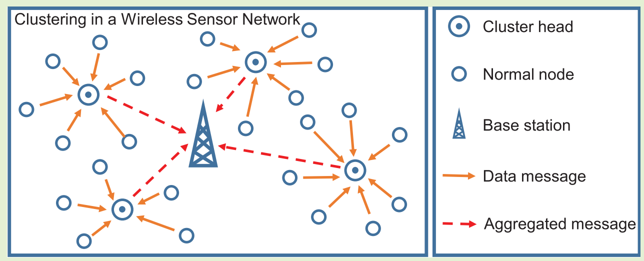
Statistical Normalization for a Guided Clustering Type-2 Fuzzy System for WSN

A. J. Yuste-Delgado, J. C. Cuevas-Martínez, and A. Triviño-Cabrera

***Abstract*—One of the main concerns in Wireless Sensor Networks is the efficient energy management of the nodes. Hierarchical techniques such as clustering have been developed in an effort to solve this problem. In this paper we present a smart evolution of a distributed clustering method that uses a turn-based scheduling cluster head selection process based on an interval Type-2 fuzzy system. The method we propose offers four main improvements. First, the setup process guided by the Base Station is adapted to tune the skip parameter during the network lifetime, which controls how many rounds the clusters are not updated. Second, the normalization of the fuzzy system input variables is carefully performed based on a statistical analysis to reduce the effect of fluctuations in edge values. Third, the value of the coefficient applied to the output of the inner Type-2 fuzzy system is updated to balance the number of cluster heads at early stages. Finally, only the strongest candidate nodes, those with the highest probability, are selected to become cluster heads. The proposed design and scheduling aim to achieve low-energy processing in the nodes. When our proposed techniques are applied, they give better results compared with other similar approaches.**



***Index Terms*—Wireless sensor networks, clustering, dynamic, normalization, Type-2 fuzzy systems.**

I. Introduction

WIRELESS sensor networks (WSNs) are now present in multiple applications linked to environmental monitoring, health monitoring, forest fire prevention, urban applications and military surveillance [1]. Wireless sensor networks are made up of numerous electronic devices known as nodes, which are connected solely by a wireless channel. In the interest area, nodes are used to sense certain magnitudes (e.g. temperature or humidity). These measurements are gathered by the nodes and sent to a special node, called the Base Station (BS), which processes the information received. In most WSN applications, the main problem is the energy consumption and lifespan of the nodes caused by finite power resources (mainly batteries), which are usually very difficult, or even impossible, to recharge or replace. These energy constraints require techniques that reduce power consumption in the nodes. Efficient communications management is the most important technique because the transmission and reception of data consume the most energy in a node. Several strategies have been proposed to cope with this. Clustering, resource allocation, opportunistic transmission, sleep-wake scheduling, routing, hierarchical formation, data gathering and optimal node deployment are the main approaches [2]. Clustering is based on a hierarchical structure in which some nodes, referred to as the Cluster Heads (CH), gather data from the nodes in their group or cluster. The CH sends only one aggregated message per cluster to the BS. By gathering data into a single message, the mean distance that the transmissions must cover is reduced. In turn, their associated energy consumption also decreases.

To increase the lifespan of the whole network, nodes that act as CHs must change during the network lifetime [3], because if the same nodes carry out this function for a long time they will deplete their batteries much faster than other nodes. Selecting active CHs in a WSN is therefore extremely important, as it can impact the network's reliability. CHs can be selected according to a centralized or a distributed approach. To select the CHs, the BS (in a centralized mode) or the nodes (in a distributed mode) can execute different types of algorithms: using turn-based approaches; randomly; based on a mathematical formulation; by a computational intelligence technique; or using any combination of these. Some of the best techniques to select the active CHs are those based on Fuzzy Rule Based Systems (FRBS) [4], [5] because of the complex mathematical models that govern these types of

AQ:1 Manuscript received November 11, 2021; revised February 4, 2022; accepted February 5, 2022. This work was supported by the "Smart and Secure EV Urban Laboratory II" through the Second Own Plan of Smart-Campus of the Universidad de Málaga. The associate editor coordinating the review of this article and approving it for publication was
AQ:2 Dr. Amitava Chatterjee. *(Corresponding author: J. C. Cuevas-Martínez.)*

A. J. Yuste-Delgado and J. C. Cuevas-Martínez are with the Department of Telecommunication Engineering, University of Jaén, 23700 Linares, Spain (e-mail: ajyuste@ujaen.es; jccuevas@ujaen.es).

A. Triviño-Cabrera is with the Department of Electrical Engineering, University of Málaga, 29071 Málaga, Spain (e-mail: atc@uma.es).

Digital Object Identifier 10.1109/JSEN.2022.3150066

1558-1748 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

systems. In particular, Interval Type-2 Fuzzy Systems (IT2FS) can cope with the high level of uncertainty in their inputs [6]. The inputs of these systems must be in the [0-1] interval to be properly combined, so they are usually pre-processed with a normalization.

The contributions of this paper are a set of four improvements to the formerly proposed Distributed Clustering Algorithm Guided by Base Station (DCAGBS) [7]. The first modification relies on an optimized normalization of the input variables of the IT2FS that is run by each node in DCAGBS. In particular, we perform a statistical and local analysis for this task, but using information inferred from the other nodes' state. We have designed the statistical functions to achieve more effective nodes that consume less energy. Also, in DCAGBS the CHs are selected in a distributed mode, but the same selected CHs are used for a number of rounds defined by the $skip$, which is dynamically set and announced by the guiding BS. Building on this, we will improve the network performance by judiciously setting the $skip$ parameter in some important moments of the network lifespan. The third contribution is the addition of a competition phase to select only the best CHs among those candidate nodes that are most likely to become a CH which avoids definitively the selection of bad CHs that could reduce the network lifetime. The fourth update is the new value for the coefficient that is applied to the output of the IT2FS that each node runs. This update reduces the probability of a node becoming a candidate at early stages when most nodes have full batteries and achieve high outputs for the IT2FS. Thus, less node become CHs and the energy spent in the process is balanced in a better way. These four actions define the contributions of this paper and enable the WSN to extend its operative lifetime from the initial version. Indeed, these changes significantly improve the lifespan of the whole WSN compared with DCAGBS, as we shall demonstrate. We apply the proposed techniques to a smart evolution of DCAGBS referred to as Statistically Adaptive with Enhanced normaliZation (SAEZ), but these techniques may also be valid for other similar clustering algorithms.

The rest of the paper is organized as follows: the next section gives details of related work, followed by a description of the proposed clustering algorithm in Section III. Simulation results are analyzed in Section IV. Finally, Section V concludes.

II. Related Work

Clustering methods for WSNs can be divided into two categories according to the procedure by which the CH is selected: stochastic, and computational intelligence-based. In stochastic methods, the CHs are selected based on a mathematical function that computes the probability of a node becoming a CH. That probability is then compared with a random number in order to make a final decision on whether or not the node is a new CH. Alternatively, those based on computational intelligence [8] use intelligent systems such as FRBS, evolutionary systems (e.g. genetic algorithms) or particle swarm optimization systems to determine whether a node is suitable to become a CH. It is also common for the output of those intelligent systems to correspond to the probability of a node becoming a CH [4], [5] [9].

One of the most important clustering methods is the Low-energy adaptive clustering hierarchy (LEACH) [3], which has been analyzed and improved by a wide variety of other approaches. LEACH is a stochastic method in which each node n becomes a CH in a round r, if a self-generated random number is lower than a threshold $T(n)$. One of the most recent versions of LEACH is presented in [10], where the original threshold $T(n)$ can be set to nine different values. The selected value for the threshold depends on the node's energy, its distance to the BS and on two scattered parameters that are related to the average energy of all the nodes in the network and their mean distance to the BS. Another recent modification to the centralized version of LEACH (LEACH-C [3]) is Hybrid approach of Firefly Algorithm with Particle Swarm Optimization (HFAPSO) [11], which tries to find the optimal cluster head selection using the firefly algorithm and particle swarm optimization.

Fuzzy systems are also used for clustering. In [12] the authors present a distributed fuzzy system with two outputs. The first one sets the maximum distance that an announcement can travel when a node announces that it has become a CH. The second output is the threshold used in the algorithm to decide whether the node has to become a new CH. A similar approach is presented in [13] where the output radius of the fuzzy system is used to reduce the transmission range of the announcements made by CHs close to the BS. The exact calculation of all the input variables is not described. In [14] the deployment field is divided into sensing zones in order to separate the possible CHs. In the BS a centralized algorithm based on a fuzzy system then chooses the CH for each zone.

The benefits of IT2FS for clustering in WSNs derive from the high level of uncertainty in these systems. Considering these advantages in centralized approaches, in [15] the authors present a Type-2 fuzzy system to select the nodes best suited to becoming CHs. They then define a communication chain with multiple CHs based on an ant colony optimization (ACO) algorithm. However, it is not clear how this affects the communication cost of the pheromone in the whole system, which could lead to worse results. Another approach based on an IT2FS is Unequal Clustering algorithm based on interval type-2 TSK fuzzy logic theory (UCT2TSK) [16], which defines an unequal clustering algorithm. The values of the output interval of the Type-2 fuzzy system are used firstly to get a rank value used to compete between nodes that are candidates to become a CH, and secondly to set the transmission distance for the announcements. Nevertheless, it is possible that with a transmission distance that does not cover the deployment area, a CH may not reach any node to announce that most nodes are dead. In [9] the authors propose a distributed IT2FS clustering algorithm, similar to [16], which selects the best CHs among the candidates within the competition radius inferred by the IT2FS. Nevertheless, the results obtained seem poor compared with other methods.

With regard to normalization, we should point out that the previous works on clustering methods do not cover the effects of the input variable normalization in the resulting algorithm.

IEEE Proof

In [12] and [13], the authors do not study normalization, while other approaches such as [15], [16] [9] merely divide the crisp value into the maximum reference value, without any justification. Alternatively, in [14] the authors design a normalization that uses the range of possible values of the variable as a denominator instead of just its maximum value.

DCAGBS [7] is a distributed BS-guided clustering algorithm, in which each node runs a Tagaki–Sugeno–Kang IT2FS in order to determine whether to become a CH or not. The Type-2 fuzzy system used to obtain the output interval in each node is based on these four input variables:

- E_r*:* residual energy of the node in round r.
- N_{CHr}*:* number of times that a node has been selected as a CH out of the total number of announcements from surrounding CHs in round r.
- Er_{CH}*:* average energy of all the nodes that were CHs in the previous selection round.
- R_{noCH}*:* the relation between the number of rounds since the last round in which the node was selected as a CH, and the total number of nodes in the network.

Selecting these variables in DCAGBS means that it can be implemented in very simple devices because it only relies on historical information instead of geolocalization hardware, estimation of Received Signal Strength Indicator (RSSI) from input messages, or the extra communications required to estimate a node's degree of centrality. The input variables are normalized taking into account only the highest possible crisp value of the magnitude. The IT2FS in each node then obtains an output interval (o_l, o_h), where $o_l < o_h$, in a round-based schedule. From that interval a threshold $T(n)$ is derived, which is used later to decide locally whether or not to become a CH. Once each node has inferred its interval and selected the threshold $T(n)$, it generates a random number and becomes a CH if that number is lower than the inferred threshold. Normal nodes then send an ascription message to the closest CH with their measurements.

To avoid excessive ascription messages from normal nodes, we do not repeat the process of selecting CHs for each round; instead we maintain the selected CHs for a number of rounds. That number is a setup parameter known as $skip$. The $skip$ value must be carefully set because if its value is too low then the selected CHs may be very similar or even the same as the previous set of CHs. However, if the $skip$ value is too high, the nodes that act as CHs will use up too much energy because they will not change for that period of time [17].

Taking into account the fact that the number of active nodes decreases during the network lifetime, in DCAGBS the BS sends two setup messages to change the $skip$ parameter. Adjusting the network behavior in this way optimizes the power consumption. At the beginning, the $skip$ is set to 5% of the number of nodes. When the first node dies, it adjusts to 2.5% and when half of the nodes have died, it changes to 2.

For message scheduling, most approaches rely on the MAC layer, particularly if it is similar to IEEE 802.15.4 as it can be adapted to use a TDMA-based scheduling to avoid synchronization messages in the upper layers [18]. In the event of asynchronous messaging, it may be necessary to retransmit lost packets in networks with a low number of nodes, whereas in networks with a high node redundancy, one packet loss can be ignored to avoid the extra cost of the new message. Nevertheless, the decision of whether or not to use retransmissions is the result of a trade-off between reliability and redundancy [19]. The cost of synchronization or retransmission messages is not usually included in clustering approaches. For synchronous messages, MAC layer is usually responsible for this synchronization [3]. For asynchronous messages, retransmissions depend on random losses, which are not taken into account either. In DCAGBS, the synchronization is achieved through the advice messages that CHs send per round, in collaboration with the messages that the BS sends to the whole network.

In the algorithm we propose in this paper – SAEZ – we apply four statistically based techniques used in DCAGBS in order to achieve a better performance by adding more significant information to properly select the CH.

III. Proposed Clustering Algorithm

The method we propose (SAEZ) employs an IT2FS in each node to obtain the probability threshold $T(n)$ of becoming a CH. As mentioned previously, using an IT2FS helps us to deal with uncertainties from input variables. For instance, measuring the exact residual energy relies on node electronics, which has intrinsic tolerance. Moreover, the other three variables depend on the correct reception of messages that could be missed or corrupt. Therefore, defining intervals in IT2FS allows us to deal better with tolerances in probes or electronics, and with errors in wireless communications. Because it takes these inaccuracies into account, IT2FS makes it easier to design the knowledge base.

A. Interval Type-2 Fuzzy System

Fuzzy logic, originally proposed by Zadeh [20], [21] is a computational intelligence technique that is applied to a wide range of applications in industry, economics and telecommunications. It is used when it is not possible to find a mathematical relationship between the inputs and the outputs of a system, or to process a studied system, particularly when all the data needing to be manipulated is affected by uncertainty and imprecision. Therefore, in fuzzy logic, that uncertainty or vagueness with regard to linguistic issues is replaced by mathematical expressions modeled by a set of IF-THEN rules and fuzzy sets. This was originally presented by Zadeh [22] as a membership function. Thus, Type-2 fuzzy sets are presumed to model that uncertainty in a better way [6]. Those Type-2 fuzzy sets define an upper membership function and a lower membership function, which can each be seen as a Type-1 fuzzy set. The interval between each membership function, which represents the uncertainty in that variable, is known as the footprint of uncertainty (FOU). In turn, this uncertainty depicts the difficulties to model or to know exactly how a process will behave. Therefore, the sources of that uncertainty can come from inaccurate measurements, random process dynamics, erroneous model specification, imprecision, etc. Thus, the larger the FOU, the higher the uncertainty in the system.

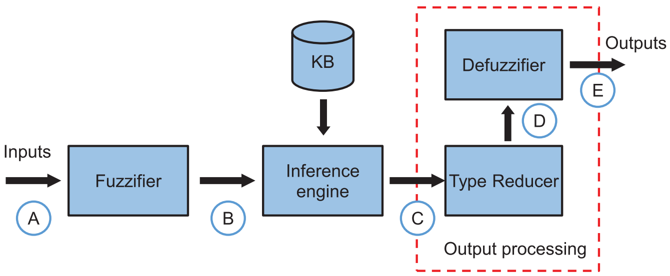


Fig. 1. System diagram of IT2FS.

The block diagram of an Interval Type-2 Fuzzy System is depicted in Fig. 1. Its operation mode is detailed as follows:

- A The crisp values of the input variables are fuzzified with Type-2 fuzzy sets. Additionally, prior to fuzzification, input values must be normalized, i.e. their values must range from 0 to 1. The normalization should depend on the range of the variable.
- B The IF-THEN rules in the Knowledge Base (KB) bind together the fuzzified inputs to obtain an output.
- C The inference process binds the inputs, the rules and the outputs.
- D The type-reducer block reduces an interval Type-2 fuzzy to a Type-1 set.
- E The last block of the system obtains the output crisp values as an interval. In our case this interval is named (o_l, o_h), as mentioned earlier. This output interval will be used to compute the probability of a node becoming a CH.

The Type-2 fuzzy sets employed in the present approach are shown in Fig. 2 where L stands for Low, M for Medium and H for High. The complete KB is composed of 81 rules that can be found in [7].

In addition, to increase the node performance and avoid running an IT2FS in each node for every round, the results of the IT2FS are obtained previously in an external machine and stored in the local memory of each node. Specifically, the IT2FS algorithm is executed for a combination of potential input data before the WSN is deployed. The nodes' memory is then updated with these results for their future use. Admittedly, the results do not cover all the possible solutions because they are obtained from a quantified set of values of the input variables, but we work with the approximated value in order to reduce the energy and the computational resources used for the complete IT2FS operation. In this way, the process of obtaining an output of the IT2FS can be accomplished exclusively by a memory access, which is a simpler operation in terms of energy and computational consumption. Moreover, our tests have demonstrated that the error produced in the output due to the quantification of the input variables is negligible.

Taking into account the main features of DCAGBS, the enhancements we propose in SAEZ affect the following aspects:

- Normalization of the input variables of the IT2FS.
- Tuning of the $skip$ parameter.
- Setup of the output of the IT2FS.
- Competition process to select final CHs.

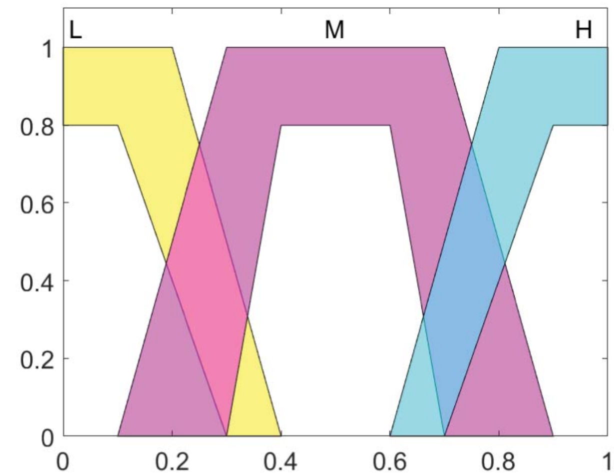


Fig. 2. Fuzzy sets for input variables of the IT2FS of DCAGBS.

Those four improvements are described in the next four subsections.

B. Normalization of the Input Variables of the IT2FS

This modification affects the normalization of three of the input variables of the IT2FS that runs in each node: E_r, N_{CHr}, and R_{noCH}, but it does not affect Er_{CH} variable because it is already a global average value that is calculated by the BS and delivered to all nodes. Considering the reviewed bibliography, clustering algorithms that employ fuzzy systems do not detail the normalization of their input variables, as mentioned previously. Moreover, the implemented normalization in revised bibliography is just a linear function whose output ranges from 0 to 1 in most cases. In contrast, SAEZ takes into account crisp values of local input variables and other information gathered from announcement messages that a candidate CH sends when trying to become a CH. Thus, the information from other nodes is used to define the 10% percentile and the 90% percentile for each of those three variables. These new fields in the announcement message, which do not significantly increase its size, are:

- Residual energy E_r of the sending node (already included in DCAGBS).
- Total number of times that the node was selected as a CH.
- Number of rounds since the last time the node was selected as a CH.

Consequently, each node computes for N_{CHr}, Er_{CH}, and R_{noCH} the 10% percentile p_{10} and the 90% percentile p_{90} having gathered all these new values from the announcements. The percentile function used is defined in (2), but it previously needs the calculation in (1).

$$x = \frac{m \cdot i}{100} \tag{1}$$

where m stands for the total number of values, or elements, gathered from the received announcement messages for that variable and i is the desired percentile, e.g. 10 or 90. Thus, the percentile i or p_i for an ordered list of elements is the element whose value is the boundary for the $(i-1)_{th}$ lower elements. The exact value of p_i is obtained as follows:

$$p_i = \begin{cases} element(E+1) & if\ D \neq 0 \\ \dfrac{element(E) + element(E+1)}{2} & if\ D = 0 \end{cases} \tag{2}$$

where E is the integer part of the previously calculated x and D is the decimal part of x.

Then a crisp value υ is normalized in a new value z for each of those three variables as follows:

$$z = \begin{cases} 0 & if\ \upsilon < p_{10} \\ \dfrac{\upsilon - p_{10}}{p_{90} - p_{10}} & if\ p_{10} \geq \upsilon \leq p_{90} \\ 1 & if\ \upsilon > p_{90} \end{cases} \tag{3}$$

We selected this type of normalization to avoid fluctuations in edge values. Therefore, crisp values for a variable under 10% percentile are considered as bad values. In the other hand, values over 90% percentile are all considered good values. Moreover, this method has low computational time required to execute it. In fact, it is implemented in two stages:

1) The received values for the announcement messages are inserted in an ordered array.
2) A division and a multiplication are applied to identify the positions of the element in the 10% position and in the 90% position.

With a counting sort algorithm, the computational cost is $\Theta(n)$, with n being the number of data associated with the variable.

Following the normalization presented in (3), we have to remark that the information used for the normalization comes for nodes that participated previously in the CH selection process. Consequently, input variables of the IT2FS are adapted to the current network status which is expected to be better than if a local data-driven linear function was used.

The 10% and 90% percentiles were selected because of the requirement that the normalization should only affect a limited range of the crisp value of each variable. This limitation is due to the fact that normalization is computed with values gathered from other nodes, which are prone to error. As a consequence, the filtered data should not be extensive. For the precise values of these lower and upper limits, we carried out a set of experiments with WSNs similar to those shown in the Results section for different percentile ranges. The selected thresholds led to the best performance. As was previously commented, those values beneath the 10% threshold that usually come from nodes with 'poor' probability of become CH will be considered too low, and with the 90% percentile it allows nodes closer to the BS, which usually have a strong chance of becoming CHs and whose batteries deplete faster, to now compete with other nodes in the top 10% with a similar probability.This will result in a better balance among the CH candidates because the energy spent on performing as a CH is divided by more nodes. Consequently, this will delay the instant at which the first node completely depletes its battery. The flowchart depicted in Fig. 3 illustrates for DCAGBS how this process is executed in each node in a round basis scheduling. As we can observe, the probability of becoming a CH $T(n)$ determines the behavior of the node as either a CH or a normal node (this process is also modified, as will be discussed later in Section III-E). We should also note that when a node receives an announcement message from a CH, it records the values that the message conveys to make the percentile calculation and normalization feasible in the next round. In addition, this process is modified in the SAEZ method as we will see in the fourth improvement to DCAGBS later in the paper.

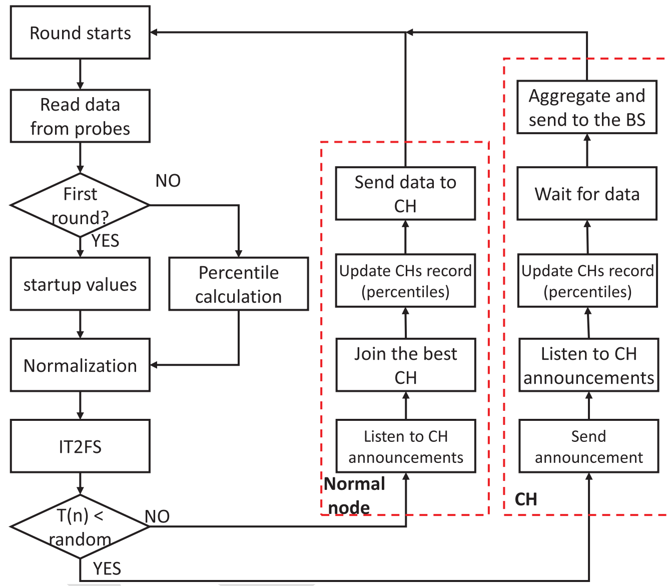


Fig. 3. Flowchart for the operation of each node in the network in DCAGBS.

C. Tuning the skip Parameter

We detected that there are periods in which nodes can frequently alternate as CHs, as they have similar amount of battery left. In contrast, there are other time intervals in which CHs do not need to change so frequently because some nodes have very low battery levels and therefore cannot take on this role. With this feature in mind, the second improvement involves tuning the $skip$ parameter differently during the network lifetime. Thus, the schedule and values of the $skip$ parameter, which continues to be announced by the BS, are updated more times than in DCAGBS. The new instants at which the BS sends the messages are carefully selected, and represent important changes in the network, as detailed below:

- The initial value of $skip$ is set to 5% of the nodes, as in DCAGBS, to allow CHs to be selected more frequently, which should balance the network's energy consumption because all the nodes are still alive.
- When the first node depletes its battery (dies), the value of $skip$ is set to 20% of the nodes (2.5% in DCAGBS). In the early stages, when most of the nodes are full of energy, a lower value of $skip$ is used to select CHs frequently and to balance the energy cost of becoming a CH. However, when the first node dies, there is a chance that several other nodes are also close to depleting their batteries. Thus, the skip value is increased to delay the selection of CHs, which helps to save the energy cost of sending and receiving unnecessary announcement

messages. Preventing these messages would effectively increase the network lifespan.

- When 75% of the nodes have died, the value of $skip$ is set to 2 because in this situation, in which most of the nodes are working on very low power and dying frequently, the rotation of CHs has to be updated more quickly in order to find the best CH among the survivors. This instant is delayed from the original DCAGBS, which sets the $skip$ to 2 when 50% of the nodes are dead. We made this change after observing that the redundancy of WSN allows for a higher $skip$ for a longer period.

D. Setting up the Output of the IT2FS

The third modification is the update of the coefficient applied to the output interval(o_l, o_h) in order to obtain the final threshold $T(n)$. In DCAGBS, the value of $T(n)$ is calculated as follows:

- if the node was selected as a CH in the previous round $T(n) = o_l/c_l$,
- in other cases $T(n) = o_h/c_h$

where $c_l = 3$ and $c_h = 2$.

Thus, if the value of c_l or c_h increases, the probability of becoming a CH decreases. Consequently, we have conducted an empirical study to determine which values would be the best for c_l and c_h. After analyzing the results, we have observed that with the same value is set for both coefficients and in particular with $c_l = 4$ and $c_h = 4$, the lifetime of the network increases.

E. Competition Process to Select the Final CHs

The fourth change to DCAGBS affects the process depicted in Fig. 3 by introducing an intermediate state after a node selects itself as a potential CH (once they have generated a random value which is lower than $T(n)$) and before it actually becomes a CH. In that new state, nodes consider themselves to be candidate nodes, not real CHs, because only those with the best $T(n)$ will be finally selected as CHs. As can be seen in the new scheduling in Fig. 4, when a node sends its announcement message it listens to the announcement messages of the other candidates. Then it compares its $T(n)$ with the other $T(n)$ values received in the announcement messages from other candidate nodes. Finally, each candidate node checks if its $T(n)$ value is in the top 5% and, if so, it becomes a CH. Candidate nodes that fall under the top 5% become normal nodes and consequently have to join the closest CH. The top 5% of nodes is chosen according to Heinzelman *et al.* in [3], which calculates that the optimum number of CHs for a network of randomly deployed sensors is 5%.

The results obtained through these four modifications are shown in the next section.

IV. Simulation and Results

Although they can be included in other clustering techniques, for evaluation purposes the four improvements are incorporated into DCAGBS leading to the SAEZ algorithm. SAEZ is compared with DCAGBS and three other distributed approaches: LEACH [3], a classical distributed stochastic method; LBFBUC [13], a fuzzy logic based on an unequal clustering algorithm for WSN; and UCT2TSK [16]. To achieve a valid comparison, a series of simulations was carried out in Matlab with all the algorithms.

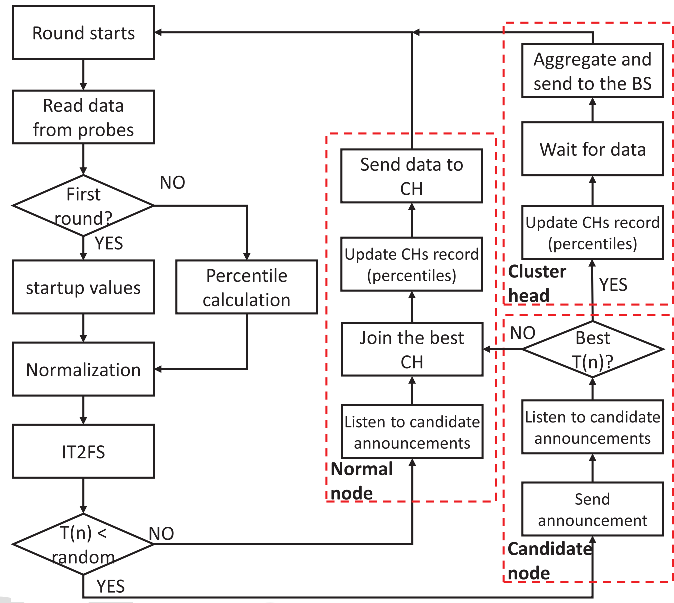


Fig. 4. Flowchart for the operation of each node in the network in SAEZ.

A. Energy Model

The experiments we carried out to compare the proposed SAEZ method with the methods listed previously use the first order radio model detailed in [3]. This model simulates data communications by radio devices, which is widely used in the related literature. It is described in equations (4)-(6). This energy model assumes that the energy spent in the transmitter and receiver can be calculated based on the length in bit of the message and the distance to the destination, taking into account free space and multipath fading channels. Thus, the energy spent in transmitting information is detailed in (4) whereas the energy consumed in receiving data is shown in (6).

$$E_{Tx}(l,d) = f(x) = \begin{cases} l \cdot (E_{elec} + E_{fs} \cdot d^2), & d \le d_0 \\ l \cdot (E_{elec} + E_{mp} \cdot d^4), & d > d_0 \end{cases} \quad (4)$$

$$d_0 = \sqrt{\frac{E_{fs}}{E_{mp}}} \quad (5)$$

$$E_{Rx}(l) = E_{elec} \cdot l \quad (6)$$

where:

- l is the number of bits of the message.
- E_{elec} is the energy in Joules that the transmitter or the receiver circuitry consumes for each bit sent or received respectively.
- d is the distance in meters between the sender and the receiver of the message.
- E_{fs} is the energy in Joules consumed by the amplifier to obtain an acceptable bit error rate according to the free space model ($d \le d_0$).
- E_{mp} stands for the energy in Joules consumed by the amplifier to obtain an acceptable bit error rate in the multi-path (mp) model ($d \le d_0$).

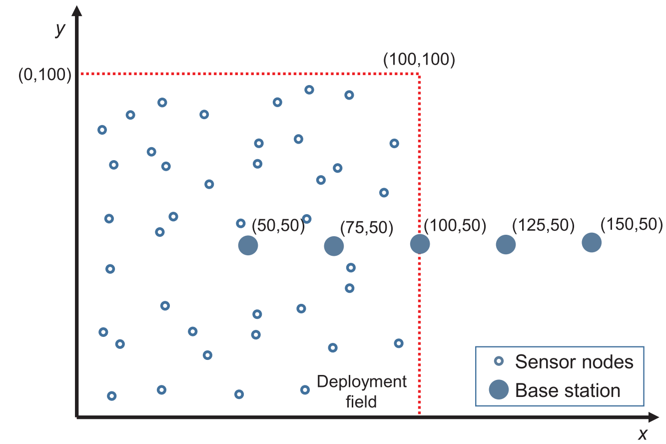


Fig. 5. Diagram for the different BS locations for the experiment.

This model also includes the energy spent on receiving and aggregating the data in a CH [23]. Aggregation is needed to avoid sending the measured data for each node independently. Thus, only a summary of those values, which is usually the result of a statistical operation, is sent to the BS. However, this process also consumes energy per bit of the message to be aggregated, which is defined in (7):

$$E_{Rx-DA} = (E_{elec} + E_{DA}) \cdot l \tag{7}$$

where:

- E_{DA} is the energy in Joules spent by the processing unit of a CH when it aggregates the data received from a contributing node.
- l is the number of bits of the aggregated message.

B. Experiment Setup

The scenario for all the simulations is a 100 m × 100 m square area where 250 homogeneous nodes were randomly deployed and all the nodes after deployment are stationary and have the same initial energy. In addition, the simulations were run with five different BS locations from the centre to outside of the deployment area as seen in Fig. 5. The BS is also stationary once it is deployed and is supplied with an unlimited power source. The exact locations of the tested BS positions are the following:

- BS located at coordinates (50,50) m.
- BS located at coordinates (75,50) m.
- BS located at coordinates (100,50) m.
- BS located at coordinates (125,50) m.
- BS located at coordinates (150,50) m.

The network model used supposes that each node can transmit and receive through a symmetric communication channel with no interference. Normal nodes, CHs and BSs send their messages in a single hop transmission using TDMA. The transmission power used by announcement messages is fixed and set to the minimum transmission power needed to reach the furthest part of the deployment field. The transmission power for messages with information from normal nodes to CHs, and from CHs to the BS, is set to the minimum power needed to reach the destination. That can be achieved because normal nodes can estimate the power needed to reach a CH by obtaining the RSSI of the announcement messages, which are always transmitted with the same transmission power. The same estimation can be accomplished by the BS, which sends a start up message to allow that calculation.

TABLE I
EXPERIMENT SETUP PARAMETERS

Parameter	Value
Initial energy of nodes	0.5 J
Length of control message	200 bits
Length of data message	4000 bits
E_{elec}	50 nJ/bit
E_{DA}	5 nJ/bit
E_{fs}	10 pJ/bit/m^2
E_{mp}	0.0013 pJ/bit/m^4

TABLE II
FND FOR DIFFERENT BS LOCATIONS

	(50,50)	(75,50)	(100,50)	(125,50)	(150,50)
LEACH	708.5	712.3	703.9	700.9	661.6
UCT2TSK	1155.7	1123.8	1030.3	923.3	798.1
LBFUC	867.8	819.5	908.7	920	928.8
DCAGBS	1541.5	1518.2	1403.5	1228.1	878.2
SAEZ	16195	1639.3	1594.4	1572.6	1488.7

TABLE III
HND FOR DIFFERENT BS LOCATIONS

	(50,50)	(75,50)	(100,50)	(125,50)	(150,50)
LEACH	859.8	858.7	851.1	834.9	800.7
UCT2TSK	1350.3	1322.1	1224.3	1095.2	948.7
LBFUC	1921.4	1926.3	1910.1	1865.3	1797.5
DCAGBS	2021.3	2000.4	1939.8	1820.9	1628.2
SAEZ	2121	2108	2098.6	2043.1	1957

The parameters for the experiments and for the first order radio model detailed in Section IV-A are shown in Table I. For all the tested methods, the communications between the nodes and the BS are assumed to operate over a TDMA-based MAC layer protocol that allows a round based scheduling with full duplex communication channels. Communications are also supposed to be free from error and interference so as to allow a fair comparison with all the methods.

The performance parameters used to compare each method, which are commonly used in the bibliography, are the following:

- The round when the first node dies or FND (First Node Dies)
- The round when half of the nodes have died or HND (Half of Nodes Die)
- The round when only 10% of the nodes are alive or LND (Last Node Dies)

To run the simulations and obtain the results, we created a set of 30 different node locations, called maps. As described earlier, in each map 250 nodes are randomly deployed within the deployment area. Thus, each method is run 5 times for the 30 different maps (once for each BS location). For each BS location, the mean values of the results of the 30 simulations are analyzed to obtain the three performance parameters: FND, HND and LND. In Tables II-IV we include the results for the average values of those 30 simulations for each method and each different location of the BS.

As can be observed in the simulation results (Tables II-IV), the proposed SAEZ method performs better overall than any other algorithm, achieving a good trade-off between the three parameters. Consequently, the four modifications help

IEEE Proof

TABLE IV
LND FOR DIFFERENT BS LOCATIONS

	(50,50)	(75,50)	(100,50)	(125,50)	(150,50)
LEACH	960.4	956.1	947.6	930.8	919.46
UCT2TSK	1521.3	1511.1	1426.3	1359.5	1251
LBFUC	2172.9	2171.7	2154.7	2140.9	2107.6
DCAGBS	2235	2216.7	2167.9	2081.6	1952
SAEZ	2236.9	2230.9	2215.3	2174.4	2115.7

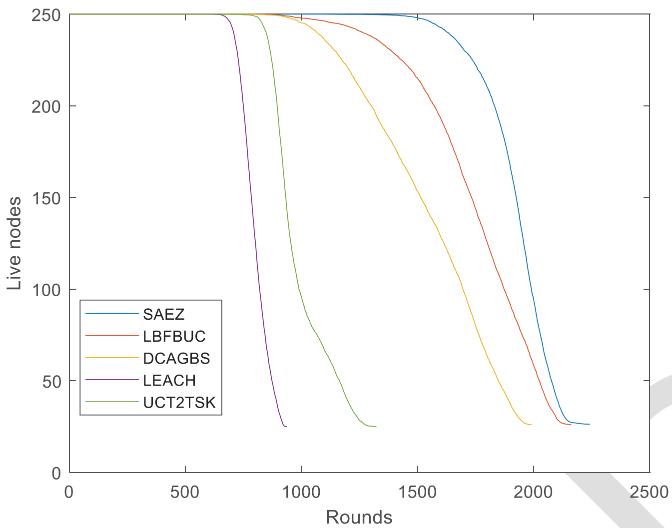


Fig. 6. Number of alive nodes per round for BS at (150,50).

TABLE V
AVERAGE SIMULATION TIME FOR BS LOCATED AT (150,50) m

LEACH	UCT2TSK	LBFUC	DCAGBS	SAEZ
16.71 s	91.09 s	20.58 s	62.05 s	30.92 s

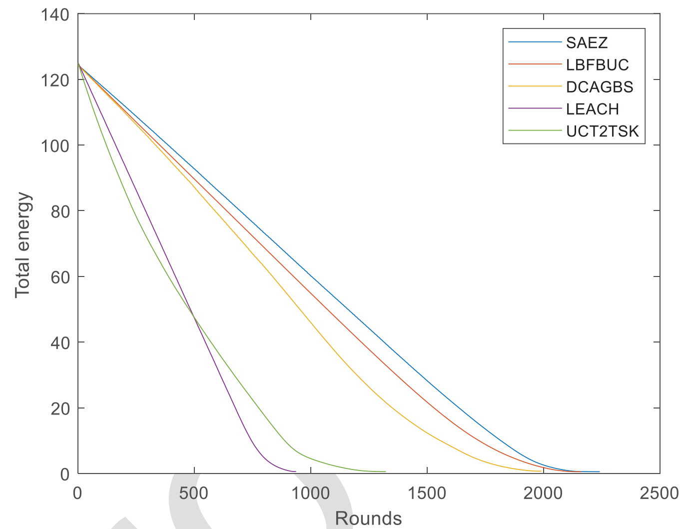


Fig. 7. Average total energy in the network per round.

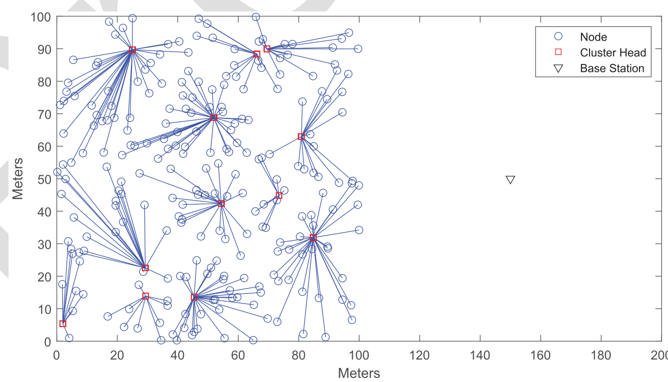


Fig. 8. Alive nodes and selected CHs in a round for BS at (150,50) m.

SAEZ to achieve more balanced results for the three parameters. As an illustrative example, those results are graphically depicted in Fig. 6 and Fig. 7 for the BS located at coordinates (150,50), which is the worst case in terms of distance and energy consumption, although the graphs for other locations are similar. Thus, Fig. 6 shows the average number of alive nodes per round for the five compared algorithms for all the simulations, while Fig. 7 shows the average total energy in the network per round. Both figures indicate that the SAEZ algorithm performs significantly better than the other methods, delaying the instant of the first death considerably and keeping almost a vertical gradient (Fig. 6) that confirms a good energy balance among the network nodes. To illustrate the CH selection process of SAEZ, Fig. 8 depicts alive nodes and selected CHs in a round at early stages for one map with the base station located at (150,50) m.

In addition, Table V shows the average time spent from the instant when the first data packet is sent to the moment when the last sensor node is dead for the 30 maps with the BS located at (150,50) m. As can be seen in the Table V, the SAEZ method achieves the best results among Type-2 fuzzy methods whereas LBFUC gets the best trade-off for performance-time.

We can conclude that statistical normalization is only effective when a significant amount of data is considered, i.e. when a relevant number of candidate nodes exchange information. As nodes die, these circumstances do not materialize and, in turn, normalization is not practical. Alternatively, the new adjustment of the $skip$ parameter has proven convenient for the middle and final periods of the network's lifetime. Selecting the best candidate nodes allows SAEZ to avoid CHs with low $T(n)$ (poor potential CHs) when there are enough candidates. However, when the number of nodes decreases, the selection could also include those candidates with poor $T(n)$ and thus balance the energy consumption among all the nodes. Finally, tuning the c coefficients reduces the number of potential candidate nodes at early stages, avoiding announcement messages that represent the main energy cost for a candidate node. As a result, SAEZ gives a more balanced performance that improves on the other methods in almost all the parameters and scenarios. As for normalization, other ranges for the percentiles have been tested but yield worse results.

V. CONCLUSION

As described, clustering is one of the main techniques used to improve the lifespan of WSNs by achieving more efficient energy use. In this field, artificial intelligence techniques are well suited due to the uncertainties of the problem. Therefore, Type-1 and Type-2 fuzzy systems have been applied in numerous clustering approaches. However, little attention has been paid to the normalization of the input values needed for those fuzzy systems. In addition, the processing complexity of fuzzy systems must be taken into consideration due to the additional energy cost that their execution incurs in a node. In this paper, we first present a new normalization method based on a statistical analysis adapted for low-energy processing. Second, we propose a dynamic setup of the $skip$ parameter to adapt the validity time (number of rounds) of the state data exchanged among the nodes. For illustrative purposes, we incorporate the two techniques into DCAGBS, an unequal clustering algorithm based on an IT2FS. The final protocol, referred to as SAEZ, achieves better results than the other methods, but with a very low increase in the complexity of the tasks performed by the nodes. The suitability of the statistical function is related to the amount of data analyzed, and an initial collection time is required to operate the system correctly.

References

[1] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, "Applications of wireless sensor networks: An up-to-date survey," *Appl. Syst. Innov.*, vol. 3, no. 1, p. 14, Feb. 2020.

[2] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. H. Hanzo, "A survey of network lifetime maximization techniques in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 828–854, 2nd Quart., 2017.

[3] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[4] I. Gupta, D. Riordan, and S. Sampalli, "Cluster-head election using fuzzy logic for wireless sensor networks," in *Proc. 3rd Annu. Commun. Netw. Services Res. Conf.*, 2005, pp. 255–260.

[5] J.-M. Kim, S.-H. Park, Y.-J. Han, and T.-M. Chung, "CHEF: Cluster head election mechanism using fuzzy logic in wireless sensor networks," in *Proc. 10th Int. Conf. Adv. Commun. Technol. (ICACT)*, vol. 1, 2008, pp. 654–659.

[6] O. Castillo, P. Melin, J. Kacprzyk, and W. Pedrycz, "Type-2 fuzzy logic: Theory and applications," in *Proc. IEEE Int. Conf. Granular Comput. (GRC)*, Nov. 2007, p. 145.

[7] A.-J. Yuste-Delgado, J.-C. Cuevas-Martinez, and A. Triviño-Cabrera, "A distributed clustering algorithm guided by the base station to extend the lifetime of wireless sensor networks," *Sensors*, vol. 20, no. 8, p. 2312, Apr. 2020.

[8] A. Kaur, P. Gupta, and R. Garg, "Soft computing techniques for clustering in WSN," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1022, Jan. 2021, Art. no. 012041.

[9] M. Adnan, T. Ahmad, and T. Yang, "Type-2 fuzzy logic based energy-efficient cluster head election for multi-hop wireless sensor networks," in *Proc. IEEE Asia Pacific Conf. Wireless Mobile (APWiMob)*, Apr. 2021, pp. 32–38.

[10] G. Kia and A. Hassanzadeh, "A multi-threshold long life time protocol with consistent performance for wireless sensor networks," *AEU, Int. J. Electron. Commun.*, vol. 101, pp. 114–127, Mar. 2019.

[11] B. Pitchaimanickam and G. Murugaboopathi, "A hybrid firefly algorithm with particle swarm optimization for energy efficient optimal cluster head selection in wireless sensor networks," *Neural Comput. Appl.*, vol. 32, no. 12, pp. 7709–7723, 2020.

[12] D. Agrawal and S. Pandey, "FUCA: Fuzzy-based unequal clustering algorithm to prolong the lifetime of wireless sensor networks," *Int. J. Commun. Syst.*, vol. 31, no. 2, pp. 1–18, Jan. 2018.

[13] D. Agrawal and S. Pandey, "Load balanced fuzzy-based unequal clustering for wireless sensor networks assisted Internet of Things," *Eng. Rep.*, vol. 2, no. 3, Mar. 2020, Art. no. e12130.

[14] A. Hamzah, M. Shurman, O. Al-Jarrah, and E. Taqieddin, "Energy-efficient fuzzy-logic-based clustering technique for hierarchical routing protocols in wireless sensor networks," *Sensors*, vol. 19, no. 3, p. 561, Jan. 2019.

[15] Q.-Y. Zhang, Z.-M. Sun, and F. Zhang, "A clustering routing protocol for wireless sensor networks based on type-2 fuzzy logic and ACO," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2014, pp. 1060–1067.

[16] Y. Tao, J. Zhang, and L. Yang, "An unequal clustering algorithm for wireless sensor networks based on interval type-2 TSK fuzzy logic theory," *IEEE Access*, vol. 8, pp. 197173–197183, 2020.

[17] J.-C. Cuevas-Martinez, A.-J. Yuste-Delgado, A.-J. Leon-Sanchez, A.-J. Saez-Castillo, and A. Triviño-Cabrera, "A new centralized clustering algorithm for wireless sensor networks," *Sensors*, vol. 19, no. 20, p. 4391, Oct. 2019.

[18] M. Anwar, Y. Xia, and Y. Zhan, "TDMA-based IEEE 802.15.4 for low-latency deterministic control applications," *IEEE Trans. Ind. Informat.*, vol. 12, no. 1, pp. 338–347, Feb. 2016.

[19] M. A. Mahmood, W. K. G. Seah, and I. Welch, "Reliability in wireless sensor networks: A survey and challenges ahead," *Comput. Netw.*, vol. 79, pp. 166–187, Mar. 2015.

[20] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.

[21] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 1, pp. 28–44, Jan. 1973.

[22] L. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning. Part I," *Inf. Sci.*, vol. 8, no. 3, pp. 199–249, 1975.

[23] G. Dhand and S. S. Tyagi, "Data aggregation techniques in WSN: Survey," *Proc. Comput. Sci.*, vol. 92, pp. 378–384, Jan. 2016.

A. J. Yuste-Delgado was born in Málaga, Spain, in 1970. He received the M.S. degree in telecommunication engineering and the Ph.D. degree in telecommunication from the University of Málaga, Spain, in 1994 and 2012, respectively. He works as an Associate Professor with the Department of Telecommunication Engineering, University of Jaén, Spain. His main research interests include wireless sensor networks and applications of fuzzy systems in computer networks. He is currently a member of the Research Group Telematics System Engineering Group and the Center of Advanced Studies in Information and Communication Technologies (CEATIC), University of Jaén.

J. C. Cuevas-Martínez was born in Mancha Real, Jaén, Spain, in 1976. He received the B.S. degree in telecommunication engineering from the University of Jaén, Spain, in 1998, the M.S. degree in telecommunication engineering from the University of Málaga, Spain, in 2002, and the Ph.D. degree in telecommunication engineering from the University of Jaén in 2014. He is also the Head of the Department of Telecommunication Engineering. He has been working as an Associate Professor with the Department of Telecommunication Engineering, University of Jaén, since 2003. His main research interests include wireless sensor networks and networking and soft computing. He is currently a member of the Research Group Advances in Intelligent Systems and Applications at the University of Jaén.

A. Triviño-Cabrera was born in Málaga, Spain, in 1978. She received the M.S. degree in telecommunication engineering and the M.S. degree in computer science engineering from the University of Málaga, Spain, in 2002 and 2008, respectively. Her thesis, which she defended in 2007, focused on wireless networking. Part of her research work has been conducted in prestigious centers, such as the Samsung Advanced Institute of Technology, South Korea, in 2004, the Laboratory of Communications, University of Coimbra, in 2007, Orange Labs Research and Development, France, in 2009, and Aston University, U.K., in 2017. In 2010, she started teaching with the University of Málaga, where she currently holds a position as an Associate Professor.

IEEE Proof