We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Chapter

Increasing the Efficiency of Rule-Based Expert Systems Applied on Heterogeneous Data Sources

Juan Ignacio Guerrero Alonso, Enrique Personal, Antonio Parejo, S. García, Antonio Martín and Carlos León

Abstract

Nowadays, the proliferation of heterogeneous data sources provided by different research and innovation projects and initiatives is proliferating more and more and presents huge opportunities. These developments create an increase in the number of different data sources, which could be involved in the process of decision-making for a specific purpose, but this huge heterogeneity makes this task difficult. Traditionally, the expert systems try to integrate all information into a main database, but, sometimes, this information is not easily available, or its integration with other databases is very problematic. In this case, it is essential to establish procedures that make a metadata distributed integration for them. This process provides a "mapping" of available information, but it is only at logic level. Thus, on a physical level, the data is still distributed into several resources. In this sense, this chapter proposes a distributed rule engine extension (DREE) based on edge computing that makes an integration of metadata provided by different heterogeneous data sources, applying then a mathematical decomposition over the antecedent of rules. The use of the proposed rule engine increases the efficiency and the capability of rule-based expert systems, providing the possibility of applying these rules over distributed and heterogeneous data sources, increasing the size of data sets that could be involved in the decision-making process.

Keywords: rule-based expert system, inference engine, heterogeneous data source integration, distributed data sources

1. Introduction

The expert systems (ESs) are one of the most traditional artificial intelligence techniques [1], providing the possibility of implementing systems which allow us to solve problems in a limited domain. However, the variety and possibilities of ESs have improved in the last years with the combination of them with other technologies such as fuzzy logic [2], Bayesian network [3], etc. Moreover, several languages and tools have been developed to provide faster developments (e.g., CLIPS, LISP, PROLOG, etc.) and deployments. Other systems are improved in collaboration with other technologies in order to expand the domain of problems and increase the knowledge base. As example, [4] improves the capabilities of an ES adding text mining, neural networks, and statistical techniques.

On the one hand, the scope of ES application is very extensive, including health [5], education [6], physics [7], chemical [8], mechanics [9], etc. Thus, the application of ES is not only restricted to solve the problem, they usually include an explaining engine, which could be used for educational purposes, operating at the same time.

The ESs have a limited domain, and the size of used data sets is smaller than in other artificial intelligence techniques. However, the improvement of availability of information, new concepts related to sensor networks, and the capability to generate and consume information in different sectors provide a different scenario, in which the ESs traditionally had a lot of information disseminated into different information resources. Although each information resources could have its own structure, the stored information could be very similar. In this scenario, it is essential to make the analysis of the distributed data sets and apply the different rules and inference engines in these distributed data sets possible.

In response to this problem, an additional layer is proposed in the current chapter which could be added to the ES engines, mainly based on rules and fuzzy logic. This new layer or middleware allows the ES to make a metadata integration of heterogeneous data sets, without making a real integration and replication of the information, allowing by mean edge, and computing the application of rules in the disseminated data sets by means of a logical decomposition of rules based on metadata integration results. This novel layer or middleware, named distributed rule engine extension (DREE) has an application protocol interface (API) to allow the communication between the ES engine and the different data sources. The DREE enables the ES engines to be applied in heterogeneous data sets disseminated in a network, by means of installation of edge computing daemon (ECD) in each distributed node.

Edge computing has some similarities with fog computing, cloud computing, etc. [10] provides a review of different similar technologies, providing a very complete survey. In the case of the proposed system, edge computing is usually related to the Internet of Things (IoT) devices. In case of the proposed solution, the edge nodes are the computers or devices, which has the information stored, by means of any type of database management system.

In the following sections, the general architecture is detailed. Firstly, the process of metadata integration from heterogeneous data sets is described. Secondly, the process of logical decomposition and how the edge computing contributes in the process of distributed application of antecedent rule are described. Finally, the experimental application of the proposed layer or middleware is shown, with the conclusion of results and future research lines.

2. Architecture overview

The application process of the DREE is performed in three stages. In the first one, the metadata integration is performed. In the second, the rules from knowledge base are logically decomposed, according to the metadata model. In the third one, the rules are carried out in the disseminated data sets by means of edge computing in the distributed nodes. In this sense, to carry out these tasks, the architecture proposed for the DREE is shown in **Figure 1**.

Each node has an edge computing daemon which performs queries in the database node to gather all information from local data sets, sending only metadata to the DREE. This information is integrated and translated in a unified metadata structure to an engine directly available for DREE.

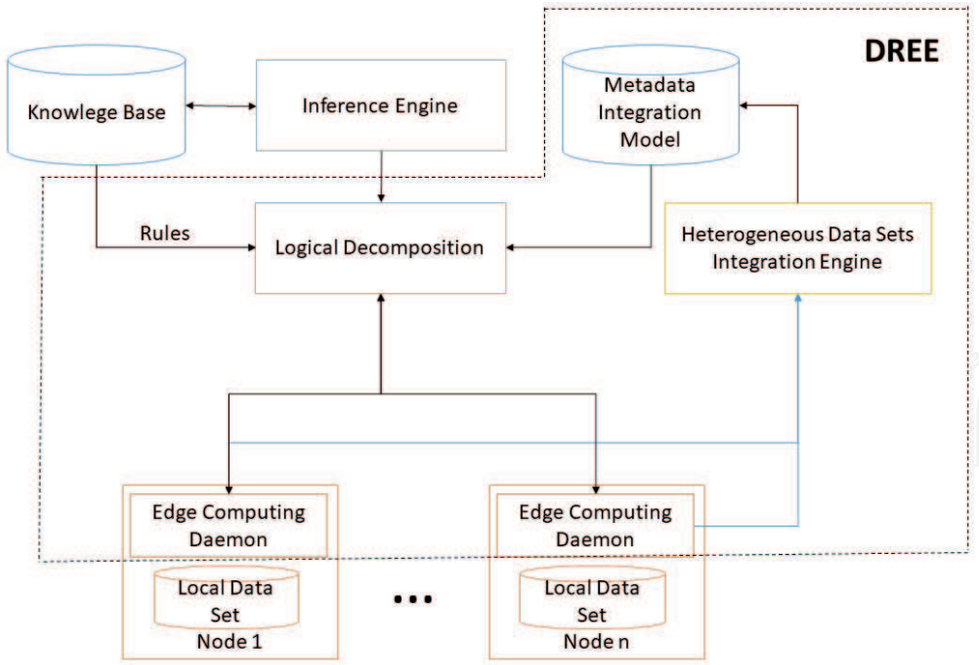

Figure 1.
DREE architecture and flow overview.

The metadata integration model database provides a map about the distribution of data between the different nodes. This information is used by logical decomposition engine (LDE) to identify the dependences between variables and the complexity of rules in the knowledge base. In some cases in which there exists a very complex logic expression with a high level of dissemination between nodes, the information would be stored in this database in order to make the rule application, but this option needs to be configured manually.

3. Heterogeneous data set integration (HDSI)

The fundamentals of heterogeneous data set integration were described in [11]. In this reference, a heterogeneous data source integration system (HDSIS) is described and applied to smart grid and health. Following this idea, a HDSIS evolution is implemented providing also a logic integration of all information at metadata level. The architecture of the proposed HDSI is shown in **Figure 2**. Specifically, the metadata from different sources is the only information that is integrated and stored.

The HDSI includes a metadata mining engine (MME), which connects with EDCs by means of characterization engine, in order to extract information from local databases in each node, getting new and integrating existing useful metadata. The dynamic extract, transform, and load (ETL) engine performs the process to integrate the metadata, previously inferred by the MME, according to the specifications gathered by metadata mining engine and the rules stored in the knowledge base. All these modules define the query engine and the rule-based expert system. The query engine is focused on performing the different tasks required to the queries in the distributed resources. The rule-based expert system (RBES) is included in the HDSI and implements the rules that perform an integration of all metadata from all disseminated resources. Therefore, it is an RBES oriented to information integration. Some references show the application of HDSIS in different problems: nontechnical losses detection [12], electric vehicle and consumption modeling in smart grids [13], etc.

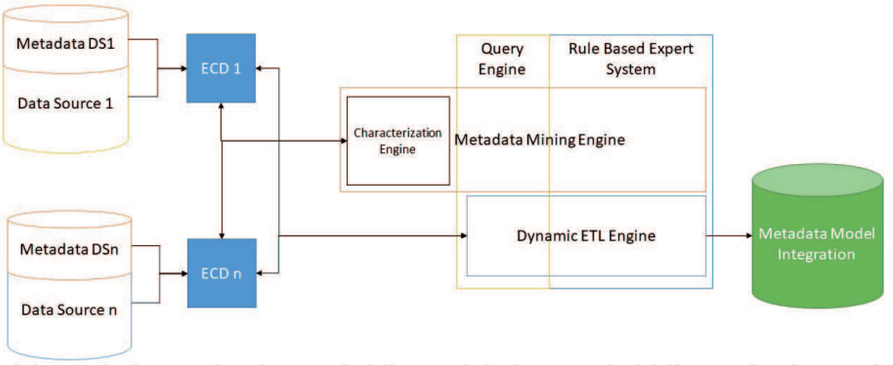

Figure 2.
Heterogeneous data set integration module overview.

The ECD includes some modules to perform the necessary queries over the database. Thus, the architecture of ECD is described in the next section.

4. Logical decomposition engine

The logical decomposition engine included two parts. The main part is in the LDE and is shown in **Figure 3**. The second part is located in the ECDs, its structure being shown in **Figure 4**.

The LDE has a first component, named logic parser. This component makes it possible to parse logic expression, which could be based on fuzzy logic or type-2 fuzzy logic, too. The logic parser works with different mathematical representation standard languages: MathML (mathematical markup language v3.0 [14]) and the OpenMath standard [15]. The logic parser serves as a RESTful web service interface, which supports extensible markup language (XML) format messages based on MathML and JavaScript Object Notation (JSON) format messages based on OpenMath.

The LDE has registered some logic expressions, which are typical equations and other previously performed expressions. LDE database stores all information about the decomposition of logic expression. In this sense, if the expression is not yet in the database, the decomposition engine checks the dependency between variables and the number of steps necessary to calculate the result. To reduce the number of steps, the decomposition engine applies a particle swarm optimization (PSO) [16] algorithm. The objective of using a genetic algorithm is to find out an equivalent logic expression with a small number of steps, reducing the dependency between variables and operations.

Thus, if the number of steps is still high or the dependence between variables could involve different data sets in disseminated nodes, the subscheduler plans the message exchange in order to establish the message sequence. This message sequence could be provided by the result of partial or complete logic expression. Thus, it is possible that some logic expressions take a long time to get the result, because of the high number of messages exchanged. In this case, the user could manually configure the HDSI to integrate the anonymized information in a server in order to reduce the time and consumption of edge computing nodes. This configuration and specification about decomposition are stored in expression database. When the LDE detects an expression of this type, the scheduler manages the decomposition according to the manual configuration.

In the ECD, the messages from the LDE are interpreted and transferred to local SQL engine and logic inference engine. The local database engine extracts the required information from local database and, this information is gathered by

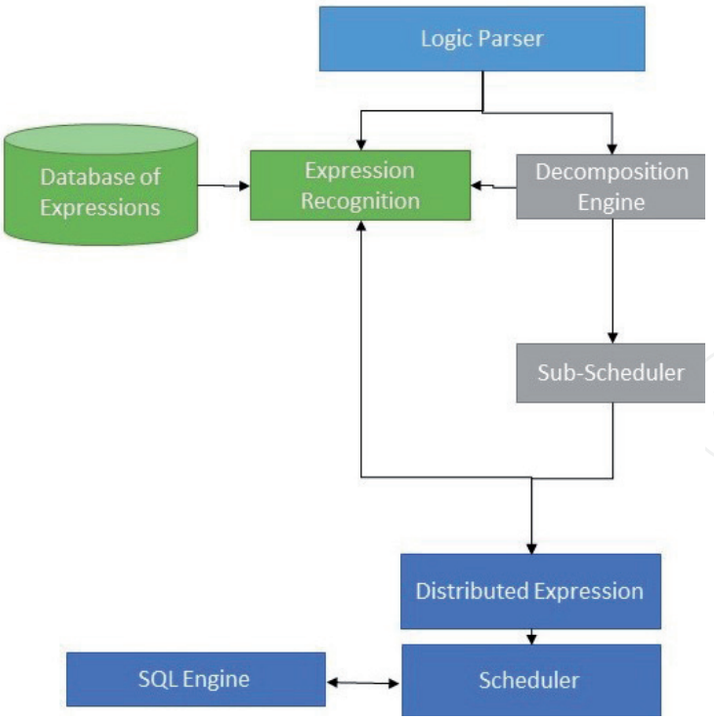

Figure 3.
The main part of LDE.

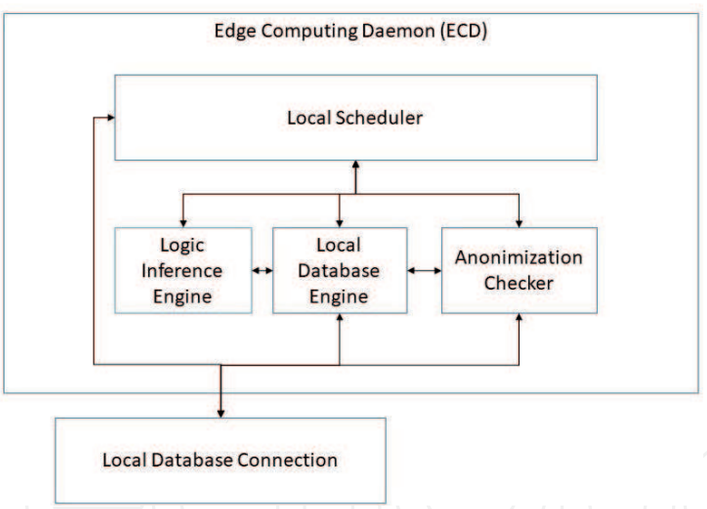

Figure 4.
Functional architecture of the edge computing daemon.

the logic inference engine which performs the expressions and returns the results. Simultaneously, the extracted information and results are checked by the anonymization checker, which is responsible for tests if the information contains data, which would be anonymized.

The local database engine is formed by two mechanisms. One engine is based on simple standard statement query language (SQL). There are several database management systems that include improvements in the SQL language. The proposed engine only applies queries on standard format, using minimized SQL queries with simple statements to ensure the compatibility. The other engine is based on NoSQL language, and it was designed to operate with unstructured databases.

Additionally, the local database engine notifies any modification in the data set to the LDE, in order to update the metadata model. In this case, the LDE

recalculates the decomposition of different expressions that involves the data from the updated or modified data set.

The logic inference engine supports logic, fuzzy logic, and type-2 fuzzy logic expressions. This engine does not perform any decompositions but only performs the expressions provided by LDE.

4.1 Particle swarm optimization

The prioritization algorithm works as a swarm intelligence algorithm. The application of the algorithm is performed after a preprocessing of information.

The prioritization algorithm is based on the parametric optimization until a solution is obtained. This optimization is executed depending on the capabilities of a system. The canonical PSO model consists of a swarm of particles, which are initialized with a population of random candidate solutions. The candidate solutions are generated by the application of different properties oriented to reduce the dependence and operations involved in the logic expression to lead to the expression, which minimizes the number of messages and distributed expression. They iteratively move through the d-dimension problem space to search for the new solutions, where fitness f can be calculated as the certain quality measure. Each particle has a position that is represented by the position-vector x_{id} (i is the index of the particle, and d is the dimension) and a velocity represented by the velocity-vector v_{id}. Each particle remembers its best position in the vector $x_{i\#}$, and its j-th dimensional value is $x_{\#ij}$. The best position-vector among the swarm is stored in the vector x_*, and its j-th dimensional value is x_{*j}. At the iteration time t, the update of the velocity from the previous velocity to the new velocity is determined by Eq. (1). The new position is determined by the sum of the previous position, and the new velocity is determined by Eq. (2):

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 \cdot \psi_1 \cdot \left(p_{id}(t) - x_i(t)\right) + c_2 \cdot \psi_2 \cdot \left(p_g(t) - x_{id}(t)\right) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2)$$

where c_1 and c_2 are constant weight factors, p_i is the best position achieved by particle i, p_g is the best position obtained by the neighbors of particle i, ψ_1 and ψ_2 are random factors in the [0,1] interval, and ω is the inertia weight. Some references denote c_1 and c_2 as the self-recognition component and the coefficient of the social component, respectively.

Different constraints can be applied to ensure the convergence of the algorithm. In this case, the operations are oriented to optimize the fitness, and the fitness is calculated based on the number of distributed operations and number of exchanged messages.

PSO algorithm:

1. Initialize particles
2. Repeat
    a. Calculate the fitness values for each particle.
    b. Is the current fitness value better than p_i?
        i. Yes. Assign the current fitness as the new p_i.
        ii. No. Keep the previous p_i.

c. Assign the best particle's p_i value to p_g.

d. Calculate the velocity for each particle.

e. Use each particle's velocity value to update its data values.

3. Until stopping criteria are satisfied

5. Testing DREE

The proposed system DREE was tested with different RBESs. These tests were performed inserting the DREE between RBES and data sets, redefining the inference engines to replace the calls to rule execution by the DREE interface. [17] provides the description of different ESs oriented to health, telecommunication, power supply, etc. The proposed architecture is shown in **Figure 5**. The information used in the different ESs was manually disseminated among five different nodes. The DREE was able to work with different ESs at the same time. The application of all rules from all ESs at the same time may involve information from different domain knowledge, because the DREE only applies the rules that the corresponding inference system provides. Thus, the state and the decision supported is performed inside of the ES. The DREE only returns the result of consequent from the fired rule. Additionally, the usage of ECD makes independent from the local data set management systems. Therefore, in these cases, the ESs improved their operational capabilities, providing the possibility to operate in real time. Additionally, the ESs may simultaneously run their own rule-based knowledge base, increasing the reliability of the different systems.

The usage of the PSO algorithm instead using the most simplified logic expression shows an increase of efficiency of the distributed operations, reducing the operations and message exchanging in 20% related to initial fuzzy and type-2 fuzzy logic expressions and in 5% related to initial logic expressions. This fact is due to how the data is disseminated by the different nodes; different distributions of data between nodes provide different optimizations. Thus, if the number of nodes or

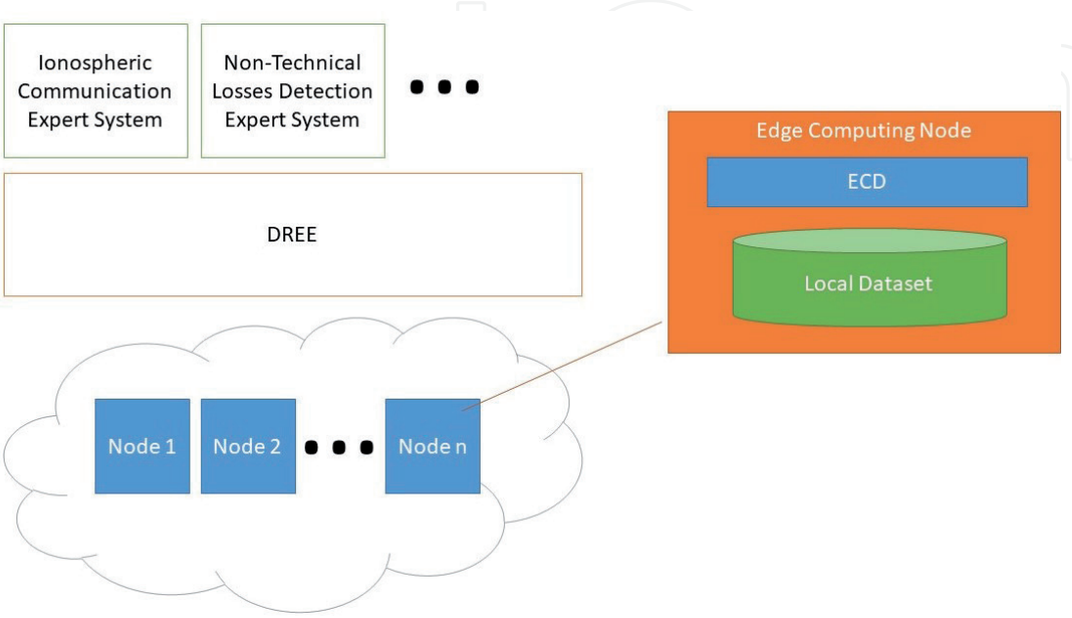

Figure 5.
Proposed architecture for application of DREE to the ESs in [17].

data sets from the nodes is updated or modified, it is necessary to recalculate the decomposition of logic expression.

In other cases, like [4, 18], the integration of the proposed DREE decreased the hardware requirements related to the storing systems.

Although the system increases the message exchange, the system avoids integrating all information in a centered data base, without using big data infrastructure, taking advantage from edge computing infrastructure and distributed nodes. Additionally, the system can quickly react to any updating or modification in any data set from the nodes involves in the DREE.

6. Conclusions

The proposed DREE provides the opportunity to integrate a great quantity of information in the inference engine, without the requirement of a big data infrastructure and the extract, transform, and load to physically integrate all the data sets. DREE makes a metadata-level integration. At this level the integration is quicker and smaller, and it does not need a great quantity of hard disk or memory space. Although the message exchanging increases the volume of the exchanged information, the load of edge computing nodes is optimized in the LDE, before the application of rules.

The deployment of DREE is simplified by adding a RESTful web service interface to access and replace the traditional services consumed by the inference engine. Additionally, the deployment on the edge nodes is summarized to install a daemon, named ECD, which simplifies the access to information in the local nodes, notifying any modification or updating in the different data sets, by updating the metadata model in the LDE. Thus, DREE reacts to any change in the data sets located on edge nodes.

Finally, the proposed DREE is independent from the ES, providing the possibility to run simultaneously several ESs. Thus, the processing load balancing is automatically provided by the information dissemination around the nodes.

Author details

Juan Ignacio Guerrero Alonso*, Enrique Personal, Antonio Parejo, S. García, Antonio Martín and Carlos León
Department of Electronic Technology, University of Seville, Seville, Spain

*Address all correspondence to: juaguealo@us.es

References

[1] Jabbar HK, Khan RZ. Survey on development of expert system in the areas of medical, education, automobile and agriculture. In: 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom); 2015. pp. 776-780

[2] D'Aquila RO, Crespo C, Mate JL, Pazos J. An inference engine based on fuzzy logic for uncertain and imprecise expert reasoning. Fuzzy Sets and Systems. 2002;**129**(2):187-202

[3] Chojnacki E, Plumecocq W, Audouin L. An expert system based on a Bayesian network for fire safety analysis in nuclear area. Fire Safety Journal. 2019;**105**:28-40

[4] Guerrero JI, León C, Monedero I, Biscarri F, Biscarri J. Improving knowledge-based systems with statistical techniques, text mining, and neural networks for non-technical loss detection. Knowledge-Based Systems. 2014;**71**:376-388

[5] Jimenez ML, Santamaría JM, Barchino R, Laita L, Laita LM, González LA, et al. Knowledge representation for diagnosis of care problems through an expert system: Model of the auto-care deficit situations. Expert Systems with Applications. 2008;**34**(4):2847-2857

[6] Nagata T, Sasaki H. Personal computer based expert system for power system operation education. International Journal of Electrical Power & Energy Systems. 1996;**18**(3):195-201

[7] Végh J. A simple "embedded" reasoning inference engine with application example in the X-ray photoelectron spectroscopy. Computer Physics Communications. 2004;**160**(1):8-22

[8] Qian Y, Li X, Jiang Y, Wen Y. An expert system for real-time fault diagnosis of complex chemical processes. Expert Systems with Applications. 2003;**24**(4):425-432

[9] Magalhães SC, Borges RFO, Calçada LA, Scheid CM, Folsta M, Waldmann A, et al. Development of an expert system to remotely build and control drilling fluids. Journal of Petroleum Science and Engineering. 2019;**181**:106033

[10] Yousefpour A, Fung C, Nguyen T, Kadiyala K, Jalali F, Niakanlahiji A, et al. All one needs to know about fog computing and related edge computing paradigms: A complete survey. Journal of Systems Architecture. 2019;**98**:289-330

[11] Guerrero JI, García A, Personal E, Luque J, León C. Heterogeneous data source integration for smart grid ecosystems based on metadata mining. Expert Systems with Applications. 2017;**79**:254-268

[12] Guerrero JI, Personal E, Parejo A, Monedero I, Biscarri F, Biscarri J, et al. High performance data analysis for non-technical losses reduction. En: Lou J, editor. Smart Grids: Emerging Technologies, Challenges and Future Directions. New York, USA: Nova Science Publishers; 2017. p. 1-45. (Energy Science, Engineering and Technology)

[13] Guerrero JI, García A, Personal E, Parejo A, Pérez F, León C. A Rule-based expert system for heterogeneous data source integration in smart grid systems. En: Ryan D, editor. Expert Systems: Design, Applications and Technology. New York, USA: Nova Science Publishers; 2017. p. 59-104. (Computer Science, Technology and Applications)

[14] Mathematical Markup Language (MathML) Version 3.0 2nd Edition

[Internet]. [cited 10th November 2019]. 2014. Available in: https://www.w3.org/TR/MathML3/

[15] The OpenMath Standard [Internet]. [cited 10th November 2019]. Available in: https://www.openmath.org/standard/om20-2019-07-01/omstd20.html

[16] Guerrero JI, Personal E, García A, Parejo A, Pérez F, León C. Distributed charging prioritization methodology based on evolutionary computation and virtual power plants to integrate electric vehicle fleets on smart grids. Energies. 2019;**12**(12):2402

[17] Monedero I, Martín A, Elena J, Guerrero J, Biscarri F, León C. A practical overview of expert systems in telecommunication networks, medicine and power supplies. In: Expert System Software: Engineering, Advantages and Applications. New York: Nova Science Publishers; 2012. p. 178-210

[18] Guerrero J, Parejo Matos A, Personal E, Monedero I, Biscarri F, Biscarri J, et al. From rule based expert system to high-performance data analysis for reduction of non-technical losses on power grids. International Journal on Advances in Intelligent Systems. 2017;**10**:136-146