Online Learning with Experts Advice in Distributed Wireless Networks

Koutsaftis Athanasios, Hamza Anwar, Muhammad Affan Javed, and Sourjya Dutta
{apk377, maj407, ha1082, sd2603}@nyu.edu

***Abstract*—This works studies expert advice framework in a distributed wireless network with nodes working in clusters. Specifically, we investigate how the choice of a cluster coordinator is influenced by the use of experts advice. We compare the performance of the static, fixed share and variable share experts algorithms in choosing cluster coordinator nodes. We compare the total energy spent by the nodes in each of the clusters and also study the variance in energy consumption within clusters. It is found that from the perspective of energy consumption, the static expert algorithm does better than fixed and variable share algorithms.**

***Index Terms*—Distributed wireless networks, experts advice, network cluster, energy consumption.**

I. Introduction

DISTRIBUTED wireless networks is a niche area of communications and networks that has seen tremendous interest within the scientific community in recent years, largely due to the advancements in electronics miniaturization and wireless communications technology. This has made developing and deploying a large number of wireless nodes distributed over a geographical area economically feasible and technologically viable. Applications of such systems encompass a wide array of platforms ranging from wireless sensor networks used to monitor volcanic and seismic activity and natural parks to military applications for security and surveillance.

In distributed wireless networks, it is common to have sensors or devices grouped according to their functionality or geographical location. We can view these nodes to be, in a sense, clustered. In this setting we focus on the scenario where each cluster has a coordinator node or a "leader" which communicates with the central access point (AP) or base station (BS). Every other node transmits data to the coordinator. Data aggregation at the local level is important because it has been shown that in energy constrained distributed networks it is highly inefficient for each node to transmit data directly to the AP [1]. Many different clustering algorithms have been explored in the context of distributed wireless networks [2], [3], [4], [5].

The role of the cluster coordinator is of utmost importance in these distributed wireless networks. A cluster coordinator performs task such as node association, authentication, and task assignment [4]. As the cluster coordinators aggregate data locally and send it on to the AP, it is critical that the choice of the coordinators is such that the network performance is not adversely affected. A key goal of these data aggregation algorithms is to gather data in an energy efficient manner so that network lifetime is enhanced [6]. Moreover, there is also the possibility of certain nodes being compromised so it is also important from a security perspective to ascertain that the nodes chosen as the coordinators are trustworthy [7]. Appropriate choices of cluster coordinators has been shown to reduce the communication overhead for both single-hop and multi-hop networks [5].

Machine learning, in general, has been extensively used in distributed wireless networks to implement more efficient ways of dealing with classical problems such as routing, security, clustering and data aggregation [8]. In this work we use an on-line learning setting, *experts advice*, to predict the cluster coordinators. In this setting, as discussed in [9], the prediction algorithm has access to a number of experts. At each iteration, based on the confidence on or weight of each of the experts the algorithm makes a prediction. Losses are computed based on the true nature and the prediction. The weights on each expert is updated based on the losses incurred. Following works [10], [11] have proposed efficient algorithms for scenarios when the number of experts is exponentially large. Moreover, a low complexity and optimal algorithm to track the best expert was presented in [12].

A variation of the weight sharing algorithms presented in [9] is postulated in [13] where the weights are shared based on past posteriors.On the other hand the authors in [14] propose a method, Learn-α, to learn the switching dynamics of the expert algorithm. Moreover, the authors also show an example of the use of this algorithm in predicting the polling time of a network node. In later scientific literatures, many works have applied the experts advice algorithm to communication and wireless networks.

The work [15] look into the channel allocation problem in wireless networks using sleeping experts. On the other hand the authors in [16] predict the future state of the network using a variant of of the fixed share expert setting developed in [9]. Thus, the unpredictability in communication systems, due to the physics of the transmission channel and the randomness of users, have the potential of using on-line learning algorithms to enhance performance. With this in mind we look into the problem of coordinator selection in wireless sensor networks using the expert advice setting.

The rest of this report is organized as follows. In Section II we describe the our model and extend it to the experts advice setting. In Section III we present our simulation design and discuss the results we obtain. In Section IV concludes the report.

II. Selecting Best Coordinator

A. Problem Setting

We consider a single cell with one access point (AP) at the center and N nodes $m_1, m_2, \ldots, m_N$ randomly distributed over a circular region (cell). A *node* denotes a wireless sensor or device which is a part of the overall distributed wireless network. We assume that all the nodes are already assigned to a particular cluster, i.e., they belong to a geographically or functionally similar group of nodes denoted by $\mathcal{C}_i$ where $i \in \{1, 2 \ldots C\}$ and C represents the total number of clusters in the system. In our model C is a known constant and the number of nodes in the i^{th} cluster are denoted by k_i.

All the nodes seek to periodically send data to the AP. However, instead of each node individually sending data to the base station, we aggregate the data locally in each cluster at one node which is defined as the cluster coordinator (CC). The CC then forwards the aggregate data, along with its transmission, to the AP.

Furthermore, we define a time period T which is known as the frame length or an epoch. At the beginning of each frame each node will assess its link with the AP and send a channel quality indicator (CQI) to the AP. A fixed period of time t_{c0} is allocated for this. Then a specified amount of time t_{c1} is allocated to, first,select one coordinator nodes for each cluster and, second, to send the identity of this node to all other nodes in the cluster. The total period at the beginning of the frame $t_c = t_{c0} + t_{c1}$ can be viewed as a control period and no data transmission takes place in this period. In our setting the AP determines the coordinator node for each cluster using the *expert advice* framework.

An AP has access to L experts. Each expert is an algorithm which assigns a probability distribution over all the nodes in each cluster indicating how likely a particular node is to be a coordinator. On receiving the advice from all the experts, the AP will decide a coordinator for each cluster for the current frame. The algorithms used by the experts are a design choice. The BS treats the experts as black boxes. We experiment with static, fixed-share and variable-share expert frameworks[9] in order to compare performance and gauge which one is the best for our case.

B. Experts Algorithm

We describe the expert algorithm used in this section.

- **Parameters:** Learning rate $\eta > 0$; For share algorithms fix $0 \leq \alpha \leq 1$. Number of clusters $C > 0$. Number of nodes in the i-th cluster k_i.
- **Initialize:** Weights on each expert

$$w^s_{1,1} = \ldots = w^s_{1,l} = \ldots = w^s_{1,L} = 1/L$$

- **Prediction:** For every epoch/frame,
  - All nodes send CQIs to the AP. The CQI of the j-th node of the i-th cluster for a given epoch is given as $q^i_{j,t}$.
  - The AP also receives C distributions from every L experts. The prediction of the l-th expert for the i-th cluster is given as,

$$p^i_{l,t}; \quad \text{where } \sum_{j=1}^{k_i} p^i_{l,t}(j) = 1. \tag{1}$$

  - For every cluster $i \in 1, \ldots, C$, the coordinator node is chosen as,

$$y^i_t = \arg\max_j \; q^i_{j,t} \left(\sum_{l=1}^{L} w_{t,l} \cdot p^i_{l,t}(j) \right). \tag{2}$$

- **Update Loss:** After selecting the coordinator node compute loss on every expert and update weights as,

$$\forall l = 1, \ldots L; w^m_{t,l} = w^s_{t,l} \exp\left(-\eta \mathcal{L}_{l,t}\right). \tag{3}$$

where

$$\mathcal{L}_{l,t} = \frac{1}{L} \sum_{i=1}^{C} \mathcal{L}(p^i_{l,t}, y^i_t).$$

- **Share Updates:** Updates are shared based on one of the three algorithms in [9] namely,
  - Static expert: $\forall l = 1, \ldots L; w^s_{t+1,l} = w^m_{t,l}$.
  - Fixed Share: $\forall l = 1, \ldots L;$

$$\begin{aligned} w^s_{t+1,l} =& (1-\alpha) w^m_{t,l} \\ &+ \frac{1}{L-1} \left(\sum_{u=1}^{L} (\alpha w^m_{t,u}) - \alpha w^m_{t,l} \right) \end{aligned}$$

  .
  - Variable Share:$\forall l = 1, \ldots L;$

$$\begin{aligned} \text{pool} &= \sum_{l=1}^{L} \left(1 - (1-\alpha)^{\mathcal{L}_{l,t}}\right) w^m_{t,l} \\ w^s_{t+1,l} =& (1-\alpha)^{\mathcal{L}(p^i_{l,t}, y^i_t)} w^m_{t,l} \\ &+ \frac{1}{L-1} \left(\text{pool} - \left(1 - (1-\alpha)^{\mathcal{L}_{l,t}}\right) w^m_{t,l} \right). \end{aligned}$$

Following the updates, normalize the weights as

$$w^s_{t+1,l} = \frac{w^s_{t+1,l}}{\left(\sum_{l=1}^{L} (w^s_{t+1,l})\right)}.$$

C. Loss Function

We define the loss function at epoch t for expert l on the prediction of cluster i as,

$$\mathcal{L}(p^i_{l,t}, y^i_t) = \sum_{j=1}^{k_i} p^i_{l,t}(j) \|x^i(j) - y^i_t\|_2^2, \tag{4}$$

where $x^i(j)$ is the coordinate of the j-th node in the i-th cluster and y^i_t is the coordinate of the chosen coordinator node. In fact the loss function measured the expected error in distance predicted by each of the experts. This loss is averaged over all the clusters for each expert at each iteration.

III. Simulations and Results

This section briefly describes the simulation performed and the results. In our simulation we consider the scenario where on access point is communicating with C node clusters with k_i nodes in each. The clusters are placed over a 2D plane randomly around the access point at the origin. One example is shown in Fig. 1 with $C = 8$ and $k_i = 20, \forall i = 1, \dots 8$. We use this setting for obtaining our results. We assume that the nodes have been pre-clustered. We do not try to provide any insight into the clustering. All the nodes communicate using a 2.4 GHz carrier over a 22 MHz channel (like 802.11b wireless LAN). The simulations are performed on MATLAB. The code is available on-line at [17].

In our setting, at the beginning of each epoch, each node within a cluster will compute the channel loss to the AP. The channel loss is a sum of the Friss path loss which scales a function of distance and a random channel loss based on the channel state. The channels is described as having 3 independent states $S \in \{0, 1, 2\}$, where a channel in state 0 has no additional loss, a channel in state 1 has an added 20 dB loss and a channel in state 2 has an added loss of 100 dB. At each epoch the channels are assigned randomly to the nodes.

On receiving the channel information from the nodes at the start of the epoch, the AP will choose a coordinator for each of the C clusters. In order to do this the AP has access to 3 experts. Each of the experts will provide a probability distribution over the nodes in each of the clusters. The AP combines these results with the channel state as described in Section II. In our simulation we consider the experts to have access to the noisy estimate of the position of the nodes and the location of the AP. "Expert 1" creates a distribution based on the distance of the node from the AP, "Expert 2" uses the average distance of one node to every other node in the cluster to produce the distribution whereas, "Expert 3" assigns an uniform distribution every time.

The channels loss between the nodes in a cluster and the coordinator is shown in the right pane of Fig. 2. Each state corresponds to the same loss. Each node transmits 100 Bytes of data to the coordinator. The coordinator aggregates the data and transmits it to the AP. At the end of each epoch the energy consumed by each node is computed.

We compare the performance of the *static expert*, *fixed share expert*, and the *variable share expert* algorithms using our simulation. We use a learning rate $\eta = 0.1$ and a share weight $\alpha = 0.1$. In Fig. 3 we show the weights assigned to the experts in each epoch for the three setting. We see that for the static expert setting, expert 1 is the best expert and the weight assigned goes to 1 while the weights on the other experts fall to 0. For the fixed-share expert, we see some degree of weight sharing, although expert 1 still has a higher weight compared to experts 2 and 3 but none of the weights go to 0. For the variable share case we find a rapid fluctuation of weights. We note that the exact behavior may vary based on the parameters chosen but it is observed that the trend remains the same.

In Fig. 4 we plot the mean of the total energy consumed by each node in each cluster along with the maximum energy

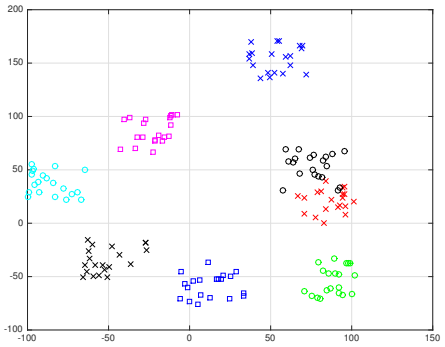

Fig. 1. Position of the node clusters in a 2D grid with the AP at $(0, 0)$.

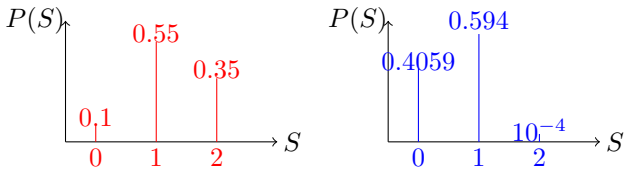

Fig. 2. Probability distribution for the 3 channels states *Left:* for channel between nodes and AP; *Right:* for channel between nodes and coordinator nodes.

consumed by any node in a give cluster. Moreover, we plot the average of the total energy with the standard deviation. What we observe is the *static* expert gives a more uniformly distributed energy consumption. This is evident as the difference between the maximum and the average is low for 7 out of 8 clusters whereas this difference is low for 5 clusters in the fixed share setting and for 6 clusters in the variable share case. The same observation is made by the plot of the standard deviation.

The result is in fact intuitive based on the nature of the channel. The channel losses are dominated by the Friss path loss. The nodes spend energy in overcoming this loss. This is the reason why the static expert performs better in terms of energy by sticking to the expert which assigns a greater probability to the node nearest to the AP. Moreover, static expert setting is using the best *linear* combination of experts, choosing the optimal policy for selecting the coordinators, rather than selecting a single(discrete) method. This agrees with the results provided in [14], in which they also observed the superiority of the static expert setting.

In a second experiment we change the way the experts make predictions. Rather than looking at the network topology, experts 1 and 2 now output a randomly generated distribution over the nodes where as expert 3 uses the same uniform distribution over all nodes. We plot the weights on the experts versus the epoch in Fig. 5. We plot the energy consumption results in Fig. 6. Like in the previous setting we notice that the static expert does a better job at minimizing the energy consumption. We have smaller variances in energy consumed on average per cluster compared to the shared algorithms.

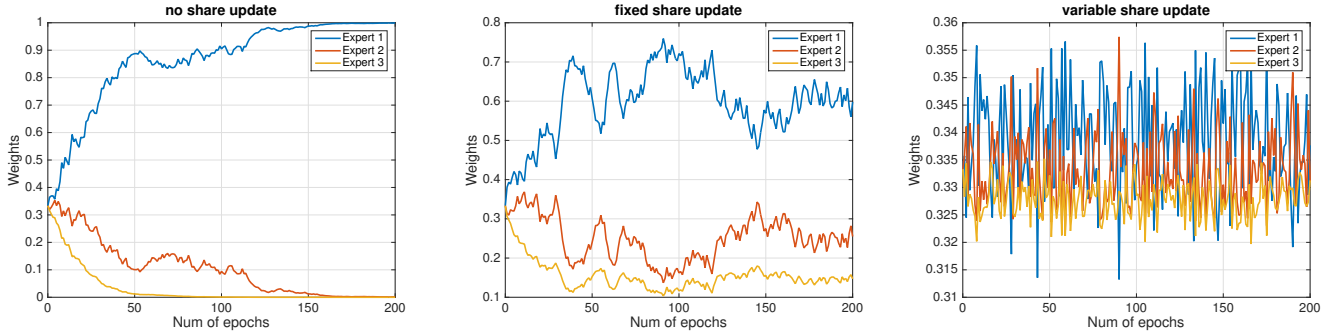

Fig. 3. Weights on the experts for *Left:* No share update i.e., the static expert setting *Middle:* Fixed α share with $\alpha = 0.1$ *Right:* Variable share with $\alpha = 0.1$. The learning rate used is $\eta = 0.1$.

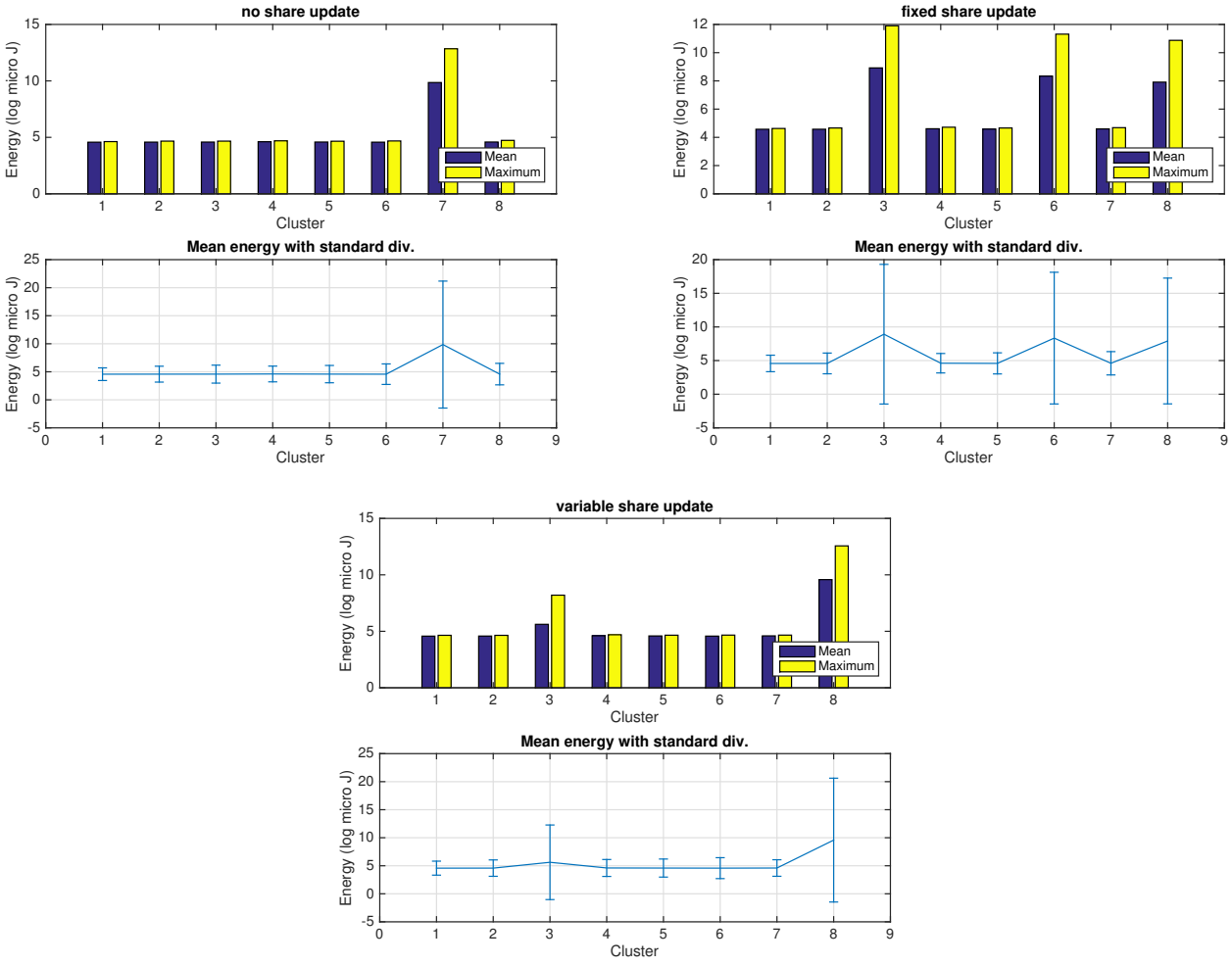

Fig. 4. Energy consumption: The figures compared the average and the maximum energy consumed per node for each cluster. The *top left* figure is for the static expert setting, *top right* is for fixed share setting with $\alpha = 0.1$, and the *bottom center* figure is for the variable share setting with $\alpha = 0.1$. Also shown along side is the standard deviation of the consumed power as error bars. The y-axis is in logarithmic in micro-Joules. Learning rate $\eta = 0.1$.

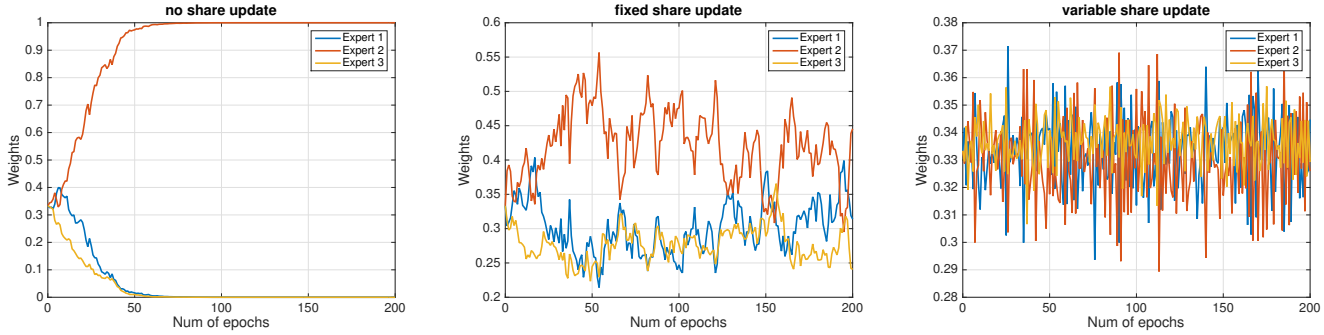

Fig. 5. Weights on the experts for *Left:* No share update i.e., the static expert setting *Middle:* Fixed α share with $\alpha = 0.4$ *Right:* Variable share with $\alpha = 0.1$. The learning rate used is $\eta = 0.2$.

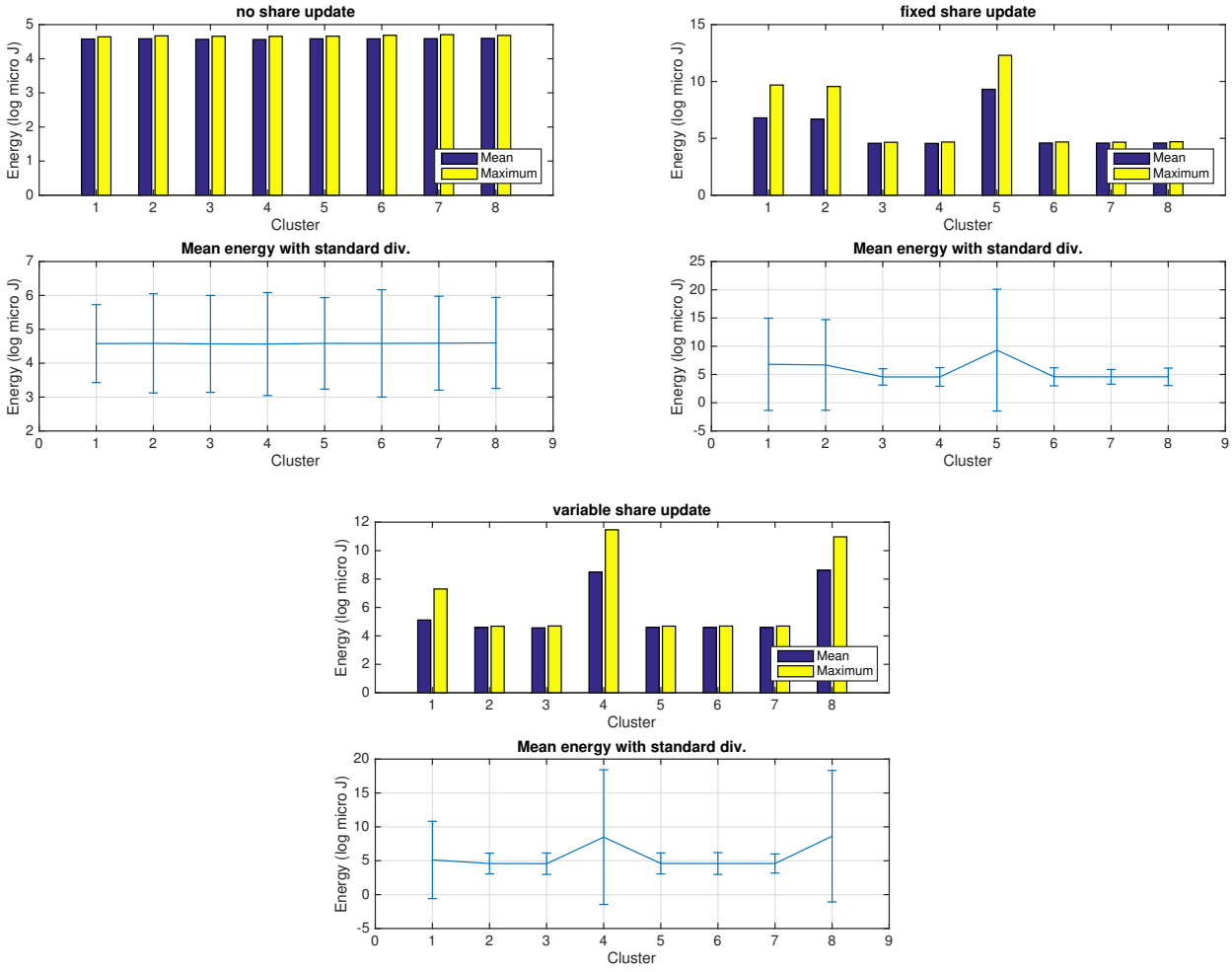

Fig. 6. Energy consumption: The figures compared the average and the maximum energy consumed per node for each cluster. The *top left* figure is for the static expert setting, *top right* is for fixed share setting with $\alpha = 0.1$, and the *bottom center* figure is for the variable share setting with $\alpha = 0.4$. Also shown along side is the standard deviation of the consumed power as error bars. The y-axis is in logarithmic in micro-Joules. Learning rate $\eta = 0.2$.

IV. Conclusion

In this report we study the expert advice setting and apply it to the problem of coordinator selection in wireless networks. We study the static, fixed share and the variable share algorithm in the wireless setting through the lens of energy consumption. Through our experiments we find that the static expert algorithm outperforms the fixed and the variable share experts algorithm.

V. Acknowledgment

This report was written as partial requirement for the course project of EL-GY 9133: Advanced Machine Learning at NYU Tandon School of Engineering.

References

[1] R. Rajagopalan,and P. K. Varshney, "Data aggregation techniques in sensor networks: A survey," available on-line at http://surface.syr.edu/, 2006.

[2] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks, *Computer Communications*, vol. 30, no. 14, pp. 28262841, Oct. 2007.

[3] O. Younis, K. Marwan, and S. Ramasubramanian, "Node clustering in wireless sensor networks: Recent developments and deployment challenges," *IEEE Network*, vol. 20, no. 3 pp. 20–25, Jun. 2006.

[4] D. Wei, Dali, *et. al.*, "An energy-efficient clustering solution for wireless sensor networks," *IEEE Trans. on Wireless Commun.*, vol. 10, no. 11 pp. 3973–3983, Nov. 2011.

[5] O. Younis, and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4 pp. 366 – 379, Oct. 2004.

[6] S. Soro and W. Heinzelman, "Prolonging the lifetime of wireless sensor networks via unequal clustering, *Proc. 19th IEEE Intl Parallel and Distributed Processing Symposium*,pp. 48, Apr. 2005.

[7] G. Crosby, N. Pissinou, and J. Gadze, "A framework for trust-based cluster head election in wireless sensor networks, *Proc. 2nd IEEE Workshop on Dependability and Security in Sensor Networks and Systems*, pp. 10-22, Apr. 2006.

[8] M.A. Alsheikh, *et. al.*, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *Commun. Surveys Tuts.* vol. 16, no. 4, pp. 1996–2018, Apr. 2014.

[9] M. Herbster, and M. Warmuth, "Tracking the best expert," *Machine Learning*, vol. 32, no. 2, pp. 151–178, Aug. 1998.

[10] A. Gyrgy, L. Tams, and G. Lugosi, "Tracking the best of many experts," *Proc. Int. Conf. on Computational Learning Theory*, pp. 204–216, Jun. 2005.

[11] A. Gyorgy, L. Tams, and G. Lugosi, "Tracking the best quantizer," *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 1604-1625, Mar. 2008.

[12] A. Gyorgy, L. Tams, and G. Lugosi, "Efficient tracking of large classes of experts," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6709– 6725, Jul. 2012 .

[13] O. Bousquet, and M. K. Warmuth, "Tracking a small set of experts by mixing past posteriors," *J. Mach. Learning Research*, vol. 3, pp. 363–396, Nov 2002.

[14] C. Monteleoni,and T. S. Jaakkola, "Online learning of non-stationary sequences," *Proc. NIPS*, pp. 1093–1100, May 2003.

[15] Dams, Johannes, Martin Hoefer, and Thomas Kesselheim, "Sleeping experts in wireless networks," *Proc. Intl. Symposium on Distributed Computing*, pp. 344–357, Oct. 2013.

[16] Y. Edalat, A. Jong-Suk, and K. Obraczka, "Network state estimation using smart experts," *Proc. 11th Intl. Conf. on Mobile and Ubiquitous Systems: Computing, Networking and Services,* pp. 11–19, Dec. 2014.

[17] https://github.com/Sourjya9015/Expert-Systems.git.