Crowdsourcing Location Sensitive Data for Dynamic Scenario by Adaptive Role Assignment

Anubhuti Garg and Amiya Nayak
School of Electrical Engineering and Computer Science, University of Ottawa, Ontario, Canada

Keywords: Participatory Sensing, Localization, Collaborative.

Abstract: The existing technique for performing crowdsourced, location-based sensing activity minimizes energy consumption by eliminating the use of GPS by some devices. For this, server detects a set of participants for the role of broadcaster which must turn-on their GPS to collect location information and broadcast it to neighbouring devices for their position calculation. However, if new devices join the region then they cannot participate in the ongoing sensing task until next localization phase when server reassigns role to all participants. In addition to this, if devices leave the region then their neighbouring devices may require a change of role. The current work does not provide solution to such dynamic scenarios. We provide time and energy efficient approach to allocate role adaptively to participants when they join or leave the region of interest. For this, we propose incremental algorithms to assign role for the new participants joining the region and for modifying the roles of existing participants when some devices leave the region. This also eliminates the need for rerunning the role-assignment algorithm over the entire set of participants for every insertion and deletion. The proposed solutions are capable of saving 95-99.9% of the role assignment time without compensating energy needs.

1 INTRODUCTION

Mobile phones have become an indispensable part of our lives and this has attracted researchers to harness its data sensing capabilities and extract valuable knowledge. Most smartphones are embedded with rich set of sensors such as accelerometer, GPS, gyroscope, microphone, camera and interfaces such as WiFi, bluetooth and other technologies (Lane et al., 2010). This has lead to number of exciting applications based on mobile phone sensing.

In this paper, our focus is on crowdsourcing data for participatory sensing system. In such a system, participants actively participate in sensing activity and collaborate to accomplish a given task (Macias et al., 2013). Participatory sensing supports various applications ranging from heath services to environment monitoring, most of which are dependent on location information. For accurate location information, devices depend on GPS which is a major source of power depletion in cell phones. Therefore, researchers focus on providing alternatives to GPS usage. In (Song et al., 2014), authors provide a device to device localization scheme to relieve some devices from using GPS and thereby saving their phone's energy. It takes into account mobility and relative positioning of devices. For accuracy, some devices need to turn on GPS and others depend on them for calculation of their location.

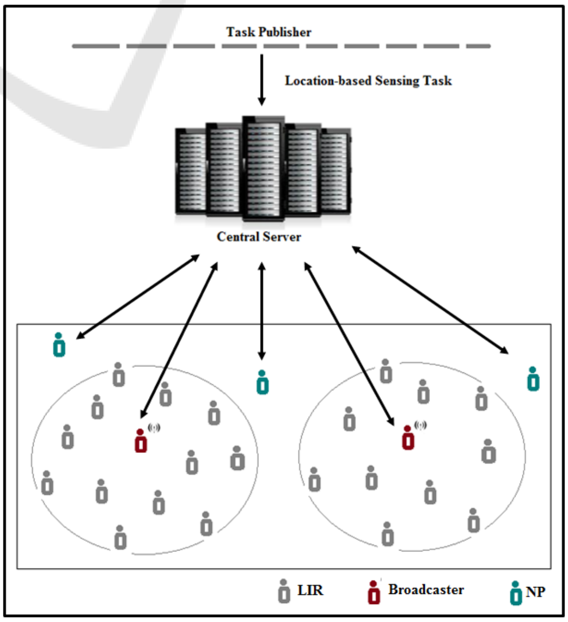

Figure 1: System Architure.

Garg, A. and Nayak, A.
Crowdsourcing Location Sensitive Data for Dynamic Scenario by Adaptive Role Assignment.
DOI: 10.5220/0006436000450054
In *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017) - Volume 6: WINSYS*, pages 45-54
ISBN: 978-989-758-261-5

Our research is motivated by the application framework shown in Fig. 1 for collaborative sensing. Similar framework was also used by authors in (Garg et al., 2017) and (Wang et al, 2016). The task publisher sends the sensing task to server which forwards the same to participating smartphones. The tasks are such that sensing requires location of the devices. These participants are assigned a role for some time period by the server after which roles are reassigned according to the updated location information. There are three categories of role that can be assigned -

- *Broadcaster*: Its function is to obtain location using GPS then broadcast this location and movement information to the surrounding participants;
- *Location Information Receivers (LIR)*: They rely on broadcasters to calculate their location using device-to-device localization method;
- *Normal Participants (NP)*: They do not receive any broadcast from their surrounding so, they depend on GPS to obtain location information.

These roles last for certain period of time then server assigns new roles according to their updated location.

The authors in (Garg et al., 2017) provide an efficient energy consumption model for participants in three roles and a sorting based method used by the server to assign role for each of the participating devices. However, there is an inherent assumption of fixed number of participants. In reality, the region can have frequent or infrequent updates because of the dynamic nature of smartphone users. For instance, consider an application that maintains gas prices at different locations in a locality. For this, participating smartphones upload prices for the gas and location whenever they make use of it. The gas stations on highways are generally used by tourists more. In such scenarios, the application must ensure that server is capable of assigning role to new participants and adapt to changes quickly and efficiently.

Updates are usually collected and applied periodically due to which these devices cannot participate in the ongoing sensing task and have to wait for next assignment round. However, we make it possible to assign role to such new devices before the task arrives without re-running greedy (Wang et al., 2016) or sorting based (Garg et al., 2017) algorithm. Due to large number of participating devices, it is desirable to assign role incrementally. This is the first study so far where we consider devices joining or leaving the region on fly. If a device joins the existing set of participants then the role is assigned based on its location, and in case device leaves the region then we reassign the role of its neighbouring devices only.

The rest of the paper is organized as follows. Section II presents related work. In Section III, we discuss various scenarios that might be possible when a device joins or leaves the region. We also present incremental insertion and incremental deletion algorithms to consider all cases. In Section IV, we present a model for the adaptive approach, taking into account both insertion and deletion algorithms. The experimental results are presented in Section V followed by conclusion and future work in Section VI. In the paper, we have used participants, devices, nodes and smartphones interchangeably.

2 RELATED WORK

Participatory sensing has shown its great potential in numerous application domains such as health care, environment monitoring, transportation, social networks, safety, industrial monitoring and maintenance, academia and government agencies. For example, AndWellness (Hicks et al., 2010) was developed as a personal behavioral and contextual data collection system to record and monitor participants' daily habits. The system could be even deployed to assess HIV+ patients through behavior and emotional distress. Similarly, Biketastic (Reddy et al., 2010) is another system that collects route experience of bikers such as terrain, noise level, scenery image. The data was collected periodically after every one second using GPS. It used accelerometer to sense noise level and roughness. The information of different routes was made available to cyclists in order to facilitate them for choosing right path.

Participatory sensing involves participation of all devices. First, the sensing task is initialized to determine the goal, what and how to sense. Then, the task is passed on to the users to collect data which can be location-based or logged or manual information. The data is then transferred to the server or cloud for processing (Goldman et al., 2009).

Energy is one of the most important issues that have been considered in the study of participatory sensing. In (Song et al., 2014), authors provide an energy-efficient participant selection algorithm based on constrained optimization problem and quality of information(QoI) which includes sensing region, time period, data granularity and quantity. They propose a behavioral model to find relationship between residual energy and willingness to participate so as to know before hand which participant will deny participating in the sensing activity. However, they consider only a subset of participants for performing required sensing task. PSense (Baier et al., 2012) is another application proposed to reduce energy consumption of mobile devices. They make use of the adaptive positioning mechanism and short range communication to exchange position related information. The server sends the set of locations that are needed to be sensed. Each mobile device then periodically fix its location to cover all queried regions. Jigsaw (Lu et al., 2010) provided an energy efficient sensing engine developed to continuously monitor human activities and contexts. But, the authors do not consider sensing tasks based on location of the devices.

Most sensing techniques which need position information rely on GPS. However, it drains sufficient power of the phones. Hence, many alternatives to GPS have been proposed compromising accuracy of the devices. In (Shafer et al., 2010), authors present an indoor WLAN-localization method using accelerometer. This is an energy-efficient technique but can work only in indoor environment. Some localization techniques are based on Bluetooth technology (Johnson et al., 2012). In (Kumar et al., 2013), a method for localization using location beacon is provided which requires fixed or mobile beacons to estimate position. (Song et al., 2014) proposed a device-to-device localization method which uses propagation model of wireless signals. The movements of devices are calculated by inertial sensors using step-up method and change in distance between devices are modelled by change in signal strength. This method was deployed in (Wang et al., 2016) for collaborative outdoor localization. A server selects set of devices which must turn on GPS while neighbouring devices rely on them for calculating location using device-to-device localization.

Authors in (Wang et al., 2016) provided a greedy algorithm which we refer to as GBS (Greedy based Broadcaster Set selection) for the role assignment. The basic idea was to select optimal set of broadcasters. The algorithm checks every participant in each iteration and assigns the role of broadcaster to the one which minimizes system energy the most. It keeps iterating until no participant can be chosen for broadcaster's role that can minimize system energy further. Devices which are not close to any other device are selected as normal participant, and rest of them are chosen for the role of location information receivers.

In (Garg et al., 2017), authors proposed a sorting based algorithm which they refer to as SBS (Sorting based Broadcaster Set selection) for the selection of broadcasters. In this, they introduced the following terms:

- *Connectivity*(κ) to represent number of devices within WiFi range of participant, and
- *Local Connectivity*(κ') to denote number of devices that participant would contribute to LIR set.

The algorithm first sorts participants on the basis of their connectivity. For every iteration, it sets an upper bound to minimize the search space for selecting the next broadcaster. Within this range, the node with highest local connectivity and that minimizes energy of system is chosen as next broadcaster. Location information receivers and normal participants are chosen in the same way as in the greedy approach.

The proposed solution for adaptive changes can be applied to both the algorithms. We have compared our algorithm with sorting based approach as its effectiveness over greedy has been proven in (Garg et al., 2017). However, results would remain same if we compare with greedy solution as well.

3 INCREMENTAL OPERATIONS

In this approach, the following two operations are considered:

- *Insertion* - new participant joins the region of interest,
- *Deletion* - existing participant leaves region of interest.

The role of new mobile device being inserted is dependent on its location. For instance, if the device is not within WiFi range of any other device then it act as a normal participant. Similarly, the change of role for the existing devices due to exit of participants is affected only within the WiFi reception range of the device being deleted.

In the following sections, we discuss various cases due to insertion and deletion of a participant and subsequently provide an algorithm for each case.

3.1 Insertion

When a participant p joins the region, new connections may be established, but none is removed. Following cases can occur:

- *(Noise)*
  If the new participant cannot become part of any broadcaster, i.e., it is not close enough to any broadcaster to receive its WiFi signal, then it becomes a normal participant. Fig. 2(case 1) depicts similar case. A node x labelled as p is being inserted to an existing set of devices. However, its WiFi range does not cover any node, hence assigned a role of a normal participant.
- *(Creation)*
  If the new participant's reception range covers few

Table 1: List of notations.

Notation	Explanation
M	Set of smartphones
$B_{t_1t_2}$	Set of broadcasters during $[t_1, t_2]$
b_m	Boolean to indicate if smartphone m is selected as broadcaster
br	Boolean to indicate if smartphone m is selected as LIR
$BLIR_{t_1t_2}$	Set of LIRs for each broadcaster for the interval $[t_1, t_2]$
κ	Connectivity of participant
κ'	Local Connectivity of a participant
$P_{t_1t_2}$	Physical connectivity matrix for interval $[t_1, t_2]$
P_{local}	Local physical connectivity matrix
I	Set of new participants being inserted
D	Set of participants to be deleted
e_b	Energy of broadcaster
e_n	Energy of normal participant
e_l	Energy of LIR
$E_{t_1t_2}$	Energy consumed during $[t_1, t_2]$
δ	Threshold for the change in database(in %)

normal nodes then it is eligible to become a broadcaster. It can then switch on its GPS, collect sensing information and update server through cellular network before task deadline. Fig. 2(case 2) illustrates this case. The green dots are used to depict normal nodes. When a new node p covers two normal nodes within its WiFi range, it is assigned a role of a broadcaster, and the covered nodes are reassigned the role of LIR.

- *(Absorption)*
  If the new participant is close enough to any of the existing broadcasters then it becomes part of it and act as a LIR node. Fig. 2(case 3) shows absorption of newly inserted node, p. It is close enough to send and receive signals from an existing broadcaster depicted by red colour. Hence, it gets absorbed and is assigned a role of LIR.

3.1.1 Incremental Insertion Algorithm

In this subsection we provide algorithm for incremental insertion to consider the cases discussed above. The for loop from Step 1 to 19 iterates through entire set of new participants joining the region. Each of the parameter, E_1, E_2, and E_3 denotes energy consumed if participant is selected as broadcaster (Step 3), LIR (Step 4) or NP (Step 5) respectively. E_1 is computed using the local connectivity of new participant that is, number of normal participants that it can cover (Step 3). E_2 is updated when the participant is within WiFi range of any broadcaster. The flag is then set to 1 and energy of LIR (e_l) is added to E_2 (Step 6 to 9). E_3 is obtained by simply adding energy of NP (e_n) in case node is selected as NP. The minimum among the three energies is used to assign role to the new node (Step 11 to 17). In case new node is assigned the role of broadcaster then, the normal participants that it covers change the role to LIR and boolean vector(br) is updated. If $br(i)$ is set to unity then it represents that i^{th} participant is assigned a role of LIR and if it is set to zero then participant can be broadcaster or normal participant.

3.2 Deletion

As opposed to insertion, when a participant p leaves the group, connections or role of existing devices might change. The trickiest case is when a broadcaster leaves the region. We discuss all possible cases below when deleting a node x, labelled as p:

- *(Removal)*
  If LIR exits such that its broadcaster still covers a set of nodes then it simply updates its broadcaster with sensing data collected so far and asks to remove it from its database. LIR is removed without affecting roles of any other device. Fig. 3(case 1.1) shows a similar scenario. Deleting node p does not affect role of any other node. The broadcaster continues to cover large set of devices (LIR).
  Similarly, if normal node decides to go out of the region of interest, then it gets deleted without affecting any other participant. When server does not listen any update then it is considered to be deleted, and the database is updated. Fig. 3(case 1.2), shows a similar scenario. The node p is initially assigned a role of normal participant (coloured green to depict its role). However, its

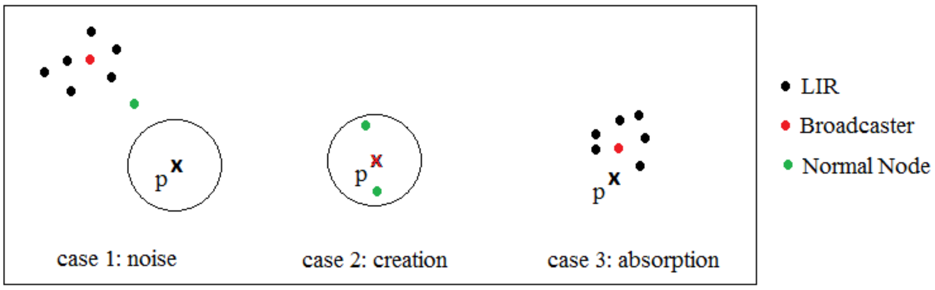

Figure 2: Different cases of the Insertion algorithm.

exit from the region does not affect role of any other device.

- *(Reduction)*
Once again, we consider the case when LIR leaves the region. We discussed previously how exit of LIR does not affect role of any other participant. The main idea for the role assignment is to minimize energy needs. If reassigning roles minimizes energy further, then it is a preferred choice. For instance, in Fig. 3(case 2), when node p gets deleted, it leaves broadcaster with no LIR node. In such case, it is better to reassign role to this broadcaster as normal participant because $e_b > e_n$.

- *(Deletion)*
In this, we consider the case when broadcaster goes out of the region of interest. Since it knows the location of every neighbouring participant (its LIR devices), it runs sorting based or greedy algorithm over its LIRs and reassign roles to them. New broadcasters are chosen if they can cover LIRs; otherwise, node is reassigned role of normal participant. Fig. 3(case 3.1), depicts the case when a node can be chosen as broadcaster which covers all LIRs of p. However in Fig 3(case 3.2), when a broadcaster node p decides to leave the region, it leads to formation of two other broadcasters to cover its LIR nodes.

The broadcaster being deleted shares its data, task and deadline with newly assigned broadcasters and normal participants. The information of the node being deleted would be updated to the server at the end of the task.

Algorithm 1: Incremental Insertion Algorithm.

Input: Initial set of participants: M,
Set of broadcaster: $B_{t_1t_2}$,
Energy $E_{t_1t_2}$ for M,
Boolean vector: br,
Set of new participants being inserted, I
Output: Role for each node in I

for each $x \in I$ **do**
  Init $flag$;
  $E_1 = E_{t_1t_2} + e_b + \kappa'_x \times (e_l - e_n)$;
  $E_2 = E_{t_1t_2}$;
  $E_3 = E_{t_1t_2} + e_n$;
  if x is physically connected to any $b \in B_{t_1t_2}$ **then**
    $flag = 1$;
    $E_2 = E_2 + e_l$;
  end if
  if $flag == 1$ && $E_2 \leq E_1$ **then**
    Assign x the role of LIR;
  else if $((flag == 1$ && $E_1 \leq E_2$ && $E_1 \leq E_3) \parallel (flag == 0$ && $E_1 \leq E_3))$ **then**
    Assign x the role of Broadcaster;
    Update br;
  else
    Assign x role of NP;
  end if
  Update $E_{t_1t_2}$;
end for

3.2.1 Incremental Deletion

In this subsection, we provide an algorithm for incremental deletion (Algorithm 2) to consider all the cases discussed above. For each deletion (Step 1), we find their existing role. If this participant x, happens to be a NP then it is simply removed(Step 39). If x was assigned a role of LIR then we remove it from LIR set by replacing one with zero in br vector(Step 3). We find its corresponding broadcaster using $BLIR_{t_1t_2}$ matrix(Step 4). The $BLIR_{t_1t_2}$ maintains the LIRs that broadcaster covers. Whenever a device is chosen as broadcaster, its index is appended in matrix $B_{t_1t_2}$. Its corresponding set of LIRs represented by Boolean vector, is appended in Boolean matrix, $BLIR_{t_1t_2}$. The bit at (i, j) is set to unity when j^{th} device is chosen as LIR for i^{th} broadcaster.

In case, the broadcaster of x no longer covers any LIR then its role is changed to normal participant(Step 6 to 9). If participant, x, was assigned the

Algorithm 2: Incremental Deletion Algorithm.

Input: Initial set of participants: M,
Set of broadcaster: $B_{t_1t_2}$,
Boolean vector: br,
Set of LIRs for each broadcaster: $BLIR_{t_1t_2}$
Physical connectivity matrix: $P_{t_1t_2}$
Energy $E_{t_1t_2}$ for M,
Set of participants to be deleted, D
Output: Role for each node in M

1: **for** each $x \in D$ **do**
2: **if** $br(x)$==1 **then**
3: $br(x) = 0$;
4: $b = x's$ broadcaster;
5: $BLIR_{t_1t_2}(b,x) = 0$;
6: **if** $sum(BLIR_{t_1t_2}(b)) == 0$ **then**
7: b becomes NP;
8: Remove b from $B_{t_1t_2}$;
9: **end if**
10: **else if** x exists in $B_{t_1t_2}$ **then**
11: i=index of x in $B_{t_1t_2}$;
12: Initialize P_{local};
13: **for** each $j : BLIR_{t_1t_2}(i,j) = 1$ **do**
14: $temp$=$BLIR_{t_1t_2}(i) \wedge P_{t_1t_2}(x)$;
15: $temp(j)$=0;
16: Insert $temp$ in P_{local};
17: **end for**
18: $\kappa'_x = sum(BLIR_{t_1t_2}(i))$;
19: Sort P_{local} in descending order of κ;
20: $iter$ = number of rows in P_{local};
21: $k = 1$;
22: **while** $\rho\ = 0$ && $k\ = iter$ **do**
23: m =Index of $P_{local}(k)$;
24: $\kappa' = sum(BLIRt_1t_2(i) \wedge P_{local}(k)$;
25: **if** $\kappa' == 0$ **then**
26: $br(m) = 0$;
27: Assign role of NP to m;
28: $\kappa'_x = \kappa'_x - 1$;
29: **else**
30: $br(m) = 0$;
31: $B_{t_1t_2} \cup \{m\}$;
32: $BLIR_{t_1t_2} = BLIR_{t_1t_2} \cup P_{local}(k)$;
33: $BLIR_{t_1t_2}(i) = BLIR_{t_1t_2}(i) \otimes P_{local}(k)$;
34: $BLIR_{t_1t_2}(i,m) = 0$;
35: $\kappa'_x = \kappa'_x - \kappa' - 1$;
36: **end if**
37: **end while**
38: **else**
39: Remove x from NP set;
40: **end if**
41: **end for**

role of broadcaster then we obtain a local physical connectivity matrix, P_{local}(Step 12). Our aim is to find new set of broadcasters that can cover its LIRs. For this, every entry of P_{local} is obtained by the operation of Boolean AND over LIR set of x and physical connectivity of its LIRs(Step 13 to 17). Next, we obtain local connectivity of x, represented by κ'_x (Step 18). We sort P_{local} in the descending order of participant's connectivity(Step 19). We then select participants from P_{local} that can cover LIRs of x(Step 30 to 35). The local connectivity of each participant is obtained by the Boolean AND operation(Step 24). In case they do not cover any LIR then they are removed from LIR set (Step 26) and assigned role of normal participant (Step 27). The local connectivity of x is decremented by 1 whenever its LIR is assigned a role of normal participant(Step 28). If participant covers some LIRs of x then it is added to the broadcaster set(Step 31), and corresponding LIR set is added to $BLIR_{t_1t_2}$(Step 32). With this, we update the local connectivity of x by removing participants that have been covered using boolean XOR operation(Step 33) and removing the participant from $x's$ LIR list that is recently chosen for broadcaster role(Step 34) and global LIR list, br. The κ'_x is then updated(Step 35).

4 PROPOSED MODEL

In dynamic scenario, we consider participants joining or leaving the region on fly and aim to assign role without rerunning greedy or sorting based algorithms for each insertion or deletion. However, our aim is to minimize energy. The proposed incremental algorithms provided in Algorithm 3, 2 do not provide optimal set of broadcasters. Hence, there is a need to rerun the algorithm when energy consumption becomes too high.

We set a threshold, δ to check the change in database. If change in database is less than δ then it is better to use the incremental technique to assign role (Step 4); otherwise, we recommend to rerun SBS or GBS algorithm (Step 6).

The percentage change in database can be calculated by following:

$$\% \text{ Change in DB} = \frac{NewData - OldData}{OldData} \times 100.$$

The proposed model is also depicted by Fig. 4. The actual SBS (or GBS) algorithm is applied to the original database to assign role to each participant(Step 1). Then, we use incremental insertion and deletion algorithm to adapt new changes to the dataset if the change is less than some threshold, δ(Step 3,4) otherwise rerun SBS/GBS(Step 6).

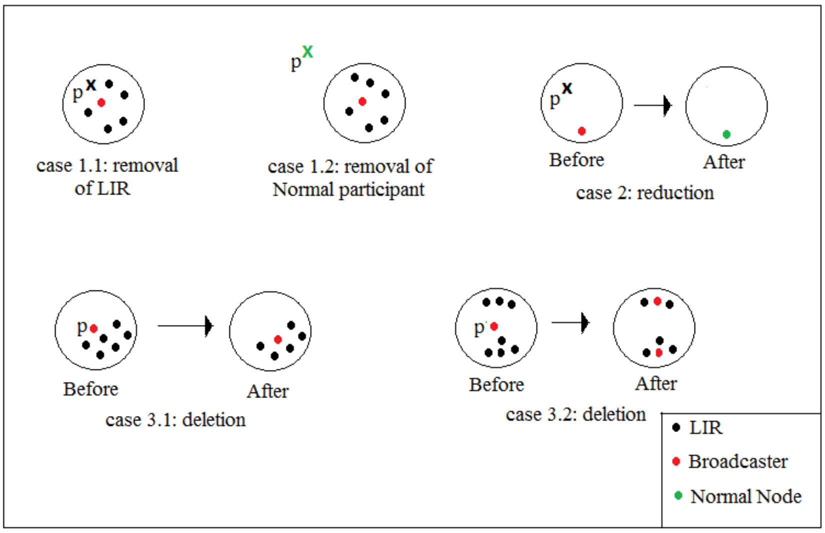

Figure 3: Different cases of the Deletion algorithm.

Algorithm 3: Adaptive Algorithm.

```
Apply SBS/GBS on the original set of participants.
for each x inserted or deleted do
    if % change in database < δ then
        Apply incremental insertion or deletion algorithm to assign role to x
    else
        Rerun SBS/GBS algorithm
    end if
end for
```

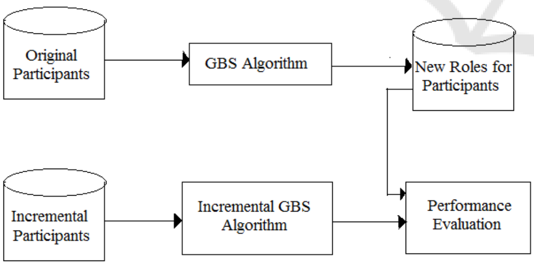

Figure 4: Proposed Model for Adaptive Algorithm.

5 PERFORMANCE EVALUATION

We evaluate the performance of proposed adaptive strategy using synthetic dataset of size 100, 300, 500 and 600. For each participant, we generate random positions in terms of x and y coordinate confined in area of $500 \times 500m^2$. For each case, a distance matrix is generated to calculate distance between every device to derive physical connectivity matrix. All parameters are set with same values as used in (Wang et al., 2016). We have used Matlab for all simulations and experiments. We have not experimented with GBS algorithm as its performance is already evaluated in (Garg et al., 2017), and the results remain same if it is compared with incremental GBS. Also, we have used energy model presented in (Garg et al., 2017) for experiments based on energy consumption. In first subsection, we discuss results of the insertion algorithm which is followed by the results of the deletion algorithm.

5.1 Incremental Insertion

In each of the results, we call the proposed algorithm as *Incremental Insertion* and *SBS* for rerunning SBS algorithm for each insertion. Each of the following results was obtained as an average of 25 runs on 25 datasets, each of the size $\{100, 300, 500, 600\}$.

In the first experiment, we aim to evaluate the time taken for assigning roles by the proposed incremental insertion and SBS algorithm. For this, the number of new participants inserted is equivalent to 5% of the dataset size. The sorting based algorithm has to be rerun for every insertion. Fig. 5 clearly depicts that our algorithm outperforms SBS.

In the next experiment, we evaluate the impact on energy consumption as a result of role assignment using the two algorithms. This is essential as SBS algorithm finds an optimal set of broadcaster so consumes minimum energy. Our aim is to check whether the new approach is efficient enough to assign roles such that it does not consume too much of energy. In this experiment also, we added new participants equivalent 5% of the data set size. Fig. 6 shows system energy consumed when roles are assigned by SBS and

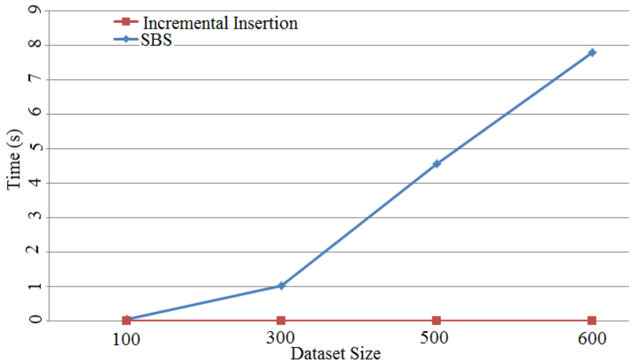

Figure 5: Time taken by SBS and Incremental Insertion Algorithms for role assignment.

incremental insertion approach. It can be observed that energy consumed by the proposed algorithm is almost equivalent to that of the optimal algorithm.

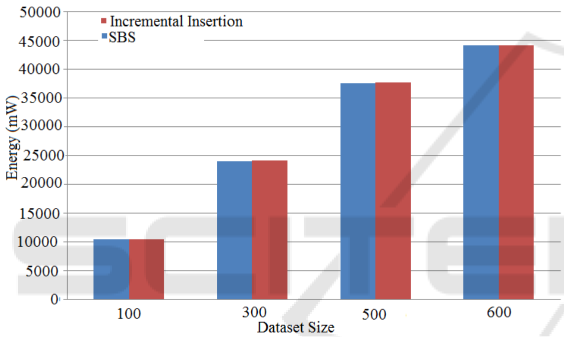

Figure 6: Energy Consumption by SBS and Incremental Insertion Algorithms.

Next, we evaluate the nature of processing time on a dataset of size 300 as we incrementally insert 15 participants. From Fig. 7, we observe that the rate of processing remains same for every insertion in proposed algorithm. However, there is slight variation for SBS algorithm which happens when an inserted node increases or decreases the number of broadcasters. We also observe that time taken by the SBS algorithm steadily increases. This is because the size of the dataset increases with every insertion.

In the last set of experiment, we analyse a threshold after which the SBS algorithm must be repeated. This is evaluated by varying percentage of insertion and observing the difference in energy between the incremental and SBS algorithms. For this experiment, we consider dataset of size 100, and incrementally insert 5%, 10%, 15% and 20% nodes. From Fig. 8, we can observe that when the number of insertions is greater than 10%, the incremental approach consumes more energy. So, 10% to 15% can be selected as a threshold for rerunning SBS algorithm.

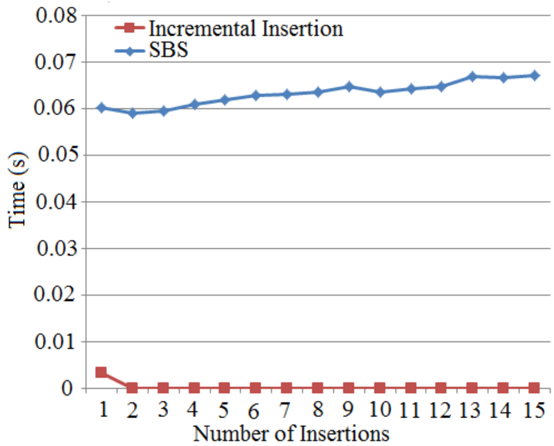

Figure 7: Time versus Number of Insertions.

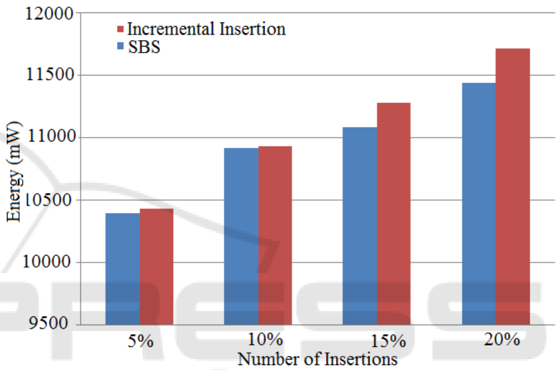

Figure 8: Energy Consumption for {5%, 10%, 15%, 20%} Insertions.

5.2 Incremental Deletion

In each of the results, we call the proposed algorithm as *Incremental Deletion* and *SBS* for rerunning SBS algorithm. Each of the following results was obtained as an average of 25 runs, each of the size {100, 300, 500, 600}. For this, we considered one data set for each of the sizes mentioned above and generated 25 sets of uniformly distributed random numbers. This provided indices of participants which were used to delete nodes.

In the first experiment, we evaluate time taken by proposed incremental deletion and SBS algorithm for role assignment. The number of participants which were deleted was equivalent to 5% of the dataset size. The sorting based algorithm had to be rerun for every deletion. Fig. 9 clearly depicts that the proposed algorithm takes much less time than SBS.

Next, we evaluate the impact on energy consumption as a result of role assignment using the two algorithms. This is required to evaluate performance of proposed algorithm over the optimal algorithm as energy is vital in mobile sensing. In this experiment,

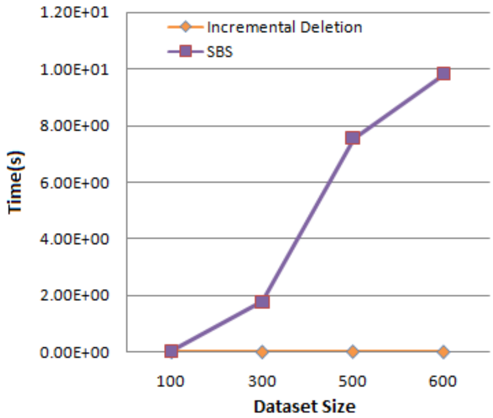

Figure 9: Time taken by SBS and Incremental Deletion Algorithms for role assignment.

we deleted 5% of the total participants in consideration. Fig. 10 shows system energy consumed when roles are assigned by SBS and incremental deletion approach. It can be observed that energy consumed by the proposed algorithm is almost equivalent to that of the optimal algorithm.

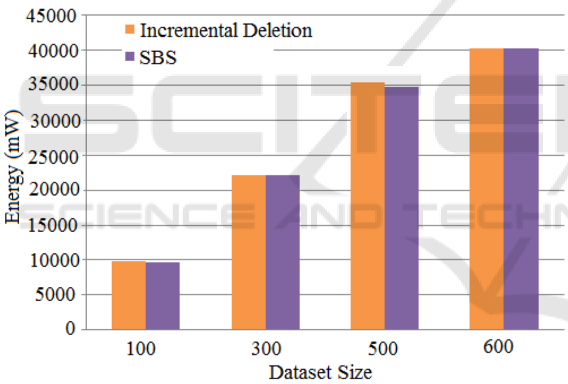

Figure 10: Energy Consumption by SBS and Incremental Deletion Algorithms.

In this experiment, we evaluate the nature of processing time on a dataset of size 300 as we incrementally delete 15 participants. From Fig. 11, we observe that the rate of processing remains same for every deletion in the proposed algorithm. However, the variation in SBS algorithm occurs due to the impact of deletion on number of broadcasters. We also observe that in contrast to insertion, the time taken by SBS algorithm steadily decreases. This is because the size of the dataset decreases with every deletion.

In the last set of experiment, we analyse a threshold after which the SBS algorithm must be repeated. This is evaluated by varying percentage of deletion and observing the difference in energy between proposed and optimal algorithm. For this experiment, we consider dataset of size 100, and incrementally delete

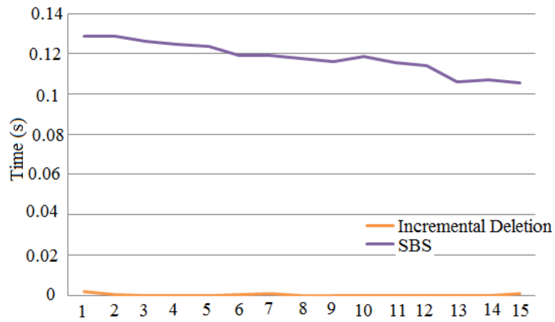

Figure 11: Time versus Number of Deletion.

5%, 10%, 15% and 20% nodes. From Fig. 12, we can observe that when the number of deletions is greater than 5%, incremental approach consumes more energy. So, a threshold between 10-15% can be chosen for rerunning the SBS algorithm.

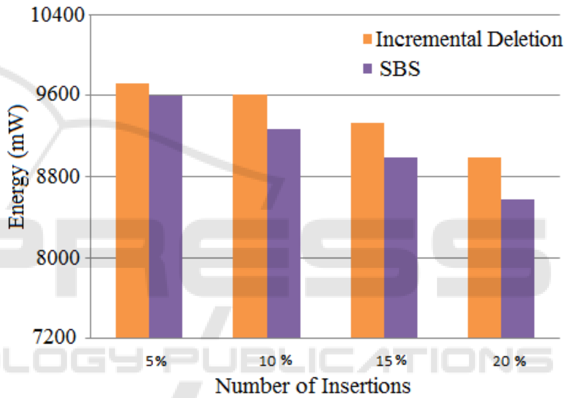

Figure 12: Energy Consumption for {5%, 10%, 15%, 20%} Deletions.

6 CONCLUSIONS

The existing techniques based on greedy and sorting based approaches are not efficient to consider the adaptive changes to the dataset for role assignment. Only way to assign role to new participant or change roles of existing devices when some devices leave the region is to rerun the algorithm. But as the number of participants can be quite large, it is not worth rerunning the algorithm for each change. In lieu of this concern, we propose incremental algorithms that help in saving 95-99.9% of the time for role assignment compared to the existing approach. Experimental results validate effectiveness of the proposed solutions. Also when change of the dataset is within [5%, 10%], the role assignment using these algorithms provide almost same energy consumption as obtained with the optimal algorithms. The current state of work does not consider velocity and movement information of participants for the role assignment. It is essentially based on their location registered to the server. This is part of our future work.

An important aspect to receive good user participation is privacy. If security is endangered then devices refrain from sharing their data. We plan to extend the work to incorporate several confidentiality and privacy concerns along with incentive mechanism to collect sufficient data samples.

ACKNOWLEDGEMENTS

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant number CRDPJ 476659-14.

REFERENCES

Anubhuti Garg, A., Nayak, A., 2017. Effective Role-Assignment for Participatory Sensing Systems, In *IEEE Percom Workshop on Crowd Assisted Sensing, Pervasive Systems and Communications*.

Wang Wang, W., Xi, T., Ngai, E.C, Song, Z., 2016. Energy-Efficient Collaborative Outdoor Localization for Participatory Sensing, In *Sensors*, vol. 16, pp. 1–12.

Survey Lane, N.D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.T., 2010. A Survey of Mobile Phone Sensing, In *IEEE Comm. Magazine*, vol. 48, issue 9, pp. 140–150.

Sensors E. Macias, A. Suarez, J. Lloret, "Mobile Sensing Systems", *Sensors*, vol. 13, pp. 17292–17321, 2013

Sayler I. Constandache, S. Gaonkar, M. Sayler, R.R. Choudhury, L. Cox, "Enloc: Energy-efficient localization for mobile phones", *Proc. of IEEE INFOCOM*, pp. 2716–2720, 2009.

Yoon C. Yoon, D. Kim, W. Jung, C. Kang, H. Cha, "AppScope: Application Energy Metering Framework for Android Smartphone Using Kernel Activity Monitoring", *Proc. of USENIX ATC*, pp. 387-400, 2012.

Geolife Y. Zheng, X. Xi, W.Y. Ma, "GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory", *IEEE Data Eng. Bull.* 33, pp. 32–39, 2010.

Song Z. Song, J. Ma, M. Dong, W. Wang, X. Gong, X. Que, "Phoneradar: Infrastructure-free device-to-device localization", *Proc. of IEEE Vehicular Tech. Conf.*, 2014.

Kumar S. Kumar, V. Sharan, R. M. Hegde, "Energy efficient optimal node-source localization using mobile beacon in ad-hoc sensor networks", *Proc. of IEEE GLOBECOM*, pp. 487-492, 2013.

Xiao B. Xiao, H. Chen, S. Zhou, "Distributed Localization Using a Moving Beacon in Wireless Sensor Networks", *IEEE Trans. on Parallel and Distributed Systems*, vol. 19, no. 5, pp. 587–600, 2008.

Davies N. Davies, A. Friday, P. Newman, S. Rutlidge, O. Storz, "Using bluetooth device names to support interaction in smart environments", *Proc. of ACM MobiSys*, pp. 151-164, 2009.

Johnson T.A. Johnson, P. Seeling, "Localization Using Bluetooth Device Names", *MobiHoc*, 2012.

Huang A. Huang, L. Rudolph, "A privacy conscious bluetooth infrastructure for location aware computing", *Proc. of SMA Symposium*, 2005.

WLAN I. Shafer, M.L. Chang, "Movement Detection for Power-Efficient Smartphone WLAN Localization", *Proc. of MSWiM*, pp. 81-90, 2010.

QoI Z. Song, B. Zhang, C.H. Liu, W. Wang, "QoI-Aware Energy-Efficient Participant Selection", *Proc. of SECON*, pp. 248-256, 2014.

Jigsaw H. Lu, J. Yang, Z. Liu, N.D. Lane, T. Choudhury, T. Andrew, "The Jigsaw continuous sensing engine for mobile phone applications", *SenSys*, pp. 71-84, 2010.

Goldman J. Goldman, K. Shilton, J. Burke, et al. "Participatory sensing a citizenpowered approach to illuminating the patterns that shape our world." *http://www.wilsoncenter.org/sites/default/f iles/participatorysensing.pdf*

Hicks J. Hicks, N. Ramanathan, D. Kim, M. Monibi, J. Selsky, M. Hansen, D. Estrin "AndWellness: An Open Mobile System for Activity and Experience Sampling." *Proceedings of the 1st Wireless Health Conference*, pp. 34–43, 2010.

Reddy S. Reddy, K. Shilton, G. Denisov, C. Cenizal, D. Estrin and M. Srivastava "Biketastic:sensing and mapping for better biking." *Proceedings of the 28th Annual CHI Conference on Human Factors in Computing Systems* pp. 1817-1820, 2010.

PSense P. Baier, F. Durr, K. Rothermel "PSense: Reducing Energy Consumption in Public Sensing Systems." *Advanced Information Networking and Applications*, pages 136-143, 2012.