NoiseSense: Crowdsourced Context Aware Sensing for Real time Noise Pollution Monitoring of the City

Joy Dutta
Department of Computer Science & Engineering, Jadavpur University
Kolkata-700032, India
joy.dutta.in@ieee.org

Pradip Pramanick
TCS Innovation Lab
Kolkata- 700156, India
pradip.pramanick@tcs.com

Sarbani Roy
Department of Computer Science & Engineering, Jadavpur University
Kolkata-700032, India
sarbani.roy@ieee.org

Abstract—**Noise pollution in urban areas is a subject of grave concern and it is being recognized globally in different countries and cities. People are facing many health-related problems because of this. Therefore, in the proposed work, we envisioned to tackle the challenge of acquiring real time and spatially fine-grained noise pollution data with a community-driven sensing infrastructure. Mobile crowdsourcing over smartphones presents a new paradigm for collecting context aware sensing data of a vast area like a city. Thus, the proposed system exploits the power of mobile crowdsourcing. The proposed system monitors the present noise level in the surroundings of the user and also generates city's noise pollution footprints. The noise map reflects the real-time pollution scenario of the city which changes with time. The prototype of the system has been evaluated with extensive experiments based on crowdsourced sensing data collected by volunteers in Kolkata city.**

Keywords— ***crowd sourcing, context aware, energy efficient, noise pollution, Cloud, Smartphone, Smart City***

I. Introduction

Noise pollution is a big concern for all of us. Traffic junctions, transport terminals, factories and industrial installations have considerable impact on the local noise pollution. People are getting various kinds of auditory and non-auditory health problems, including different types of heart related problems. Because of the side effect of this pollution, average sleep time is decreasing as a result affects human health. For our case study, we have chosen Kolkata, the city of joy for monitoring noise. In a recent study in India, Kolkata, once the country's model city for noise control is now a top offender. In Kolkata, a report [1] from the Central Pollution Control Board, which measured noise levels using eight automatic noise measurement stations during the period April 10 to 16, 2017 shows that noise levels are way above the limits in both day and night times (as shown in Fig.1). The troubling fact is the silence zones near hospitals shows much more noise than specified limits and some key commercial and residential areas show very high noise levels 70 to 76 decibels. In the traditional approach, noise pollution can be monitored by networks of static measurement stations operated by official authorities. These stations are highly reliable and can accurately measure a wide range of air pollutants and different levels of noise. Although the extensive cost of acquiring and operating these stations severely limits the number of installations and results in a limited spatial resolution of the published pollution maps.

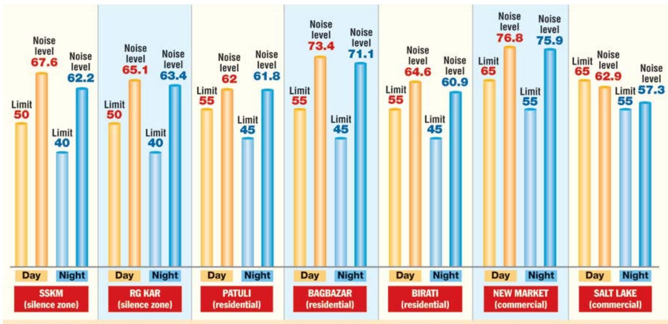

Fig 1: Noise monitoring in different parts of the Kolkata city by Central Pollution Control Board, India [1]

However, such infrastructure is neither cost-effective nor very scalable from deployment and maintenance point of view. On the other hand, mobile devices such as smartphones and tablets are increasingly becoming an essential part of human life as the most effective and convenient communication tools these days. Thus, using participatory sensing with mobile phone as a sensor is a cost effective, scalable way to do the real-time noise pollution monitoring. Moreover, participatory sensing involves people, which can play a key role in raising public awareness. Given the broad availability of personal GPS equipped smartphones, we aim to involve citizens in monitoring noise pollution using their smartphones for participatory sensing. To get a city pollution footprint, we need to collect huge streaming noise data in the city using participatory sensing. GPS of the phone is used to get the spatial information, which is a prime source of battery consumption. The contributions of this paper are twofold:

i. Crowdsourced context aware sensing based system called NoiseSense is proposed to monitor the present noise level in the surroundings of the user as well as to generate city's noise pollution footprints in real time.
ii. An intelligent grouping technique is developed for both power optimization and error reduction. Since the application is a crowd sourced, grouping of people can be utilized when it comes to power usage optimization. Moreover, this approach helps in reducing error and monitors the noise more accurately.

The structure of the work is presented here as follows. Section II reviews some of the representative works on the noise monitoring system along with their limitations. This section also highlights the motivation and contributions of our work. The architecture of the proposed system is discussed in Section III and the design of the system is elaborated in Section IV. Section V contains the implementation details. Finally, the proposed system is evaluated in Section VI and concluded in Section VII.

II. Related Work

There has been several works on monitoring ambient noise. In EarPhone [2] the authors present an end-to-end noise pollution mapping system which uses Microphone, GPS receiver and System Clock in a mobile. They have shown differences in noise levels when the phone was carried in hand, bag, pocket and waist. They have also profiled CPU, RAM and Power usage for running the application. However, the proposed system was for Nokia N95 using the J2ME platform which is obsolete now. Also the system does not have any calibration capability for generic mobile phone models. Whereas, NoiseSPY [3] is a real time environmental noise monitoring system which is implemented using the Symbian C++ platform targeted to Nokia series 60 devices. They have conducted experiments and shown real time noise maps in the city of Cambridge. In NoiseTube [4], open source noise sensing application for modern operating systems, Android and iOS has been developed. The mobile application can sense the noise level in dB(A) and users can tag the measurements with the sound source, level of annoyance. Their calibration technique uses a number of device profile manually calibrated and stored in the server. When a user using a known device uses the application the calibration profile is downloaded, and the device is automatically calibrated. However, the number of device profile stored in their server is relatively small compared to the huge number of devices by device manufacturers in present day. The need for auto-calibration of any generic devices is still needed. In DDNM [5] the authors have implemented a similar system inspired by the above and they have added the functionality of downloading maps through WiFi for offline usage. But, in WideNoise [6] system is focused on awareness and learning of participatory noise sensing. In this system the users are helping to better understand the noise levels intuitively by allowing the users to guess the current noise level in his vicinity and then compare the actual measurements. The system also allows the user to tag their current feelings about the noise.

Limitations

As mentioned above, pollution monitoring in urban areas is usually carried out using networks of fixed and deployable sensors or monitoring stations at a sparse set of locations e.g. traffic junctions and industrial estates. However, this practice has a number of limitations:

i. The cost of traditional pollution monitoring unit is high due to the need of expertise and human resources, the deployment of expensive equipments and the processing effort.
ii. Data collection at sparse locations in a city hardly scales to meet the requirements of data with high granularity in both time and space.
iii. Unable to measure an individual's pollution footprint and hence its effect on health of the citizens. Only, effective to view the effects on a macro scale.
iv. Indoor pollution level in a building, especially in workplaces cannot be measured. But, most people spend a significant portion of their time indoors.

Motivation and Contributions

So, it is evident that there are several limitations of the currently proposed systems which can be improved for better performance as well as for user experience. Hence, the primary objective of the proposed work is to design and implement a system which tries to overcome the above mentioned issues. The idea is to employ the participatory sensing in real time pollution monitoring. Usually common people do not take part in the traditional approach of pollution monitoring and their individual pollution exposure is never measured. Probably, for this reason citizens remain unaware of how much noise pollution they are being exposed to and what harm can it do. The proposed system NoiseSense will encourage the citizens to participate in a crowd-sensing initiative, which could be a backbone of any smart city [11]. Thus, using participatory sensing with mobile phone is a cost effective, scalable way to do noise pollution monitoring in both indoor and outdoor environments at macro and micro level [12][13]. In the proposed approach we have also focused on making the noise sensing more accurate, device independent, context-aware and energy efficient usage of GPS. Actually, there is no provision for automatic calibration and context-aware sensing in the currently available systems. In NoiseSensee, we have incorporated these two things in our work for better performance. Also, in the related systems little work has been done on power optimization in crowd sensing environment [14], removing data redundancy and error correction. In this work, we present an effective technique to do that by dynamically creating local group, leader election and rotation of the leader. As shown in EarPhone [2] and NoiseSpy [3], GPS is the prime source of power consumption. Our proposed scheme greatly reduces it by balancing GPS usage in a group of users. Also, we have used here a generic calibration technique [10] for sensing noise accurately independent of the device.

III. Architecture

The architecture of NoiseSense is depicted in Fig. 2, which consists of three layers, namely crowd layer, local processing layer and the cloud layer.

Crowd Layer

The bottom layer is named as crowd layer, which is used for collecting the data in participatory sensing mode. Crowd layer utilizes mobile phones to offer unprecedented observational capacity at the scale of the individual. In the proposed approach, the crowd can be accessed either in the group or in single user mode. Here, we are assuming that the users are using android device where NoiseSense app is running in the background. In the single user mode, the device is standalone and directly sends the data to the cloud. As a result, the participatory device may drain its energy rapidly because of the extensive GPS usage for getting the spatial information. However, to resolve this, [14] can be followed for energy saving which is a proven solution in this case. But, when the users are traveling in a group or in a crowded place where users can form a group (free WiFi zone in public places/offices etc.), there to avoid energy draining, along with [14], our collaborative approach can be used, i.e., group mode for further energy optimization. Here, we have evaluated the system using our proposed group mode algorithm (discussed in Section IV), elected the leader, and the collected data is sent to the cloud server by the leader.

Local processing layer

The smartphones come with different sets of sensors, however reading from two smartphones differs even when both senses the same environment. Thus, pre-processing is essential for data gathered by the smartphone before pushing it to the cloud. Preprocessing includes duplicate removal and error correction. Reliability of data gathered is very important as far as correct service delivery is considered. Sensing using mobile phone is power consuming and as a result user may lose interest in taking active participation in sensing activities. In this context, collaborative approach can play a significant role. As mentioned earlier, the system can operate in both single user mode and group mode. In group mode, beside pre-processing steps there is leader election phase, which runs periodically. The main idea is to form a group of devices in the vicinity and deciding a leader, which can collect all the data from the group members. In this mode, the leader will take the burden of preprocessing of the collected data and will send that data to cloud for further analysis.

The group mode approach can save energy consumption significantly. In [7] it has been shown that GPS is the prime source of power consumption in any location based applications. In NoiseSense, users who are staying in close proximity, can form a group using a local wireless (WiFi) network or hotspot, like in offices, schools, other organizations, railway stations, bus stand, corporate sectors, office buildings, public transports and other social gathering places where people tend to spend some time together. So, in such cases group members can avoid GPS for location sensing. The location of the leader will be treated as the location of group members.

Moreover, here group members will send their sensed data directly to the leader, therefore data transfer will also be cost effective (from the energy consumption point of view). Most of the existing solutions to smart city scenario, assume that the connectivity to the Internet is always-on and provides high bandwidth. But from Indian perspective, we generally have an intermittent connection to the Internet and quite often network bandwidth is poor than demanded. Due to high population density in urban areas often the peak demand of bandwidth is very high and often infeasible. According to [8], the energy consumption characteristic of a phone is closely related to the nature of the workload. That means, energy consumption is not only depends on the total transfer data size. For example, transferring a few hundred bytes data intermittently on 3G/4G can consume more energy than transferring a megabyte in one attempt. It is also reported in [8] that 3G and GSM incur a high tail energy overhead because the phone stay in high power states after the completion of a transfer. On the other hand, WiFi on phones typically uses the power save mode and thus consumes comparatively less energy. In the proposed approach, WiFi is used to transfer data in the group, thus the energy consumed here due to data transfer is proportional to the size of the data transfer and the transmit power level, which is significantly less than 3G/4G.

Cloud Layer:

It is quite clear that the proposed participatory sensing based application has high potential to gather streaming data of noise levels using smartphones which can actually be beneficial for the citizens. Thus, our requirement here is to provide the storage and processing facilities to all these real time and large volume of data. As shown in Fig. 2, cloud is used for both data storage and processing.

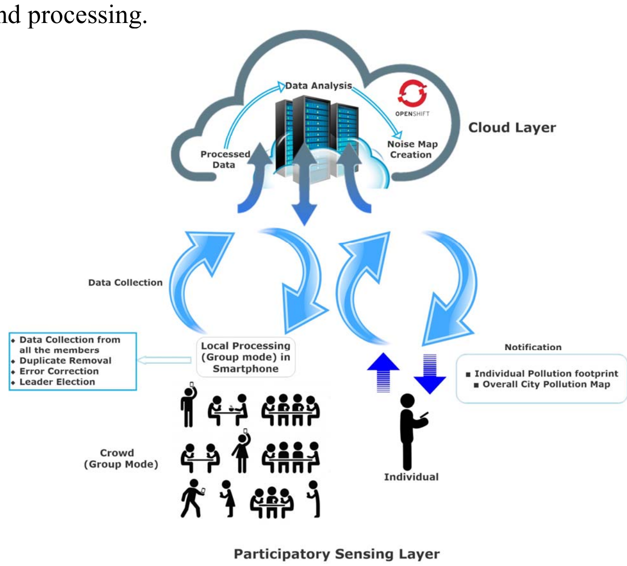

Fig. 2: NoiseSense system architecture

The proposed NoiseSense system has specific access demands, some of them are highlighted here: i) fast query responsiveness i.e., millisecond latency; ii) random access to indexed subsets of data; iii) real time expressive ad-hoc queries and aggregations against the data; iv) updating fast-changing data in real time as users interact with online applications, without having to rewrite the entire data set. Here, we have used MongoDB to store data. However, for fast response, partial data is stored in mobile devices or smartphones. To reduce delay in the service delivery, processing (usually not very high computation) can be done on mobile devices. For instance, in group mode, the leader node collects data and performs initial processing before sending it to the cloud. After processing the crowd data by performing the necessary aggregation and statistical model fitting in cloud, NoiseSense generates user's exposure to noise as well as a noise heat map of the city.

IV. SYSTEM DESIGN

This section presents the design of NoiseSense. For sensing the ambient noise, the device that we are utilizing here handheld smartphone and collecting data in the way as shown in Fig. 3. Microphone of the smartphone is used for ambient noise monitoring. The mobile application component of the NoiseSense running in the smartphone records the sound signal using the microphone of the phone and then generates the decibel values of the sensed noise.

Along with this data, the system adds a time stamp from the phone clock and the location information from the GPS subsystem. After collecting the ambient noise data along with timestamps and GPS location, the data is either forwarded to cloud directly if they are not in group mode, otherwise collected data is gathered in the group leader's mobile (following our proposed algorithm), which in turn updates the cloud server.

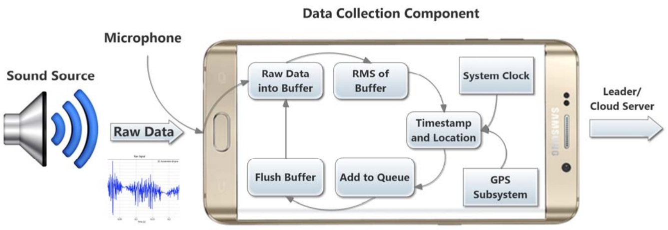

Fig. 3: Participatory data collection layer

To make crowdsensing context-aware, the data collection component runs with some constraints. Data collection stops when any of the following conditions is valid and restarts when all of the two are invalid: a) proximity sensor reports a close proximity value; b) the user is using the mobile phone for a call. The proximity sensor is present nowadays in most of the smartphones, which can be used to detect whether the phone is out in the environment or inside a pocket, bag, etc. It makes sense because it is reasonable to collect only environmental noise data only when the mobile phone is actually out in the environment. Similarly, when user is using the smart phone for a call, the user speaks very close to the on-device microphone, resulting in inaccurate high dB samples which obviously needs to be eliminated.

Each device is calibrated with the help of a reference sound level meter or another already calibrated device. We have used generic calibration technique from Google's implementation of the personal SPL meter which allows the user to adjust a gain parameter to be multiplied with p^2 of Eq. 1.

As mentioned earlier, analysis of data to generate a uniform pollution map of the city for any application user is managed in cloud layer. The main noise indicators for noise mapping are L_{day}, $L_{evening}$, L_{night} and L_{den} (day–evening–night), obtained by averaging sound levels for corresponding time intervals over the period of one month. To be more precise, the sound level required is the so called A-weighted equivalent continuous sound pressure level $L_{A,eq}$; the quantity put forward as the international standard for environmental noise assessment. $L_{A,eq}$ is itself an average, time of instantaneous air pressure differences (i.e. sound) over arbitrary periods of time [4]:

$$L_{A,eq} = 10\log_{10}\frac{1}{T}\int_0^T \frac{p^2}{p_0^2}dt \quad [\text{dB}] \tag{1}$$

Where, p_0 is the reference sound pressure, taken as 20 μPa. The subscript A denotes that the equivalent sound pressure level is to be weighted with the A-weighting curve, which approximates the sensitivity of the human ear to different frequencies (low and high frequencies are attenuated). By choosing time intervals appropriately equivalent sound pressure levels can be computed for periods of interest, such as peak hours or during the night time. In this way, L_{day}, $L_{evening}$ and L_{night} are found by taking the following time intervals in Eq. (1): a 12 hour day period, 4 hour evening period, and 8 hour night period respectively. The composite day–evening–night sound pressure level L_{den} is a logarithmic, weighted average over these three periods, as follows [4]:

$$L_{den} = 10\log_{10}[\frac{1}{24}(12\times10^{\frac{L_{day}}{10}} + 4\times10^{\frac{L_{evening}+5}{10}} + 8\times10^{\frac{L_{night}+10}{10}})] \tag{2}$$

In summary, the maps that are produced as a result of the Environmental Noise Directive (END) [9] present L_{den} and L_{night} values following Eqn. 2 and (1) respectively for railway station, school, college, university, hospital, industrial zones and other related sources of noise, in color-coded bands. The END states specifically that "the values of L_{den} and L_{night} can be determined either by computation or by measurement". In our approach, the values are coming from the measurement only. However, in practice, sound levels are generally computed because of scalability issues, indeed, it has hitherto been unfeasible to consider measurement devices at all places and times

In group mode of the NoiseSense, a leader is elected periodically based on parameters phone's power level P_0, remaining battery power (mAh). However, other parameters like CPU clock frequency, RAM and storage capacity can also be considered. Only the leader can use GPS while other connected members assuming location of the leader which is effective in some sense because members and the leader are always in close proximity to remain connected via wireless network. The leader is rotated after some time to balance power usage among the group. Here, we propose an algorithm to elect a leader among a number of mobile devices connected via any wireless network like WiFi. The process of data sending to leader and server is also included in the same algorithm. In this algorithm initially messages are broadcasted, so there is no need of early exchange of addresses and power level information. Also, there is no need for reliable communication channel because leader periodically announces its presence for new member to join. If a device fails to receive any message it gets the next one. Here, we have considered the leader failure case also because it is crucial to the group and handled that carefully.

Assumptions:

1. Devices are already connected and can exchange messages.
2. The algorithm works in three phases. INIT, MEMBER and LEADER. All the devices run the same algorithm starting from INIT phase.

3. Messages can be of three types: NOMINATION_MESSAGE, LEADER_MESSAGE, & DATA_MESSAGE.
4. Two timers are used:
    a. LISTEN_TIMEOUT: Time window specified to receive broadcast messages from other devices.
    b. LEADER_TIMEOUT: Expiry time for a leader.
5. P_o is own power level on a scale of 0 to 100. P_i is a vector containing power levels of all NOMINATION_MESSAGE received.
6. R_o is a positive random integer number generated in case of a tie.

Algorithm

Input: N connected devices
Output: A group formed by N participating devices with a unique Leader device.

Init ( )

1. *Broadcast NOMINATION_MESSAGE with P_o*
2. *while LISTEN_TIMEOUT !=0 do*
3. *Receive Messages*
4. *LISTEN_TIMEOUT - -*
5. *end while*
6. *If (LEADER_MESSAGE received) then*
7. *Member()*
8. *else temp ← max(P_i)*
9. *If(P_o > temp)*
10. *Leader()*
11. *else if (P_o = temp)*
12. *Multicast NOMINATION_MESSAGE with P_o+ R_o*
13. *go to step 1*
14. *end if*
15. *end if*

Member ( )

1. *Read LEADER_TIMEOUT from LEADER_MESSAGE*
2. *while (LEADER_TIMEOUT != 0)*
3. *Send DATA_MESSAGE with data sample to Leader*
4. *LEADER _TIMEOUT← LEADER_TIMEOUT- Elapsed time*
5. *end while*
6. *Init()*

Leader ( )

1. *Initialize LEADER_TIMEOUT and broadcast LEADER_MESSAGE*
2. *while (LEADER_TIMEOUT!= 0)*
3. *Receive DATA_MESSAGE , M_i from all members*
4. *Read current location, add it to each Mi and send to server*
5. *Send own DATA_MESSAGE to the server*
6. *LEADER _TIMEOUT ← LEADER_TIMEOUT- Elapsed time*
7. *Re-broadcast LEADER_MESSAGE with LEADER_TIMEOUT*
8. Goto step 2
9. *end while*
10. *Init()*

The algorithm guarantees that a leader is formed among the devices, taking part in the election and there is always a leader in a group even if the current leader fails. It also guarantees the leader is changed in each election provided that in the new election current leader's L(P_o) becomes lesser than any other node's P_i. If the leader fails, or removed from the system, the same iteration occurs and eventually a leader always remains in the group. Thus, the maximum delay, i.e. missing noise values in a certain location can never be more than *LEADER_TIMEOUT.*

V. IMPLEMENTATION

Implementation of NoiseSense involves implementing a mobile application and a server end. A mobile application using the Android platform was implemented. Android Studio 2.3.1 was used as IDE for both design and development of the application. In Android Studio user interface elements are generated using XML and Java is used as a programming language.

The main functions of the cloud layer are data storage, processing, data analysis and giving feedback to the user a noise map of the city. In our work, we have used OPENSHIFT as our Cloud Service provider.

VI. SYSTEM EVALUATION

The proposed NoiseSense application is tested in various conditions. We have noted that instead of using the phone's microphone directly, the use of a collar clip microphone is giving more accurate result as it removes ambient sound pressure error from the collected noise reading with respect to the ground truth value. We have used the same to collect our noise data. In the application, we have used the automatic calibration setting [10] to get proper ambient noise.

Noise exposure while moving in the city is shown in the Fig. 4(a) and the same on the map is shown in 6(a). Here we are getting individual's noise pollution exposure both its variation graph with time and on the map. The personal noise exposure experiment reported in this paper was done on 14/05/2017 Sunday morning and was measured on a bus trip near Rabindra Sadan, Kolkata. Temporal effect of noise at Jadavpur University, Salt Lake campus on 28/06/2017 is reported in Fig. 4(b).

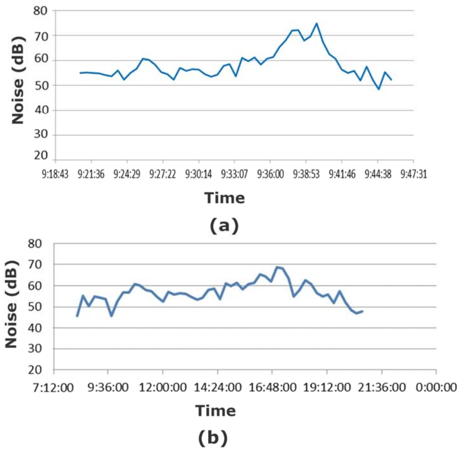

Fig. 4: Personal Noise Exposure experiment (a) plot of noise variation with time on the road while travelling (b) plot of noise variation with time in JU Salt Lake Campus

As we have mentioned earlier, that both group mode and individual modes can be applied in this participatory mode of data collection. If the devices are not forming any group, then the power consumption is high because of continuous use of GPS, where as if they are forming a group, then power consumption is reduced significantly, and this reduction is proportional to the number of people present in the groups.

In our experiment, we have used Lenovo A6020a46(3GB), Samsung GT-S7392 (512 MB), Motorola Moto E XT1021(1GB), Asus Zenphone MAX(2GB), Redmi 3S Prime (3GB) having battery capacity 2750, 1500, 1980, 5000 and

4100 mAh smartphones respectively. We have done this experiment for both single mode and group mode for an hour duration. Now, when we start testing our single mode vs Group mode, in both cases the device is connected with WiFi because the energy saving is different for Cellular and WiFi. Now, let, single mode energy consumption is E_s and group mode energy consumption is E_g, then the energy saving percentage is calculated as:

$$\text{Energy Save \%} = \frac{(E_s - E_g)}{E_s} \times 100\% \quad (3)$$

Following Eq. 3, we have calculated energy savings in our case studies. Using the above mentioned set of mobiles, we note that for a group of 3 people, we are saving 14.5% energy and saving 26.8% for a group of 5 devices as compared to E_s for running this NoiseSense application. Also note that the overlapping region in the graph presented in Fig. 5(b), clearly shows the effect of change in group leader. In Fig. 5, high slope denotes relatively faster discharge of smartphone due to the leader role. It is also clear that the nodes that are just participating in this crowd sourced application and running this algorithm in the background, are not affected, but getting benefitted both in terms of battery usage and by getting city's overall noise pollution overview.

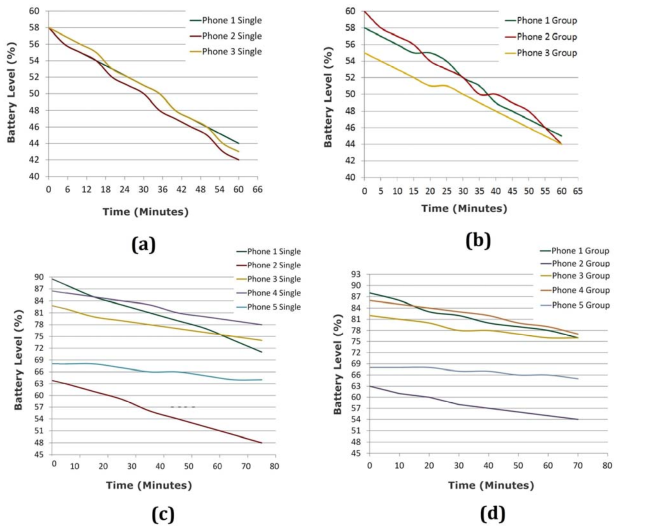

Fig. 5: Energy consumption comparison in Single mode vs Group mode (a) 3 mobiles in single mode (b) 3 mobiles in group mode and (c) 5 mobiles in single mode (d) 5 mobiles in group mode

Finally, incorporating all the collected data, from all the users, we are getting a real-time noise pollution footprint of the city as shown in Fig. 6 (b).

VII. Conclusion

Here, we proposed a crowd sourced energy efficient real-time noise monitoring system called NoiseSense which gathers context aware temporal and spatial data from the citizen to generate the noise pollution map of the city. However, we have generated these maps based on the collected data from the citizen, but there are many missing points even after rigorous data collection from this crowd sourcing. To address this problem, noise maps can be simulated on the basis of statistics on sources present, source specific propagation rules and information on the urban layout that portion which we plan to focus in our next work.

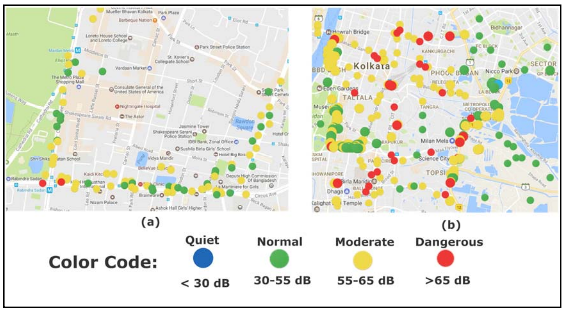

Fig. 6: Noise Pollution exposure (a) Personal exposure to noise during a city bus trip in maps (b) overall city's noise pollution in maps

Acknowledgement

The research work of the first author is funded by "Visvesvaraya PhD Scheme, Ministry of Communications and IT, Government of India."

References

[1]. https://www.telegraphindia.com/1170526/jsp/calcutta/story_153581.jsp

[2]. R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu., EarPhone: An End-to-End Participatory Urban Noise Mapping System, In proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), pp. 105–116, April 2010.

[3]. Eiman Kanjo, NoiseSPY: A Real-Time Mobile Phone Platform for Urban Noise Monitoring and Mapping, Mobile Network Application, Volume 15, Issue 4, pp. 562-574, August 2010.

[4]. Nicolas Maisonneuve, Matthias Stevens, Maria E. Niessen and Luc Steels, NoiseTube: Measuring and mapping noise pollution with mobile phones, In Proceedings of the 4th International ICSC Symposium, Thessaloniki, Greece, May 2009.

[5]. Hong Yao, Guang Yang, Changkai Zhang, Chengyu Hu, Qingzhong Liang, DDNM: Monitoring Environment Noise using Smart Phones, In IEEE 7th International Symposium on Embedded Multicore/Manycore System-on-Chip, 2013.

[6]. Becker M, Caminiti S, Fiorella D, Francis L, Gravino P, et al. Awareness and Learning in Participatory Noise Sensing. PLoS ONE 8(12), 2013.

[7]. Kjærgaard MB, Langdal J, Godsk T, Toftkjær T. Entracked: energy-efficient robust position tracking for mobile devices. In Proceedings of the 7th international conference on Mobile systems, applications, and services (pp. 221-234). ACM. June 22, 2009.

[8]. Balasubramanian N., Balasubramanian A., and Venkataramani A., Energy consumption in mobile phones: a measurement study and implications for network applications. In Proceedings of the 9th ACM SIGCOM Conference on Instrument Measurement, pp. 280-293, 2009.

[9]. Noise in Europe 2014, EEA Report No 10/2014, ISBN 978-92-9213-505-8, ISSN 1977-8449, 2014

[10]. Google SPL Meter, Weblink: https://github.com/mondain/android-spl-meter

[11]. J. Dutta and S. Roy, "IoT-fog-cloud based architecture for smart city: Prototype of a smart building," 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, Noida, India, pp. 237-242. Jan. 2017. doi: 10.1109/CONFLUENCE.2017.7943156

[12]. J. Dutta, F. Gazi, S. Roy, C. Chowdhury, "AirSense: Opportunistic Crowd-Sensing based Air Quality monitoring System for Smart City," in the proceedings of the IEEE SENSORS 2016, pp. 976-978, Oct. 2016.

[13]. J. Dutta, C. Chowdhury, S. Roy, A.I. Middiya, F. Gazi, "Towards Smart City: Sensing Air Quality in City based on Opportunistic Crowd-sensing," International Conference on Distributed Computing and Networking (ICDCN) 2017, ACM, DOI: http://dx.doi.org/10.1145/3007748.3018286.

[14]. J. Dutta, P. Pramanick, S. Roy, "Energy Efficient GPS Usage in Location based Applications." Proceedings of the 6th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA). AISC, Springer. Oct. 2017.