



# An aggregator-less distributed smart sensor network with selective data exchange

Mathieu Chêne, Benoit Larras, Antoine Frappé, Andreas Kaiser

## ► To cite this version:

Mathieu Chêne, Benoit Larras, Antoine Frappé, Andreas Kaiser. An aggregator-less distributed smart sensor network with selective data exchange. 2023 IEEE International Symposium on Circuits and Systems (ISCAS), May 2023, Monterey, United States. pp.1-5, 10.1109/ISCAS46773.2023.10181765 . hal-04189293

**HAL Id: hal-04189293**

**<https://hal.science/hal-04189293>**

Submitted on 10 Nov 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# An aggregator-less distributed smart sensor network with selective data exchange

Mathieu Chêne, Benoit Larras, Antoine Frappé, Andreas Kaiser

Univ. Lille, CNRS, Centrale Lille, Junia, Univ. Polytechnique Hauts-de-France, UMR 8520 -IEMN

Lille, France

mathieu.chene1@junia.com

**Abstract**—Sensor Networks (SN) could be defined as networks of autonomous devices that can sense and/or act on physical or environmental conditions cooperatively. In sensor networks, data is typically sensed and sent to an aggregator that will process it with local AI or send it to a cloud with larger AI. This centralized architecture has drawbacks such as the aggregator having to receive and process a potentially huge amount of data, which results in power consumption that can be significant. In addition, the transmission of all the sensor data results in extra consumption of energy due to communication. To reduce the impact of this last point, on-edge computing allows data to be pre-processed at the sensor level. It is often used in the architecture of distributed sensor networks in which each node receives data from other nodes in the network and processes it with its local data. In this work, a distributed sensor network model aims to solve this problem while reducing the impact of data transmission on energy consumption. The proposed model is able to reduce the number of transmitted bits per node by 90% in a node-to-node directed communication scenario. It is also capable of working with different AI paradigms depending on the required balance between energy consumption, application configurability, and accuracy. Finally, this model is capable of converging even in a network without complete interconnections between nodes.

**Index Terms**—Wireless Sensor Networks, Distributed Intelligence, Smart Sensors, On-edge Computing

## I. INTRODUCTION

Sensor Networks (SN) could be defined as networks of autonomous devices that can sense and/or act on physical or environmental conditions cooperatively [3]. In most SNs, data measured by the sensor are sent to an aggregator that will pre-process them and send them to the cloud (or the fog) to be processed by a centralized AI. Posture recognition (PR) is an interesting use case for SNs like wireless body area network (WBAN) [2]. Posture is defined as the manner in which the skeleton and muscles hold the body during different activities [1]. Figure 1 shows different WBAN architectures and their power consumption distribution. However, as shown in [4], centralized structures, such as presented in a) in Figure 1, have three main drawbacks:

- 1) The aggregator has to be able to process a potentially enormous amount of data in real-time. This leads to a trade-off between the energy consumption and the computing capabilities of the system.

This work was supported in part by the AI Engineering PhD@Lille program and by french ANR : ANR IA/JUNIA/ ISITE

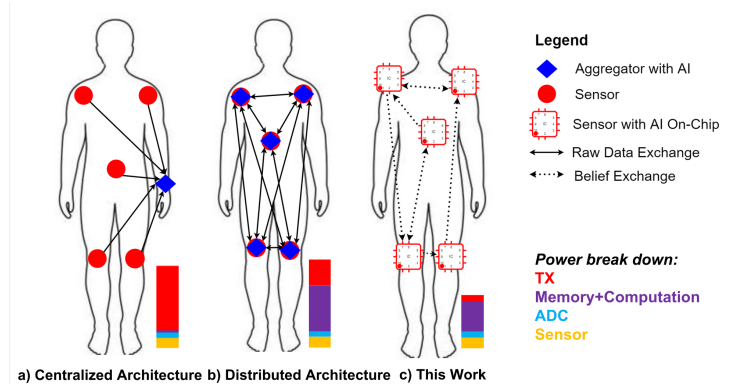


Fig. 1: Centralized, distributed and belief-based (this work) SN architectures and their power consumption breakdown per sensor, inspired from [5]

- 2) Transmitting all the sensors' data is another problem that can hurt the battery lifetime of the sensor network nodes. Indeed, the energy consumption overhead due to communication is usually large when sending raw data to the aggregator.
- 3) Finally, transmitting the sensed data does not guarantee information privacy in case of interception by a malicious individual.

To reduce the impact of the first two drawbacks, “on-edge” computing allows pre-processing of the data at the sensor level with AI and, therefore, reduces the amount of data to broadcast to the aggregator. On-edge computing is often used in distributed models of smart autonomous SNs where the intelligence is split among the sensors of the networks and where no aggregators act as a master device, as presented in b) in Figure 1. However in [5], He *et al.* present a power breakdown of a node and compare centralized and distributed architectures. It shows that even if TX consumption is lower than in the centralized model, the distributed model is not as interesting as the centralized architecture in terms of power consumption due to memory and computation parts.

In [4], Larras *et al.* present a distributed architecture of a Clique-Based Neural Network (CBNN), in which only the index of the activated neuron is transmitted among the nodes. Therefore, the amount of transmitted data reduces to

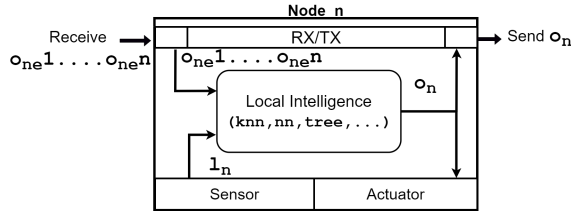


Fig. 2: Model of the node for the Distributed Sensor Network.

$\log(N_{Cluster}) + \log(N_{Neurons})$ , with  $N_{Cluster}$  and  $N_{Neurons}$ , the number of possible global states (clusters) and number of local states (neurons), respectively. However, CBNNs rely on a specific architecture because the cliques need to be *a priori* defined.

In the case of an aggregator-less distributed SN architecture, this paper proposes to investigate the impact of the reduction of the volume of exchanged data to unique beliefs, which are nodes' local decisions/classifications of the global context. This is illustrated in c) in Figure 1. Section II introduces the proposed model by presenting the structure of a node in the SN and the training and test setups in the context of posture recognition. Then, the performance of this architecture is detailed in Section III-A and compared with the distributed approach in Section III-B. Section III-C shows the impact of limited network connectivity on the performance. Finally, this work is compared with other works in Section III-D and concluded in Section IV.

## II. PROPOSED MODEL

### A. Nodes of the Network

As presented in Figure 2, each node is composed of a sensor and/or an actuator, an AI processing block (classification tree, neural network, ...), and a TX/RX interface used to communicate with neighboring nodes within communication range. As explained in Section I, the model aims at reducing energy consumption at the node level. One of the leads to achieving this would be to avoid transmitting raw data or pre-processed data. The solution explored here is that each node only broadcasts  $o_n$ , the local classification of the posture. This  $o_n$  is named *belief* because, after one iteration of communication within a graph, it can lack information and be wrong at first, but it will get more and more precise as the nodes exchange their beliefs. Another benefit of only transmitting  $o_n$  as a representation of the global context is that it largely reduces the amount of data to transmit compared with a system that exchanges raw or pre-processed data. Indeed, with this model,  $k$  bits are used to exchange  $2^k$  contexts. However, when exchanging only beliefs, the global classification accuracy can be altered. This work studies application scenarios and techniques to show that the possible gain in energy is considerable with a minimal drop in accuracy.

### B. Test Setup

To analyze the model of smart sensors presented in Section II-A, tests were performed using the PR dataset presented in [6], and available in [7]. It is composed of data from 3 accelerometers and 3 gyroscopes placed by pairs on the

chest, the knee, and the back of 2 types of populations: *healthy* and *unhealthy* subjects. Data is aggregated via Arduino and processed using classification algorithms to determine the posture: sitting, standing, or sleeping. This dataset, including both types of population, is here referred to as the global dataset. This global dataset is shuffled and separated into two sub-datasets: the global training dataset and the testing dataset. The training dataset is composed of 80% of the global one and the testing one of the remaining 20%.

The network is represented as a directed graph in which edges represent communication from one node to another. As each node of the network embeds its own local AI, it should be trained independently. Thus, local training datasets should be extracted from the global training one. Local dataset extraction is performed on *Matlab*<sup>®</sup>. The process is divided into 4 steps:

- 1) The possible local outputs for all combinations of the local inputs are retrieved from the global dataset. At the node level, the local inputs are sensed values, while the local outputs are the node beliefs.
- 2) The neighbors' label combinations associated with the local outputs are identified. For each output found in step 1, the script finds in the global dataset the combination of neighbors' local values that lead to it. Then, it determines the neighbors' beliefs, creates entries for all combinations of neighbors' beliefs and local inputs, and stores this in a cell array. Steps 1 and 2 are repeated for each local input value of a node present in the global dataset.
- 3) The local dataset is assembled by concatenating all cell arrays of a specific node. Duplicate lines are identified and removed. This is done by comparing the combinations of local inputs and neighbors' beliefs. Different cases exist:
  - If the local inputs are different, then each entry is kept.
  - If The local input are identical and their outputs are identical, then one entry is kept and the other one discarded.
  - If the local inputs are identical and their outputs are different, then the decision is more tricky. If one of the possible outputs is present in the majority of the neighbors' beliefs, then the entry with this output is kept while the others are discarded. Otherwise, the entry to keep is randomly selected among the possibilities.
- 4) The dataset is verified and cleaned. The script automatically adds a reset signal in the local dataset. It is a binary value associated with each local input, which indicates if the node has sensed a new value. A 0 means that the local input has not changed. If it is 1, the local value has changed, and an entry is then added to the dataset. In this new entry, the neighbors' beliefs are set to a reset state "R" and the output is randomly chosen among the possible label associated with the local input. Random selection is done to avoid output imbalance in

the training dataset. The script also checks the balance of each output and it duplicates lines if the balance is not respected. Finally, it shuffles the local training dataset.

During inference, each node of the network reads the associated quantized input value in the test dataset and performs the inference to obtain  $o_n$ . Then, it broadcasts this output to its neighbors and receives their beliefs. When the network has converged to one of the possible contexts, the average accuracy over all the nodes in the graph is computed. Then, each node reads the next value in the test dataset, and so on.

### III. RESULTS

#### A. Embedded AI Model

The complete system is modeled using *Matlab*<sup>®</sup>, according to the test setup presented in Section II-B. The initial setup is a full-connectivity SN with 6 nodes. Each node of the SN represents one of the 3 accelerometers or 3 gyroscopes. Even though they are placed by pairs (an accelerometer combined with a gyroscope) in the experiment, the setup here considers a single sensor per node to effectively demonstrate the applicability of the proposed distributed model. Table I compares three different AI paradigms that could be implemented as local intelligence in a node. To determine which paradigm is the more suitable for the proposed model, the considered metrics are the test accuracy averaged over all nodes of the SN, the number of bits transmitted per node when new information is sensed (referred to as stimulus), the number of MAC operations per node and per stimuli (to define the required complexity of the local AI, and thus the associated consumption), and the configurability to address different applications, such as the number of layers, nodes, clusters, etc. The number of bits transmitted per node and per stimulus and the number of MAC operations per node and per stimulus are defined as follows.

$$BTS = So_n * C \quad (1)$$

Where  $BTS$  is the number of bits transmitted per stimulus,  $So_n$  is the size of  $o_n$  in bits, and  $C$  is the number of directed communications, from a node to its neighbors, required to converge. Comparisons will be done using directed node-to-node communications.

$$MAC_{sti} = MAC_{Inf} * I \quad (2)$$

Where  $MAC_{sti}$  is the number of MAC operations per node and stimulus,  $MAC_{Inf}$  is the number of MAC operations per inference, and  $I$  is the number of inferences needed to converge.

Each model has been designed to reach an accuracy higher than 99.5%. Classification trees offer the best performance regarding the metrics. However, a specific tree is required per node and per localization, which largely impacts the configurability for different scenarios. K-nearest neighbors algorithms are more flexible than classification trees, but they need  $146\times$  more  $MAC_{sti}$ . The neural network used in this work is a feed-forward network composed of 3 fully-connected layers of dimensions 9 (3 local inputs, 1 reset flag and 5

TABLE I: Comparison of AI models

AI Paradigm	Classification Tree	K Nearest Neighbors	Neural Network
Average test accuracy	99.99%	99.79%	99.75%
$BTS$	60.07	60.53	73.36
$MAC_{sti}$	149.5	21876.8	396.8
Configurability over application	Low	High	High

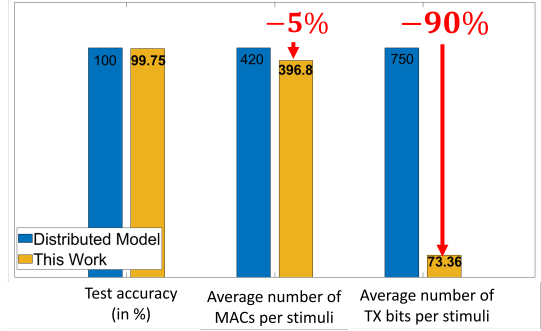


Fig. 3: Comparison of performance between Distributed architecture and this work (Matlab).

neighbors' beliefs), 10, and 3, respectively. While offering excellent reconfigurability, it shows similar performance to classification trees in terms of communication overhead. The drawback is the higher complexity highlighted by the number of MAC operations per stimulus. From this comparison, neural networks seem to be the best compromise between performance and flexibility, especially when a node is removed or displaced. In this case, to adapt to the new situation, only the local and neighbors' neural networks need to be retrained and not the complete system. This is particularly beneficial when the number of nodes is extremely large.

#### B. Comparison with distributed architecture

Figure 3 compares this proposed belief-based distributed architecture with a traditional distributed architecture, in the case of a fully-connected graph. For each model, the data from the sensors is identically normalized on  $[-1; 1]$  and quantized into 5 bits. These values are the inputs of neural networks as described in Section III-A. The metrics presented in (1) and (2) are used to compare the performance between the two architectures. Thanks to the fact that the nodes only exchange beliefs, the local intelligence block obviously needs to process a reduced amount of data, compared to a system that exchanges raw data. In the case of PR with the dataset presented in [6], it leads to a reduction in the number of  $MAC_{sti}$  of about 5%, while the amount of  $BTS$  drops by a factor of 10 (90%), in the case of a node-to-node communication scenario. Hence, it is shown that if a small variation in accuracy is tolerable, a belief-based distributed model helps to reduce both the computations and communication needs of a single node. It is an excellent lever to reduce the global energy consumption of an autonomous decision-making node, in an aggregator-less distributed SN architecture.

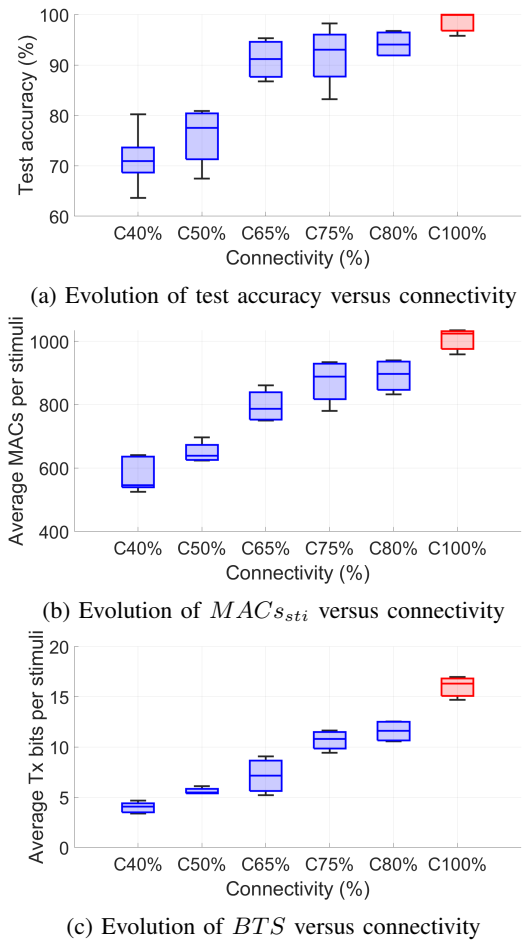


Fig. 4: Evolution of test accuracy, average  $MAC_{s_{sti}}$  and  $BTS$  versus connectivity of the SN.

### C. Impact of Connectivity

In this section, the robustness of the model is studied. Starting from a fully-connected directed graph with  $n$  nodes, a script randomly removes the edges one by one until it reaches a certain percentage of connectivity  $C$ , which equals the number of edges in the final graph divided by the number of edges in the fully connected version. However, to consider the final graph valid, a few rules should be respected:

- 1) The final graph must be unique. Since edges are randomly removed, any final graph that is composed of two distinct graphs should be discarded.
- 2) Each node should keep its self-connection to be able to understand some recurrent relations if needed.
- 3) Each node should have at least 1 predecessor and 1 successor (without counting its self-connection) to ensure all information is transmitted.

The goal is to avoid working with too specific topologies of networks and evaluate the robustness of the proposed model in a general case.

The impact is evaluated using 4 sets of 6 randomly generated graphs per set. Each series has connectivity  $C$  of 40%, 50%, 65%, 75%, 80% and 100% (the reference, shown in red in the figures). Figure 4 presents the evolution of the

TABLE II: Comparison with the state of the art

Reference	[Gupta, GLOBECOM 21] [6]	[Larras, JETCAS 20] [4]	This Work
SN Architecture	Centralized	Distributed	Fully Distributed
AI Model	KNN	CBNN	Fully-Connected Neural Network
Test Accuracy	100%	99.72%	99.75%
TX data width at the node level	48 bits	2 bits	2 bits
Number of $MAC_{s_{sti}}$ over the 6 sensors	8100	Not Indicated	2380.68
On-chip implementation	No	Yes	Compatible

metrics presented in (1) and (2) with respect to the connectivity value. As expected, Figure 4a shows that reducing the SN connectivity degrades the test accuracy. A large reduction of accuracy is observed for  $C$  lower than 65%. Figure 4b and 4c show that as the connectivity decreases,  $MAC_{s_{sti}}$  and that the number of  $BTS$  decrease.

Hence, the proposed model is able to converge even when all nodes are not connected to each other. However, there is a trade-off between accuracy and power consumption. This conclusion is only valid for a reduced number of nodes, such as in this posture recognition scenario. This should be explored further in scenarios with a greater number of nodes.

### D. Comparison With the State of the Art

Table II compares this model in the case of  $C100\%$  with other state-of-the-art solutions<sup>1</sup>. The proposed distributed AI model reduces the number of MAC operations per stimulus by up to 70% compared to the centralized network. Thanks to its distributed architecture, this model also has the advantage of being implementable on-chip. It significantly reduces the latency that exists in a centralized model that relies on an aggregator. Finally, in the presented model, the SNs do not need to be fully connected in order to converge to a classification result, unlike the CBNN.

## IV. CONCLUSION

In this work, a new distributed intelligence model that aims to reduce the energy consumption of smart sensors in SNs is developed. The belief-based model reduces the amount of data to be transmitted per stimulus by 90% compared to the common model of distributed SNs while providing decent accuracies at the system level. Finally, it is also shown that the proposed architecture is compatible with sensor networks with reduced connectivity.

<sup>1</sup>for the works of Gupta *et al* and Larras *et al*, the results were extrapolated as if they were used with the present dataset

## REFERENCES

- [1] Ivanenko, Y., & Gurfinkel, V. S. (2018). Human postural control. *Frontiers in neuroscience*, 12, 171.
- [2] Movassaghi, S., Abolhasan, M., Lipman, J., Smith, D., & Jamalipour, A. (2014). Wireless body area networks: A survey. *IEEE Communications surveys & tutorials*, 16(3), 1658-1686.
- [3] R. V. Kulkarni, A. Förster and G. K. Venayagamoorthy, "Computational Intelligence in Wireless Sensor Networks: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 13, no. 1, pp. 68-96, First Quarter 2011, doi: 10.1109/SURV.2011.040310.00002..
- [4] B. Larras and A. Frappé, "On the Distribution of Clique-Based Neural Networks for Edge AI," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 4, pp. 469-477, Dec. 2020, doi: 10.1109/JETCAS.2020.3023481.
- [5] Y. He et al., "A 28.2  $\mu$ C Neuromorphic Sensing System Featuring SNN-based Near-sensor Computation and Event-Driven Body-Channel Communication for Insertable Cardiac Monitoring," 2021 IEEE Asian Solid-State Circuits Conference (A-SSCC), 2021, pp. 1-3, doi: 10.1109/A-SSCC53895.2021.9634787.
- [6] Ritwik Duggal, Jahnvi Gupta, Mukesh Kumar, Nitin Gupta, Joel J.P.C. Rodrigues, November 12, 2020, "Dataset for IoT Assisted Human Posture Recognition", IEEE Dataport, doi: <https://dx.doi.org/10.21227/gk5q-gw14>.
- [7] <https://ieee-dataport.org/documents/dataset-iot-assisted-human-posture-recognition>