1}(prob,\lambda) + RPC - \lambda \tag{7}$$

The parameter LIM_U calculated in Eq. (7) represents the increase of the amount of requests/messages that can be managed by the nodes, calculated by the dynamic component.

$$GR(t) = \{x : x = req(t) - req(t-1)\} \tag{8}$$

$GR(t)$ in Eq. (8) represents the request/message increase rate.

$$GR_{+1}(t) = \{x : x = req(t) - req(t-2)\} \tag{9}$$

$GR_{+1}(t)$ is the growth rate of the request/message that is calculated using $t-2$, to determine the limit for the dynamic component, as represented in Eq. (9).

$$Ov(x,y) = \sum_{t=x}^{y} |LIM_U(t) - GR(t)| \tag{10}$$

The parameter $Ov(x,y)$ calculated in Eq. (10) represents the overload in an interval between the instants x and y, in other words, the increase in the amount of requests/messages goes beyond the capacity of the dynamic part with a fixed confidence interval, $prob$.

Fig. 3 shows the Poisson probability distribution and Fig. 4 shows the prediction built on this distribution. In the Poisson distribution figure, each point represents the possibility of receiving a specific

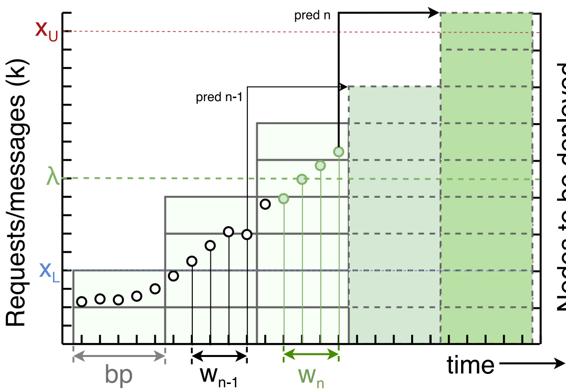

Fig. 4. Dynamic approach behavior — time vs. requests/messages and nodes to deploy.

amount of requests/messages in discrete distribution. The Poisson distribution depends on λ (as defined in Eq. (1)), that is determined considering the values requested in the time window w_n. The value X_U is obtained as defined in Eq. (4) on the basis of a confidence or error interval, as shown in Eq. (5). X_U represents the maximum amount of requests/messages that can be predicted within the confidence interval.

In Fig. 4, the points represent the number of requests/messages received until the current moment. As explained previously, the booting period is represented by bp and it refers to the time needed to deploy a node. It must be noted that the result of a prediction obtained in an instant t, is used for the instant $t + bp$. Rectangles whose border is a continuous line represent the amount of nodes that have already been deployed, and the rectangles whose border is a dashed line indicate that the nodes have been predicted depending on the values of the variable X_U. Variable X_L represents the minimum amount of requests/messages predicted with the given probability.

The inverse cumulative probability is then estimated over an established confidence interval (in our case the value chosen is 95%), the amount of requests/messages whose probabilities are below that amount is assumed to be the total number of requests/messages handled by the system.

Once the number of requests/messages is estimated by the two approaches, the system proceeds to calculate the required nodes to provide the result to those requests/messages.

3.2.3. Number of nodes to deploy

To determine the amount of nodes that needs to be deployed to attend all the requests/messages, the predictions provided by the SVR predictor and the Poisson predictor are taken into account. Given that the system's priority is to ensure all the requests are replied to in under two seconds, the amount of nodes to be deployed is calculated as shown in Eq. (11).

After the combination of the two approaches (static and dynamic), the amount of nodes to deploy is calculated using the higher of the two predicted values. This solution encourages the assurance that requests/messages are handled and processed to the detriment of saving on the cost of processing capacity.

$$n = \left\lceil \frac{\max(req_R, req_P)}{RPN} \right\rceil \quad (11)$$

where n represents the predicted amount of nodes to deploy at the moment $t + bp$, the variable req_R represents the amount of requests/messages predicted by the static approach (SVR), the variable req_P is the number of requests predicted by the dynamic approach, and finally, $requestsPerNode$ is the highest amount of requests/messages that every single node can serve, ensuring that the response is dispatched in less than two seconds.

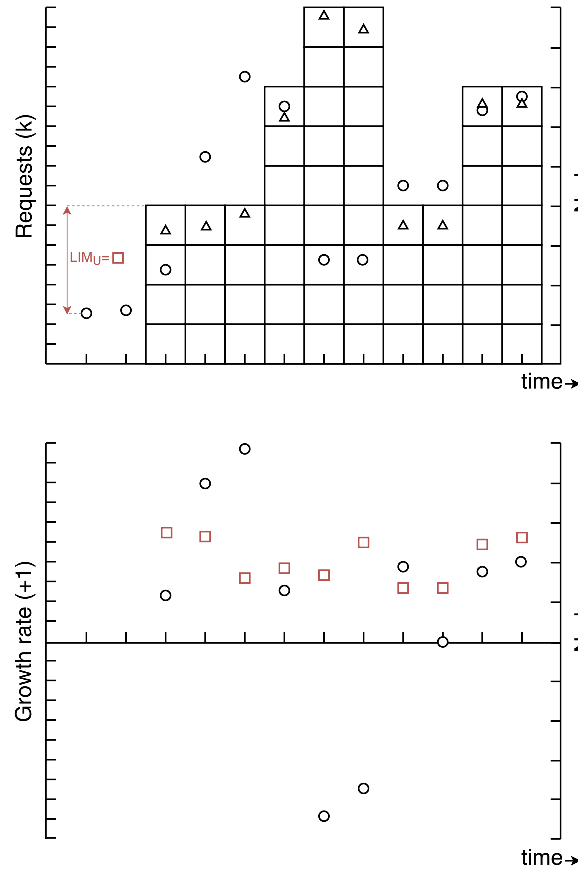

Fig. 5. Representation of the dynamic system limits.

3.2.4. Node deployment system evaluation

Fig. 5 presents two charts: (i) the upper one shows the number of requests/messages is represented as circles, the X_U predictions as triangles and the number of nodes predicted like rectangles; (ii) the lower graphic shows the growth rate+1 following Eq. (9) as diamonds. The squares represent the LIM_U as shown in Eq. (7), that is also the distance shown in the upper graphic.

It is important to notice that when a diamond is over a rectangle it means that the number of deployed nodes is not enough to satisfy the two seconds restriction. In this way, we have a measure of the system performance because when there is overload respect to the restriction the system begins to recover as soon as the growth of petitions is below the limit. So the overload of the system can be calculated as defined in Eq. (10).

In this case, despite not being able to satisfy the condition of dispatching all requests/messages in less than two seconds as marked as restriction, the designed system guarantees that all requests are dealt with in due order even though there are overload points that imply that the result is available later than the established two seconds.

One of the main advantages of the proposed system is that the system can support a volume adapted to the needs, without the need to have a static upper limit. In all the tests carried out, the upper limit has been marked by the limitations of the hardware available to support, not having found any restrictions of the system in terms of scalability beyond the availability of hardware resources.

4. Results: Two citizen-oriented case studies

In this section, we describe the two case studies that have been conducted to verify the performance of the proposed platform. In particular, the case studies were designed to test the "Smart home

Fig. 6. Monitoring the status of a smart home by means of the "Smart home platform" application.

platform" and the "Information analysis platform", as illustrated in Fig. 1. Moreover, we have developed more applications in other fields (such as industry 4.0, city incident management, etc.) to validate the smooth performance and scalability of the platform.

4.1. Smart home platform

In this case study we describe a new IoT platform for improved energy efficiency in homes. This platform has also been tested in multiple industrial environments as an Industrial IoT (IIoT) solution, although, due to data privacy reasons and similarity to the IoT scenario, no specific cases of IIoT have been described.

Thanks to the developed smart home platform, unlimited installations can be deployed in homes with a limitless number of sensors that collect a range of data, including: the status of the doors and windows, temperature and humidity of each room, real-time power consumption or presence in each of the rooms. The sensors can follow different protocols, such as Z-wave, M-Bus, DALI, BACnet or Wi-beee over KNX. Several algorithms that use Jade-developed smart agents, are applied to the collected information [35]. Those agents use a set of rules and inferences to recommend changes in the energy consumption habits of the inhabitants of a home. The main objective is to reduce energy consumption with all that this entails.

The application allows the user to monitor the state of their home in real time, receive notifications, suggestions and define their preferences to optimize energy consumption according to their comfort criteria (see Fig. 6).

The system also obtains information from the Internet about energy prices and tariffs available for the location, as well as information about the user's behavior (only if they approve of this kind of information being collected). On the basis of the retrieved information, the rules engine is fed to make new suggestions to the user, such as changing the behavior of users in the domestic environment or changing the electricity tariff. Suggestions for change of behavior may include doing household chores at times when the price of energy is the lowest. This is achieved through the use of the "Internet information recovery module", which is also used in the case study presented below.

In this case study, a model designed by the BISITE research group has been used. The model simulates the activity of a smart city so it can be used in related case studies. different case studies related to, such as case studies on energy optimization in homes or buildings, case studies on transport or on the energy grid or on the sustainability of cities, Fig. 7. In this specific case, only the residential area has been used, which is the upper central zone marked in the image.

The present case study, Fig. 8 has proven that data management is made possible by the architecture since it simulates the different patterns of behavior. In this way, it is possible to verify if an optimization of the energy takes place in the city or not.

The Automatic Lighting Management Module (ALMM) using the developed IoT platform has made it possible to simulate lighting management in 60 homes using the previously presented model. The lighting in 30 of the 60 homes was managed automatically when the system detected that the light intensity was lower than 200Lux, so that the inhabitants of this house had sufficient light to carry out their daily tasks. In the other 30 dwellings, the lighting was managed according to real behavior (homes of the members of the BISITE Research Group who have volunteered to replicate the daily lighting patterns in their homes).

Table 1
Results of energy consumption in the different steps of the experiment.

	Baseline period (kWh)	Simulation period (kWh)	Difference (kWh)	Savings (%)
1st gr.	19.71	14.24	5.47	−27,75
2nd gr.	18.66	13.65	5.01	−26.84
3rd gr.	16.68	12.24	4.44	−26.61

The use of this platform has allowed for an easy and fast integration of the application for intelligent lighting management and has made it possible to evaluate its effectiveness in 30 homes. The results of the simulation are shown below, grouping the dwellings into three groups of ten dwellings each (Average consumption). As can be seen in Table 1 there is a reduction in the three groups with respect to the same period if they had not used the ALMM with an average percentage of 27%.

As shown in Table 2, the energy consumption is notably lower in the case study. We can see how the p-value is always under 0.05, which ensures the effectiveness of ALMM and the platform in integrating various management modules into Smart Cities or intelligent buildings.

4.2. Information analysis platform

This case study has used all the publicly available information on the Internet about users. This type of analysis can be very useful in applications such as terrorist searches, analysis of suspect profiles, etc. The system creates a series of crawlers that retrieve public data from all pages that fulfill the criteria specified in the input form shown in Fig. 9. Only two prior fields are required, which are the first and last name, and the system is responsible for retrieving all the existing data.

Once all the data has been downloaded, a series of studies are carried out on the sources from which they have been retrieved, the information and the images they contained. As a result, it is possible to create a set of clusters that classify the results into possible identities. For example, hundreds of people called 'John Doe' can be found and the system is able to classify them according to the analyzed results. From each cluster, we obtain their profiles on social networks, their most relevant images (we analyze the pictures in which their faces are visible and compare the faces in different images in existing libraries to check if they are the same person), their addresses, emails, companies with which they are associated, related users, phone numbers, nationality, etc. At the moment, the system works in Spanish and English, however, since it mainly uses the Stanford NLP development [36], it is adaptable to new languages.

Fig. 7. BISITE Research Group Mock-Up to simulate transport and energy case studies.

(a) Automatic lighting management through the platform of 30 homes.

(b) Automatic lighting management through the platform of 25 homes.

(c) Automatic lighting management through the platform of 20 homes.

(d) Automatic lighting management through the platform of 15 homes.

Fig. 8. Different stages of consumption simulation in intelligent buildings of the mock-up.

Table 2
Result of the Student's t-test and Levene's test performed to assess the difference of means and variances between the Baseline data and the Evaluation period for the three groups.

	Baseline period		Evaluation period					
	Mean (kWh)	Stdr. Deviation (kWh)	Mean (kWh)	Stdr. Deviation (kWh)	*t*	Sig. (2-Tailed)	F	Sig.
1st gr.	19.71	2.40	14.24	5.16	5.25	0.000	9.42	0.003
2nd gr.	18.66	2.09	13.65	4.63	5.39	0.000	10.19	0.002
3rd gr.	16.68	2.41	12.24	4.22	5.00	0.000	7.73	0.007

Although Kafka has some limitations in the passage of messages, its use to connect IoT devices in the platform proposed for the management of Smart Cities offers certain improvements over existing technologies, as can be seen in Table 3. The data shown in the table have been retrieved from [37].

It can be observed in the comparative table that the proposal presented in this work allows for an unlimited connection of devices which is not offered by the rest of the state of the art platforms. In addition, the data transfer is far superior to its strongest competitor AWS IoT, having the ability to send 10,000 messages/second. These

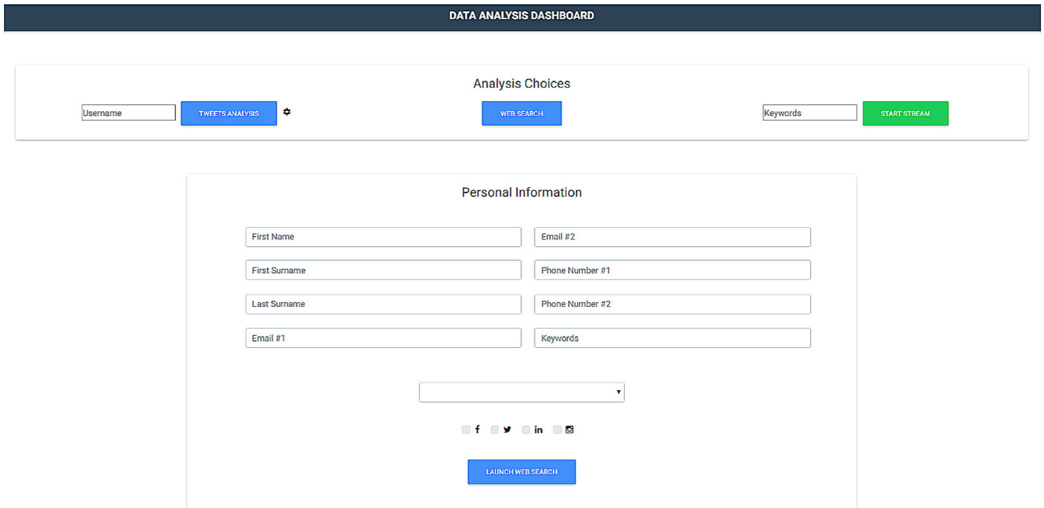

Fig. 9. Input form shown in the frontend of the "Information analysis platform" application.

Table 3
IoT platforms comparison.

	Max. allowed devices	Data transference
Azure IoT Hub [38]	500	8000 messages/day
AWS IoT [39]	No limit	8300 messages/day
Watson IoT [40]	500	200 MB/month
Proposed platform	No limit	10,000 messages/second

features are a prerequisite for managing the data produced in a city such as those used in the case study.

5. Conclusions and future work

A distributed and modular platform has been developed. Thanks to its modularity, it is possible to provide the citizen with a set of applications that reuse functionalities. Not only citizens benefit but also the developers who can use the technologies they know best or those that adapt best to the problem they want to solve. Furthermore, those technologies can be integrated transparently in the platform. The required scalability and computing capacity, as well as other aspects, directly depend on the preferences of the system administrator or on the city's needs. Nevertheless, the infrastructure that supports the proposed platform is completely transparent for the system, making it possible to use the module that best adapts to the needs of each city. Despite the great benefits of using Kafka, it also has some limitations, especially in large cities. The maximum number of messages that can be handled per second is 100,000. Nevertheless, the system could be scaled for large cities so as to make it capable of dealing with high information traffic. Kafka also has a limitation on the number of topics that can be created, which is less than a thousand. However, this does not impede the scalability of the system, because if it is necessary to create more than a thousand topics in a large city, different information transmission systems can be created in a single city using Kafka.

Although there are certain limitations due to the use of the selected technology, those limits can be overcome easily. As it has been shown, the architecture has extra computing capacity for the analysis of large amounts of data. Although only two case studies have been conducted, there are currently more than ten functionalities that the different applications can perform almost automatically. We continue working on providing new functionalities to meet all the possible needs that may arise in any type of city. One of the use cases on which we are currently working, is the integration of Salamanca's public transport information, which includes buses (15 lines) and bicycles for rent (more than 160). We are working on developing the applications fully, so that they can be offered to end-users. These applications will help improve the quality of life in the city.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Pablo Chamoso: Conceptualization, Methodology, Software, Writing - review & editing. **Alfonso González-Briones:** Data curation, Writing - review & editing. **Fernando De La Prieta:** Investigation, Writing - review & editing. **Ganesh Kumar Venyagamoorthy:** Methodology, Supervision. **Juan M. Corchado:** Methodology, Supervision.

References

[1] C. Richter, S. Kraus, P. Syrjä, The smart city as an opportunity for entrepreneurship, Int. J. Entrep. Ventur. 7 (3) (2015) 211–226.
[2] A.J. Sánchez, S. Rodríguez, F. de la Prieta, A. González, Adaptive interface ecosystems in smart cities control systems, Future Gener. Comput. Syst. 101 (2019) 605–620.
[3] P. Cardullo, R. Kitchin, Smart urbanism and smart citizenship: The neoliberal logic of 'citizen-focused' smart cities in europe, Environ. Plan. C (2018) 0263774X18806508.
[4] T.J.H. Gracia, A.C. García, et al., Sustainable smart cities. creating spaces for technological, social and business development, in: Boletín Científico de las Ciencias Económico Administrativas del ICEA, Vol. 6, 2018.
[5] S. Hashim Raza Bukhari, S. Siraj, M. Husain Rehmani, Wireless sensor networks in smart cities: Applications of channel bonding to meet data communication requirements, Transp. Power Grid Smart Cities: Commun. Netw. Serv. (2018) 247–268.
[6] J. Toutouh, E. Alba, Parallel multi-objective metaheuristics for smart communications in vehicular networks, Soft Comput. 21 (8) (2017) 1949–1961.
[7] A. González-Briones, P. Chamoso, H. Yoe, J. Corchado, Greenvmas: virtual organization based platform for heating greenhouses using waste energy from power plants, Sensors 18 (3) (2018) 861.
[8] A. González-Briones, J.A. Castellanos-Garzón, Y. Mezquita Martín, J. Prieto, J.M. Corchado, A framework for knowledge discovery from wireless sensor networks in rural environments: A crop irrigation systems case study, Wirel. Commun. Mobile Comput. 2018 (2018).
[9] A. González-Briones, J. Prieto, F. De La Prieta, E. Herrera-Viedma, J. Corchado, Energy optimization using a case-based reasoning strategy, Sensors 18 (3) (2018) 865.
[10] A. González-Briones, J. Prieto, J.M. Corchado, Y. Demazeau, Enervmas: Virtual agent organizations to optimize energy consumption using intelligent temperature calibration, in: International Conference on Hybrid Artificial Intelligence Systems, Springer, 2018, pp. 387–398.

[11] P. Chamoso, A. González-Briones, S. Rodríguez, J.M. Corchado, Tendencies of technologies and platforms in smart cities: A state-of-the-art review, Wirel. Commun. Mobile Comput. 2018 (2018).

[12] A. González-Briones, F. De La Prieta, M. Mohamad, S. Omatu, J. Corchado, Multi-agent systems applications in energy optimization problems: A state-of-the-art review, Energies 11 (8) (2018) 1928.

[13] A. Enayet, M.A. Razzaque, M.M. Hassan, A. Alamri, G. Fortino, A mobility-aware optimal resource allocation architecture for big data task execution on mobile cloud in smart cities, IEEE Commun. Mag. 56 (2) (2018) 110–117.

[14] V. Potdar, S. Batool, A. Krishna, Risks and challenges of adopting electric vehicles in smart cities, in: Smart Cities, Springer, 2018, pp. 207–240.

[15] A. Rivas, P. Chamoso, A. González-Briones, J. Corchado, Detection of cattle using drones and convolutional neural networks, Sensors 18 (7) (2018) 2048.

[16] J. Toutouh, E. Alba, A swarm algorithm for collaborative traffic in vehicular networks, Veh. Commun. 12 (2018) 127–137.

[17] A.G. Briones, P. Chamoso, A. Rivas, S. Rodríguez, F. De La Prieta, J. Prieto, J.M. Corchado, Use of gamification techniques to encourage garbage recycling. a smart city approach, in: International Conference on Knowledge Management in Organizations, Springer, 2018, pp. 674–685.

[18] P. Chamoso, A. González-Briones, A. Rivas, F. Bueno De Mata, J. Corchado, The use of drones in spain: Towards a platform for controlling uavs in urban environments, Sensors 18 (5) (2018) 1416.

[19] A. González-Briones, P. Chamoso, F. De La Prieta, Y. Demazeau, J. Corchado, Agreement technologies for energy optimization at home, Sensors 18 (5) (2018) 1633.

[20] S. Srikanth, P. Pramod, K. Dileep, S. Tapas, M.U. Patil, et al., Design and implementation of a prototype smart parking (spark) system using wireless sensor networks, in: 2009 International Conference on Advanced Information Networking and Applications Workshops, IEEE, 2009, pp. 401–406.

[21] C.G. García, E.R.N. Valdez, V.G. Díaz, B.C.P.G. Bustelo, J.M.C. Lovelle, A review of artificial intelligence in the internet of things, IJIMAI 5 (4) (2019) 9–20.

[22] V.K. Solanki, M. Venkatesan, S. Katiyar, Conceptual model for smart cities: Irrigation and highway lamps using iot, IJIMAI 4 (3) (2017) 28–33.

[23] S.M.S. Trimberger, Three ages of fpgas: A retrospective on the first thirty years of fpga technology: This paper reflects on how moore's law has driven the design of fpgas through three epochs: the age of invention, the age of expansion, and the age of accumulation, IEEE Solid-State Circuits Mag. 10 (2) (2018) 16–29.

[24] V. García-Díaz, J.B. Tolosa, B.C.P. G-Bustelo, E. Palacios-González, Ó. Sanjuan-Martínez, R.G. Crespo, Talisman mde framework: an architecture for intelligent model-driven engineering, in: International Work-Conference on Artificial Neural Networks, Springer, 2009, pp. 299–306.

[25] J. Pascual Espada, V. García-Díaz, E.R. Núñez-Valdéz, R. González Crespo, Real-time force doors detection system using distributed sensors and neural networks, Int. J. Intell. Syst. 34 (9) (2019) 2243–2252.

[26] A.A. Pereira, J.P. Espada, R.G. Crespo, S.R. Aguilar, Platform for controlling and getting data from network connected drones in indoor environments, Future Gener. Comput. Syst. 92 (2019) 656–662.

[27] K.A. Paskaleva, The smart city: A nexus for open innovation? Intell. Build. Int. 3 (3) (2011) 153–171.

[28] A. Ferraris, N. Erhardt, S. Bresciani, Ambidextrous work in smart city project alliances: unpacking the role of human resource management systems, Int. J. Hum. Resour. Manage. (2017) 1–22.

[29] F.D. Sandulli, A. Ferraris, S. Bresciani, How to select the right public partner in smart city projects, R&D Manage. 47 (4) (2017) 607–619.

[30] E.A. Mohamed, A.-A.A. Mohamed, Y. Mitani, Genetic-moth swarm algorithm for optimal placement and capacity of renewable dg sources in distribution systems, Int. J. Interact. Multimedia Artif. Intell. 5 (7) (2019) 105–117.

[31] A. González-Briones, F. De La Prieta, M. Mohamad, S. Omatu, J. Corchado, Multi-agent systems applications in energy optimization problems: A state-of-the-art review, Energies 11 (8) (2018) 1928.

[32] P. Chamoso, A. Rivas, R. Sanchez-Torres, S. Rodriguez, Social computing for image matching, PLoS One 13 (5) (2018) e0197576.

[33] T. Mahmud, M. Hasan, A. Chakraborty, A.K. Roy-Chowdhury, A poisson process model for activity forecasting, in: 2016 IEEE International Conference on Image Processing (ICIP), IEEE, 2016, pp. 3339–3343.

[34] W.-C. Hong, Y. Dong, L.-Y. Chen, S.-Y. Wei, Svr with hybrid chaotic genetic algorithms for tourism demand forecasting, Appl. Soft Comput. 11 (2) (2011) 1881–1890.

[35] F. Bellifemine, A. Poggi, G. Rimassa, Jade: a fipa2000 compliant agent development environment, in: Proceedings of the Fifth International Conference on Autonomous Agents, ACM, 2001, pp. 216–217.

[36] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, D. McClosky, The stanford corenlp natural language processing toolkit, in: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55–60.

[37] V.M. Vicente-Cebrián, Comparison of free versions on iot platforms. sofia2 iot platform blog, 2019, https://about.sofia2.com/2017/04/27/comparison-of-free-versions-on-iot-platforms. Accessed: 2019-03-12.

[38] Microsoft, Iot hub — microsoft azure, 2020, https://azure.microsoft.com/es-es/services/iot-hub/, Accessed: 2020-01-12.

[39] AWS, Internet of things — platform as a service — aws iot, 2020, https://aws.amazon.com/iot/, Accessed: 2020-01-12.

[40] IBM, Explore the internet of things (iot) — ibm whatson iot, 2020, https://www.ibm.com/internet-of-things, Accessed: 2020-01-12.