Intelligent Environments 2016
P. Novais and S. Konomi (Eds.)

doi:10.3233/978-1-61499-690-3-318

Data Similarity Based Dynamic Node Clustering Using Bio-Inspired Algorithm for Self-Organized Wireless Sensor Networks

Misbahuddin[1] and Riri Fitri Sari
Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Indonesia

Abstract. Wireless sensor networks have been used by various environmental sensing applications as an interface between physical and computational environments. The wireless data gathering techniques commonly use the node clustering approaches to reduce energy consumption in the network. Moreover, it can also organize the network to be a stable network, improve the scalability, and prolong the lifetime of the cluster head through load balancing. However, few node clustering methods consider the data similarity based cluster formation referred on the spatial correlation to reduce the size of the large amounts of data and to detect the anomalous events. Furthermore, it can also manage the network through a self-organized approach to reduce the fault in the network. In this article, we propose an approach for data similarity based node clustering through a bio-inspired algorithm as a self-organizing mechanism. The simulation results show that the firefly-inspired synchronicity can synchronize the nodes on the network in order to perform together actions simultaneously. The clustering technique can establish the clusters based on the spatial similarity readings and can detect the anomalous data robustly.

Keywords. Data similarity, self-organized, node clustering, bio-inspired algorithm

1. Introduction

Wireless Sensor Networks (WSNs) have been used in various disciplines due to their ease of installation for both offline and online applications. WSNs are one of the major data sources for the Internet of things. WSNs play an important role to support various high-potential applications such as agriculture [1], healthcare, intelligent transportation system [2], military surveillance [3], industry automation [4], and industrial robotics [5]. WSN consists of a set of autonomous sensor nodes that are spatially deployed over the area of interest. The sensor node is a device that can capture a physical parameter, forwarding it into a processor for preprocessing, and then transmitting it through RF transceivers to the base station.

The node sensors are powered by battery as a limited energy source therefore, their life span is finite. Instead, WSNs have the main duty to collect data accurately. Therefore,

[1] Corresponding Author, Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Kampus Baru Depok UI, 16424, Indonesia; E-mail: misbahuddin@ui.ac.id.

WSNs as the self-contained systems have to consider the energy saving in conducting their services.

The WSN's activity consuming large amount energy is the data communication. Transmitting the redundant data consumes the valuable energy of the node, so that it can waste the energy. Moreover, such redundant data does not contribute any information benefit [6]. Accordingly, the redundant data transmissions have to be minimized in order to increase the energy saving significantly. Data similarity is a way to discover the redundant data, which is measured based on either the temporal relation or spatial relation. The temporal relation is the similarity degree between successive observation of the node over the time, which generate the node's data series. Such data series obtained through the higher sampling rate can produce smooth grained data quality. However, it results in a significant amount of redundant data. Likewise, the data similarity between adjacent nodes over a region is known as the spatial relation. To cover the whole sensed region of WSN, nodes are often closely deployed. This case can increase the spatial relation between neighboring nodes. Consequently, the spatial redundant data also increases along with the energy consumption and the network bandwidth.

Clustering techniques are commonly used for better data aggregation with more efficient energy consumed through organizing nodes into a hierarchically logical cluster. Furthermore, they are also used to increase better network scalability, and prolong the life span of network through the load balancing. However, a few clustering approaches consider the data similarity referred on either temporal or spatial relation to eliminate the data redundancy [7], [8]. In many previous approaches, the temporal redundancy is reduced by collecting and storing the data in an array to be compressed at the node itself. Such approach is unsuitable for real-time data aggregation. The spatial redundancy is eliminated by selecting the data based on its relative importance, and omitting simply the insignificant data.

There are some difficulties to design the clustering construction that are able to provide an adaptive operation in keeping continuous nodes grouped together based upon the similarity readings. Ideally, its operation should avoid a coordinator like a sink to construct the clustering. However, it requires a distributed and self-organized way. Such requirements can be found in various biological systems, whose behaviors have inspired the distributed systems such as the firefly-inspired synchronicity mechanism in the distributed networks.

In this work, our proposed clustering approach does not emphasize the energy saving because this issue has been explored by other researchers [9], [10]. Nevertheless, we propose a dynamic node clustering approach in WSN to support more efficient data aggregation referred to the spatial similarity relation, and to detect anomalous events. Thus, our main contributions consist of: (i) the usage of reachback firefly synchronicity through a clustering schema approach to synchronize the node's action in sensing environmental information. This approach will be faster for convergence than the flat network approach; (ii) the node clustering based upon the data similarity operates dynamically in a self-organized way without guided by the sink; (iii) the usage of a fuzzy aggregation technique to measure the data similarity degree can be more independent of the absolute threshold. The absolute threshold is a value highly depending on the data type.

The reminder of the paper is organized as follows: Section 2 presents the related works. Section 3 describes the proposed scheme for developing the dynamic node clustering based on spatial data similarity. Section 4 presents the simulation results. Finally, Section 5 concludes this paper and ideas of future work.

2. Related Works

The data aggregation techniques ware previously conducted on a flat sensor network [11]. However, due to the advantage of clustering approaches on energy efficiency, fault tolerance, scalability, and load balancing, most data aggregation techniques are based on the node clustering approach [12]. Furthermore, due to increase the need of WSN's applications, the node clustering techniques are also used for handling various applications such as the target tracking systems [13], the continuous data collection systems that can be applied in the environmental assessments and industrial process systems [7], and the monitoring systems of the real-time environmental condition pattern [8].

The clustering techniques to tackle the problem in the real world should be adapted based upon the requirements of application. For example, the target tracking system in [13] uses a hybrid clustering based data aggregation mechanism, i.e. static and dynamic node clustering. The data aggregation based on the static node clustering is used when the sensed target is closer to the sink, therefore, the data packets can be sent directly to the sink without constructing a cluster. Instead, when the location is far from the sink, the data aggregation based on the dynamic node clustering is chosen. accordingly, the data aggregation and forwarding to the sink are only handled by a cluster head.

The continuous data collection system uses the node clustering method based on the temporal similarity correlation to reduce the data redundancy. The method developed by [14] eliminated the data redundancy by compressing the data, where an array in the node is used to collect and store its data. Another method developed by [7] conducted in three stages, i.e. (i) least mean squares (LSM) filter is used to construct the data projection in each node; (ii) digital filters are used to interpret the data on the time domain; and (iii) a multilevel lossless data compression scheme is used to be applied in inter-cluster communication. This method can work in terms of energy efficiency, data accuracy and scalability. However, these methods may not be suitable for real-time data collection.

A protocol supporting in data similarity aware node clustering proposed by [8] has different objective to the previous mentioned methods. The protocol was developed to support more efficient data aggregation and to detect the anomalous events. However, this protocol needs to improve its performance in the convergence rate of the synchronicity, the sensitivity of abnormal event's detection, the balance of the cluster load.

3. Proposed Scheme

3.1 Network Model

The network model is initially formed as a flat network that consists of randomly deployed sensor nodes. Next, the flat network is converted into a clustered network. The network is divided in some of the logical clusters established based on the spatial data similarity by assigning roles: *cluster head* (leader of the cluster, abbreviated CH), member cluster (nodes merging in a cluster), and gateway (node belonging to a cluster but connects two CHs). The CHs forward the data aggregation from the member cluster to a base station called as the sink through a multi hop backbone of the CHs.

3.2 Firefly-inspired Synchronicity Mechanism

Many nature phenomena around us have inspirited the researchers to find the solution of the WSNs characteristics such as synchronicity [15], time synchronization [16], self-configuration [17], and self-localization [18].

One of the natural systems that are classified as the bio-inspired system is a firefly-based approach for synchronicity or time synchronization in WSNs. Mirollo and Strogatz [19] are the first one that introduced the fireflies firing as a firefly-inspired model for the synchronization through Pulse Coupled Oscillators (PCO). This model is called the M&S model. However, this model does not satisfy the realistically practical WSNs because the node responds immediately to the firing message sent by their neighbors without considering the unpredictable delay prior to. Accordingly, Reachback Firefly Algorithm (RFA) introduced by [15] was used to address the realistic wireless transmission problems through the PCO model. In our work, we developed a firefly-inspired synchronicity model that adopts the RFA model by dividing nodes in several clusters in order to reduce the convergence time. Our model synchronizes the nodes in the network in three stages: (1) cluster construction based on phase proximity; (2) intra-cluster synchronicity; and (3) inter-cluster synchronicity. The stages of our model are as follows:

3.2.1 Cluster Construction Based on Phase Proximity

The cluster construction divides the nodes in some logical clusters based on the initial phase of node. The following steps are used to construct the logical cluster.

1. Assign the number of clusters desired k, and the center point of clusters ($\emptyset c_1$, $\emptyset c_2$, …, $\emptyset c_j$, …, $\emptyset c_k$). Next, each node will internally find its own cluster through a nearest distance to the center points of the cluster that can be calculated by the following Euclidean equation:
$$d\emptyset_{ij} = \sqrt{\emptyset_i^2 - \emptyset c_j^2} \tag{1}$$
where $\emptyset_i$ is the i^{th} node's phase, $\emptyset c_j$ is the j^{th} node's phase, and the node's phase is $\emptyset_i \in [0\ 1]$. For illustration, let's assume that there are two clusters (k = 2) with the center points are $\emptyset C_1$ = 0.15 and $\emptyset C_2$ = 0.45 respectively. If a node has phase = 0.2, the node is grouped in the first cluster because its phase closer to $\emptyset C_1$ = 0.15 than to $\emptyset C_2$ = 0.45.
2. Each node stores all its neighbor node's addresses internally that exist in the same cluster.
3. To know the neighbor nodes in the same cluster, each node broadcasts its address and cluster number. The receiver nodes will store the node transmitter node's address if their cluster numbers are the same.

3.2.2 Intra-cluster Synchronicity

After all virtual clusters based on the initial phase are constructed, so that each virtual cluster can start to conduct the intra-cluster synchronicity concurrently. In our work, as the reachback response mechanism, we use the RFA approach to overcome the message delay in realistic transmission. For instance, let's suppose that in a virtual cluster, there is a node x broadcasting a firing message to other nodes in the same cluster where a node y receives it at time t_1 or phase $\emptyset(t_1)$. In response, the node y adjusts its internal oscillator's phase by smoothly jumping to a new phase of $\emptyset_{new}(t_1)$ as depicted in Figure 1. The new phase can be calculated using the following firing function:

$$\emptyset(t_1)_{new}) = \min(1, (1+\varepsilon)\emptyset(t_1)) \tag{2}$$

Thus, the phase jump value of the node y is:

$$\begin{aligned}\Delta\emptyset(t_1) &= \min(1, (\emptyset(t_1)_{new} - \emptyset(t_1))) \\ &= \min(1, \varepsilon\emptyset(t_1))\end{aligned} \tag{3}$$

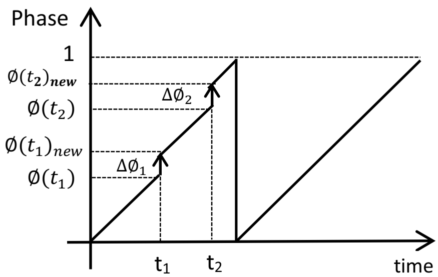

Figure 1. Updating Node y's phase in the M&S model

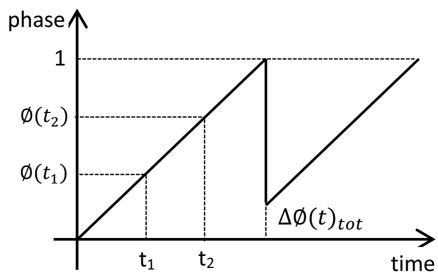

Figure 2. Updating Node y's phase in the RFA model

In reachback response mechanism, the node y responds to all firing messages by storing each phase jump until the end of period. Next, the sum of phase jumps in the previous period is used to update the new phase in the beginning of the next period. This mechanism is illustrated in Figure 1 and 2. Node y receives two firing messages at time t_1 and t_2 respectively. It updates its new phase at time t_3 in the beginning of the next period. The new phase updating can be obtained through the sum of phase jumps as follows:

$$\Delta\emptyset(t)_{tot} = \sum_{k=1}^{n} \Delta\emptyset(t)_k \tag{4}$$

where n is the number of firing messages received by a node in a previous period.

3.2.3 Inter-cluster Synchronicity

To achieve the converging synchronicity for all nodes in the network each cluster is considered as a converging group of the node. Therefore, it can also be assumed as a candidate of the firing node if its phase gets near to one. In this case, the inter cluster synchronicity also uses the reachback response mechanism to reach the converging synchronicity of the network. The converging node group has a phase of 1, and will transmit a firing message to other node groups. When there are two or more node groups that have the same phase, and will merge. Finally, there is only one group that considered as a converging synchronicity of the network.

The number of jumps in the G_y group receiving the firing messages from the group G_x in a period is equal in the number of the G_x's members. Thus, the sum of phase jumps of the group G_y is as follows:

$$\Delta\emptyset_{(tot),G_y} = \sum_{i=1}^{n_x} \Delta\emptyset_{i,G_y} \tag{5}$$

where n_x is the number of G_x's members. Because the sum of phase jump can be more than phase $\emptyset = 1$ caused by the numerous firing events, an overshooting jump will occur within the G_y group. To avoid this problem, a Late Sensitive Window (LSW) can be added in RFA mechanism. This concept is introduced by [20], proposing a window as a prevention against the response of a node in the firing messages falling outside of the window. When the group G_y reaches phase $\emptyset = 1$, it will adjust the updating phase based on the sum of phase jump in the previous period, and will jump at the beginning of next period. The updating phase of the group G_y can be obtained using the following equation:

$$\Delta\emptyset_{G_y} = \begin{cases} 0 & \emptyset \leq LSW \;\; and \; \varepsilon\emptyset \; \geq 1 \\ \varepsilon\emptyset & \emptyset \geq LSW \; and \; \varepsilon\emptyset \; \leq 1 \\ 1 & \quad Otherwise \end{cases} \tag{6}$$

where LSW ∈ [0, 1].

3.3 Spatial Data Similarity Based Dynamic Node Clustering

Generally, WSNs applications need spatially dense node deployment to satisfy coverage. Consequently, the data readings observed by the adjacent sensors tend to have high correlation. This case is called as the spatial correlation in which the correlation degree depends on the distance between the adjacent nodes. Moreover, the data readings have many redundancies that cause a large amount of traffic on the wireless channel and consume numerous energy of battery.

The spatial correlation along with a bio-inspired algorithm will bring significantly the potential advantages in the development of an efficient communications protocol that well suited to reduce the redundant data in WSNs. In our work, we use the fuzzy aggregation technique to measure the data similarity degree as follows [21]:

$$s(a,b) = \exp\frac{-\|a-b\|^2}{2 * \sigma^2} \tag{7}$$

where $s(a,b) \in [0,1]$, σ is a determiner of width of the Gaussian Kernel with σ = 1.74. The data similarity is measured based on its similarity degree. The data *a* is the same as the data *b*, when $s(a,b) = 1$. In contrary, both data are highly different when $s(a,b)$ is nearly zero.

To construct the dynamic node clustering based on the data similarity, we develop a new model that is a modification of our previous model in [22]. The modified important part is usage a clustered network schema to synchronize the nodes to increase convergence rate. Besides, the Absolute threshold-based similarity function is changed by the data similarity degree using the fuzzy aggregation technique as written in equation (7). This technique is a popular and powerful kernel used in the different problems of classification, data compatibility, data integration, and data completion [21].

The overview of the modified dynamic node clustering protocol is described as follows:

1. There are two data structures in each node: (i) *spatialNeighbor* is a simple data structure used to store the information about its similar neighbor candidate; and (ii) *similarNeighbor* is a vector data structure used to store the information

about its similarity neighbors. The spatial neighbors are its adjacent nodes, and the similar neighbors are nodes satisfying the data similarity degree threshold. There are three local variables in each data structure, address (*addr*), current data readings (*cR*), and the number of the similar neighbors (*nNeigh*).

2. There are two algorithms that have to exist in each node, i.e. Algorithm 1 and 2. In the Algorithm 1, the **Broadcasting** function (line 1-3) is used by a transmitter node to broadcast periodically a beacon message. This function consists of three sub functions. (i) The Send sub function (line 1) transmits the information about its identifier address (*addr*), its current data reading (*cR*), and the number of its similar neighbors (*nNeigh*). (ii) The Delay sub function intended to avoid simultaneous transmissions. (iii) The TimeExpired sub function is the limited time to broadcast.

Algorithm 1 Beacon Message

```
Broadcasting()
1. Send(addr, cR, nNeigh)
2. Delay(interval+rand())
3. TimeExpired()
```

Algorithm 2 Receiving data for node clustering

```
Receiving(addr, cR, nNeigh)
1.  spatialNeighbor  simNeighCand
2.  simNeighCand ← CreateSimNeighborCandidate(addr, cR, nNeigh)
3.  sigma ← 1.74
4.  a ← cR
5.  b ← DataReading()
6.  s ←  exp (- || a – b||^2)/(2 * sigma^2))
7.  if (s >= sThresh)
8.     if(!CheckExistSimilarNeigh(simNeighCand)
9.        AddSimilarNeighbor(simNeighCand)
10.       IncrNumSimilarNeighbor()
11.    else if (CheckExistSimilarNeigh(simNeighCand)
12.       DelSimilarNeighborbor(simNeighCand)
13.       DecrNumSimilarNeighbor()
14.     end if
15. end if
```

3. The **Receiving** function in Algorithm 2 is used by node to receive a beacon message. This message is to identify whether a transmitter node includes as a similar neighbor node. The transmitter node's information containing of *addr*, *cR*, and *nNeigh* is used to create a similar neighbor candidate (line 1-2). The data similarity degree is measured by the fuzzy aggregation equation (line 5). If the current data readings (*a*) of the transmitter node and the current data readings (*b*) of the receiver node satisfy the similarity degree threshold (*sThresh*) and also the similar neighbor candidate has not existed within the data structure, it is added in the data structure as one of the members of the similar neighbor (line 9), and the number of similar neighbor (*nNeigh*) is incremented (line 10). In contrast, if the similar neighbor candidate has existed within the data structure, it is deleted as the similar neighbor in the data structure (line 12), and the number of similar neighbor (*nNeigh*) is decremented (line 13).

4. Simulation Results

The performance of our proposed dynamic node clustering model is implemented in a network simulator (NS-3 version 3.25) to evaluate the performance. Furthermore, the design of the network topology represents a scenario of the realistic environment sensing application. The network consists of 54 sensors deployed in a small rectangular area sized 640mx480m. Each sensor node reads data simultaneously at every 30 seconds. The used realistic data is the humidity readings gathered by the Intel Berkeley Research Lab [23].

In our work, we evaluated two stages, i.e. the synchronicity and the dynamic node clustering. Figure 3 shows the performance of the firefly-inspired synchronicity. It is evaluated some variation of number of clusters and variation of coupling strengths in the range of 0.01 through 0.2. The evaluation shows that the number of cluster equal to two (# clusters = 2) will generate the best performance of the synchronicity mechanism for all scenarios of the variation of the coupling strengths.

In the evaluation of dynamic node clustering, the parameter settings used to assess the constructed clusters based on the data similarity readings are a variety of similarity degree thresholds (*sThresh*) in the range of 0.1 through 0.9. Moreover, the evaluated metrics is the number of clusters, number of nodes per cluster, and number of lone nodes without cluster.

Figure 4 presents the number of the established clusters tends to form a normal distribution in which the number of clusters are fewer in the low and high *sThresh* value than in the middle *sThresh* value. The number of clusters will be fewer in the lower *sThresh* value because more distant readings will be considered similar. On the contrary, the number of clusters will increase in higher *sThresh* because closer readings will be not assumed similar. Figure 5 shows an opposite graph pattern of Figure 4 in which the number of nodes per cluster decreases in the meddles part of similarity degree threshold.

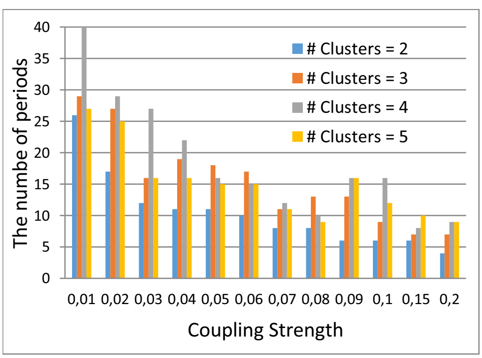

Figure 3. Comparison the converging synchronicity rate among the number of clusters

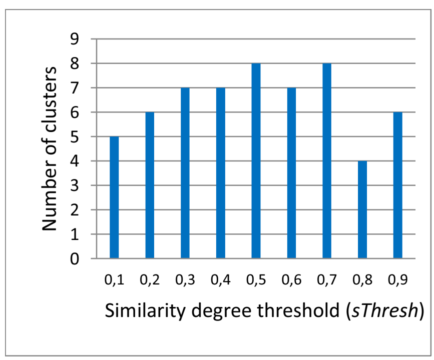

Figure 4. Relation between similarity degree threshold and number of clusters

Figure 6 presents that there is not the lone node in *sThresh* = 0.1 and 0.2 as both of them are the lowest similarity degree in which all data will be considered similar, but in higher *sThresh* emerges a few lone nodes except in *sThresh* = 0.8. The emergence of the lone nodes because there are anomalous data. Thus, node clustering can detect robustly the abnormal data that is read by the sensors.

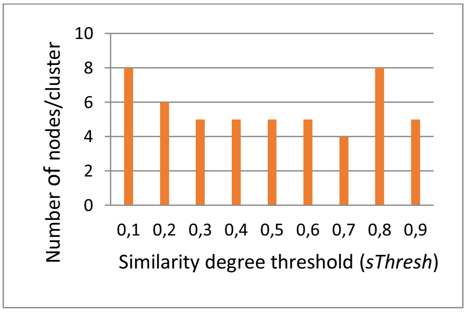

Figure 5. Relation between similarity degree threshold and number of nodes/cluster

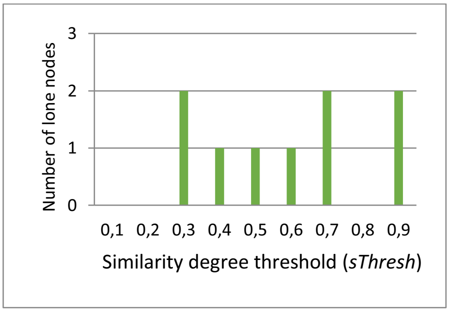

Figure 6. Relation between similarity degree threshold and number of lone nodes

5. Conclusions

The firefly-inspired synchronicity can synchronize the nodes on the network in order to perform together actions simultaneously. This mechanism can be conducted successfully by the self-organized WSNs. The synchronicity is executed through a clustering approach in order to increase the converging synchronicity rate. In our work, we find that the two clusters show the best performance all scenarios of the variation of the coupling strengths.

The dynamic node clustering technique can group the nodes based on the spatial data similarity readings where the number of established clusters tend to form a normal distribution. The number of clusters increases in the middle similarity degree threshold. Instead, the number of nodes per cluster decreases in the middle similarity degree threshold. The technique can detect robustly the anomalous data that is read by a sensor.

As future work, we will increase the capacity of self-organized network utilizing the bio-inspired techniques for selecting the cluster head and developing a routing mechanism in order to forward the data aggregation to the base station.

References

[1] L. Tuan Dinh and T. Dat Ho, "Design and deploy a wireless sensor network for precision agriculture," in *Information and Computer Science (NICS), 2015 2nd National Foundation for Science and Technology Development Conference on*, 2015, pp. 294–299.

[2] H. Adachi, H. Suzuki, K. Asahi, Y. Matsumoto, and A. Watanabe, "Estimation of bus traveling section using wireless sensor network," in *Mobile Computing and Ubiquitous Networking (ICMU), 2015 Eighth International Conference on*, 2015, pp. 120–125.

[3] R. Gupta, K. Sultania, P. Singh, and A. Gupta, "Security for wireless sensor networks in military operations," in *Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference on*, 2013, pp. 1–6.

[4] M. Kollam and S. R. B. S. Shree, "Zigbee Wireless Sensor Network for better Interactive Industrial Automation," in *Advanced Computing (ICoAC), 2011 Third International Conference on*, 2011, pp. 304–308.

[5] L. V Nguyen, S. Kodagoda, R. Ranasinghe, and G. Dissanayake, "Mobile robotic wireless sensor

networks for efficient spatial prediction," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, 2014, pp. 1176–1181.

[6] M. Arunraja, V. Malathi, and E. Sakthivel, "Distributed Similarity based Clustering and Compressed Forwarding for wireless sensor networks.," *ISA Trans.*, vol. 59, pp. 180–92, Nov. 2015.

[7] M. Arunraja, V. Malathi, and E. Sakthivel, "Distributed Similarity based Clustering and Compressed Forwarding for wireless sensor networks.," *ISA Trans.*, vol. 59, pp. 180–192, 2015.

[8] F. Gielow, G. Jakllari, M. Nogueira, and A. Santos, "Data similarity aware dynamic node clustering in wireless sensor networks," *Ad Hoc Networks*, vol. 24, Part A, no. 0, pp. 29–45, 2015.

[9] J. Tsiligaridis and C. Flores, "Reducing Energy Consumption for Distributed EM-based Clustering in Wireless Sensor Networks," *Procedia Comput. Sci.*, vol. 83, pp. 313–320, 2016.

[10] L. Kong, Q. Xiang, X. Liu, X.-Y. Liu, X. Gao, G. Chen, and M.-Y. Wu, "ICP: Instantaneous clustering protocol for wireless sensor networks," *Comput. Networks*, vol. 101, pp. 144–157, Jun. 2016.

[11] R. Rajagopalan and P. K. Varshney, "Data-aggregation techniques in sensor networks: A survey," *IEEE Commun. Surv. Tutorials*, vol. 8, no. 4, pp. 48–63, 2006.

[12] M. M. Afsar and M.-H. Tayarani-N, "Clustering in sensor networks: A literature survey," *J. Netw. Comput. Appl.*, vol. 46, no. 0, pp. 198–226, 2014.

[13] B. S. Mathapati, S. R. Patil, and V. D. Mytri, "Energy Efficient Reliable Data Aggregation Technique for Wireless Sensor Networks," in *Computing Sciences (ICCS), 2012 International Conference on*, 2012, pp. 153–158.

[14] J. Luo, L. Xiang, and C. Rosenberg, "Does Compressed Sensing Improve the Throughput of Wireless Sensor Networks?," in *Communications (ICC), 2010 IEEE International Conference on*, 2010, pp. 1–6.

[15] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," *Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM, San Diego, California, USA, pp. 142–153, 2005.

[16] N. Lipa, E. Mannes, A. Santos, and M. Nogueira, "Firefly-inspired and robust time synchronization for cognitive radio ad hoc networks," *Comput. Commun.*, vol. 66, pp. 36–44, Jul. 2015.

[17] H. Araujo, W. L. T. de Castro, and R. Holanda Filho, "A proposal of self-configuration in Wireless Sensor Network for recovery of broken paths," in *Sensors Applications Symposium (SAS), 2010 IEEE*, 2010, pp. 245–250.

[18] K. Kyungmi, K. Hyunsook, and H. Youngchoi, "A Self Localization Scheme for Mobile Wireless Sensor Networks," in *Computer Sciences and Convergence Information Technology, 2009. ICCIT '09. Fourth International Conference on*, 2009, pp. 774–778.

[19] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillator," *SIAM J. Appl. Math.*, vol. 50, no. 6, pp. 1645–1662, 1990.

[20] C. Lin and W. HongPeng, "Reachback Firefly Synchronicity with Late Sensitivity Window in Wireless Sensor Networks," in *Hybrid Intelligent Systems, 2009. HIS '09. Ninth International Conference on*, 2009, vol. 1, pp. 451–456.

[21] A. Ghaddar, T. Razafindralambo, I. Simplot-Ryl, S. Tawbi, and A. Hijazi, "Algorithm for data similarity measurements to reduce data redundancy in wireless sensor networks," in *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*, 2010, pp. 1–6.

[22] Misbahuddin and R. F. Sari, "Firefly-Inspired Algorithm for Data Similarity Aware Node Clustering in Low Rate Wireless Personal Area Network," *Adv. Sci. Lett.*, vol. 21, no. 11, pp. 3463–3467, 2015.

[23] Berkeley, "<http://db.csail.mit.edu/labdata/labdata.htm>," 2012. .