Journal of Information Systems and Telecommunication
Vol.10, No.4, October-December 2022, 259-269
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

A Hybrid Approach based on PSO and Boosting Technique for Data Modeling in Sensor Networks

Hadi Shakibian[1*], Jalal A. Nasiri[2]

[1]. Department of Computer Engineering, Faculty of Engineering, Alzahra University, Tehran, Iran
[2]. Department of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran

Received: 14 Jul 2021/ Revised: 04 Dec 2021/ Accepted: 29 Jan 2022

Abstract
An efficient data aggregation approach in wireless sensor networks (WSNs) is to abstract the network data into a model. In this regard, regression modeling has been addressed in many studies recently. If the limited characteristics of the sensor nodes are omitted from consideration, a common regression technique could be employed after transmitting all the network data from the sensor nodes to the fusion center. However, it is not practical nor efferent. To overcome this issue, several distributed methods have been proposed in WSNs where the regression problem has been formulated as an optimization based data modeling problem. Although they are more energy efficient than the centralized method, the latency and prediction accuracy needs to be improved even further. In this paper, a new approach is proposed based on the particle swarm optimization (PSO) algorithm. Assuming a clustered network, firstly, the PSO algorithm is employed asynchronously to learn the network model of each cluster. In this step, every cluster model is learnt based on the size and data pattern of the cluster. Afterwards, the boosting technique is applied to achieve a better accuracy. The experimental results show that the proposed asynchronous distributed PSO brings up to 48% reduction in energy consumption. Moreover, the boosted model improves the prediction accuracy about 9% on the average.

Keywords: Wireless sensor network; Distributed optimization; Particle swarm optimization; Regression; Boosting.

1- Introduction

In wireless sensor networks (WSNs), keeping massive ongoing data is an expensive task due to the limited power supply and capacity of the sensor nodes. Moreover, this data is expected to be analyzed in order to extract more useful information about the phenomenon of interest. In this regard, regression modelling has been addressed as an efficient approach for abstracting [1], [2] and analyzing the network data [3], [4].
Distributed data and limited characteristics of the sensor nodes impose major challenges on performing regression over WSNs. A naive simple solution is to gather all the network data in the fusion center and obtain the network regressor using a well-known technique [5], [6]. Although a high accuracy is achieved, a huge data transmission from the sensor nodes to the fusion center is needed which makes this solution inapplicable, especially when the network grows in size.
To overcome both the communication and the computation constraints of the sensor nodes, several Learning/optimization algorithms have been proposed in many research papers.
A distributed sub-gradient algorithm with uncoordinated dynamic step sizes has been proposed for multi-agent convex optimization problems [7]. In this algorithm, each agent i can utilize its estimation of the local function value. Theoretical analysis show all the agents reach a consensus on the optimal solution. The gradient methods have also been studied by [8], [9], [10] over a network with communication constraints.
In [11], the information theoretic optimality of the distributed learning algorithms has been addressed in which each node is given i.i.d. samples and sends an abstracted function of the observed samples to a central node for decision making.
The use of machine learning algorithms in clustered WSNs has been studied by [12] in order to decrease data communications and make use of the features of WSNs. Different applications of machine learning algorithms in

✉ **Hadi Shakibian**
h.shakibian@alzahra.ac.ir

the context of WSNs has been recently reviewed by [13], [14], [15].

A kernel regression algorithm has been introduced in [16] to predict a signal y_t defined over the N network nodes with a series of T regularly sampled data points. A Laplace approximation is proposed to provide a lower bound for the marginal out-of-sample prediction uncertainty to address the large problems.

Logistic regression fusion rule (LRFR) has been proposed in [17] in which the coefficients of the LRFR is learnt at first, and then, it is used to make a global decision about the presence/absence of the target.

In [18], a quantized communication based distributed online regression algorithm has been proposed. Also, a distributed quantile regression algorithm has been proposed by [19], where, each node estimates the global parameter vector of a linear regression model by employing its local data as well as collaboration with the other nodes. Due to the sparsity of numerous natural and artificial systems, they have introduced $l1-$distributed quantile regression algorithm to exploit the sparsity and consequently to improve the performance of the method.

An energy-efficient distributed learning framework has been proposed using the quantized signals in the context of IoT networks [20], [21]. This is a recursive least-squares algorithm that learns the parameters using low-bit quantized signals and requires low computational cost.

Some distributed learning algorithms have also been suggested based on linear and polynomial regression models [22], [23].

On the other hand, several distributed regression models have been proposed in WSNs in which the learning problem is formulated as an optimization task [24]. To solve it, Incremental Gradient (IG) algorithm has been proposed in which the parameter to be estimated is circulated through the network. Along the way, each sensor node adjusts the parameter by performing a sub-gradient [25] based on its own local data set. Increasing the network cycles, the accuracy might be improved. In [26], IG has been proposed with the addition of quantization technique which can be used in the presence of low bandwidth to reduce the bits of transmitted data. In [27], a cluster-based version of IG has been developed. It brings a better energy efficiency and robustness. Incremental Nelder-Mead Simplex (IS) has been proposed in [28] and [29] with the addition of boosting and re-sampling techniques, respectively. They introduce a better accuracy and convergence rate.

In [30] a new evolutionary based approach has been proposed based on the PSO algorithm, denoted as Distributed PSO (DP). In DP, the network is partitioned into a number of clusters, dedicating a swarm of particles for which. Then the regressor of each cluster is trained by employing PSO algorithm distributively within the cluster. The final model is obtained after combining the clusters models by the fusion center. This approach obtains a model closer to the centralized case, and decreases the latency significantly. However, its synchronous processes are in contrast with autonomous nature of WSNs. In addition, different clusters have their own cluster size and data pattern which are not taken into account by DP.

IVeP [31] is another PSO based distributed approach that learns the network regression model using a multi-objective optimization technique. They employ VEPSO model to perform the optimization task through inter- and intra-cluster cycles. The results show high prediction accuracy with moderate energy consumption.

In this paper, a modified version of DP algorithm is proposed that can simultaneously decrease the communication overheads as well as improves the final prediction accuracy. Firstly, Asynchronous DP (ADP), has been proposed by defining a diversity threshold for the particles within each cluster swarm. As a result, each cluster regressor is learned regardless of the status of the other clusters. Defining diversity thresholds, the number of transmissions is reduced. However the final accuracy might be decreased on the other hand. In this regard, Boosted ADP (BADP) has been introduced which boosts the clusters regressors and keeps the overall accuracy in high. The proposed algorithms have been compared with IG- and IS-based algorithms as well as IVeP and centralized approaches in terms of the accuracy, latency, and communication cost. The results show that ADP and BADP bring the lowest latency. Moreover, thanks to the boosting technique, BADP learns a model closer to the centralized approach while the communication cost still remains considerably acceptable. The contributions of this paper are:

- Asynchronous DP algorithm is proposed in which in-cluster optimization is performed asynchronously based on the size and data patterns of the cluster. While this is in accordance to the autonomous operations of the sensor networks, it brings more energy efficiency.
- The obtained model by ADP is boosted to improve the overall accuracy even further. Accordingly, Boosted ADP algorithm is proposed that obtains a high accurate network model and closer to the centralized approach with quite acceptable communication requirements.

The rest of this paper is organized as follows. Distributed regression problem is formally stated in section 2. The proposed approach is introduced in section 3. Evaluation and experimental results are discussed in section 4 and the last section is concluding remarks.

2- Distributed Regression in WSNs

Consider a sensor network with n nodes and m measurements per node spatially distributed in an

area. Every sensor node is expected to capture the phenomenon of interest in pre-defined time intervals [32]. Each measurement is stored as a record as:

$$< locx_i, locy_i, time_{i,t}, l_{i,t} >$$

in which ($locx_i, locy_i$) denotes to the i -th node's location, $time_{i,t}$ is epoch number, and $l_{i,t}$ is the captured measurement. Now, considering

$$A = \{locx_i, locy_i, time_{i,t}\}_{i=1,\dots,n}^{t=1,\dots,m}$$

as the feature space and:

$$B = \{l_{i,t}\}_{i=1,\dots,n}^{t=1,\dots,m}$$

as the labels, the aim of the parametric regression is to learn the coefficients of the mapping function $g: A \to B$, i.e. θ, such that the RMS error be minimized:

$$RMSE\big(g(A|\theta)\big) = \sqrt{\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}[g\big(locx_i, locy_i, time_{i,t}|\theta\big) - l_{i,t}]^2} \qquad (1)$$

Throughout this paper the following assumptions will be held:

- The learning process starts by disseminating a query from the fusion center to cluster heads.
- Every sensor node can localize itself by executing a well-known localization algorithm [33], [34].
- Since clustering is not the subject of this paper, it is assumed that the network is partitioned into C clusters via a well know clustering algorithm [35], [36], designating a cluster head for each cluster, $CH_1, \dots, CH_C$.
- The member nodes belonging to the cluster j are denotes as $\{sn_1^{(j)}, \dots, sn_{n_j}^{(j)}\}$ where n_j is the size of the cluster.
- The local data set of $sn_i^{(j)}$, cluster data j, and global network data are denoted as $LD_i^{(j)}$, $CD_j = \cup_{i=1}^{n_j}\{LD_i^{(j)}\}$, and $GD = \cup_{j=1}^{C}\{CD_j\}$, respectively.
- The L denotes the size of the parameter under estimate.

Table 1 shows the Nomenclature used in this study.

Table 1. Nomenclature used in this study.

Symbol	Definition
n	Number of sensor nodes
m	Number of sensor measurements
C	Number of clusters
CH_j	Cluster head j
$sn_i^{(j)}$	Sensor node i in cluster j
$LD_i^{(j)}$	The local data of $sn_i^{(j)}$
CD_j	The cluster data j
GD	The global(network) data
g_j	The cluster regression model j
g_{net}	The network regression model
A	The feature space
N_s	The swarm size
N_d	The problem dimensionality
$S^{(j)}$	The diversity of swarm j
$p_{i,d}$	The dimension d of particle i
τ_i^j	The weight of the local repressor of $sn_i^{(j)}$

3- The Proposed Approach

In Algorithm1, the basic idea of DP algorithm [13] has been recalled. The ADP is introduced afterwards. In summary, DP has the following steps:

1. Inside each cluster, every sensor node is given a swarm of particles to learn the cluster model. To do this, each cluster node obtains the model of its local data and sends it to other cluster nodes. Then, every cluster node employs the received local models to regenerate the whole cluster data. Now, every sensor is ready to start its local PSO to learn a candidate cluster regressor.
2. During the in-cluster optimizations, in order to guarantee the convergence of different swarms of the cluster nodes, the best particles are exchanged inside the cluster.
3. After completing the in-cluster optimization process, the cluster models are transmitted from the cluster heads to the fusion center.
4. The final network model is obtained by a weighted combination technique.

Algorithm 1: Distributed PSO (DP) [13]

Fusion Center disseminates the desired model
for each cluster j **do**
  data_view_unification()
  parameters_initialization()
  for i in range 1:$N_{migrations}$
    for each cluster node i **do**
      sn_i^j runs a local PSO
      sn_i^j sends its best particle to the CH_j
    end for
    CH_j sends the best of the best particles to its members
  end for
  CH_j sends g_j and its RMSE to the fusion center
end for
The fusion center obtains G_{net} by weighted averaging

3-1- Asynchronous DP (ADP)

The major drawback of DP is that the migration steps should be synchronized for all clusters. In more words, the particles of a particular cluster might be converged before the final migration, while more migration steps might be required in another cluster. This is because different clusters have different data patterns and cluster size. By eliminating the extra migrations inside the converged clusters, the energy consumption is reduced. Furthermore, the synchronized clusters are in contrast with the autonomous nature of WSNs. To resolve these issues, asynchronous DP (ADP) is introduced in this section.

Attractive and Repulsive PSO, called as ARPSO, is a variant of PSO model in which the particles can switch between two phases [37], [38]. This approach is based on the diversity guided evolutionary algorithm (DGEA) developed by [39]. In ARPSO, the particles obey from the diversity of the swarm to alternate between an attraction and repulsion phases to make a proper exploitation-exploration tradeoff. Accordingly, the swarm diversity is defined as:

$$diversity\big(S(z)\big) = \frac{1}{N_s}\sum_{i=1}^{N_s}\sqrt{\sum_{d=1}^{N_d}(p_{i,d}(z) - \overline{p_d}(z))^2} \quad (2)$$

where N_s is the swarm size, N_d is the dimensionality of the problem, and $\overline{p_d}$ is the average of the dimension d over all the particles, i.e.

$$\overline{p_d}(z) = \frac{\sum_{i=1}^{N_S} p_{i,d(z)}}{N_s} \quad (3)$$

Although ARPSO was originally applied to one swarm, nothing prevents its application to sub-swarms [40].

The diversity equation in ARPSO has been adopted in ADP for measuring the diversity of the clusters swarms. In cluster j, the diversity is calculated using only the best particles received from the cluster nodes:

$$diversity\left(S^j(z)\right) = \frac{1}{n_j}\sum_{i=1}^{N_j}\sqrt{\sum_{d=1}^{N_d}(gbest_i^j(z) - \overline{gbest_d}(z))^2} \quad (4)$$

where:

$$\overline{gbest_d}(z) = \frac{\sum_{i=1}^{n_j} gbest_{i,d}^j(z)}{n_j} \quad (5)$$

If the diversity (Eq. 4) be greater than a threshold φ, the in-cluster optimization is stopped, and the cluster regressor is transmitted to the fusion center as well as the corresponding RMS error. The final model is obtained by the fusion center similar to the idea proposed in DP algorithm. The steps of ADP is shown in Algorithm 2.

3-2- Boosted ADP (BADP)

Defining smaller thresholds, the quality of the clusters models are expected to be increased in ADP algorithm. However, it brings more communication cost. In this regard, in order to keep both energy efficiency and high accuracy, a boosting technique is applied on ADP inspiring from [28]. In Boosted ADP (BADP) algorithm, firstly, the clusters regressors are obtained using a diversity threshold, as explained in ADP. Then, each cluster model is boosted before transmitting to the fusion center. To do this, within the cluster j, the cluster head broadcasts the final obtained regressor and the size of the

Algorithm 2: Asynchronous PSO (ADP)

Fusion Center disseminates the desired model
for each cluster j **do**
  data_view_unification()
  parameters_initialization()
  for each cluster node i **do**
    sn_i^j runs a local PSO
    sn_i^j sends its best particle to the CH_j
  end for
  CH_j calculates the cluster diversity, i.e. Eq. 4
  if the diversity is larger than φ **then**
    CH_j sends the best of the best particles to its members
  else
    CH_j sends g_j and its RMSE to the fusion center
  end if
end for
The fusion center obtains G_{net} by weighted averaging

cluster data to its member nodes. Each member node, e.g. s_i^j ,tests the cluster model on its own local data set and calculates a partial weight for it, ω_i^j .Afterwards, a new learner, υ_i^j, is trained over data points labeled incorrectly by the cluster model [28]. Similarly, the new learner is test over the local data set and a local weight is computed [11]:

$$\tau_i^j = \frac{\#\ of\ truly\ labaled\ data\ points}{|CD_j|} \qquad (6)$$

The new learner acts as a weak learner when applying on the cluster data, as it has been trained over a small data set. So, it should be combined with the new leaners obtained by the other cluster nodes to build a second stronger regressor. For this pupose, s_i^j sends $\tau_i^j \times \upsilon_i^j$ as well as ω_i^j to the cluster head. The cluster head aggregates the received partial weights to compute the weight of its regressor, ω_j . Now, a new boosted cluster model is obtained as:

$$g_j^{boosted} = w_j \times g_j + \sum_{i=1}^{n_j} \tau_i^j \times \upsilon_i^j \qquad (7)$$

The last step is to calculate the in-cluster RMS error of the new boosted model using the cluster data as explained in ADP algorithm. Finally, the boosted model and its RMS error are sent to the fusion center, and the global model is obtained. Algorithm 3 describes the steps of BADP algorithm. Although the computational complexity has not been found as a major concern in WSNs, we can provide an estimation of the computational complexity for a single

Algorithm 3: Boosted ADP (BADP)

Fusion Center disseminates the desired model
for each cluster j **do**
  data_view_unification()
  parameters_initialization()
  for each cluster node i **do**
    sn_i^j runs a local PSO
    sn_i^j sends its best particle to the CH_j
  end for
  CH_j calculates the cluster diversity, i.e. Eq. 4
  if the diversity is larger than φ **then**
    CH_j sends the best of the best particles to its members
  else
    CH_j sends g_j and the cluster data size to its members
    for each cluster node i **do**
      sn_i^j tests g_j^{ADP} on LD_i^j and obtains two data partitions as $LD_{true,i}^j$ and $LD_{false,i}^j$
      sn_i^j computes ω_i^j, the partial weight of g_j^{ADP}
      sn_i^j runs a local PSO over $LD_{false,i}^j$ to learn v_i^j
      sn_i^j computes τ_i^j, the weight of v_i^j as Eq. 6.
      sn_i^j sends $\tau_i^j \times v_i^j$ and ω_i^j to CH_j
    end for
    CH_j computes ω_j using $\{\omega_i^j\}_{i=1}^{n_j}$
    CH_j computes its final regressor $g_j^{boosted}$ as Eq. 7
    CH_j sends g_j and its RMSE to the fusion center
  end if
end for
The fusion center obtains G_{net} by weighted averaging

sensor node belonging to the cluster j in DP, ADP, and BADP algorithms. For DP algorithm we have:

$$T(DP) = \mathrm{T}(\text{ data_view_unification step}) + \mathrm{T}(\text{ optimization })$$

where:

$$T(\text{ data_view_unification }) = t(\text{local PSO}) + \mathrm{O}(mk)$$

in which $t(\text{local PSO})$ denotes the computational time of running the PSO algorithm over the cluster data by the sensor node and $\mathrm{O}(mk)$ is the required time for resampling of m measurements using a k-parameters data model and:

$$\mathrm{T}(\text{ optimization }) = \mathrm{O}\big(M \times t(local\ PSO)\big)$$

where M denotes the number of migration steps. Similarly, for ADP and BADP algorithms we have:

$$\mathrm{T(ADP)} = \mathrm{O(local\ PSO)} + \mathrm{O}(mk) + \mathrm{O}(M_j \times t(local\ PSO))$$

where M_j is the required migration steps for the corresponding cluster j and:

$$T(BADP) = t(local\ PSO) + O(mk) + O(M_j \times t(local\ PSO)) + t(local\ PSO\ for\ the\ boosting\ task)$$

Totally, the computational time complexity of a sensor node in DP algorithm could be simplified as:

$$\mathrm{T(DP)} = O(M \times t(local\ PSO)) + \mathrm{O}(mk)$$

and for ADP and BADP we would have:

$$\mathrm{T(ADP)} = \mathrm{T(BADP)} = O(M_j \times t(local\ PSO)) + \mathrm{O}(mk)$$

4- Evaluation and Results

The proposed algorithms have been compared with their distributed counterparts, IG, IS, BIS, IS-Resampling, IVeP and the centralized approach. In all of these algorithms, the learning problem is formulated as an optimization task, as discussed in Section 2. Two datasets have been used for comparison. In the first one, Berkeley Intel Lab network [41], there are 54 sensor nodes with two corrupted ones. Mica2Dot sensors with weather boards capture humidity, light, voltage, and temperature in every 31 seconds. As mentioned before, regression modeling has been performed only over temperature readings. Two portions of the network data, named as DS1 and DS2, have been chosen such that each sensor node has 100 and 2880 (measuring for one day) data points, respectively. So, the global data, GD, has 5200 and 149760 data points in total, respectively. The second network, denoted as DS3, is an artificial network with 100 sensors distributed uniformly over a square of 100 m^2. Each sensor has collected a dataset of size 200. The phenomenon under study, temperature, is sensed in each epoch from 1 to 200. This data is generated using Eq. (8) with an additive Gaussian noise of mean 0 and variance 1, $N(0,1)$. The coefficients of the model are randomly chosen in the range of (-10, 10).

Ten Fold Cross Validation method (10-CV) has been adopted for each approach. In 10-CV, the data set is divided into 10 partitions for 10 times. Each time, one of the partitions is used as the test data and the learning will be executed with the remaining parts. Finally, the average results will be found. In [42], some polynomial models have been suggested for the Berkeley network data set. It was reported that a linear space and quadratic time model can be fitted more accurately. Accordingly, in order to fit a model on the network data, a spatiotemporal model with linear in space and quadratic in time has been chosen as:

$$G_{net}(locx, locy, time|\theta) = \theta_1 locx + \theta_2 locy + \theta_3 time^2 + \theta_4 time + \theta_5 \quad (8)$$

where the node location $(locx, locy)$ and epoch number ($time$) introduce the feature set while the captured temperature per epoch is the label. Therefore, the proposed approach aims to learn the coefficients of a set of basis functions as $\{locx, locy, time^2, time, 1\}$. Accordingly, the RMS error could be calculated as:

$$RMSE\big(G_{net}(A|\theta)\big) = \sqrt{\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}[\theta_1 locx_{ij} + \theta_2 locy_{ij} + \theta_3 time_{ij}^2 + \theta_4 time_{ij} + \theta_5 - T_{ij}]^2} \quad (9)$$

In order to have a good exploration-exploitation tradeoff, the inertia weight, w ,is usually decreased during the time as [43]:

$$w(z) = w_{min} + \frac{m_i - z}{m_i}(w_{max} - w_{min}) \quad (10)$$

where m_i is the maximum number of iterations and z denotes to the current iteration number. The particles starts with a maximum value w_{max} and linearly decrease their inertia weights to a pre-defined minimum value w_{min}.

As the problem addressed in this study is a data-centric application, the discrete-event simulators are not required. Accordingly, all the algorithms have been implemented with Java using Eclipse IDE and the experiments were performed on an Intel dual core processor with 4 GB RAM memory.

4-1- Prediction Accuracy

The prediction accuracy of different approaches have been shown in Table 2. As all the data points are available for the centralized approach, a good accuracy is achieved at the end of the learning process. In practice, IG suffers from a low convergence rate and requires to pass several cycles in order to obtain an average accuracy. In our experiments, the accuracy of IG has been obtained through 40 cycles. While IS based approaches obtain better results within one network cycle.

In ADP, thanks to (i) learning several candidate models and (ii) the high accuracy of each cluster model, a good accuracy is achieved. However, integrating the boosting technique with the ADP algorithm leads the final accuracy becomes much closer to the centralized case and consequently BADP outperforms its distributed counterparts in most cases. Moreover, BADP

shows more stable accuracies in both networks rather than the other methods. This indicates how the boosted

Table 2. The final RMS error of different approaches based on each data set.

Approach	DS1	DS2	DS3
IG	17.481	21.549	110.678
IS	8.206	5.059	13.452
BIS	6.268	5.011	9.647
IS-Res.	5.806	3.104	11.714
IVeP	2.219	3.009	3.348
ADP	2.060	4.311	3.448
BADP	1.892	2.917	3.017
Central	0.835	2.536	1.005

Table 3. The RMSE comparison of BADP algorithm with [44].

Algorithm	RMSE
BADP	2.917
[44]-LG	≅ 2.35
[44]-PV	≅ 2.75
[44]-UV	≅ 3.02
[44]-RA	≅ 2.91

regressor can accurately predict those parts of the phenomena that labeled inaccurately by the first learner.

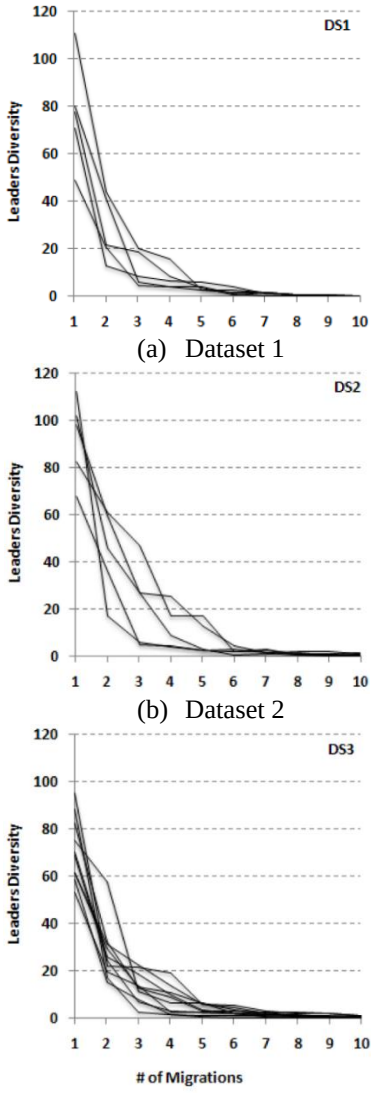
The prediction accuracy of the BADP has also been compared with the reported results of [44] where a multi-objective sensor placement algorithm has been proposed. The performance of the state estimation of the temperature measurements has been evaluated based on the RMSE. As shown in Table 3, the proposed BADP could obtain better or completely close prediction accuracy compared to [44] which is not a distributed algorithm.

In Figure 1, the convergence rate of different cluster swarms have been depicted based on three data sets. As it is expected, some cluster swarms converge faster than the other ones due to their size and data patterns. As a result, lesser in-cluster communications would be required. To show how the swarms diversity could decrease the energy consumption, 5 diversity thresholds have been defined for each dataset and shown in Table 4.

The prediction accuracy using each defined diversity threshold has been demonstrated in Figure 2 along with the corresponding amount of energy saving in comparison with the DP algorithm. It is concluded from the Figure 2 that the energy consumption of DP algorithm could be decreased up to %48 by the proposed approach while the final RMSE is quite high. As it was shown, by using more tight diversity thresholds, the final prediction accuracy is increased. However, a trade-off should be made between the energy consumption and the prediction accuracy using diversity thresholds. It should be noticed that the reported results in Table 1 are based on φ_4.

Table 4. Swarms diversity thresholds

Dataset	φ_1	φ_2	φ_3	φ_4	φ_5
DS1	3.50	2.25	1.25	0.75	0.50
DS2	4.75	2.50	2.25	1.50	1.00
DS3	3.75	2.50	1.75	1.25	0.75



(a) Dataset 1

(b) Dataset 2

(c) Dataset 3

Figure 1. The convergence rate of different clusters' swarm

4-2- Latency

Regarding to the ongoing sensor measurements, the network model is valuable for some pre-specified periods of time. Thus, when the measurements is refreshed, it is required to train the regression model with the new network data. In this regard, the required time to rebuild the model is important, known as the latency metric: *the*

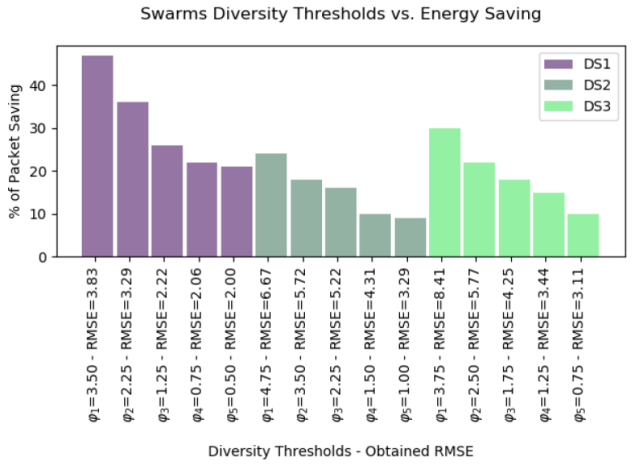


Figure 2. The impact of varying swarms diversity thresholds on the energy consumption.

number of iterations to visit all the network data for the first time [27]. The latency of different approaches has been compared in Table 5.

The centralized approach just needs one iteration to visit the whole network data, and so, its latency is $O(1)$. In IG and IS based approaches, the latency is $O(n)$, as they need one network cycle to get access to the network data. In IVeP algorithm, one in-cluster cycle is required and thus its latency is $O(n^*)$ where n^* is the size of the largest cluster. If we let the number of clusters equals to $\sqrt{n}$, then the latency of IVeP would be $O(\sqrt{n})$. In ADP and BADP, the training process is started synchronously in all clusters. Consequently the latency would be the same as the centralized method, i.e. $O(1)$.

4-3- Communication Cost

Bit/meter metric has been used for comparing the communication requirements of the approaches. According to [24], assume the network has been deployed in a unit square area. Having the size and average distance of each transmission, the communication requirements of each approach can be evaluated. Every transmission in all the studied approaches falls in one of the following transmission types and the corresponding average distance can be achieved similar to [24]:

cluster node - cluster node:

Table 5. Latency comparison

Algorithm	Latency
IG	$O(n)$
IS	$O(n)$
BIS	$O(n)$
IS-Resampling	$O(n)$
IVeP	$O(\sqrt{n})$
ADP	$O(1)$
BADP	$O(1)$
Centralized	$O(1)$

$$d_1 = O(\sqrt{log^2 n/n})$$

cluster head - cluster head:

$$d_2 = O(\sqrt{log^2 \sqrt{n}/\sqrt{n}})$$

sensor node - fusion center:

$$d_3 = O(1)$$

cluster head - cluster node:

$$d_4 = O(1/\sqrt{n})$$

cluster head - fusion center:

$$d_5 = O(1/\sqrt{n})$$

The size of the parameter(s) transmitted between two consecutive sensors in IG and IS based approaches are as follows:

- In IG algorithm, a double vector of size L is transmitted [24].
- In IS algorithm and the first pass of BIS, $|LD_k|$ (an integer of size 1) and a double vector of size L are transmitted [28].
- In the second pass of BIS, three parameter transmissions are happened: the partial weight of the learned regressor (a double value), the size of the global data, $|GD|$ (an integer of size 1), and the partial weighted combination of local regressors which is a double vector of size L.
- In IS-Resampling algorithm, a double vector of size L and a double vector of size 2 $(locx_k, locy_k)$ are transmitted.

In the centralized approach, m data values are transmitted between a sensor node and the fusion center. Thus, n transmissions of size υ would be required. As each data point in our experiments contains three features with a label, we have $\upsilon = 4$.

The communication requirements of the ADP algorithm is similar to DP. Firstly, a parameter of size L is transmitted from the cluster nodes to cluster

Table 6. Types and size of transmissions of different approaches

	Sensor-Sensor	Sensor-Fusion
IG	$\mathcal{C}_{IG} \times (n-1)L$	L
IS	$\mathcal{C}_{IS} \times (n-1)(L+1)$	L
BIS	$cost(IS) + 2(n-1)(L+1)$	$2L$
IS-R	$\mathcal{C}_{IS-R} \times (n-1)(L+2)$	L
Centralized		nmv
	Sensor-CHead	CHead-Fusion
ADP	$3nL + n - LC + (2L+1)\sum_{j=1}^{C} l_j n_j$	$C(L+1)$
BADP	$4nL + 2n - 2LC - C + (2L+1)\sum_{j=1}^{C} l_j n_j$	$C(L+1)$

Table 7. Comparing the communication order. Without loss of generality, we follow [27] and let $\mathtt{C} = \sqrt{\mathtt{n}}$ and K denotes the average number of iterations in ADP and BADP algorithms.

Approach	Communication requirement	Rank
IG	$\mathcal{O}((\mathcal{C}_{IG})(L)(\sqrt{n}.logn + 1))$	7
IS	$\mathcal{O}((\mathcal{C}_{IS})(L)(\sqrt{n}.logn + 1) + L)$	1
BIS	$\mathcal{O}(IS) + \mathcal{O}((L)(\sqrt{n}logn + 1))$	3
IS-Resmapling	$\mathcal{O}((\mathcal{C}_{IS-Res})((L+2).\sqrt{n}.logn) + L)$	2
IVeP	$\mathcal{O}(n(L\sqrt{n}) + 3\sqrt{n}L + \sqrt{\sqrt{n}L}\log n)$	6
ADP	$\mathcal{O}((\mathcal{K} + 2L\mathcal{K} + 3L + 1)\sqrt{n} + 1)$	4
BADP	$\mathcal{O}((\mathcal{K} + 2L\mathcal{K} + 4L + 2)\sqrt{n} - L)$	5
Central	$\mathcal{O}((n)(m)(v)(1))$	8

head, and vice versa. Then, a driver message by size of $2L + 1$ is transmitted from the cluster head to the members. Afterwards, during the in-cluster optimization, the best particle of each cluster node by size of $2L$ with an RMS error of size 1, and the best of the best particles by size of $2L$ are transmitted between the cluster head and the cluster members at each migration step. Then, each cluster node sends its regressor with the corresponding RMS error to the cluster head by size of L+1. Finally, each cluster head sends the obtained cluster model with its RMS error by size of $L + 1$ to the fusion center. In BADP algorithm, every cluster node transmits an extra parameter of size $L + 1$, new learner plus its partial weight, to the cluster head. In Table 6, the parameter transmissions of all approaches have been summarized based on transmission size and type. For IVeP algorithm, the communication cost analysis is recalled from [31]. Accordingly, the total communication cost of different approaches are obtained as shown in Table 7.

In this regard, the communication cost of the centralized approach is the highest due to a huge data transmission. As mentioned before, IG practically needs to meet a large number of cycles to obtain an average accuracy. As a result, its energy consumption is higher than the other distributed approaches while IS has the lowest transmissions. From Table 6, it can be understood that in-cluster communications consume less energy due to a smaller average distance. On the other hand, the main part of the transmissions in ADP as well as BADP has been spent in clusters. Thanks to this property, ADP and BADP both work moderately in terms of the energy consumption, as shown in Table 7.

5- Conclusion

A novel distributed data modeling approach has been proposed based on multi-swarm PSO algorithm. In the proposed approach, the task of learning the regression model of a cluster is assigned to a swarm of particles. Each swarm executes an in-cluster optimization process to learn the cluster regressor asynchronously. The most important feature of this approach is that the optimization of each swarm is terminated according to the size and data pattern of its cluster. This property leads to save up to 48% of energy consumption by eliminating extra migration steps while the accuracy is high. In order to improve the prediction accuracy even further, a boosting technique is also employed in a distributed manner. The proposed approach has been evaluated against two real and artificial network data and compared to common distributed regression modeling techniques as well as the centralized approach. The results show the boosted model improves the prediction accuracy about 9% on the average. Due to the recent advances in the sensor nodes technologies, more complex machine learning algorithms, such as Deep Learning, could be employed to achieve higher prediction accuracies in the context of WSNs [45], [46].

References

[1] Sharma, Himanshu, Ahteshamul Haque, and Frede Blaabjerg. "Machine Learning in Wireless Sensor Networks for Smart Cities: A Survey." Electronics 10.9 (2021): 1012.

[2] Liu, Longgeng, et al. "An algorithm based on logistic regression with data fusion in wireless sensor networks." EURASIP Journal on Wireless Communications and Networking 2017.1 (2017): 1-9.

[3] Deng, Yulong, et al. "Temporal and spatial nearest neighbor values based missing data imputation in wireless sensor networks." *Sensors* 21.5 (2021): 1782.

[4] Zuhairy, Ruwaida M., and Mohammed GH Al Zamil. "Energy-efficient load balancing in wireless sensor network: An application of multinomial regression analysis." International Journal of Distributed Sensor Networks 14.3 (2018): 1550147718764641.

[5] Kumar, D. Praveen, Tarachand Amgoth, and Chandra Sekhara Rao Annavarapu. "Machine learning algorithms for wireless sensor networks: A survey." Information Fusion 49 (2019): 1-25.
[6] Ghate, Vasundhara V., and Vaidehi Vijayakumar. "Machine learning for data aggregation in WSN: A survey." International Journal of Pure and Applied Mathematics 118.24 (2018): 1-12.
[7] Ren, Xiaoxing, et al. "Distributed Subgradient Algorithm for Multi-Agent Optimization With Dynamic Stepsize." IEEE/CAA Journal of Automatica Sinica 8.8 (2021): 1451-1464.
[8] Doan, Thinh T., Siva Theja Maguluri, and Justin Romberg. "Convergence rates of distributed gradient methods under random quantization: A stochastic approximation approach." IEEE Transactions on Automatic Control (2020).
[9] Zhang, Peng, and Gejun Bao. "An incremental subgradient method on Riemannian manifolds." Journal of Optimization Theory and Applications 176.3 (2018): 711-727.
[10] Berahas, Albert S., Charikleia Iakovidou, and Ermin Wei. "Nested distributed gradient methods with adaptive quantized communication." 2019 IEEE 58th Conference on Decision and Control (CDC). IEEE, 2019.
[11] Xu, Xiangxiang, and Shao-Lun Huang. "An information theoretic framework for distributed learning algorithms." 2021 IEEE International Symposium on Information Theory (ISIT). IEEE, 2021.
[12] Perumal, T. Sudarson Rama, V. Muthumanikandan, and S. Mohanalakshmi. "Energy Efficiency Optimization in Clustered Wireless Sensor Networks via Machine Learning Algorithms." Machine Learning and Deep Learning Techniques in Wireless and Mobile Networking Systems. CRC Press, 2021. 59-77.
[13] Kumar, D. Praveen, Tarachand Amgoth, and Chandra Sekhara Rao Annavarapu. "Machine learning algorithms for wireless sensor networks: A survey." Information Fusion 49 (2019): 1-25.
[14] Mohanty, Lipika, et al. "Machine Learning-Based Wireless Sensor Networks." Machine Learning: Theoretical Foundations and Practical Applications. Springer, Singapore, 2021. 109-122.
[15] Pundir, Meena, and Jasminder Kaur Sandhu. "A systematic review of Quality of Service in Wireless Sensor Networks using Machine Learning: Recent trend and future vision." Journal of Network and Computer Applications (2021): 103084.
[16] Antonian, Edward and Peters, Gareth and Peters, Gareth and Chantler, Michael John and Yan, Hongxuan, GLS Kernel Regression for Network-Structured Data (August 9, 2021). Available at SSRN: https://ssrn.com/abstract=3901694
[17] Liu, Longgeng, et al. "An algorithm based on logistic regression with data fusion in wireless sensor networks." EURASIP Journal on Wireless Communications and Networking 2017.1 (2017): 1-9.
[18] Wang, Heyu, Lei Xia, and Chunguang Li. "Distributed online quantile regression over networks with quantized communication." Signal Processing 157 (2019): 141-150.54532343WE332
[19] Wang, Heyu, and Chunguang Li. "Distributed quantile regression over sensor networks." IEEE Transactions on Signal and Information Processing over Networks 4.2 (2017): 338-348.
[20] Danaee, Alireza, Rodrigo C. de Lamare, and Vitor H. Nascimento. "Energy-Efficient Distributed Recursive Least Squares Learning with Coarsely Quantized Signals." 2020 54th Asilomar Conference on Signals, Systems, and Computers. IEEE, 2020.
[21] Danaee, Alireza, Rodrigo C. de Lamare, and Vitor H. Nascimento. "Energy-efficient distributed learning with coarsely quantized signals." IEEE Signal Processing Letters 28 (2021): 329-333.
[22] Hellkvist, Martin, Ayça Özçelikkale, and Anders Ahlén. "Generalization error for linear regression under distributed learning." 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC). IEEE, 2020.
[23] Shuman, David I., et al. "Distributed signal processing via Chebyshev polynomial approximation." IEEE Transactions on Signal and Information Processing over Networks 4.4 (2018): 736-751.
[24] M. Rabbat and R. Nowak, "Distributed Optimization in Sensor Networks," in Proceedings of the 3rd international symposium on Information processing in sensor networks, Berkeley, California, USA, (2004), pp. 20-27.
[25] Bertsekas, Dimitri P., "Incremental gradient, subgradient, and proximal methods for convex optimization: a survey," Optimization for Machine Learning, No. 85, pp. 1-38, 2011.
[26] M. Rabbat, and Nowak, R. "Quantized Incremental Algorithms for Distributed Optimization," IEEE Journal on Selected Areas in Communications, 23 (4) (2006), pp. 798-808.
[27] S.H. Son, M. Chiang, S. R. Kulkarni, and S. C. Schwartz, "The Value of Clustering in Distributed Estimation for Sensor Networks," in proceedings of International Conference on Wireless Networks, Communications and Mobile Computing, Maui, Hawaii, 2 (2005), pp. 969-974.
[28] P.J. Marandi, N.M. Charkari, "Boosted Incremental Nelder-Mead Simplex Algorithm: Distributed Regression in Wireless Sensor Networks," IFIP Joint Conference on Mobile and Wireless Communications Networks, France, (2008), pp. 199-212.
[29] P.J. Marandi, M. Mansourizadeh, N. M. Charkari, "The Effect of Resampling on Incremental Nelder-Mead Simplex Algorithm: Distributed Regression over Wireless Sensor Network," in Proceedings of the Third International Conference on Wireless Algorithms, Systems, and Applications, LNCS, 5258 (2008), Dallas, Texas, pp. 420-431.
[30] H. Shakibian and N. Moghadam Charkari, "D-PSO for Distributed Regression over Wireless Sensor Networks," Iranian Journal of Electrical and Computer Engineering, Vol. 11, No. 1, pp. 43-50, 2012.
[31] Shakibian, Hadi, and Nasrollah Moghadam Charkari. "In-cluster vector evaluated particle swarm optimization for distributed regression in WSNs." Journal of network and computer applications 42 (2014): 80-91.
[32] Zhao, Jijun, Hao Liu, Zhihua Li, and Wei Li., "Periodic Data Prediction Algorithm in Wireless Sensor Networks,"

In Advances in Wireless Sensor Networks, pp. 695-701, 2013.
[33] Cheng, Long, et al. "An Indoor Localization Algorithm based on Modified Joint Probabilistic Data Association for Wireless Sensor Network." IEEE Transactions on Industrial Informatics (2020).

[34] Shahbazian, Reza, and Seyed Ali Ghorashi. "Distributed cooperative target detection and localization in decentralized wireless sensor networks." The Journal of Supercomputing 73.4 (2017): 1715-1732.
[35] Zandhessami, Hessam, Mahmood Alborzi, and Mohammadsadegh Khayyatian. "Energy Efficient Routing-Based Clustering Protocol Using Computational Intelligence Algorithms in Sensor-Based IoT." Journal of Information Systems and Telecommunication (JIST) 1.33 (2021): 55.
[36] Daanoune, Ikram, Baghdad Abdennaceur, and Abdelhakim Ballouk. "A comprehensive survey on LEACH-based clustering routing protocols in Wireless Sensor Networks." Ad Hoc Networks (2021): 102409.
[37] Qi, Hong, et al. "Inversion of particle size distribution by spectral extinction technique using the attractive and repulsive particle swarm optimization algorithm." Thermal Science 19.6 (2015): 2151-2160.
[38] Mo, Simin, Jianchao Zeng, and Weibin Xu. "Attractive and repulsive fully informed particle swarm optimization based on the modified fitness model." Soft Computing 20.3 (2016): 863-884.
[39] Ursem, Rasmus K. "Diversity-guided evolutionary algorithms." International Conference on Parallel Problem Solving from Nature. Springer, Berlin, Heidelberg, 2002.
[40] A.P. Engelbrecht, Computational Intelligence: An introduction, 2ed., Wiley, 2007.
[41] Madden, S., 2003. Intel Berkeley research lab data. USA: Intel Corporation, 2004 [2004-06-08]. http://berkeley, intel-research, net/labdata, html.
[42] C. Guestrin, P. Bodi, R. Thibau, M. Paskin, and S. Madde, "Distributed Regression: An Efficient Framework for Modeling Sensor Network data," in Proceedings of third international symposium on Information processing in sensor networks, Berkeley, California, USA, (2004), pp. 1-10.
[43] Y. Shi, R.C. Eberhart, "Empirical study of particle swarm optimization," in Proceedings of the IEEE International Congress on Evolutionary Computation, 3 (1999), pp. 101-106.
[44] Xu, Zhaoyi, Yanjie Guo, and Joseph Homer Saleh. "Multi-objective optimization for sensor placement: An integrated combinatorial approach with reduced order model and Gaussian process." Measurement 187 (2022): 110370.
[45] Premkumar, M., and T. V. P. Sundararajan. "DLDM: Deep learning-based defense mechanism for denial of service attacks in wireless sensor networks." Microprocessors and Microsystems 79 (2020): 103278.
[46] Mohanty, Sachi Nandan, et al. "Deep learning with LSTM based distributed data mining model for energy efficient wireless sensor networks." Physical Communication 40 (2020): 101097.