Optimizing Smartphone Energy Consumption in Sensor Data Collection and Data Transmission

by

Devika Sondhi

Under the Supervision of Dr. Pushpendra Singh

Indraprastha Institute of Information Technology Delhi

June, 2015

Optimizing Smartphone Energy Consumption in Sensor Data Collection and Data Transmission

By

Devika Sondhi

Submitted in partial fulfilment of the requirements for the degree of Master of Technology

to

Indraprastha Institute of Information Technology Delhi

June, 2015

Certificate

This is to certify that the thesis titled "Optimizing Smartphone Energy Consumption in Sensor Data Collection and Data Transmission" being submitted by Devika Sondhi to the Indraprastha Institute of Information Technology Delhi, for the award of the Master of Technology, is an original research work carried out by her under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

June, 2015

Dr. Pushpendra Singh
Department of Computer Science
Indraprastha Institute of Information Technology Delhi
New Delhi 110020

Acknowledgements

I take this opportunity to sincerely thank my advisor, Dr. Pushpendra Singh, for providing me with all the valuable guidance and suggestions that have helped me tremendously in honing my skills. I would also like to express my gratefulness to the institute, IIIT-Delhi, for providing me with all the necessary facilities to carry out my thesis.

I thank all those who kindly volunteered for the data collection and generously lent their smartphones to carry out the evaluation of my work.

Lastly but importantly, I thank my family for their encouragement and support in times of difficulties.

Abstract

Sensors that come built in smartphones are extensively used by various apps and for analysis purpose. Some of these sensors, especially GPS and Microphone, are energy expensive. Hence, there is a need to use these judiciously. To study the trade-off between data accuracy and energy consumption due to sensor usage, we experiment with duty cycling while collecting sensor data. In many applications, the sensor data is required to be sent to a cloud or a server for storage or further computations. The size of the data is correlated to the amount of energy consumed in data transmission as the number of bytes to be transferred contributes significantly to the amount of energy consumed in data transfer. The key here is to reduce the size of data but at a minimal cost of accuracy. We propose a representative clustering approach to represent the dataset and transmit these representatives in place of the entire dataset. Further, we evaluate the amount of energy conserved by using this approach by transmitting representative data over different wireless technologies.

Contents

List of Figures

List of Tables

1. Introduction

Smartphones come with a variety of sensors used for a variety of applications such as traffic predictors, navigators, fitness apps, local search and recommendation system(such as Foursquare), noise detectors etc. . These sensors drain a lot of energy as they are polled frequently in the process of data collection. Often there is a possibility that the data collected is repetitive or redundant. For instance, the GPS data continuously being collected in a moving car versus GPS data collected at a restaurant would vary in the degree of repetitiveness of the readings obtained. The latter would have a higher degree of repeated values and hence, logging a value less frequently would work in this case and would be an optimal choice in terms of energy conservation.

For analysis or further computations the sensor data may need to be sent to a cloud or distant server for which the internet connectivity is required. The process of transmitting data consumes a lot of energy of the device. The size of the data to be transmitted contributes significantly to the total energy consumed in data transmission[3]. Hence, a possible solution is to reduce the size of the data. However, direct reduction in the dataset may lead to absence of the data points that may represent a critical event in the whole dataset. In other words, direct reduction may cause high inaccuracy. However, we can attempt to minimize the degradation in accuracy by selecting the representatives in a systematic manner.

As a proposed solution, we make use of clustering algorithms- CURE, BIRCH, DBSCAN applied on our sensor data(GPS, Accelerometer, Audio) to select representatives of each cluster. Hence, the representatives from each cluster altogether represent the whole data instead of just a part of it. So instead of sending the entire data over the network, sending the representatives would suffice, hence allowing some amount of energy to be conserved at a minor cost of accuracy. Note that our approach works for applications that deal with non-real time data. In case of real time data transmission, each time the data transmitted is already small in size and hence no significant improvement would be provided by our proposed solution. The amount of energy conserved depends on factors like the original data size, the percentage of data reduction and the wireless network technology used for data transmission. We shall elaborate on how the results vary with these factors in the upcoming sections.

The following sections are divided as follows.
Section 2 briefly describes the three algorithms- CURE, BIRCH, DBSCAN and gives the background of how we have extended the three clustering algorithms to obtain representatives. The section also discusses other work done in the domain of smartphone energy optimizations in sensor data applications.

Section 3 describes our methodology of data collection for all the three sensors. Section 4 presents the results and conclusions of our study on applying duty cycling while collecting data. Section 5 discusses the results of applying the three clustering algorithms-CURE, BIRCH, DBSCAN on the three sensor datasets.

Section 6 evaluates the work that we present and Section 7 concludes the study and the work.

2. Background and Related Work

I. *Background Work*

As discussed, we have worked with three clustering algorithms to obtain our dataset representatives.

We have used these three particular clustering algorithms as they make use of entirely different clustering approaches. CURE makes use of a representative based clustering approach. On the other hand, DBSCAN makes use of density based clustering and BIRCH uses of hierarchical clustering.

- CURE[12]: CURE(Clustering Using Representatives) subsamples a large database selectively and performs clustering on smaller dataset which is later transferred to larger dataset. CURE starts with each point as a cluster. With a predefined sample size c, which is determined using chernoff bounds, c well scattered points are selected from the cluster. These selected points, called the representatives of the clusters, are then shrunk towards the centroid of the cluster by a factor 'a'. Closest clusters are merged and new representatives are selected. Distance between two clusters is the distance between the closest pair of representative points- one from each cluster. The clusters are merged until the desired k number of clusters is obtained. In this process, the outliers, that are clusters of size smaller than a threshold, are eliminated. The representatives chosen in the last round of clustering are considered as the final representatives of the data.

- DBSCAN[14]: DBSCAN(Density-based spatial clustering of applications with noise) is a density based clustering algorithm. Before we get into the clustering approach, there are a few definitions to consider.
    - Eps-Neighbourhood: The neighbourhood within a radius of 'Eps' units of an object/point.
    - Minpts.: Minimum number of points in the Eps-neighbourhood of that point p.
    - Core point: If for a point p, number of points in its Eps-neighbourhood is at least Minpts then p is a core point.
    - Directly density reachable Point: A point p is directly density reachable from point q if p is in Eps-neighbourhood range of q and q is a core point.
    - Density reachable point: A point p is density reachable from a point q w.r.t. Eps and MinPts if there is a chain of points P_1 ..... P_n, $P_1 = q$, $P_n = P$ such that P_{i+1} is directly density-reachable from P_i.

If the Eps-neighbourhood of a point p contains more than MinPts, a new cluster with a core object is created. DBSCAN iteratively collects directly density reachable objects from these core objects which may involve the merge of a few density-reachable clusters. The process terminates when no new point can be added to any cluster. Once the final clusters are obtained, well scattered representatives from each cluster are selected to represent the dataset.

- BIRCH[13]: The acronym BIRCH stands for Balanced Iterative Reducing and Clustering using Hierarchies. Before getting into the algorithm, there are some terminologies that are required to be understood.
    - Clustering Feature(CF): Contains summary of statistics of a cluster. CF entry of a cluster is defined as a triple {no. of data points in the cluster, linear sum of the data points of the cluster, square sum of the data points of the cluster}
    - CF-Tree: It is a height balanced tree with two parameters- Branching Factor(number of entries in each node) and Threshold T(diameter of all entries in each node)

  Broadly, the following steps are involved in the BIRCH algorithm.

  Starting with the root we find the CF entry in the root closest to the data point and move to that child and repeat the process until a closest leaf entry is found. At the leaf, if the point can be accommodated in the cluster, the entry is updated; else if this addition violates the threshold T, we split the entry; if this violates the limit imposed by L(Branching factor for leaf node), we split the leaf. If its parent node is full, split that and so on. Update the CF entries from the leaf to the root to accommodate this point.

  This insertion algorithm is carried out for every data point. If, in the middle of the above step, the size of the CF tree exceeds the size of the available memory, the value of threshold is increased. Then we convert the partially built tree into a new tree. The above steps are repeated until the entire dataset is scanned and a full tree is built. Further, global clustering algorithm is applied to the sub-clusters as provided by leaf entries of the CF tree. At the end, the entire dataset is scanned to label the data points. Once the final clusters are obtained, well scattered representatives from each cluster, at leaf entries, are selected to represent the dataset.

II. *Related Work*

Extensive work has been done in the domain of energy optimization on smartphones in sensor data collection and cellular data transmission. Some approaches, such as TailEnder, concentrate on scheduling transmissions to minimize energy consumption due to tail energy generated. Tail energy is the energy spent in high-power state after completion of data transfer. Another approach, presented by Mirco Musolesi et al.[2], attempts to reduce the upload activities by developing a forecasting system at the server end. Instead of continuously uploading data, the forecast system makes use of the available data at the server-end to predict the next state.

The work in [4] focuses on optimizing the data collection phase itself by replacing the power hungry sensor such as GPS with a low power ambient sensors like pressure, thermometer, hygrometer and light sensors deployed on smartphone to infer sematic locations such as home, office, shop etc..

[5] presents an approach of opportunistically selecting between the wireless interfaces- Wifi and cellular network without powering up the network interface to scan for Wifi connection availability. Instead, it makes use of context information such as history, time and cellular network condition to predict Wifi availability.

Other approaches such as [6] and [7] make use of context aware approaches to draw inferences from historically known data or collect data by making use of certain context knowledge instead of continuous sensing- for instance, collect data at a particular place or time.

Several approaches as in [8], [9] and [10] make use of clustering, but the authors cluster the sensor nodes that are likely to provide similar sensor due to spatial correlation. In [9] the sensor nodes are clustered once the sink node computes the correlation between the sensor nodes. After clustering, the sink schedules the nodes for data transmissions, which happens when the predicted value at the sink is significantly different from the actual value at the sensor node. Note that the predictor runs concurrently at source as well as the sink. Techniques of data aggregation and compressive sensing have also been put to use as discussed in [8] and [10].

The work presented in [11] makes use of distributed clustering technique meant for distributed datasets. The data collected at every node is clustered locally using K-Means. Using the contours as the representatives of each cluster, these representatives are transferred to a leader node where merging of clusters coming from each node takes place until no overlapping clusters are obtained. However, for a high variability in the shape and density of clusters, using contours as the

representatives would miss out on some critical data points located in the core of the cluster, causing inaccuracy in representation. Unlike contour based representation, we choose the cluster representatives as well scattered points from each cluster and analyse the effect on the data accuracy on selecting different percentage of data points as representatives. Also for the approach taken in [11], an overhead is involved in electing an appropriate leader and mapping nodes to leader such that the resultant reduced clusters obtained give the best possible accuracy.

To the best of our knowledge, no work implements a representative based clustering approach on sensor data to represent a dataset that would work for a standalone device at an acceptable accuracy. Our work attempts to satisfy all these properties in order to reduce the energy consumed in data transmission.

3. Data Collection

We have worked with three sensors-GPS, Accelerometer, Microphone. In our experimental setup, we logged the readings from each sensor along with the timestamp.

I. *GPS Sensor Data*

No. of points in Original Data	81216	40272	20081	10043	5021
File Size of Original Data	2.8MB	1.43MB	731KB	366KB	183KB
Distance covered(in km.)	833	417	208	104	52

Table I: GPS Sensor Data

We collected GPS data at various events, spanning over a week. Apart from latitude and longitude values, we captured the timestamp (in milliseconds). Thus coordinates of a data point represent [Longitude,Latitude,Timestamp]. The data has been collected while driving on road or while travelling in the metro. The data has been collected with and without duty-cycles. We set the duty cycle intervals to 5 seconds, 15 seconds, 30 seconds, 1 minute and 2 minutes.

Two types of datasets were obtained.

Dataset 1: This dataset contains the GPS sensor data collected without duty cycling as well as with duty cycling at the described intervals.

The main purpose of collecting this dataset is to study the trade-off between the localization accuracy and the energy consumption due to sensor usage. The GPS readings have been collected on a 10km. stretch while travelling by metro for duration of 21 minutes. On road, the uniformity of speed and acceleration cannot be guaranteed during data collection over different duty cycle intervals. The environment requirements are relatively favourable over a metro. Hence, metro has been selected as the mode of transport while collecting the data. In all, 6 data samples (one for each duty cycle interval and one for no duty cycling) were obtained for this dataset.

Dataset 2: This dataset contains GPS data samples of sizes varying from approx. 5000 points to 81000 points. This data has been collected for certain duration each day for at most two weeks. The data collected is a continuous one, that is, no duty cycling has been applied in this case. This data has been obtained either while driving or in a metro. The description of the samples thus obtained has been described in Table I. These samples have been used to evaluate the

performance of the three clustering algorithms (DBSCAN, BIRCH, CURE) in representing the data collected.

II. Accelerometer Sensor Data

No. of points in Original Data	84940	50964	33976	16988	8494
File Size of Original Data	2.08MB	1.25MB	860KB	433KB	217KB
No. of Brakes	102	56	33	10	5

Table II: Accelerometer Sensor Data

The accelerometer sensor readings have been taken while travelling in the metro. The x axis of the smartphone was aligned along the direction of motion while collecting the data. The data readings contain the x axis values of the accelerometer and the timestamp. On polling the accelerometer sensor, the rate of delivering the sensor events was set to the default which is SENSOR_DELAY_NORMAL as it is considered suitable for screen orientation changes. Note that no duty cycling has been applied in case of accelerometer sensor since this sensor is always active. Hence the study of power consumption by this sensor at various duty cycling intervals may not be of relevance here.

The dataset consists of data samples with sizes varying from approx. 8400 points to 84000 points. For each sample, the number of brake events has been noted that shall be used to evaluate the performance of clustering algorithms in the following sections. The details of the dataset have been described in Table II.

III. Audio Sensor Data

No. of points in Original Data	153773	63480	32976	16006	7489
File Size of Original Data	2.69MB	1.17MB	625KB	303KB	143KB
No. of Sound Events	12	5	3	1	1

Table III: Audio Sensor Data

For audio data collection we played a 20 seconds long tune at every 25 seconds interval. The event of playing the tune is referred to as the sound/noise event. Each reading, i.e, a data point consists of the amplitude value and the timestamp. The sampling rate, on polling the microphone, was set to

11025Hz and the channel configuration was set to CHANNEL_IN_MONO. The microphone sensor is activated at various duty cycle intervals. These intervals were set to 5 seconds, 15 seconds, 30 seconds and 1 minute. Apart from duty cycling, we have also collected data without duty cycling, i.e., when the microphone is active throughout the data collection period.

Similar to the GPS data collection we have two types of datasets for audio data as well.
Dataset 1: This dataset contains audio data collected at the described duty cycles as well as without duty cycling. The sound is played repeatedly every 25 seconds for about 20 minutes with no other prominent surrounding sound. This environment is recreated to take audio recordings at every duty cycle interval as well as for recording without any interval. In all, 6 data samples (one for each duty cycle interval and one for no duty cycling) were obtained for this dataset. The purpose of this dataset is to study the trade-off between energy consumed on using microphone sensor and sound detection accuracy at different duty cycle intervals.

Dataset 2: For this dataset the same environment was created for data collection. However, the duration (which was 20 minutes in case of Dataset 1) has been varied according to the number of data points we wished to obtain in a sample. Note that, no duty cycling has been applied while recording this dataset. All of this data has been collected with the microphone continuously listening for the audio data. Thus, on recording the audio data for different durations we obtained 5 data samples, the details of which have been described in Table III. We apply the three clustering algorithms on these data samples to evaluate their performance.

4. Data Collection Analysis

Our objective to carry out this study is to analyse how the energy consumption is affected in the process of data collection. In order to reduce energy consumption, the sensor usage may be duty cycled instead of continuously keeping the sensor active. However, duty cycling would affect the quality and accuracy of data collected by missing certain data that maybe critical in nature. Ideally, we would want to minimize the energy consumption and maximize the accuracy(i.e minimize inaccuracy). However, as explained, this may not be possible and some intermediate combination of energy consumption and accuracy should be chosen. Hence, a tradeoff is involved and which point of energy consumption and accuracy to select depends on the nature of application.

I. GPS Data Analysis

We analysed the data collection approach by collecting the data at certain intervals as described for Dataset 1 for GPS. Table IV contains the results of duty cycling while collecting GPS data. Fig. 1 depicts the results for the same. The average power and total energy have been measured using PowerTutor app on Xperia L device, running Android v4.1.2.

To study the quality of data obtained for every setup of duty cycle interval, we measured the average localization inaccuracy. We computed the average localization inaccuracy for a dataset *d* as the average of the distance between every point in the dataset collected without duty cycling and the closest point to it in the dataset *d*. Table IV indicates that, as expected, the localization inaccuracy increases with increasing the duty cycle interval.

As can be seen, the average power, computed over 20 iterations, declines with increasing the duty cycle interval. However, if the total energy consumption is to be observed, the value increases with an increase in the duty cycle interval upto 15sec interval and from there on the energy consumption declines. However, as compared to 10.7MJ of energy consumption in case of no duty cycling, the energy consumption at a 2min. duty cycling interval is 101MJ which is much higher than without duty cycling. This indicates that there isn't really an advantage in going for duty cycling. Before we get into the explanation to this observed trend, it is important to note that a GPS signalling process consists of three stages:

1) Acquisition: It's the first stage in GPS receiver start up. During this stage, the GPS receiver searches for the satellites to receive data from. Note that it is the most energy intensive of all the stages.
2) Tracking: After the satellite acquisition, the receiver locks to the satellites to track the satellites signals.
3) Decoding: With successful tracking, the data packets received from the satellites can be successfully decoded to pass on to the main processor for location calculation.

In case of continuous mode of data collection, the acquisition stage occurs only once in the whole process of data collection. However, on setting up the duty cycle intervals, the GPS receiver receives data and logs it and then goes into the standby mode until it is woken up after the set interval for re-acquisition. Hence, with duty cycling, the acquisition stage occurs several times in the process of data collection. As discussed, acquisition is an energy intensive stage. Several occurrences of it cause the overall energy consumption, due to GPS sensor, to increase.

Hence, clearly, setting up of fixed duty cycle intervals, up to 2min, is not advisable here as it not only decreases the localization accuracy but also increases the energy consumption.

Duty Cycle Interval	**Average Power**	**Total Energy**	**Inaccuracy w.r.t w/o Duty Cycling**
0sec.	450mW	10.7MJ	0m
5sec.	450mW	10.7MJ	23.9m
15sec.	300mW	805.3MJ	44.9m
30sec.	225mW	401.6MJ	82m
1min.	122.5mW	175MJ	125.6m
2min.	56.25mW	101MJ	263.2m

Table IV: Trade-off on Applying Duty Cycling over GPS Data Collection

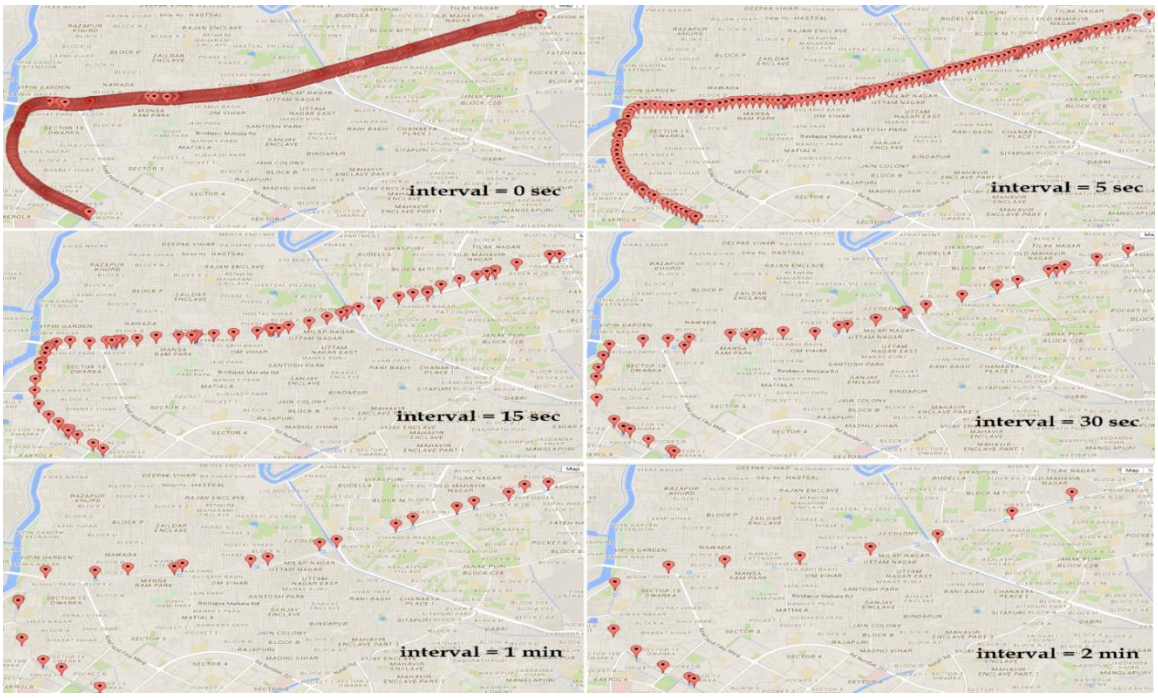

Fig. 1. GPS data when mapped at different duty cycle intervals

II. Audio Data Analysis

The data has been collected as described for Dataset 1 for Audio. The outcome of duty cycling is shown in Table V. The PowerTutor app has been used to measure average power and energy, over 20 iterations.

A total of 30 noise events existed in the experiment setup while collecting data. The accuracy column in the table shows the number of these events captured in each dataset collected over the same experiment setup at different duty cycle intervals.

The average power and the total energy consumed show the expected trend of declining with the increase in duty cycle interval and so does the accuracy.
Fig. 2. illustrates the dataset obtained at every setup.

Duty Cycle Interval	Average Power	Total Energy	Accuracy (No. of Noise Events Captured)
0sec.	910mW	860J	30 (of 30)
5sec.	265mW	268J	22 (of 30)
15sec.	70mW	81.4J	22 (of 30)
30sec.	46mW	50.1J	13 (of 30)
1min.	14mW	15J	8 (of 30)

Table V: Trade-off on Applying Duty Cycling over Audio Data Collection

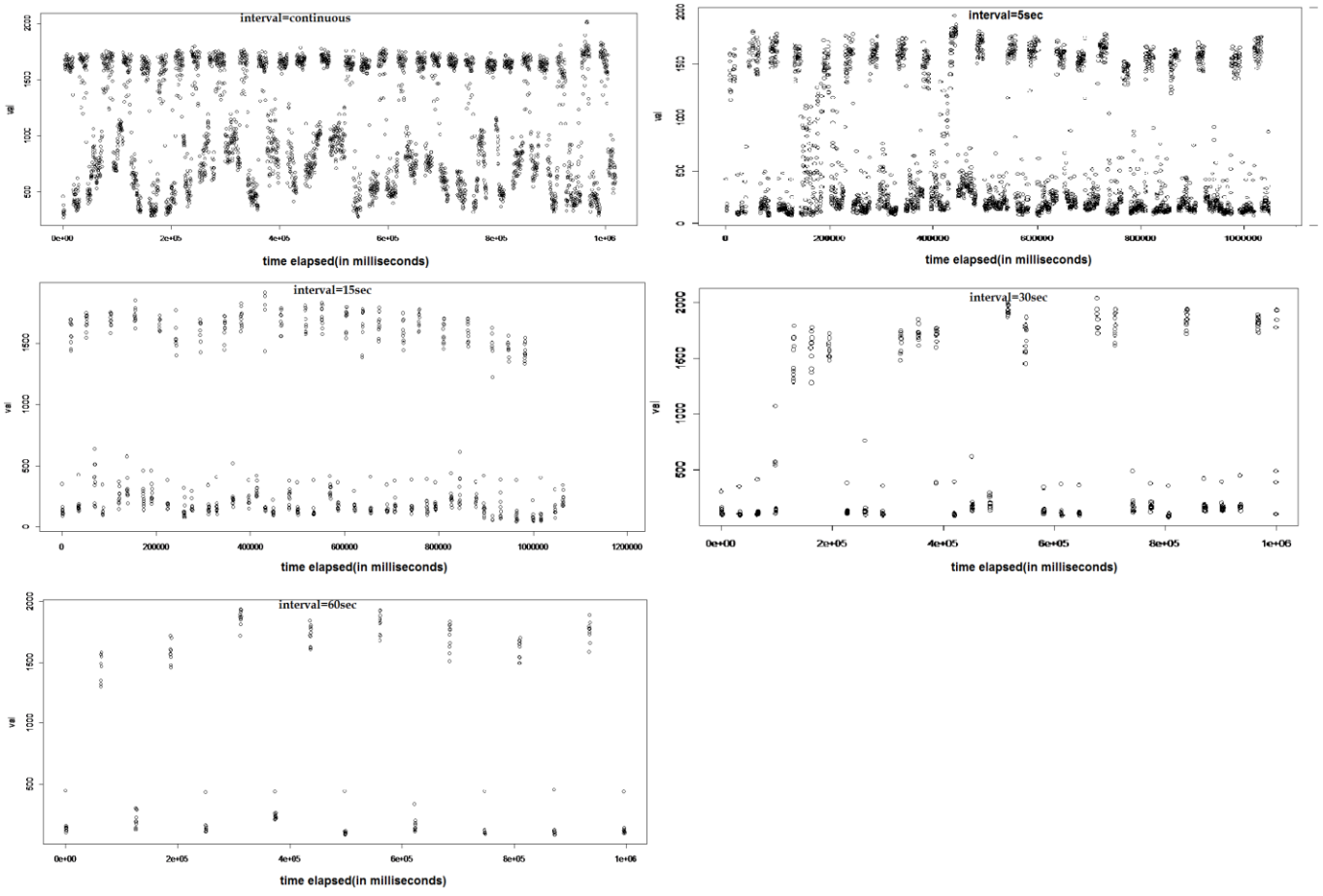

Fig. 2. Audio data points at different duty cycle intervals

5. Applying Clustering Algorithms on Data

The three clustering algorithms, namely CURE, DBSCAN and BIRCH, were applied on the data samples of the three sensors. They were run on Sony Xperia Z3 device with 3GB RAM and 2.5 GHz Qualcomm Quad-core processor running on Android v4.4.4 (KitKat).

Selection of Representative: A detailed description of how we selected the representatives for every clustering algorithm has been provided in Section 2. In general, the representative selection approach used for every algorithm is to select few well scattered (random) points from every cluster once the clusters are obtained. The number of well scattered points to select depends on the reduction factor to obtain.

Time complexity of CURE, DBSCAN and BIRCH are $O(n^2 \log n)$, $O(n \log n)$ and $O(k*n)$ respectively where k is the number of clusters and n is the number of points.

I. *GPS Data*

The parameters passed in the 3 clustering algorithms were tuned to reduce each of the 5 GPS data samples(as described in Data Collection Section) to 50%(+-1%) of the data size.

Parameters used for CURE

For CURE, the minimum representative count in each cluster was set to 12, the shrink factor was initialized to 0.01. Number of partitions was set to 5 and the reducing factor for each partition was set to 2. Using these parameters, the initial sample size has been computed.

Parameters used for BIRCH

For BIRCH, the maximum node entries were set to 14 and the threshold distance was set to 0.001.

Parameters used for DBSCAN

The Minpts. was set to 5 and the eps was set to 1000.

The execution time was observed for every algorithm over each sample size. As expected from the time complexities, Fig. 3. shows that BIRCH took the least execution time and CURE took the maximum execution time for every data sample.

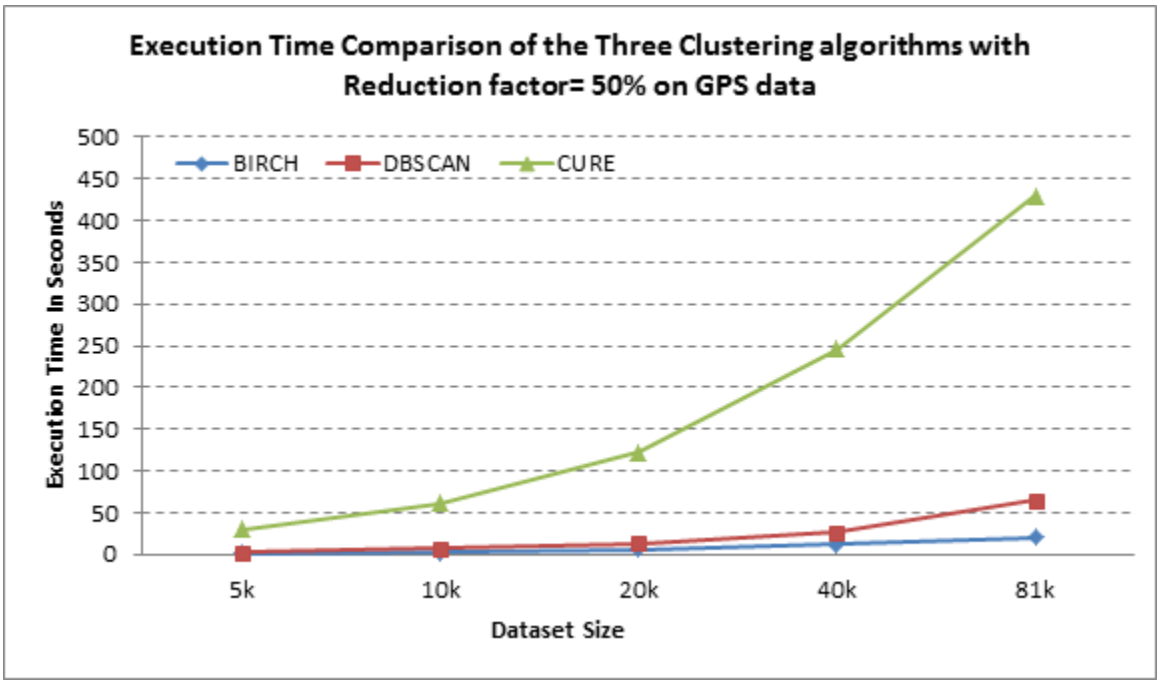

Fig. 3. Comparison of clustering algorithms based on program execution time

Apart from the execution time based comparison, the algorithms have been compared on the basis of clustering accuracy and the accuracy of selecting data points that represent the whole data sample.

For GPS data samples we evaluate the localization inaccuracy when the data is represented by 50% of the data points as obtained by the three algorithms.

As can be observed from Fig. 4. the representatives obtained by BIRCH and DBSCAN give a relatively stable inaccuracy of around 4.4 meters and 5.8 meters respectively, for almost all the samples. BIRCH also provides representatives with the least inaccuracy among the three algorithms and this holds across all sample sizes. CURE, on the other hand gives unstable inaccuracy values across the data samples. This can be attributed to the nature of CURE algorithm as it selects a random sample of data points in its pre-clustering stage.

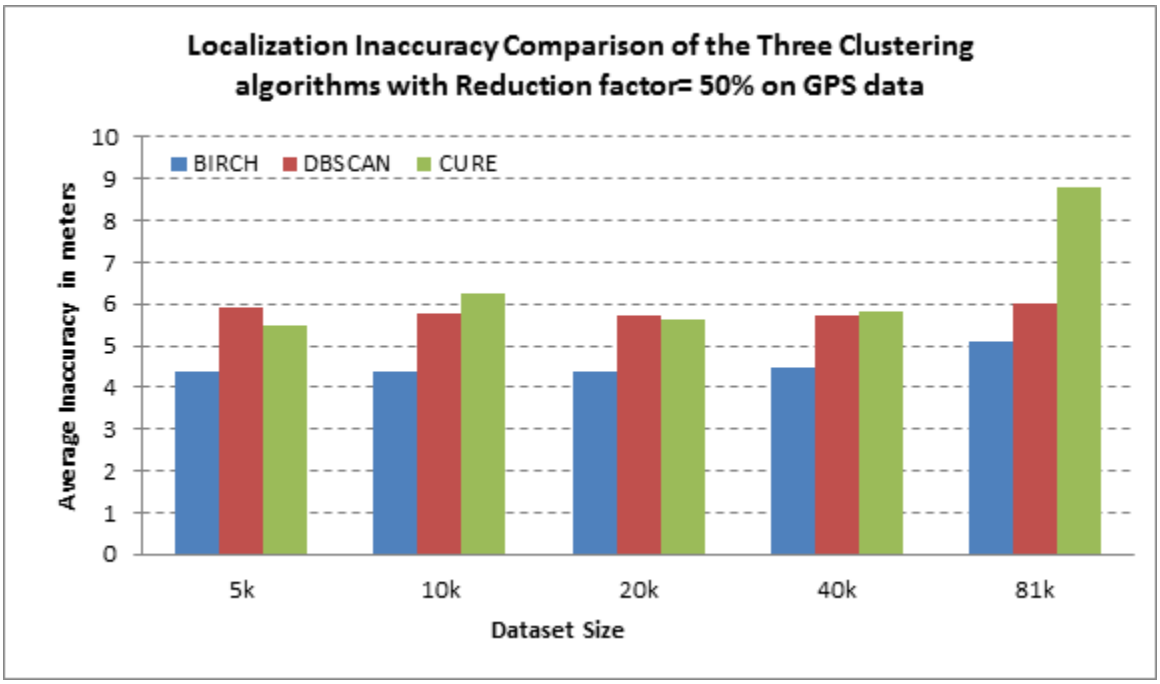

Fig. 4. GPS Accuracy

II. *Accelerometer Data*

Similar as in case of GPS data, the accelerometer samples were also reduced to 50%(+-1%) by tuning the input parameters passed in CURE, BIRCH and DBSCAN.

Parameters used for CURE

For CURE, the minimum representative count in each cluster was set to 12, the shrink factor was initialized to 0.01. Number of partitions was set to 20 and the reducing factor for each partition was set to 2. Using these parameters, the initial sample size has been computed.

Parameters used for BIRCH

For BIRCH, the maximum node entries were set to 14 and the threshold distance was set to 0.01.

Parameters used for DBSCAN

The Minpts. was set to 8 and the eps was set to 1000.

Due to the time complexities as discussed before, the execution time was the least for BIRCH and maximum for CURE across all data samples even for the accelerometer sensor data. Refer to Fig. 6 for the results.

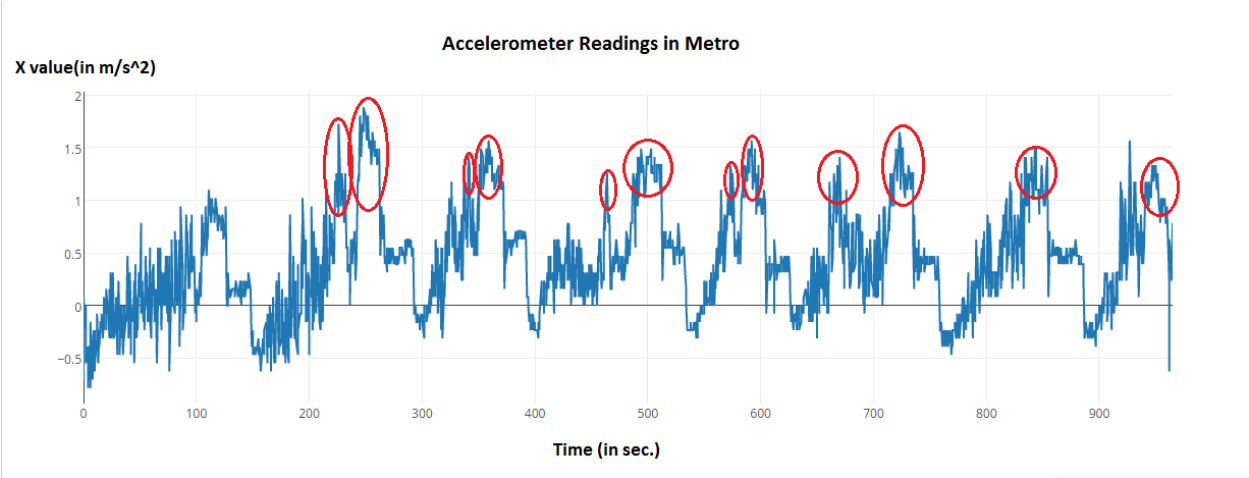

Fig. 5. Brake events in an accelerometer data sample

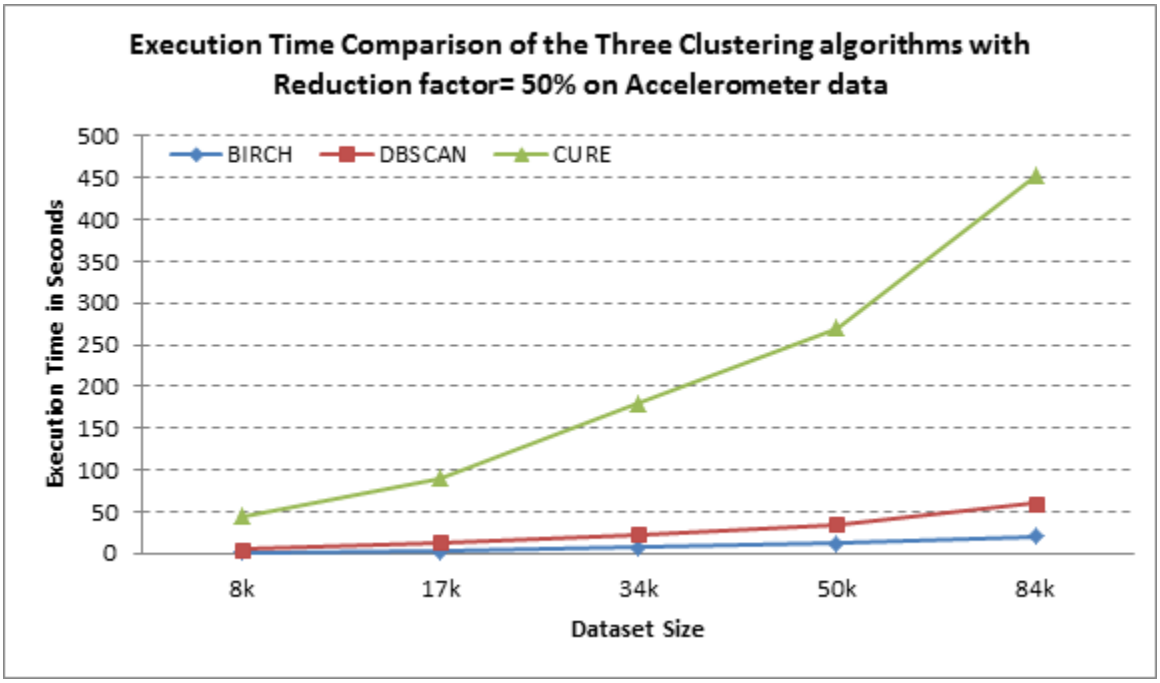

Fig. 6. Comparison of clustering algorithms based on program execution time

To evaluate how well the representatives represent the original data, we check for the number of brake events that could be successfully detected even at a 50% reduction of data. A brake event has been detected as positive if the following condition is satisfied: the average accelerometer

value is greater than 1m/s^2 over a time window of 3 seconds, which indicates a sharp deceleration. Fig. 5 shows a data sample consisting of 12 brake events (encircled) recorded while travelling in the metro. The details of number of brakes events in each sample are provided in Table II.

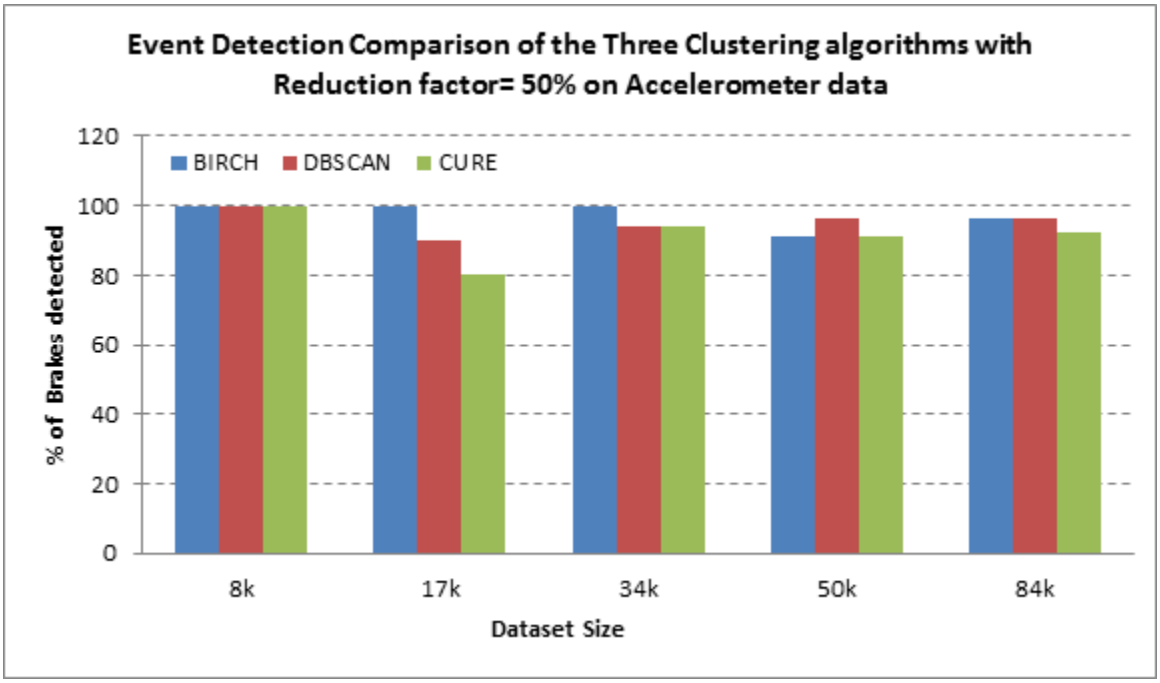

Fig. 7. Accelerometer Accuracy

The detection accuracy for the representatives produced by each of the clustering algorithm is shown in Fig. 7. The graph shows the percentage of brake events detected for each of the data sample. For almost all data samples, except for the 50k sample, the accuracy of BIRCH is maximum or at par with the other two algorithms. In case of the 50k sample, BIRCH was able to detect 51 out of 56 brake events whereas CURE and DBSCAN were able to detect 51 and 54 out of 56 brakes. On inquiring, we found out that in the original data sample there existed 3 brake events such as the one shown in Fig. 8. This shown brake event was captured by 9 points in the original data sample as compared to other detected brake events that were captured by as many as 115 points. On reduction to 50% data size, these 9 points were represented by just 3 points when BIRCH was implemented and by just 2 points by CURE implementation. Hence, due to this insignificant number, the brake went undetected.

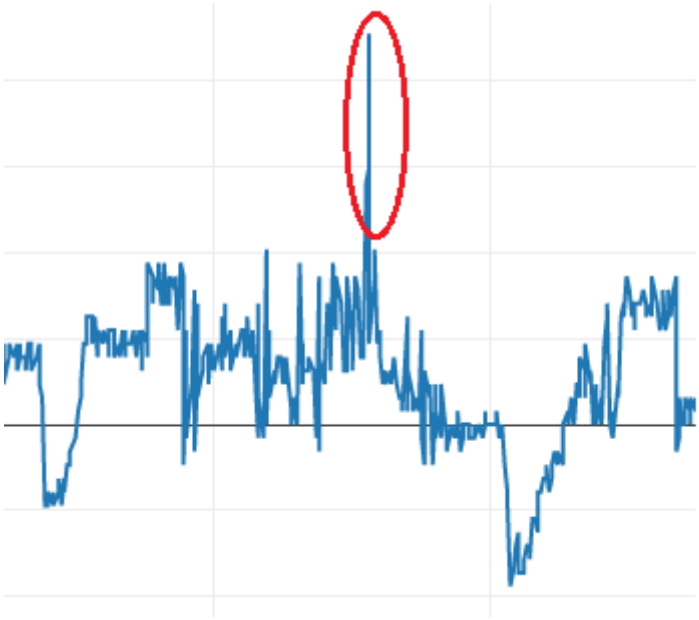

Fig. 8. Undetected Brake

III. *Audio Data*

A 50% reduction of audio data samples was implemented for 5 data samples by tuning the parameters of CURE, BIRCH and DBSCAN.

Parameters used for CURE

For CURE, the minimum representative count in each cluster was set to 15, the shrink factor was initialized to 0.01. Number of partitions was set to 20 and the reducing factor for each partition was set to 2. Using these parameters, the initial sample size has been computed.

Parameters used for BIRCH

For BIRCH, the maximum node entries were set to 13 and the threshold distance was set to 0.05.

Parameters used for DBSCAN

The Minpts. was set to 12 and the eps was set to 1000.

The execution time results for the three algorithms were no exception for the audio sensor. As can be seen in Fig. 9, BIRCH took the least execution time for all the data samples. CURE took significantly longer to execute, relative to other two algorithms.

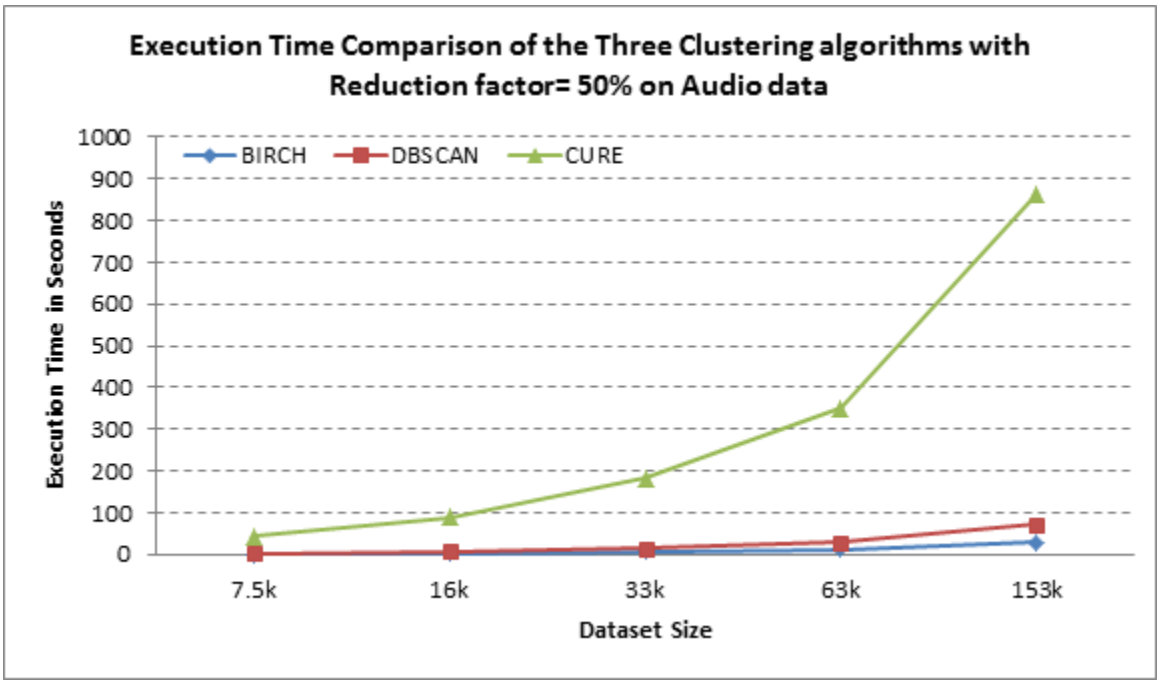

Fig. 9. Comparison of clustering algorithms based on program execution time

To evaluate the quality of representatives obtained by the three algorithms, the approach used is similar to the one used over accelerometer dataset. We have defined noise events as discussed in Data Collection Section. We declare a noise event to have occurred if the average amplitude, in a time window of 2 seconds, exceeds 4000 units. The number of such noise events in every data sample is described in Table III.

Fig. 10. shows the percentage of events detected at a 50% reduction using the representative clustering by the three algorithms. It was found that across all data samples and across all three algorithms, even at 50% data, 100% events could be detected. This outcome could be explained by the high sampling rate of the data collected. A noise event, as defined in Data Collection Section, is captured by as many as 4500 points. Hence, even at a 50% reduction the number representatives suffice to be able to detect the noise.

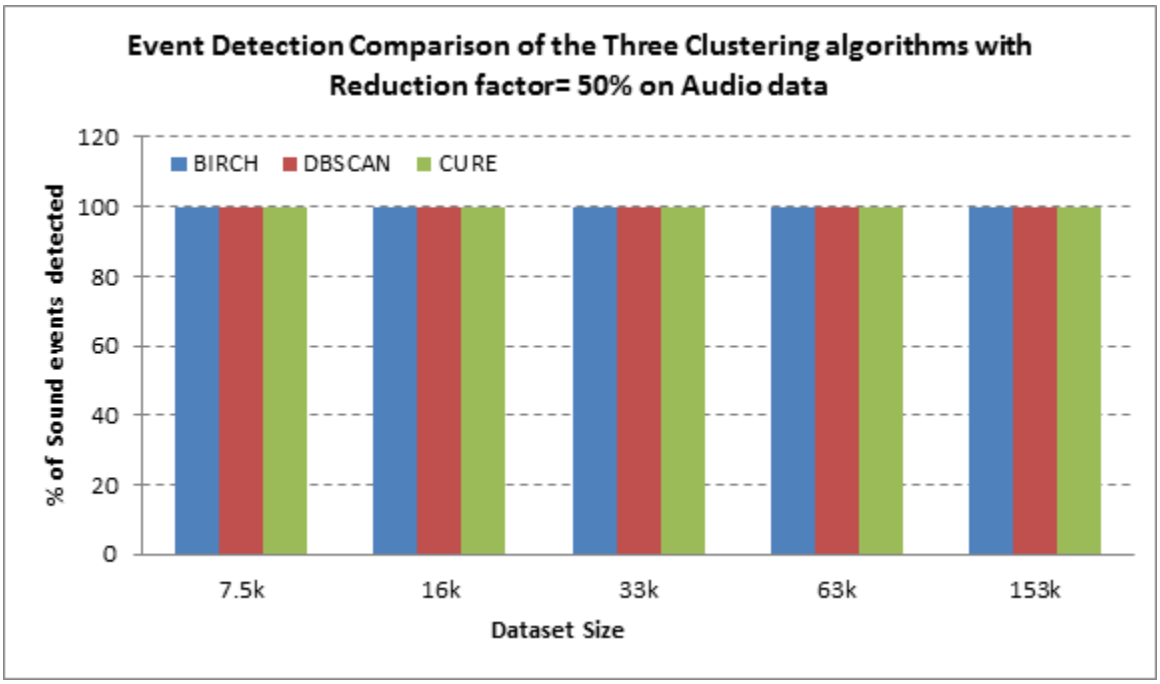

Fig. 10. Audio Accuracy

6. Evaluation

As discussed in the previous section, at a 50% reduction, BIRCH seems to be the most suitable clustering algorithm in terms of the execution time as well as the accuracy across all three sensors and data samples. Hence, for further evaluation we conduct all the other experiments on BIRCH algorithm.
This section discusses the performance of BIRCH at various other reduction percentages. The evaluation has been done based on three parameters- Execution time, accuracy on reduction and the energy consumption in execution. Further, this section evaluates how the overall study that we conducted is advantageous or disadvantageous when the reduced data is to be transmitted over various wireless technologies.
Unless stated explicitly in the following subsections, all of these evaluations have been conducted on Xperia Z3 device running Android v4.4.4 (KitKat). For energy and power monitoring, we have used an app called PowerTutor.

I. Performance of BIRCH on Different Devices

In order to evaluate how BIRCH performs on different devices with varied memory, CPU and other specifications we have implemented BIRCH on three devices- Xperia Ray, Moto G and Xperia Z3.
For every collected data sample, as described in Dataset 2 for GPS, we implemented BIRCH so as to reduce the data to 50%. All the other apps running in the background were killed beforehand. Fig. 11. shows how the algorithm performs on the basis of execution time for GPS data samples on the three devices. Xperia Z3 with a 3GB RAM took the least time in all cases, followed by MotoG(1GB RAM) and Xperia Ray(512MB RAM). With the RAM size and processing speed improving with the advancement of smartphone technology the execution time ranges from 2 seconds to reduce a data size of 5021 points to 50% to at most 21 seconds to reduce a data size of 81216 points to 50% as observed on Xperia Z3, averaged over 20 iterations.

A similar evaluation was carried out for accelerometer and audio datasets as well. A similar trend was observed as can been in Fig. 12 and Fig. 13. Note that, for an audio data sample size as large as 153773 points, BIRCH took 30 seconds to cluster and reduce it to 50% representatives.

Clearly, the accuracy of data reduction is not dependent on the device type and hence the accuracy observed was same across all devices for each data sample. The accuracy obtained has already been discussed before as depicted in Fig. 4, Fig. 7 and Fig. 10.

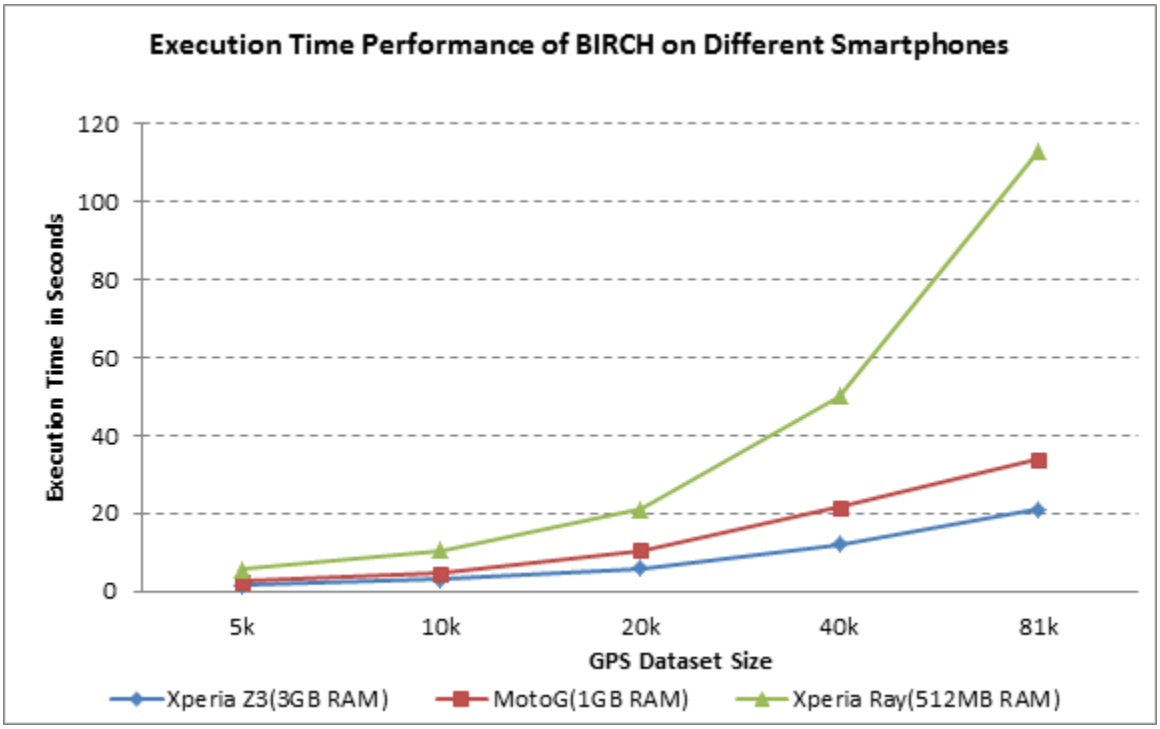

Fig. 11. Execution Time of BIRCH on GPS data over different devices

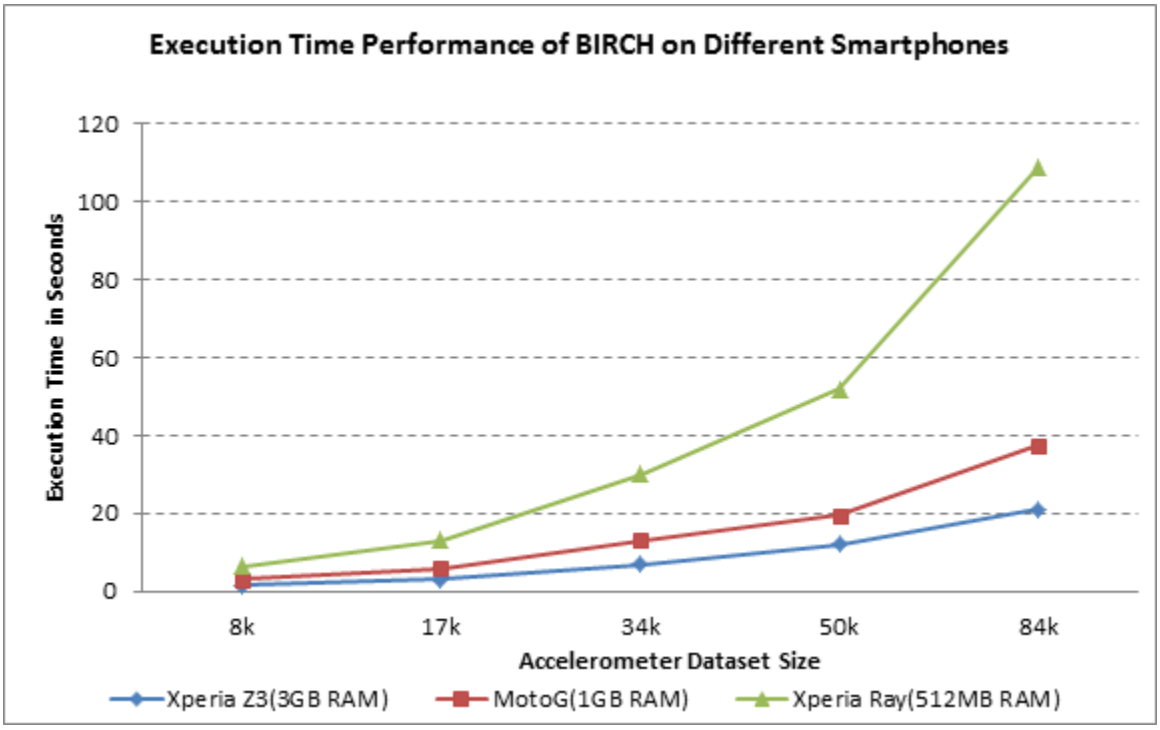

Fig. 12. Execution Time of BIRCH on Accelerometer data over different devices

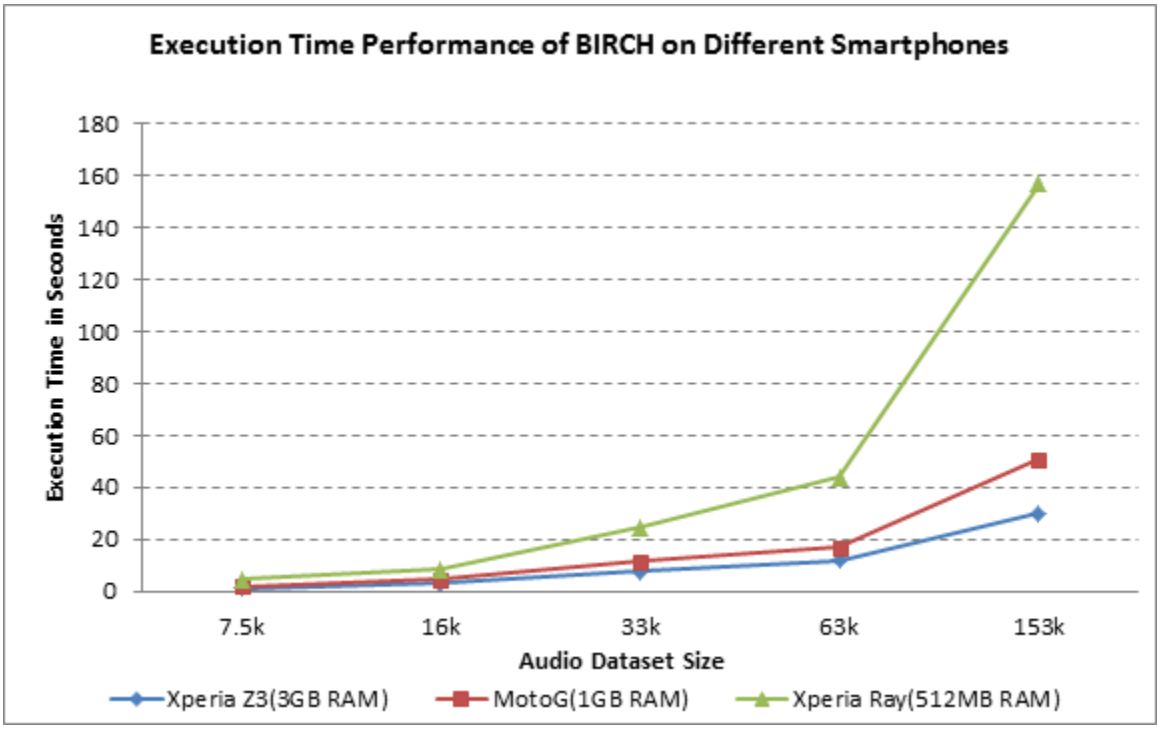

Fig. 13. Execution Time of BIRCH on Audio data over different devices

II. Energy Consumption in Executing BIRCH

Now we move on to the evaluation of BIRCH based on its energy consumption when implemented on different data sample sizes. The energy consumption has been evaluated on Xperia Z3. The BIRCH parameters have been tuned to reduce the data to 50% of data points. While running the experiment, all the other apps, except for PowerTutor, have been killed. Fig. 14, Fig. 15 and Fig. 16 show the energy consumption of BIRCH across different data samples for GPS, accelerometer and audio sensors respectively.

The energy consumption ranges from 1 Joule to around 16 Joules for different samples of GPS and accelerometer data, increasing with the size of data sample. In case of audio data samples the trend is the same. The largest audio data sample of around 153773 points consumes 23.5 Joules of energy to be reduced to 50% representatives by BIRCH clustering.

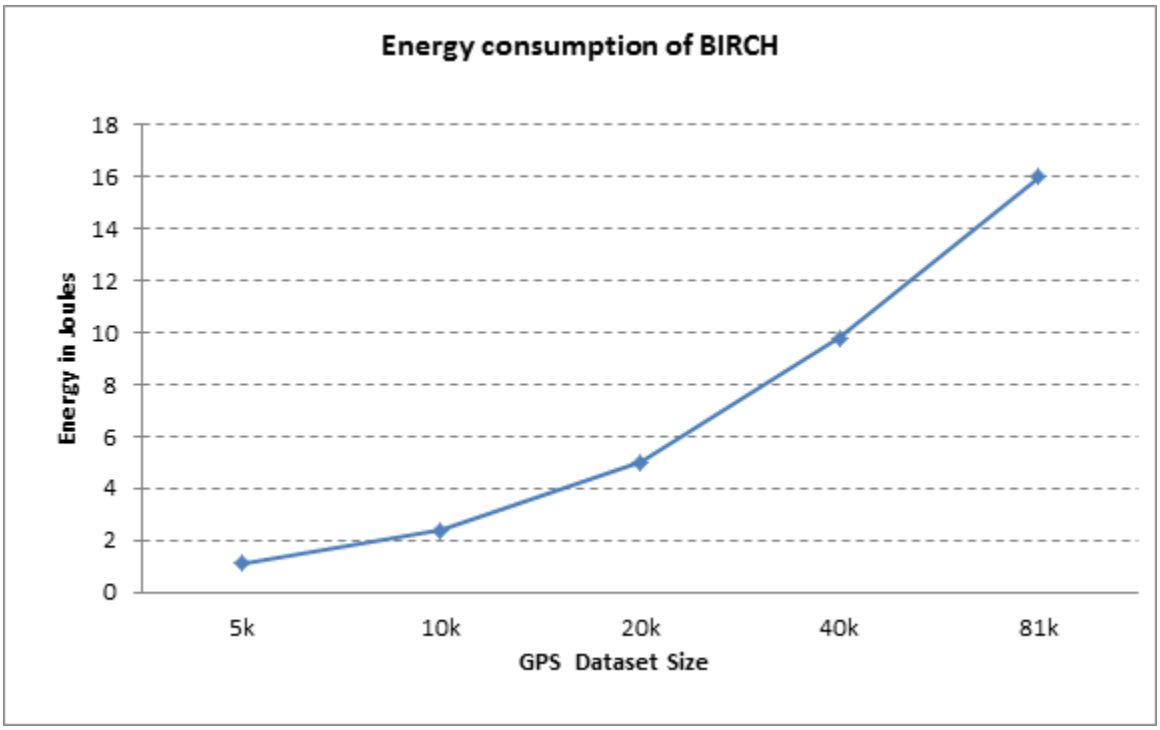

Fig. 14. Energy consumed in executing BIRCH on different data sizes (GPS data)

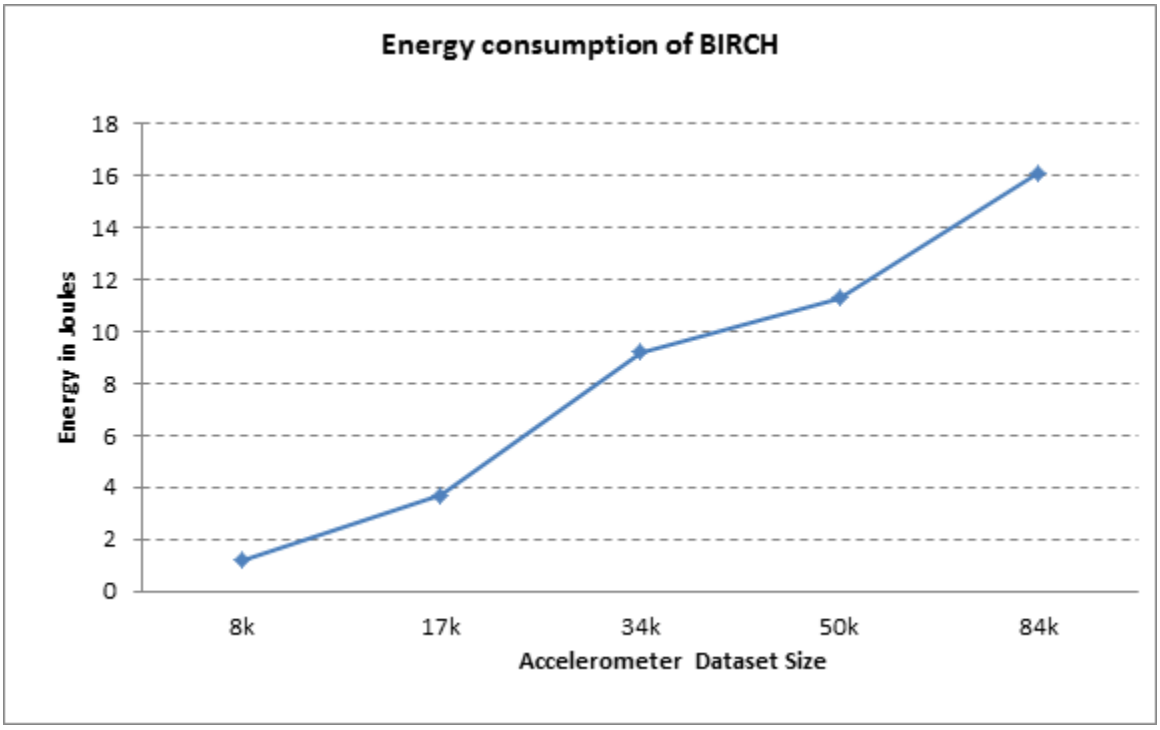

Fig. 15. Energy consumed in executing BIRCH on different data sizes (Accelerometer data)

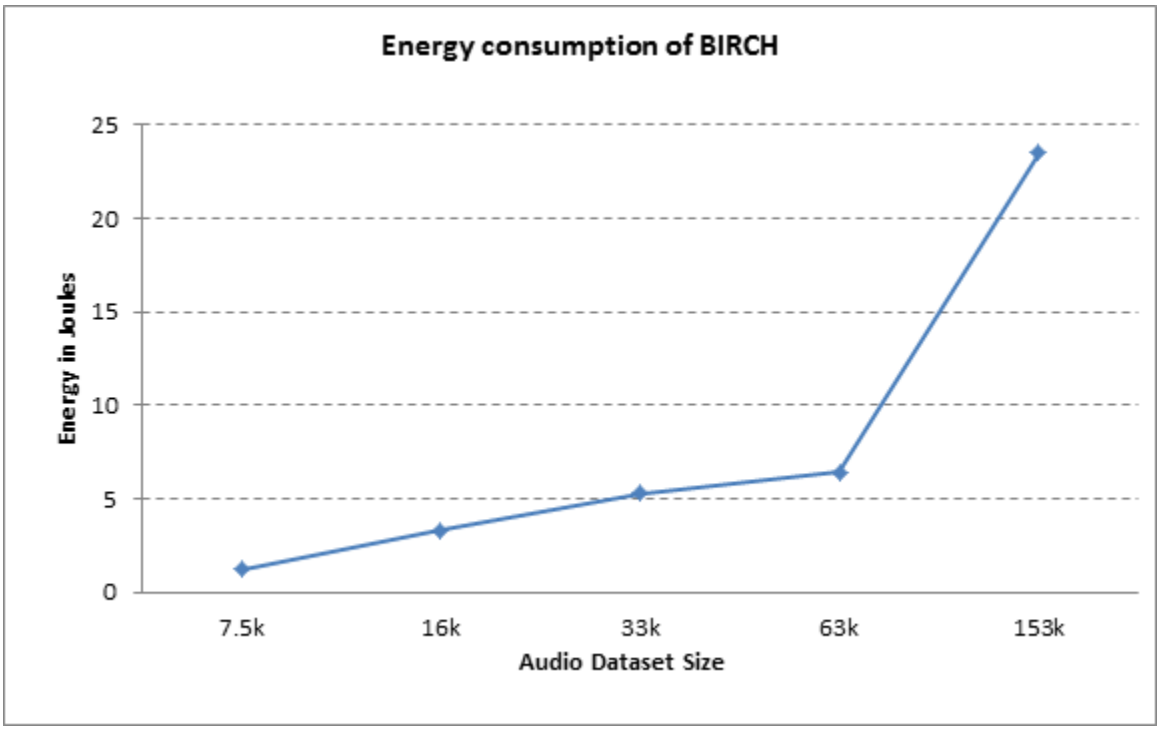

Fig. 16. Energy consumed in executing BIRCH on different data sizes (Audio data)

III. Performance of BIRCH at other Reduction Percentages

So far, our experiments have concentrated on reducing the data samples to 50% size. What if reducing it further would still give acceptable accuracy with lesser energy consumption and execution time? Else, if accuracy is the critical requirement of an application, reducing it lesser(say to 75% instead of 50%) may give better results. We shall discuss how these factors are affected with different reduction percentage.

We applied BIRCH on one of the GPS data samples, which is of size 81216 points. This was run on Xperia Z3. 20 such experiments were conducted for each reduction percentage and the averaged values have been provided in Table VI. As the reduction percentage increases, the execution time increases, though insignificantly. The execution energy also increases with the percentage increase and the localization inaccuracy decreases. Even at 25% data, the inaccuracy is of about 12.66m, which is a moderate value if the application does not require a very high accuracy.

Reduction to	Execution Time	Inaccuracy	Execution Energy
25%	18sec	12.66m	13.36J
50%	21sec	5.08m	16.56J
75%	24sec	2.6m	20.2J

Table VI: Performance of BIRCH on GPS data at different reduction percentage

The same experiment was conducted for accelerometer and audio data samples of size 84940 and 153773 points respectively. The results in Table VII and Table VIII show the same trend as for GPS data. The accuracy obtained is within the acceptable limits even at 25% data where it could detect 92.1% of the brake events in the resultant accelerometer data and 100% noise events in the resultant audio data.

Reduction to	Execution Time	Accuracy(Brakes detected)	Execution Energy
25%	19sec	94(of 102)	13.7J
50%	20sec	98(of 102)	17J
75%	23.5sec	102(of 102)	19.2J

Table VII: Performance of BIRCH on Accelerometer data at different reduction percentage

Reduction to	Execution Time	Accuracy(Noise Events detected)	Execution Energy
25%	26sec	12(of 12)	20.26J
50%	28.5sec	12(of 12)	22J
75%	40sec	12(of 12)	31.2J

Table VIII: Performance of BIRCH on Audio data at different reduction percentage

IV. Energy Consumption in Data Transmission

Now we move on to the most important evaluation. Our objective to carry out this work was to conserve smartphone energy that is consumed in the process of sensor data collection and transmission. For optimizing energy consumption in sensor data collection we have discussed the duty cycling approach before. Now let us study how the energy is conserved in data transmission. Network data transmission, especially in 2G and 3G network, drains a lot of the device's energy. The energy that we target to conserve here is the data transfer energy.

To carry out the evaluation of transfer energy, we use the smartphone device to upload the data files and mail them. These data files are uploaded over four types of wireless technologies- Wifi, EDGE, UMTS and HSPA+. Note that, over Wifi, we do not include the energy consumed in scanning and associating with the access point in our analysis.

There are two categories of data that we have taken for each sensor- the original data sample and the resultant data sample after 50% reduction by BIRCH. There are 5 data samples taken for each of the sensor. These samples have been described in the data collection section as datasets obtained without duty cycling.

Fig. 17, Fig. 18 and Fig. 19 show the total energy consumption in transferring the data over the four wireless networks. The figures show three types of energy consumption-

1. Energy consumed in transferring the original data,
2. Energy consumed in transferring the reduced data,
3. Energy consumed in BIRCH execution to reduce the data + Energy consumed in transferring the reduced data

It is the (3) energy consumption that we ideally want to be lesser than the (1). A positive difference between (1) and (3) indicates that the clustering based approach to reduce the data has benefitted in saving some energy.
The figures show that in case of data transfer over EDGE and UMTS technology, we save some overall energy. The amount of energy conserved increases as the data sample size grows. However, in case of data transfer over Wifi and HSPA+, we do not save any overall energy.

Hence, applying a clustering based approach to reduce the data is of use if the medium of data transfer is over EDGE or UMTS. In EDGE and UMTS the transfer energy makes up for a significant fraction of the overall energy[3]. The amount of transfer energy consumed clearly depends upon the bytes of data to be transmitted. Hence, by reducing the data size, a significant fraction of energy can be saved. In case of Wifi and HSPA+, the transfer energy does contribute to the overall energy consumed but the fraction it makes up is insignificant as compared to the energy that goes into associating with the access point or in tail energy or ramp energy.

A lot of applications that make use of the three sensors(GPS, Accelerometer and Microphone) are used in places where Wifi network may not be available. For instance, navigation based or

location based apps make use of GPS sensor and the data collection and transmission is mostly done while travelling where a Wifi network is usually not available. Under such circumstances the app would make use of the cellular network technologies. Hence, for such apps that would use EDGE or UMTS as the wireless network for data transfer, especially when the data size is large, our work would help conserve some energy at a minor cost of accuracy.

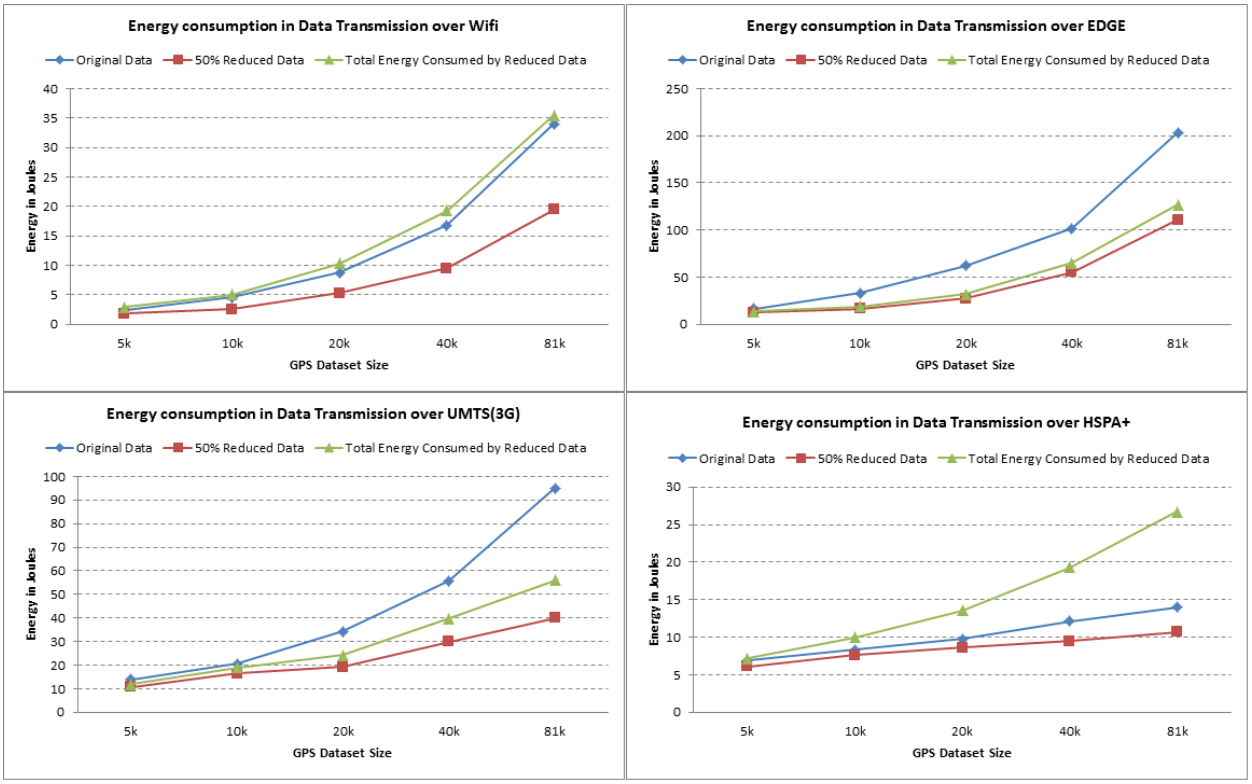

Fig. 17. Energy Consumption in GPS data transmission over different wireless technologies. Energy saved/lost by applying data reduction

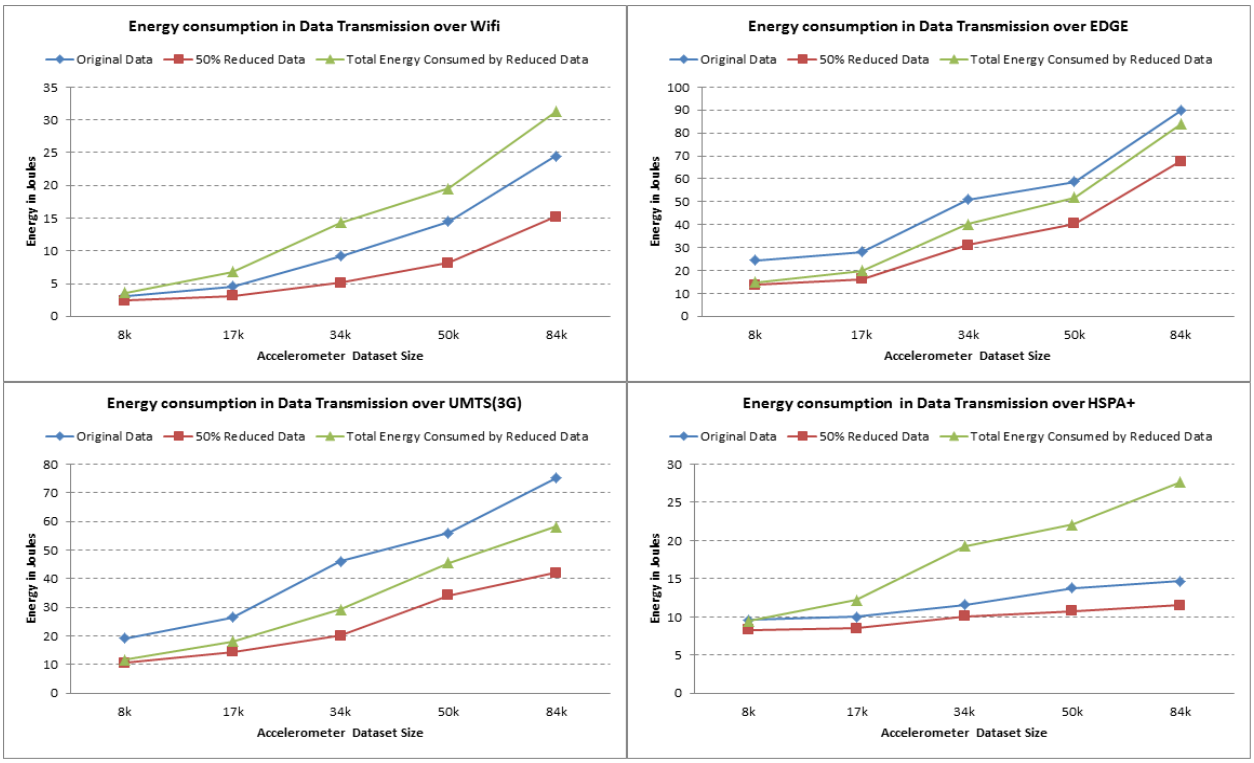

Fig. 18. Energy Consumption in Accelerometer data transmission over different wireless technologies. Energy saved/lost by applying data reduction

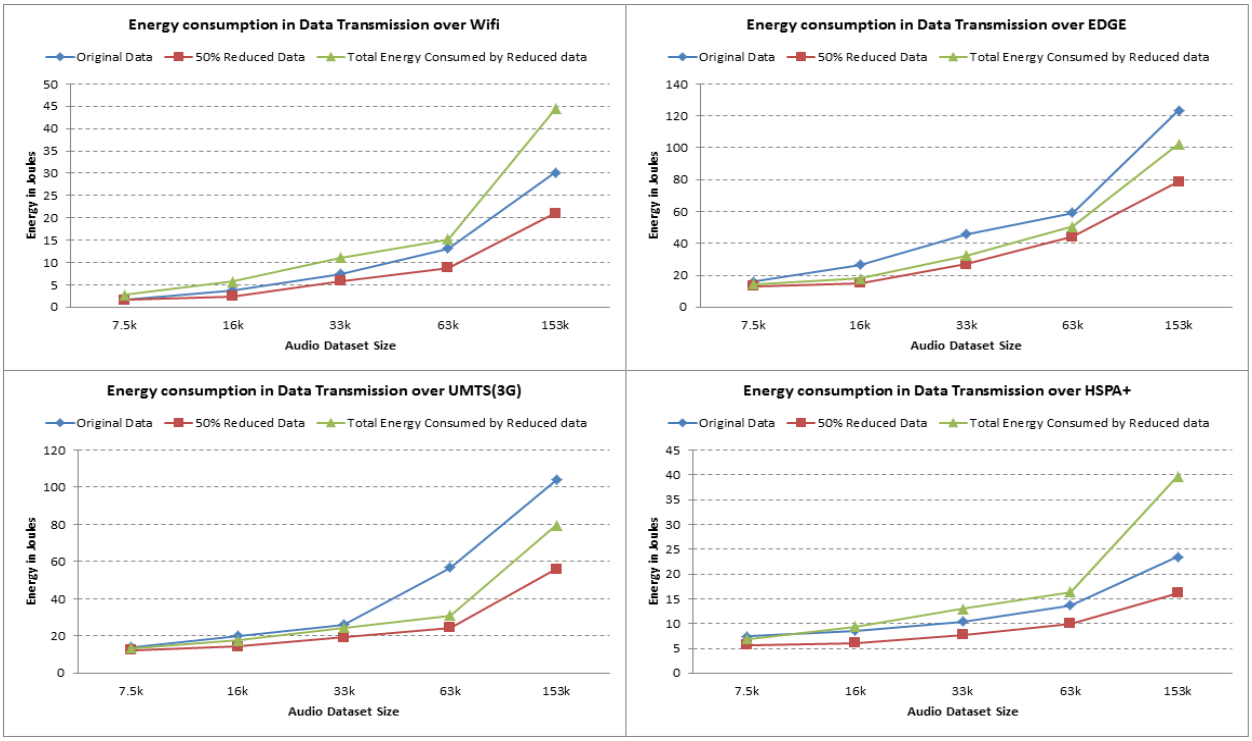

Fig. 19. Energy Consumption in Audio data transmission over different wireless technologies. Energy saved/lost by applying data reduction

Conclusion

To reduce the energy consumption due to sensor data, measures can be taken at two stages. Either the energy consumption could be restricted in the process of sensor data collection. This may or may not work by applying duty cycling on polling the sensors as we have already discussed.

The second stage where the energy consumption can be controlled is while transmitting the collected sensor data over the wireless network. This is the stage where we have worked intensively and made use of the knowledge that the data file size is correlated to the energy consumed in transmission. Hence, reducing the file size would reduce the energy consumption. We evaluated the performance of three clustering algorithms- CURE, BIRCH and DBSCAN when applied on GPS, Accelerometer and Microphone sensor data. From the clusters obtained, we obtained the representatives of the dataset. BIRCH gave the best results in terms of execution time and accuracy of the obtained representatives. Further we concentrated on BIRCH to evaluate the energy saved in executing the data reduction and transmitting the representative data over various wireless networks- Wifi, EDGE,UMTS, HSPA+. A decline in total energy consumption (in BIRCH execution + data transmission) was observed for representative data over EDGE and UMTS data transmission. For all three sensors, the same trend was observed. Hence, using our approach would be beneficial in conserving transmission energy if the data is to be transmitted over these two wireless technologies, especially in case of large data size.

References

1. Yong Cui, Shihan Xiao, Xin Wang , Minming Li, Hongyi Wang and Zeqi Lai. “Performance-aware Energy Optimization on Mobile Devices in Cellular Network”.
2. Mirco Musolesi, Mattia Piraccini, Kristof Fodor, Antonio Corradi, and Andrew T. Campbell. “Supporting Energy-Efficient Uploading Strategies for Continuous Sensing Applications on Mobile Phones”.
3. Niranjan Balasubramanian, Aruna Balasubramanian, Arun Venkataramani. “Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications”.
4. Sinziana Mazilu, Ulf Blanke, Alberto Calatroni, and Gerhard Troster. “Low Power Ambient Sensing In Smartphones for Continuous Semantic Localization”.
5. Ahmad Rahmati and Lin Zhong. “Context-for-Wireless: Context- Sensitive Energy-Efficient Wireless Data Transfer”.
6. Ralf Herrmann, Piero Zappi, Tajana Simunic Rosing. “Context Aware Power Management of Mobile Systems for Sensing Applications”.
7. Suman Nath. “ACE: Exploiting Correlation for Energy-Efficient and Continuous Context Sensing”.
8. Kavi Khedo, Rubeena Doomun, Sonum Aucharuz. “READA: Redundancy Elimination for Accurate Data Aggregation in Wireless Sensor Networks”.
9. C.Tharini and P. Vanaja Ranjan. “An Energy Efficient Spatial Correlation Based Data Gathering Algorithm For Wireless Sensor Networks”.
10. Ruitao Xie and Xiaohua Jia. “Transmission-Efficient Clustering Method for Wireless Sensor Networks Using Compressive Sensing”.
11. Malika Bendechache, M-Tahar Kechadi, Chong Cheng Chen. “Distributed Clustering Algorithm for Spatial Data Mining”.
12. Sudipto Guha, Rajeev Rastogi and Kyuseok Shim, “CURE: An Efficient Clustering Algorithm for Large Databases”.
13. Tian Zhang, Raghu Ramakrishnan and Miron Livny. “BIRCH: A New Data Clustering Algorithm and Its Applications”.
14. Martin Ester, Hans-Peter Kriegel, Jorg Sander and Xiaowei Xu. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”.
15. Prashanth Mohan, Venkata N. Padmanabhan and Ramachandran Ramjee. “Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones”

INDRAPRASTHA INSTITUTE of INFORMATION TECHNOLOGY DELHI

*IIIT-Delhi was created as a State University by an act of Delhi Government (**The IIIT Delhi Act, 2007**) empowering it to do research and development and grant degrees.*

Devika Sondhi

DOB: September 15th, 1992
email: devika10028@iiitd.ac.in, Mob: 9818489219

Education

2010-present (2015 pass out)	IIIT-Delhi Dual Degree(B.Tech + M.Tech) (CSE), M.Tech specialization- Mobile Computing B.Tech CPI 8.15/10 M.Tech CPI 9.2/10
2009-10	Loreto Convent School, Delhi Cantt. CBSE- Class 12th 90% (aggregate)
2007-08	Loreto Convent School, Delhi Cantt. CBSE- Class 10th 88.8% (aggregate)

Skills

Expertise Area	Developing Mobile Applications (Android platform), Application Development, REST based Applications, Developing Cloud Services, Issue Tracking/Bug Testing
Programming Language	C, Java, Python, SQL, PHP, HTML, Android Programming
Tools and Technologies	Matlab, Wireshark, Eclipse, MySQL, Adobe Dreamweaver, Java ME SDK, TeXworks, Xilinx
Technical Electives	**Mobile Computing:** Mobile Computing, Ad Hoc Wireless Networks, Advance Mobile Computing, Distributed System Security, Software Engineering, Programming Cloud Services for Mobile Application, Cellular Data Networks, Program Optimization **Information Security:** Digital and Cyber Forensics, Foundations in Computer Security, Network Security, Privacy and Security in Online Social Media **Data Engineering:** Data Mining, Database System Implementation

Internship

Research	Privacy in Location Based Services Guide: Dr. Vikram Goel, IIITD	(June,2011- July, 2011) Team Size -2
	Learnt existing algorithms to protect the user location privacy for location-based services in the Euclidean space and implemented algorithms in **C language.**	
Research/Industrial	At Tata Consultancy Services, Delhi, Head Office	(June-July 2013)
	Developed an Order Flow Tracking Web-Portal in .NET Framework. Demonstrated over video conferencing at all India level.	

Projects

1. **M.Tech Thesis** (August,2014-June,2015)
   Guide: Dr. Pushpendra Singh

   Topic: Optimizing smartphone energy consumption in sensor data collection and data transmission.

2. **Security in Adhoc Network (as B.Tech Project)** (Sept,2013-Apr, 2014)
   Guide: Dr. Pushpendra Singh Team Size-2

   Implemented Black Hole, Flooding and Impersonation attacks on real systems in an adhoc network. Analyzed the effects of these attacks on throughput, packet loss and packet delay. Further, developed real time attack detection mechanisms for these attacks

3. **Personal Translator** (Feb-Apr, 2015)
   Guide: Dr. Pushpendra Singh Team Size-2

   Implemented a chat application on Android platform that supports user's preferred language. It also supports speech to text and text to speech conversion so the user can speak up the message to send and also listen to the received messages. The cloud between the clients end stores user's preferences and uses language translation.

4. **Medical Report Tracker** over smartphone (Android)
   Guide: Dr. Pushpendra Singh (Feb-Apr, 2014)
   Team Size-2

   Performs OCR on clicked images of physical reports and maintains medical history of a person at the server end. The history can be downloaded any time as per user's requirement.

5. Developed **Android Application to Detect Fake Documents**(for v2.2 and above) (Feb-Apr, 2014)
   Guide: Dr. Gaurav Gupta Team Size-2

DocCheck app makes use of QR code reader and OCR to check if a document is fake or genuine

6. Developed **Mobile Application on Android Platform**(for v2.2 and above) (Sept – Nov,2012)
   Team Size-2
   Guide: Dr. Vinayak Naik

   Photo Editor, 'FunFoto N Frames', that allows the user to click pictures with a set timer and set a selected fancy frame to it at the time of clicking the picture. The app also provides an option to share these pictures on Facebook and Twitter.

Awards and Achievements

Successfully taken Microsoft Technology Associate (MTA) Certification Exam
Certificate of All Round Excellence in Class 10th and 12th

Interests and Hobbies

Have provided Community Services by teaching a group from underprivileged children.
Playing Piano, Guitar
Discovering Latest Apps and Gadgets

Declaration: The above information is correct to the best of my knowledge.

Devika Sondhi (June, 15th, 2015)