Neural Computing and Applications (2021) 33:4217–4228
https://doi.org/10.1007/s00521-020-05253-3

ORIGINAL ARTICLE

Source localization in resource-constrained sensor networks based on deep learning

S. Hamed Javadi[1] · Angela Guerrero[1] · Abdul M. Mouazen[1]

Received: 7 April 2020 / Accepted: 25 July 2020 / Published online: 3 August 2020

Abstract
Source localization with a network of low-cost motes with limited processing, memory, and energy resources is considered in this paper. The state-of-the-art methods are mostly based on complicated signal processing approaches in which motes send their (processed) data to a fusion center (FC) wherein the source is localized. These methods are resource-demanding and mostly do not meet the limitations of motes and network. In this paper, we consider distributed detection where each mote performs a binary hypothesis test to detect locally the existence of a desired source and sends its (potentially erroneous) decision to FC during just one bit (1 indicates source existence and 0 otherwise). Hence, both processing and bandwidth constraints are met. We propose to use an artificial neural network (ANN) to correct erroneous local decisions. After error correction, the region affected by the source is specified by nodes with decision 1. Moreover, we propose to localize the source by deep learning in FC which converts the network of decisions 1 and 0 to a black and white image with white pixels in the locations of motes with decision 1. The proposed schemes of error correction by ANN (ECANN) and source localization with deep learning (SoLDeL) were evaluated in a fire detection application. We showed that SoLDeL performs appropriately and scales well into large networks. Moreover, the applicability of ECANN in delineation of farm management zones was illustrated.

Keywords Artificial neural network (ANN) · Decentralized detection · Deep learning · Error type I · Error type II · Internet of things (IoT) · Source localization · Target tracking · Wireless sensor networks (WSN)

1 Introduction

The applications of the networks of wireless smart motes with sensing, processing, and communication capabilities are expanding. These wireless sensor networks (WSNs) are the basis of the emerging technology of internet of things (IoT) and specifically fit very well in surveillance applications [15, 16] where detecting an event source in a region of interest (ROI) and tracking it are important challenges.

For event detection and localization in a region, nodes[1] send their either raw or processed observations to a fusion center (FC) where the final assessments are carried out. Due to the network bandwidth and the nodes' power limitations, it is desirable for nodes to send as less data as possible. The extreme case is decentralized detection [21] where each node decides locally about the event occurrence and sends its decision to FC using just one bit.

The state-of-the-art solutions to source localization are mostly based on information theoretic approaches. In [54], an initial coarse estimation of the source location is obtained based on the data of pre-specified anchor nodes. Then, a set of non-anchor nodes with maximum mutual information (MI) between the source location and their measurements are activated. The source location is tuned after several iterations. The complexity of the MI-based method grows exponentially in the network size. To

✉ S. Hamed Javadi
h.javadi@ugent.be

Angela Guerrero
Angela.Guerrero@ugent.be

Abdul M. Mouazen
abdul.mouazen@ugent.be

[1] Department of Environment, Faculty of Bioscience Engineering, Ghent University, 9000 Gent, Belgium

[1] Hereinafter, motes are referred to as nodes as it is more common in the literature of WSN.

alleviate the scalability problem, another sensor selection scheme for source localization based on conditional posterior Cramer-Rao Lower Bound (PCRLB) was proposed by [22, 37, 55].

Another approach to source localization is to track it during time, i.e., tracking. To that end, statistical filters with cumbersome computations are exploited. These computationally heavy algorithms should be run by each network node with severe limitations of processing power, memory, and communication. The nodes' estimations of the source track are transmitted to FC which obtains a tuned estimation of the track by resorting to an appropriate fusion rule. Some of existing fusion rules are: independent likelihood pool (ILP) [36], covariance intersection (CI) fusion [30], information graph [11], track-to-track fusion [10], and consensus-based fusion (distributed MTT–DMTT) [6].

In this paper, the goal is to develop machine learning-based methods for tackling the source localization problem over WSNs. In any WSN, we encounter distributed agents gathering data for a specific task. Since the size of data collected by WSNs is usually enormous, learning methods may be applied to solve the related problems. Instances of applying learning methods on different aspects of WSNs—such as node localization, channel allocation, and routing—have been reviewed in [4].

A simple learning-based source localization method is the fault recognition (FR) method [34] in which the decision of each node is adjusted according to the decision taken by a majority of the node's neighbors. FR attempts to correct possibly wrong decisions of nodes by a data exchange among them. Then, the region of event may be obtained by nodes with decision 1 (indicating event occurrence). The centroid of this region may be considered as the source location.

Javadi et. al. in [24] have used the support vector machine (SVM) [33] learning method for source localization. In their method, referred to as Red-S, the locations of the network nodes and their decisions are used as the training data and their labels, respectively, based on which the SVM parameters are trained. Finally, the decision of each node is corrected by applying its location to the SVM classifier. The nodes with decision 1 depict the region of the event whose centroid is considered as the source location. To improve the accuracy of Red-S, it has been proposed in [24] to apply twin SVM (TWSVM) [29] to the resultant of Red-S.

In this paper, we propose two source localization methods based on ANN and deep learning. The contributions presented by this paper are as follows:

- Decision fusion and machine learning algorithms are exploited collaboratively in order to: i. detect event occurrence; ii. estimate the event location; and iii. to specify the region affected by the event;
- We show that the erroneous decisions of sensors—which are due to their observation noise—can be corrected by using an artificial neural network (ANN). By error correction by ANN (ECANN), the region of the event is specified by sensors with decision 1. Then, the centroid of the event region is considered as an estimation of event location;
- Source localization with deep learning (SoLDeL) is proposed by considering the network layout as a block and white image with white pixels in locations of nodes with decision 1. We show that SoLDeL is scalable to any network size and works appropriately in both source localization and tracking;
- The performance of ECANN and SoLDeL is evaluated in a fire detection application in different network sizes with different noise and false alarm rates. To that end, a fire ignition model is presented which can be adopted for assessment of any IoT-based fire detection implementation.
- We examine the ECANN application in a real scenario of delineation of a farm into two management zones. For this purpose, the soil calcium of different locations of a farm is used and the whole farm is classified into low-calcium and high-calcium zones using ECANN.

The remaining of this paper is organized as follows. The models and assumptions used throughout this manuscript are discussed in Sect. 2. Section 3 provides the required background knowledge. Section 4 presents the proposed source localization methods using ANNs [5, 18, 45, 39] and deep learning (SoLDeL). The performances of the proposed methods are evaluated in Sect. 5. Finally, the paper is concluded in Sect. 6.

Notations: Lower-case bold letters denote vectors with a_i representing the ith element of $\boldsymbol{a}$. The 2-norm of vector $\boldsymbol{a}$ will be denoted by $\|\boldsymbol{a}\|$, $\mathbb{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, and $\mathbb{U}(a, b)$ denotes a uniform distribution with support $[a, b]$. Finally, the symbol $\sim$ means "distributed as."

2 System model and assumptions

For source localization, the system model shown in Fig. 1 is used throughout this paper. In what follows, this model is discussed in detail.

2.1 Node model

A network of K randomly deployed wireless smart motes (referred to as "nodes") is considered that are programmed

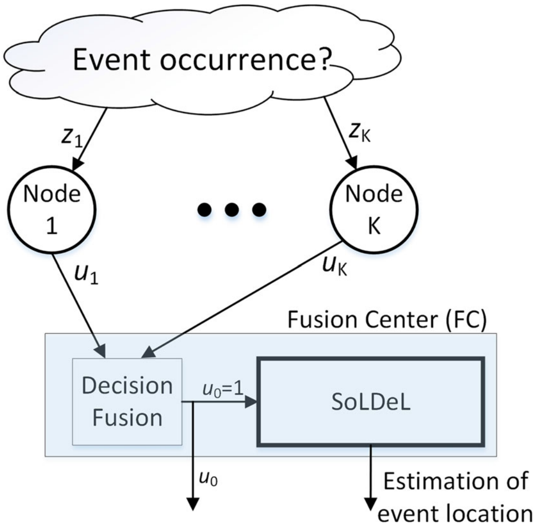

Fig. 1 Model adopted in this paper for source localization with deep learning (SoLDeL)

for collaborative detection of a desired event occurrence. The nodes decide about either event occurrence (denoted by hypothesis $\mathcal{H}_1$) or the normal condition in which no event has occurred (denoted by hypothesis $\mathcal{H}_0$). More specifically, node $i \in \{1, \ldots, K\}$ observes its local region for detection of possible appearance of a desired scalar parameter $\theta(\boldsymbol{x}_s)$ originating from the source location $\boldsymbol{x}_s$. The observation model of node i is given by:

$$z_i = \begin{cases} g(\theta(\boldsymbol{x}_s)) + v_i, & \mathcal{H}_1 \\ v_i, & \mathcal{H}_0 \end{cases} \qquad (1)$$

where v_i is a zero-mean additive white Gaussian noise (AWGN) with variance σ^2 (i.e., $v_i \sim \mathbb{N}(0, \sigma^2)$), and $g(.)$ is a mapping function (e.g., a function indicating the attenuation of an acoustic signal originating from position $\boldsymbol{x}_s$). The noises of network nodes are assumed to be spatially and temporally independent.

Node i takes decision u_i regarding event occurrence using a decision rule—later discussed in Sect. 3.3.1—and sends it to FC (as shown in Fig. 1). Note that the scalar observation model (1) has been adopted just for simplicity in our discussions. If the desired parameter (signal) is multi-dimensional, just the decision rule would change. Nevertheless, we adopt the scalar observation model since detection is not the focus of this study.

2.2 Network model

The location of node $i \in \{1, \ldots, K\}$ is denoted by $\boldsymbol{x}_i$ and is assumed to be known by FC. The positions may be obtained by using an appropriate node localization method such as those presented in [20, 35].

The parallel configuration has been adopted for data communication between nodes and FC, as shown in Fig. 1. This is the most popular configuration in the literature of WSNs. However, nodes may transmit their decisions to FC indirectly through intermediate nodes in a hop-by-hop manner (since their communication range is usually limited). This configuration can be represented by the parallel configuration as well if the communication channels are assumed to be error-free [25, 51, 52].

2.3 Communication channel model

The communication channels between nodes and FC are assumed to be ideal and error-free. In practical implementations of decentralized detection over WSNs, there are two sources of error: (1) erroneous local decisions, which are due to either local false alarms or misses of nodes, and (2) erroneous received decisions due to the faulty nature of wireless channels. We integrate both error sources into just the first type and ignore the communication error rate since it does not affect the implementation and evaluation of the proposed algorithms.

2.4 Problem statement

After the event occurrence is detected by FC, the problem is to estimate the location of event by having the nodes' locations and their one-bit decisions. Specifically, defining $\mathcal{X} \triangleq \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}$ and $\mathcal{U} \triangleq \{u_1, \ldots, u_K\}$ as the sets of the nodes' locations and their decisions, respectively, the goal is to estimate the location of the source, $\hat{\boldsymbol{x}_s}$ so that the mean squared error is minimized, i.e.,

$$\hat{\boldsymbol{x}_s} = \arg\min \mathrm{MSE}(\boldsymbol{x}_s; \mathcal{X}, \mathcal{U}) , \qquad (2)$$

where $\boldsymbol{x}_s$ is the true location of the source.

3 Background

In this section, the required backgrounds are briefly reviewed.

3.1 Artificial neural network (ANN)

There are problems that machines, despite humans, are not capable of solving by resorting to classical algorithms. Instead, inspired by the structure of the brain, a connection of artificial neurons may be used by machines [7]. Similar to the brain of human being, the ANNs must be trained in order to be able to accomplish a specific task.

As shown in Fig. 2a, an artificial neuron is simply a function of the biased weighted sum of its inputs. The

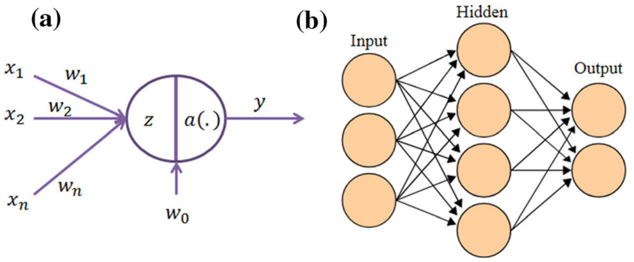

Fig. 2 **a** Structure of an artificial neuron. **b** Artificial neural network (ANN) as a connection of artificial neurons

function $a(.)$, referred to as the *activation function*, may be either of sigmoid ($g(x) = 1/(1 + e^{(-x)})$), rectified linear (ReLU) ($g(x) = \max(0, x)$), or any other function based on the application for which the ANN is used. After a configuration is adopted for the ANN (i.e., the number of its layers as well as the number of neurons in each layer, as shown in Fig. 2b), it should be trained (i.e., its parameters should be set) by minimizing a loss function with regard to a given training dataset:

$$[w_j] = \arg\min \sum_{i=1}^{K} l(y^{(i)}, \hat{y}^{(i)}) , \tag{3}$$

in which w_j indicates the jth parameter of the ANN, $y^{(i)}$ is the ith, $i \in \{1, \ldots, K\}$ label of the dataset with the ANN estimate denoted by $\hat{y}^{(i)}$, K is the size of the training set, and $l(.)$ is a loss function.

3.2 Deep learning

The basis of deep learning is to first extract simple and complex features of high-dimensional input data and then to apply them to an ANN, so that the artificial network is deepened and would be capable of accomplishing complicated tasks.

One method of feature extracting is convolutional neural networks (CNNs) [19]. A CNN basically consists of several convolutional layers each of them usually followed by a pooling layer. In each convolutional layer, some features of the input image are extracted by scanning it with appropriate filters (e.g., 3 × 3 matrices). In fact, convolutional layers facilitate deep learning by mitigating fully connection of neurons as well as using shared weights among the edges. The output of each convolutional layer is given by $a(\sum_i \alpha_i z_i + b)$, where z_i denotes the ith input to the convolutional filter with α_i as its corresponding weight, b is a bias, and $a(.)$ is an activation filter. Pooling layers are used for decreasing the dimensions of the output of convolutional layers (Fig. 3).

The deep network is trained (i.e., its parameters, including the weights and biases the ANN and the filters of the convolutional layers, are set) by using a sufficiently large dataset in order to minimize an appropriate loss function.

3.3 Decentralized detection

In detection applications of WSNs, a final decision about either an event occurrence (labeled as "hypothesis $\mathcal{H}_1$") or the normal condition (denoted by "hypothesis $\mathcal{H}_0$") must be taken by FC using an appropriate data fusion scheme [21, 28].

Each node transmits either raw or processed observations to FC. In order to save bandwidth and energy, the nodes are usually programmed to decide locally and inform FC about their decision by sending just one bit. To implement this practically popular scheme—known as "decentralized detection"—, two types of decision making rules must be designed: a local decision rule for each node and a decision fusion rule for FC. These two kinds of decision rules are discussed in what follows.

3.3.1 Local decision rule

Network nodes observe a desired signal θ according to model (1). In practical scenarios, the source signal is not known (e.g., there is no information about the location and the strength of the acoustic signal), and hence, reaching an optimum local detector is not tractable [32]. Nevertheless, a good sub-optimal rule would be the generalized likelihood ratio test (GLRT) if the statistical information of the sensing noise (v_i in (1)) is available [32]. Accordingly, the decision rule for node i with observation model (1) is given by [32]:

$$u_i = \begin{cases} 0, & z_i^2 < \tau_i \\ 1, & z_i^2 > \tau_i \end{cases} \tag{4}$$

with $u_i = 0$ ($u_i = 1$) indicating $\mathcal{H}_0$ ($\mathcal{H}_1$), and τ_i being the detection threshold given based on a local false alarm rate (i.e., $\Pr(u_i = 1|\mathcal{H}_0)$).

3.3.2 Fusion rule

The decisions of nodes are transmitted to FC where a decision fusion rule should be implemented for taking a final decision. Assuming independence among the decisions of the nodes conditioned on each hypothesis, the optimal fusion rule needs local detection performances [9]. When local detection performances are not known at FC, an alternative is the counting rule (CR) [41] in which the sum of received decisions is simply compared against a threshold:

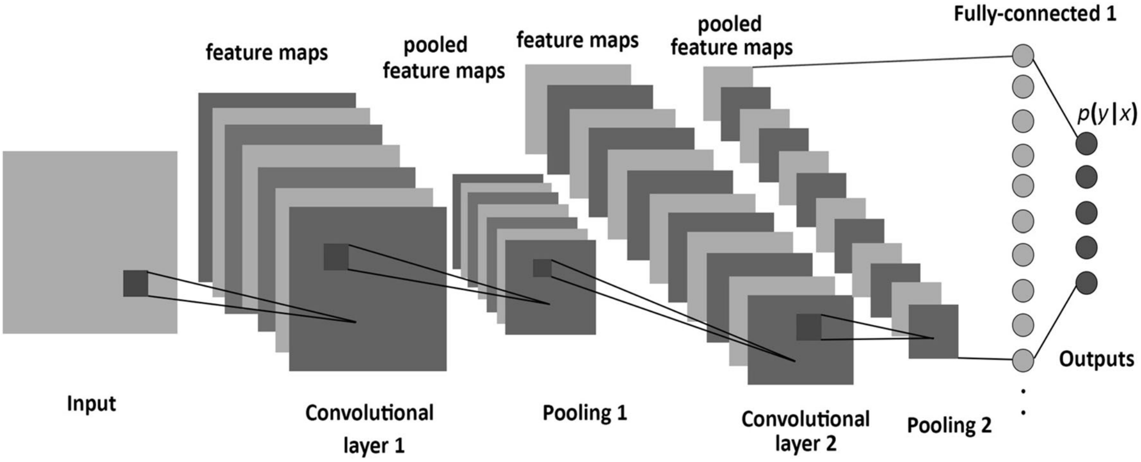

Fig. 3 Structure of a convolutional neural network (CNN) [2]

$$\Lambda \triangleq \sum_{i=1}^{K} u_i \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} T\,, \tag{5}$$

where K indicates the network size and T is the detection threshold which is obtained according to a desired network false alarm rate. Under a common threshold for sensors and the same noise statistics (i.e., in homogeneous WSNs), Λ under $\mathcal{H}_0$ (i.e., $\Lambda|\mathcal{H}_0$) is binomial-distributed while it follows a Poisson-binomial distribution in more general cases [13]. Threshold T is computed according to this distribution.

While simple, CR maintains robustness [12, 14] and can reach almost the optimum detection performance in large network sizes [21] (i.e., in sufficiently large values of K), the overall detection performance of the counting rule is improved in its modifications such as LVDF [31] and WDF [28, 26]. Each node in LVDF modifies its decision based on the majority of the decisions of its neighbors. In WDF, the decision of each node is weighted based on the node's sensed signal-to-noise ratio (SNR), and then, the weighted decisions are counted. Moreover, the network's detection performance can be more improved by considering the existing correlation among the nodes' decisions [23, 27].

4 Learning-based source localization

In this section, we propose two source localization methods based on machine learning.

4.1 Correction of errors type I and II with ANN

Error type I and error type II for node i are, respectively, defined as:

$$p_{fa,i} \triangleq \Pr(u_i = 1|\mathcal{H}_0)\,, \tag{6}$$

$$p_{m,i} \triangleq \Pr(u_i = 0|\mathcal{H}_1)\,. \tag{7}$$

In this section, the error correction using an ANN (referred to as ECANN) is proposed with the aim to classify network nodes into two categories: 1. in-event-region and 2. not-in-event-region. After errors are corrected, the region of event is obtained by nodes with decision 1 whose centroid can be considered as an estimation of the event location.

Correction of errors type I and II is performed as follows:

1. As shown in Fig. 1, the nodes send their decisions to FC. Upon event detection by FC, source localization procedure is initiated.
2. A fully connected ANN (with a predefined structure) is trained by using nodes' locations and their decisions as the training set. More specifically, the training set will be:

   $$\begin{aligned}\mathcal{X} &= \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}\,,\\ \mathcal{U} &= \{u_1, \ldots, u_K\}\,,\end{aligned} \tag{8}$$

   with $\mathcal{X}$ and $\mathcal{U}$ being the training sets of input samples and their labels, respectively.
3. After the ANN is trained, it gives the corrected decision $\hat{u}_i$ of node i by applying the node's location into the input of the ANN. The outputs of the ANN for all nodes are collected in the set of the corrected decisions:

   $$\hat{\mathcal{U}} = \{\hat{u}_1, \ldots, \hat{u}_K\}\,. \tag{9}$$
4. Denoting the set of the nodes with *corrected* decisions 1 by $\hat{\mathcal{U}}_1$, the estimation of source location $\hat{x}_s$ is given by:

$$\hat{x}_s = \frac{1}{|\hat{\mathcal{U}}_1|} \sum_{i \in \hat{\mathcal{U}}_1} x_i \,, \qquad (10)$$

where |.| denotes the cardinality.

4.1.1 Discussion

The ANN attempts to classify nodes according to their decisions (labels) (see Fig. 4). However, it cannot perform classification because of existing errors type I and II. As if the ANN understands that something is wrong with the training set, it attempts to correct the wrong decisions.

As an illustrative example, a 300-node network intended for fire detection in an environment is shown in Fig. 4 with a fire occurrence in the center of the field center. The nodes have been observing a fire ignition with model (11) and standard normally distributed measurement noises. They have been set up with local false alarm probability of 0.1. As shown, there are errors type I and II throughout the network upon fire ignition.

In this example, a two-layer ANN with 7 and 3 neurons in, respectively, the first and second hidden layers were exploited. ReLu was used as the activation function of the two hidden layers, while the sigmoid function was considered for the output neuron. As shown, this ANN was successful in correcting errors with an accurate estimate of the source location.

4.1.2 Advantages

The advantages of ECANN are listed as follows:

- The errors type I and II are corrected appropriately. Therefore, ECANN is applicable for any problem including decision errors. Such problems include diagnosing a special disease based on blood tests, recruiting in companies, crime cases, etc. In these examples, a collection of n-dimensional (instead of the two-dimensional nodes' locations in our case) data together with their labels are encountered.
- In many applications, detection of the event region is crucial. For example in fire detection in forests and detection of poisonous gas in a company, it is vital to detect exactly the event region in order to necessary actions to be taken. ECANN is capable of detection of the event region in addition to source localization.

4.1.3 Disadvantages

The disadvantages are:

- The main drawback is that the ANN must be trained whenever an event is detected. Training the ANN needs a lot of epochs and is time-consuming. As an instance, the ANN used in Fig. 4 was being trained by 250,000 epochs. Therefore, ECANN does not satisfy real-time applications.
- ECANN localization is not accurate enough when the event is occurred at the network edges since the centroid of nodes with corrected decisions 1 is never located at edges.

4.2 Source localization with deep learning (SoLDeL)

In this subsection, source localization with deep learning (SoLDeL) is proposed. As discussed in the previous subsection, using ANN for source localization needs much time during which the source location may vary. Therefore, it is not suitable for source localization and tracking in a field. Instead, the network may be trained to localize the source if it appears in any place in the field. The procedure is as follows:

1. As shown in Fig. 1, the network nodes observe their local environment for source detection using decision rule (4) and send their decisions to FC. FC takes the

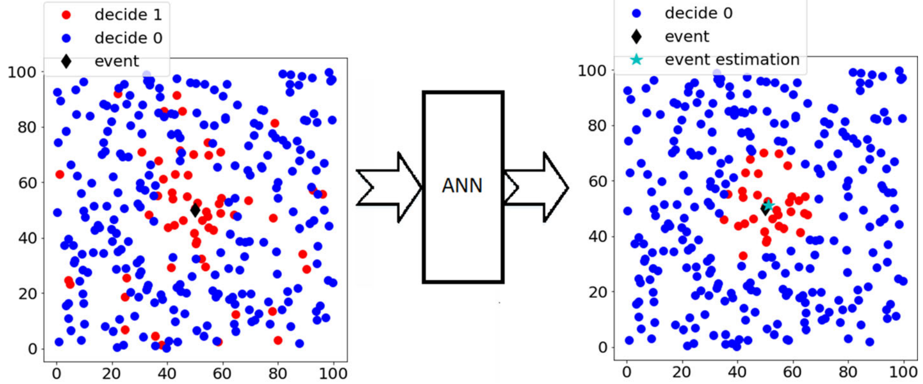

Fig. 4 Correction of errors type I and type II with an ANN (ECANN)

final decision by an appropriate fusion rule such as the counting rule (5). It is assumed that the network size is sufficiently large so that the overall detection performance of the network is optimum [32, 43] (i.e., there is neither a false alarm nor a miss in network level).

2. Upon source detection (i.e., if $u_0 = 1$), FC localizes the source by running SoLDeL as follows:
   (a) Network nodes with different decisions establish a network layout in which the locations of nodes with decision 1 (hot nodes) are highlighted (see Fig. 5).
   (b) The network layout is converted to a grayscale image with a predefined size whose all pixels are black except the locations of hot nodes that are denoted by white.
   (c) Applying the image to a CNN followed by a fully connected ANN (FCN), the source location is estimated.

For more discussion about the above steps, note to Fig. 5. When an event occurs, a network layout is established accordingly. The network layout is converted to an image with the pixels related to the hot nodes in white. The image is applied to the deep network input. The features of the image are extracted by the CNN. Finally, the source location is estimated by the FCN. The example of Fig. 5 is elaborated in Sect. 5.

4.2.1 Training deep network

A sufficiently large dataset is required for training the deep learning network. To this end, different network layouts are built based on known source locations and the information related to the network, source signal propagation model, and the statistical measurement model of nodes as well as their settings. The network layouts are converted to images with a specific size. Each image is labeled by its specific source location. The obtained images together with their labels are used for training the deep network. Training is needed just once before the network is going to be used, and the obtained parameters for the deep network will stay stationary unless the information that the network has been already trained based on them changes.

4.2.2 Advantages

The advantages of SoLDeL are as follows:

- Despite ECANN, SoLDeL needs training just once, and hence, it performs source localization in real-time.
- It performs uniformly throughout the network field, even at the edges.
- Despite the Red-S method [24] whose complexity grows with the cubed network size, SoLDeL scales very well to larger network sizes. In fact, the performance of source localization is improved in larger network sizes with no more computational burden.

4.2.3 Disadvantage

The main drawback of SoLDeL is that its performance degrades if the situation of either of network, nodes, or the source signal deviates from what was considered in training.

5 Evaluation

In this section, we first evaluate and compare the performance of the proposed localization methods. Then, the performance of SoLDeL is examined in a tracking application. Finally, we apply ECANN in a real scenario of farming where the farm is categorized into two management zones.

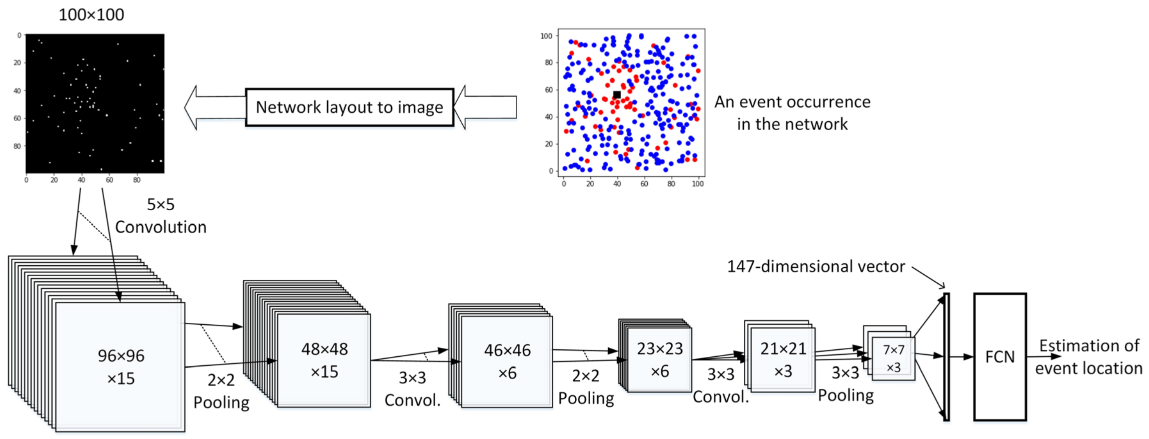

Fig. 5 Source localization based on deep learning (SoLDeL). The location is estimated by the fully connected network (FCN) at the latest stage

5.1 Performance evaluation in fire detection

5.1.1 Setup

Immediate detection of fire occurrence and its region is an important applications of WSNs, and it is crucial for protecting forests and natural resources. Accordingly, we present here a fire ignition model that can be adopted for cost and performance assessment of any WSN implementation for fire detection.

Fire ignition impacts its environment temperature according to the following model [44, 46]:

$$\theta_i(d_{is}) = \frac{1}{4}\theta_{\mathrm{f}}\left(\frac{h_{\mathrm{f}}l_{\mathrm{f}}}{2\pi d_{\mathrm{is}}{}^2}\right)^{1/4} \tag{11}$$

where $\theta_i(d)$ is the temperature at the location of sensor i with distance d_{is} from the flame in Kelvin, θ_{f} is the flame temperature in Kelvin, h_{f} and l_{f} are the height and the length of the flame, respectively. Model (11) is obtained using the Stefan-Boltzmann law [17, 44, 46] in equilibrium.

The temperature is contaminated by a measurement noise when reported by sensor i:

$$z_i = \theta_i + v_i\,. \tag{12}$$

in which v_i is the measurement noise which is assumed to be temporally independent and distributed according to the normal distribution, i.e., $v_i \sim \mathbb{N}(0, \sigma^2)$. The normal assumption holds in many commercial-off-the-shelf (COTS) temperature sensors [3, 38, 48]. These sensors can be connected to LoRaWAN [1] modules, such as [49] with the unit price less than 100€, which send their local decisions to a gateway.

We consider a network of K wireless temperature sensors randomly deployed over a $100\,\mathrm{m} \times 100\,\mathrm{m}$ region in order to detect any fire occurrence. ECANN and SoLDeL were evaluated during 1000 Monte Carlo runs. In each run, random values of $h_f \sim \mathcal{U}(1.5, 1.8)$, $l_f \sim \mathcal{U}(0.9, 1)$, and $\theta_{\mathrm{f}} \sim \mathcal{U}(1800, 2100)$ were used. These values are similar to those used in [44]; however, random values were adopted in order to assess different situations.

To implement ECANN for fire detection, an ANN with two hidden layers, each consisting of seven and three neurons, respectively, was used. ReLu and sigmoid were used as the activation functions of the hidden layers and the output neuron, respectively. The ANN was trained during 250,000 epochs in each scenario. An instance of ECANN application in estimation of the fire occurrence and its region is shown in Fig. 4.

In order to evaluate SoLDeL, a CNN followed by a fully connected network (FCN) was used for estimating the source location. As shown in Fig. 5, the network layout is converted to a 100×100-pixel image, i.e., one pixel for each $1\,\mathrm{m} \times 1\,\mathrm{m}$ area. The image is used as the input of the CNN. In the CNN, three layers of convolution together with pooling were used. The sizes of the convolution filters as well as the pooling layers were chosen such that no padding was needed. At the end of the CNN, the obtained features are flattened into a 147-dimensional vector. This vector is applied to a two-layer FCN including 35 and 20 neurons in the first and the second hidden layers, respectively. This structure of deep network was obtained after several rounds of try and error.

For training the structure of Fig. 5, 600,000 samples of random network layouts were generated and converted into images. In order to train the CNN structure for different situations, network sizes between 300 and 500 and false alarm rates between 0.08 and 0.22 were used in database generation. The obtained dataset was broken into batches of size 100 and learned the deep network during 20 epochs.

The performances of ECANN and SoLDeL are compared in different scenarios against that of Red-S [24]. Red-S uses the SVM classifier for detection of the region of the event. The parameters of SVM are trained by the locations of nodes as the training data and their decisions as their classes. After having the SVM classifier computed, the locations of nodes are applied to it in order to obtain their corrected decisions.

Evaluation and comparison of the methods have been carried out in terms of mean squared error (MSE) [5, 18, 45, 39] defined by:

$$MSE = \mathbb{E}\left(\|\boldsymbol{x}_s - \hat{\boldsymbol{x}}_s\|^2\right), \tag{13}$$

where $\boldsymbol{x}_s$ and $\hat{\boldsymbol{x}}_s$ are, respectively, the exact and the estimated source location, and $\mathbb{E}(.)$ denotes the statistical expectation. The MSEs of the methods were obtained through 1000 Monte Carlo runs in different scenarios. The results are depicted in Fig. 6. In each Monte Carlo run of SoLDeL, the fire location was considered as random. However, it was assumed to be located at the center of the field in the ECANN and Red-S evaluations since their performance degrades significantly if the fire occurs in the network edges.

5.1.2 Results and discussion

Figure 6a shows that ECANN and SoLDeL presented a stable performance in different network sizes while the Red-S performance improves in larger networks. The performance of SoLDeL degraded in larger network sizes because the CNN was trained for network sizes between 300 and 500, and any deviation from that situation may worsen its performance. Though the evaluation result conveys that ECANN outperforms SoLDeL, however,

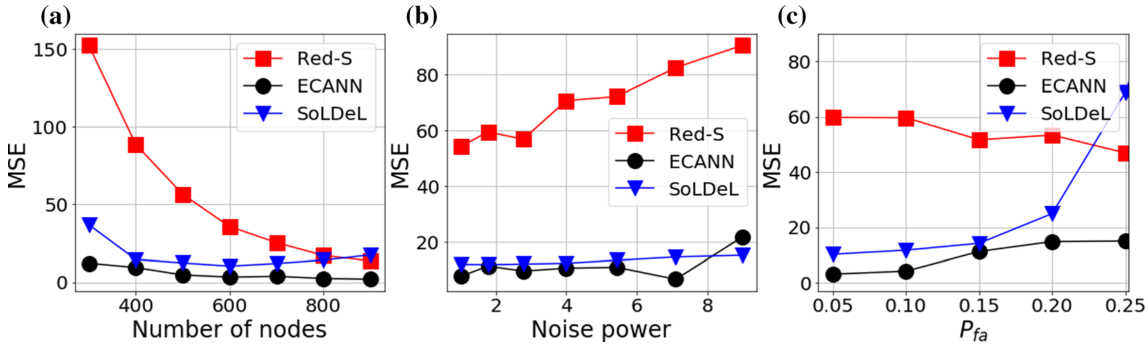

Fig. 6 Comparison of mean squared error (MSE) of the learning-based localization methods. The nodes are deployed randomly in a 100 m × 100 m field for detection of a fire with a random temperature, height, and length of $\theta_f \sim \mathcal{U}(1800, 2100)$ K, $h_f \sim \mathcal{U}(1.5, 1.8)$ m, and $l_f \sim \mathcal{U}(0.9, 1)$ m, respectively. **a** Local false alarm rate $P_{fa} = 0.1$ with the standard normal distributed observation noise. **b** Network size $K = 500$ with $P_{fa} = 0.1$. (c) $K = 500$ with the standard normal distributed observation noise

ECANN does not perform well in network edges wherein the SoLDeL accuracy is not affected. Note that the edge effect was neglected in obtaining the performance curves of ECANN and Red-S.

The performance of the localization methods in different values of noise power, σ^2, is compared in Fig. 6b. As shown, ECANN and SoLDeL, with a relatively stable accuracy, outperformed Red-S. Red-S presented more sensitivity to the noise than the other two methods. The reason is that noisier network layouts makes the classification more difficult by SVM with a linear nature. Moreover, as shown in Fig. 6b, SoLDeL also lost performance in higher noise power values. The reason is that learning the location by the deep network is much harder in images with more noise. In fact, the deep network should be learned with much more samples when the situation turns noisier while its training procedures were the same as in the high SNRs in the presented simulations.

The learning-based localization methods have been compared in different settings of local false alarm probabilities in Fig. 6c. As seen, the SoLDeL MSE has risen in $P_{fa} = 0.25$ since it is outside the range that was used for its training.

5.2 SoLDeL application in tracking

In addition to the fire detection application, we were interested to examine the performance of the proposed localization methods in tracking a moving target. To that end, we considered an acoustic source with the isotropic model moving from the lowest left corner of the network field to its highest right in a constant speed. Sensor i within the sensing range of the source observes $z_i = \sqrt{P_0/(1 + d_{si}^c)} + v_i$ with P_0 and c being the strength of the source in $1m$ distance from the target and an attenuation coefficient, respectively. This model has been extensively adopted in the literature [24, 42] As shown in Fig. 7, SoLDeL performs appropriately in target tracking, especially since it can estimate source locations in network edges.

5.3 ECANN application in farming

Precision agriculture (PA) uses technology for site-specific application of farm production inputs (e.g., fertilizers, pesticides, and seeds) to maximize yield and optimize nutrient use efficiency [8]. In PA applications, regions with relatively homogeneous combination of yield-limiting factors are delineated for which a single rate of a specific input is optimal to maximize the output. These regions are referred to as management zones (MZ) [53]. Proximal soil sensors facilitate the assessment of key soil properties at high sampling resolution to quantify the within-field spatial

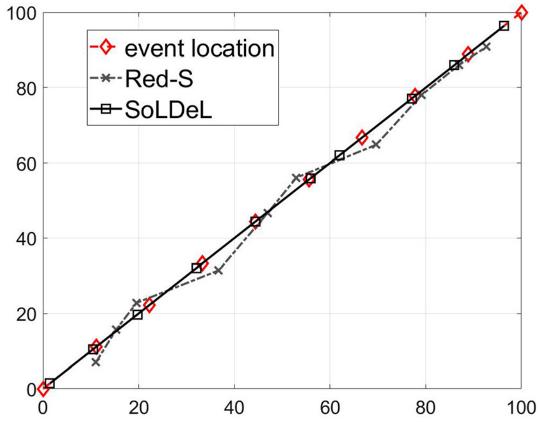

Fig. 7 Comparison of the performance of different learning-based methods in tracking a source that moves in a constant speed from the lowest left corner to the highest right. A 500-node network with a random deployment over a 100 m × 100 m field with $P_0 = 1000$ and local false alarm rate of 0.1 has been used. The observation noise of the nodes has been assumed to follow the standard normal distribution

and temporal variability [50] that MZ delineation requires. Among proximal soil sensors, diffuse reflectance spectroscopy techniques are highly used to assess macro and micro-nutrients in the soil [47].

In this subsection, our goal is to examine the applicability of ECANN in MZ delineation. To this end, the estimated calcium (Ca) values of Kouter field were used. This commercial field is located in a farm in Huldenberg (50°48′38.1″N4°34′46.9″E)in Flanders, Belgium. It has an area of 13 hectares, a loam to light loam soil texture, and annual crop rotation of wheat, barley, potato, and sugar beet with a short duration autumn cover crop. This field was scanned with an online multisensor platform designed and developed by [41]. The platform consists of a subsoiler, which penetrates the soil, creating a ditch at 15 to 25 cm depth range. Attached to the heal of the subsoiler chisel, an optical probe connected to a visible and near-infrared (vis-NIR) spectroscopy sensor (Tec5 Technology for Spectroscopy, Germany) with a measurement range of $305-1700$ nm is used to collect soil spectra in diffuse reflectance mode. Additionally, the multisensor platform included a differential global positioning system (DGPS) (version CFX-750, Trimble, USA) and a semi-rugged laptop computer (Toughbook, Panasonic, Belgium) to store the collected soil spectra and DGPS readings at 1 Hz, using MultiSpec pro-II software (Tec5 Technology for spectroscopy, Germany).

The location-based values of the soil calcium can be considered as a sensor network, as shown in Fig. 8. The decision rule (4) with threshold 55000 was used for detection of high calcium regions. Figure 8a shows the obtained network layout in which there exists several sparse high-calcium zones which might be due to the measurement noise. After applying this network layout to the same ECANN structure explained in Sect. 5.1.1, the whole farm was well classified into high-calcium and low-calcium zones, as depicted in Fig. 8b. Note that there exists no event occurrence here and ECANN was applied just in a binary classification of the whole region of interest.

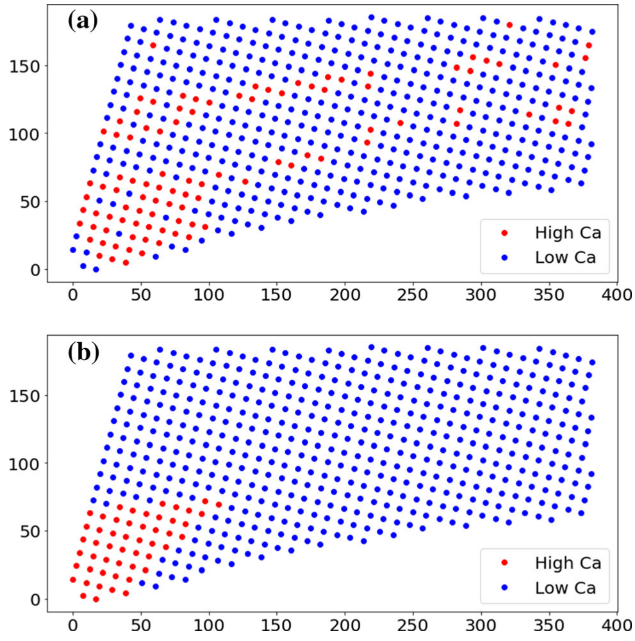

Fig. 8 Classifying a farm into high- and low-calcium (Ca) zones using error correction by artificial neural network (ECANN)

6 Conclusions and future directions

In this paper, two source localization methods based on decentralized detection and learning-based approaches were proposed. Nodes of a wireless sensor network (WSN) send their decisions about an event occurrence to a fusion center (FC). FC takes the final decision about the event occurrence by using an appropriate decision fusion rule. If the event occurrence is detected, FC localizes it by resorting to a deep learning structure. To that end, the situation of the WSN is converted to a gray-scale image in which the locations of nodes that have already detected the event occurrence are denoted by white pixels. The deep network yields an estimation of the current source location. It was shown through simulations that SoLDeL performs appropriately while its computational complexity does not grow with network size.

Moreover, we proposed using an ANN for specifying the region affected by the event source. To that end, the wrong decisions (errors type I and type II) of network nodes are corrected by the ANN. After having the event region specified, its centroid is considered as an estimation of the source location. The drawback of this method was that the ANN need to be trained each time an event occurrence is detected.

In this paper, localization of a single source was studied and two approaches were proposed and examined in two prominent applications including fire detection and a real scenario of smart farming. Localization and tracking of multiple sources by using the *k*-means clustering method or its extension—Bayesian non-parametric models—can be considered as an interesting future study.

Acknowledgements Authors acknowledge the financial support received from the European Commission for SIEUSOIL project (No. 818346).

Compliance with ethical standards

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

References

1. About lorawan | lora allience. https://lora-alliance.org/about-lorawan
2. Albelwi S, Mahmood A (2017) A framework for designing the architectures of deep convolutional neural networks. Entropy 19(6):242
3. Alippi C (2014) Intelligence for embedded systems: a methodological approach. Springer, Cham
4. Alsheikh MA, Lin S, Niyato D, Tan H (2014) Machine learning in wireless sensor networks: algorithms, strategies, and applications. IEEE Commun Surv Tutor 16(4):1996–2018
5. Aquino G, Rubio JDJ, Pacheco J, Gutierrez GJ, Ochoa G, Balcazar R, Cruz DR, Garcia E, Novoa JF, Zacarias A (2020) Novel nonlinear hypothesis for the delta parallel robot modeling. IEEE Access 8:46324–46334
6. Battistelli G, Chisci L, Farina A, Graziano A (2013) Consensus CPHD filter for distributed multitarget tracking. IEEE J Sel Top Signal Process 7(3):508–520
7. Bishop CM (1996) Neural networks for pattern recognition. Oxford University Press, New York
8. Bongiovanni R, Lowenberg-Deboer J (2004) Precision agriculture and sustainability. Precis Agric 5(4):359–387. https://doi.org/10.1023/B:PRAG.0000040806.39604.aa
9. Chair Z, Varshney P (1986) Optimal data fusion in multiple sensor detection systems. IEEE Trans Aerosp Electron Syst AES–22(1):98–101
10. Chang KC, Saha RK, Bar-Shalom Y (1997) On optimal track-to-track fusion. IEEE Trans Aerosp Electron Syst 33(4):1271–1276
11. Chong CY, Mori S, Chang K (1990) Distributed multitarget multisensor tracking. In: Bar-Shalom Y (ed) Multitarget-multisensor tracking: advanced applications, chapter 8. Artech House, Norwood
12. Ciuonzo D, De Maio A, Salvo Rossi P (2015) A systematic framework for composite hypothesis testing of independent Bernoulli trials. IEEE Signal Process Lett 22(9):1249–1253
13. Ciuonzo D, Romano G, Salvo Rossi P (2013) Optimality of received energy in decision fusion over Rayleigh fading diversity MAC with non-identical sensors. IEEE Trans Signal Process 61(1):22–27
14. Ciuonzo D, Salvo Rossi P (2014) Decision fusion with unknown sensor detection probability. IEEE Signal Process Lett 21(2):208–212
15. Ciuonzo D, Salvo Rossi P (2017) Distributed detection of a noncooperative target via generalized locally-optimum approaches. Inf Fusion 36:261–274
16. Ciuonzo D, Salvo Rossi P, Willett P (2017) Generalized Rao test for decentralized detection of an uncooperative target. IEEE Signal Process Lett 24(5):678–682
17. Duffie JA, Beckman WA (2013) Solar engineering of thermal processes. Wiley, New York
18. Elias I, Rubio J, Cruz D, Ochoa G, Novoa J, Martinez D, Muñiz S, Balcazar R, Garcia E, Juarez C (2020) Hessian with mini-batches for electrical demand prediction. Appl Sci 10:2036
19. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
20. Gustafsson F, Gunnarsson F, Lindgren D (2012) Sensor models and localization algorithms for sensor networks based on received signal strength. EURASIP J Wirel Commun Netw 2012(1):1–13
21. Javadi SH (2016) Detection over sensor networks: a tutorial. IEEE Aerosp Electron Syst Mag 31(3):2–18
22. Javadi SH, Farina A (2020) Radar networks: a review of features and challenges. Inf Fusion 61:48–55. https://doi.org/10.1016/j.inffus.2020.03.005
23. Javadi SH, Mohammadi A, Farina A (2019) Hierarchical copula-based distributed detection. Sig Process 158:100–106
24. Javadi S, Moosaei H, Ciuonzo D (2019) Learning wireless sensor networks for source localization. Sensors 19(3):635
25. Javadi SH, Peiravi A (2012) Reliable distributed detection in multi-hop clustered wireless sensor networks. IET Signal Process 6(8):743–750
26. Javadi SH, Peiravi A (2015) Fusion of weighted decisions in wireless sensor networks. IET Wirel Sensor Syst 5(2):97–105
27. Javadi SH, Mohammadi A, Farina A (2019) Serial Plackett fusion for decision making. IEEE Trans Aerosp Electron Syst (in press) (2019)
28. Javadi SH, Peiravi A (2013) Weighted decision fusion vs. counting rule over wireless sensor networks: a realistic comparison. In: 2013 21st Iranian conf. electr. eng. (ICEE), pp 1–6
29. Jayadeva, Khemchandani R Chandra S (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5:905–910
30. Julier SJ (2008) Fusion without independence. In: IET seminar on target tracking and data fusion: algorithms and applications
31. Katenka N, Levina E, Michailidis G (2008) Local vote decision fusion for target detection in wireless sensor networks. IEEE Trans Signal Process 56(1):329–338
32. Kay SM (1998) Fundamentals of statistical signal processing, volume 2: detection theory. Prentice Hall, Upper Saddle River
33. Ketabchi S, Moosaei H, Razzaghi M, Pardalos PM (2019) An improvement on parametric -support vector algorithm for classification. Ann Oper Res 276:155-168
34. Krishnamachari B, Iyengar S (2004) Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. IEEE Trans Comput 53(3):241–250
35. Liu C, Fang D, Yang Z, Jiang H, Chen X, Wang W, Xing T, Cai L (2016) RSS distribution-based passive localization and its application in sensor networks. IEEE Trans Wirel Commun 15(4):2883–2895
36. Manyika J, Durrant-Whyte H (1994) Data fusion and sensor management: a decentralized information-theoretic approach. Ellis Horwood, Hempstead
37. Masazade E, Niu R, Varshney PK, Keskinoz M (2010) Energy aware iterative source localization for wireless sensor networks. IEEE Trans Signal Process 58(9):4824–4835
38. Maxim Integrated: SOT temperature sensors with period/frequency output (2014). Rev. 1
39. Meda-Campana JA (2018) On the estimation and control of nonlinear systems with parametric uncertainties and noisy outputs. IEEE Access 6:31968–31973
40. Mouazen AM (2006) Soil Survey Device. International publication published under the patent cooperation treaty (PCT). World Intellectual Property Organization, International Bureau. International Publication Number: WO2006/015463; PCT/BE2005/000129; IPC: G01N21/00; G01N21/0
41. Niu R, Varshney PK (2005) Distributed detection and fusion in a large wireless sensor network of random size. EURASIP J Wirel Commun Netw 2005(4):462–472
42. Niu R, Varshney PK (2008) Performance analysis of distributed detection in a random sensor field. IEEE Trans Signal Process 56(1):339–349
43. Niu R, Varshney PK, Cheng Q (2006) Distributed detection in a large wireless sensor network. Inf Fusion 7(4):380–394
44. Rossia JL, Chetehounab K, Collinc A, Morettia B, Balbia JH (2010) Simplified flame models and prediction of the thermal radiation emitted by a flame front in an outdoor fire. Combust Sci Technol 182(10):1457–1477
45. Rubio dJ (2009) Sofmls: online self-organizing fuzzy modified least-squares network. IEEE Trans Fuzzy Syst 17(6):1296–1309

46. Rybicki GB, Lightman AP (1979) Radiative processes in astrophysics. Wiley-Interscience, New York
47. Stenberg B, Viscarra Rossel RA, Mouazen AM, Wetterlind J (2010) Visible and near infrared spectroscopy in soil science. Adv Agron 107(C):163–215. https://doi.org/10.1016/S0065-2113(10)07005-7
48. Texas Instruments: analog temperature sensor, RTD and precision NTC Thermistor IC (2015)
49. Uc11-n1 lorawan sensor node. https://www.ursalink.com/en/n1-lorawan-sensor-node/
50. Viscarra Rossel RA, Adamchuk VI, Sudduth KA, McKenzie NJ, Lobsey C (2011) Proximal soil sensing. An effective approach for soil measurements in space and time, vol 113. Elsevier Inc, Amsterdam. https://doi.org/10.1016/B978-0-12-386473-4.00010-5
51. Viswanathan R, Thomopoulos SCA, Tumuluri R (1988) Optimal serial distributed decision fusion. IEEE Trans Aerosp Electron Syst 24(4):366–376
52. Viswanathan R, Varshney PK (1997) Distributed detection with multiple sensors: part Ifundamentals. Proc IEEE 85(1):54–63
53. Vrindts E, Mouazen AM, Reyniers M, Maertens K, Maleki MR, Ramon H, De Baerdemaeker J (2005) Management zones based on correlation between soil compaction, yield and crop data. Biosyst Eng 92(4):419–428. https://doi.org/10.1016/j.biosystemseng.2005.08.010
54. Williams JL, Fisher JW, Willsky AS (2007) Approximate dynamic programming for communication-constrained sensor network management. IEEE Trans Signal Process 55(8):4300–4311
55. Zuo L, Niu R, Varshney PK (2011) Conditional posterior Cramer Rao lower bounds for nonlinear sequential Bayesian estimation. IEEE Trans Signal Process 59(1):1–14