RoadSoundSense: Acoustic Sensing based Road Congestion Monitoring in Developing Regions

Rijurekha Sen, Pankaj Siriah, Bhaskaran Raman
{riju, pankaj, br}@cse.iitb.ac.in
Indian Institute of Technology, Bombay

Abstract

Road congestion is a common problem all over the world. In many developed countries, automated congestion detection techniques have been deployed, that are used in road travel assisting applications. But these techniques are mostly inapplicable in many developing regions due to high cost and their assumptions of orderly traffic. Efforts in developing regions have been few. In this paper, we present RoadSoundSense, an acoustic sensing based technique, for near real time congestion monitoring on chaotic roads, at a moderate cost.

We present the detailed design of an acoustic sensing hardware prototype, which has to be deployed by the side of the road to be monitored. This unit samples and processes road noise to compute various metrics like amount of vehicular honks and vehicle speed distribution, with speeds calculated from honks using differential Doppler shift. The metrics are sent to a remote server over GPRS every alternate minute. Based on the metric values, the server can decide the traffic condition on the road.

Data from deployment of this prototype in six different Mumbai roads, validated against manually observed ground truth, shows feasibility of per minute congestion monitoring from a remote server. K-means clustering gives on average 90% accuracy to group unlabeled data on a new road into two clusters of congested and free-flow. Deployment data from one road for six days shows the temporal variation in traffic state for that road. Though we test our technique in Mumbai, we believe that most of our claims and experimental results can be extended to city roads of other developing regions as well.

1 Introduction

The average number of vehicles on Indian roads is growing at an enormous rate [1]. This is leading to increasing levels of road congestion, longer and unpredictable travel times and wastage of time and fuel for commuters. Growth in infrastructure has been slow due to various reasons such as high cost, lack of space, bureaucracy, etc.

We define traffic on a road to be *congested* if drivers have to slow down vehicles or stop and go, because of presence of several other vehicles on road. Automated congestion monitoring can help in making traffic signal timings more efficient. Secondly, if some roads or junctions show regular trends of becoming congested, new infrastructure such as flyovers and freeways can be planned there. Thirdly, this can be used to design mobile applications for on road commuters to help them avoid congested routes and reduce travel time.

Automated congestion monitoring is widely in use in developed countries. But in developing countries, the non-lane based disorderly traffic conditions, make direct application of the existing techniques difficult. We elaborate on the difficulties in Section 2. On the other hand, unlike traffic in developed countries, chaotic traffic in developing regions is very noisy. One of the characteristic sounds comprising this noise is vehicular honks, which drivers use excessively to alert other drivers and pedestrians. We have made a list of about 50 video clips [2], shot on Indian roads, from which the noisiness of chaotic traffic becomes apparent. In *RoadSoundSense*, we seek to exploit this distinctive feature of "excessive noise" to do congestion detection on chaotic roads.

In our prior work [3], vehicle honks recorded in roadside recorders were used to estimate vehicle speeds using differential Doppler shift. Metrics like vehicle speed distribution and amount of vehicle honks gave 70% accuracy in classifying traffic state into congested and free-flow using threshold based classification. But can this technique, involving computation intensive acoustic signal processing, be implemented on an embedded sensor platform, to be used for on-road sensing? Can the sensing and processing be done in near real time? Will the cost be low enough? These are some *implementability issues* of *RoadSoundSense*. Will the system be able to detect congestion on a wide variety of roads? Will the traffic classification model vary from road to road? In that case, what will be the training overhead of our system on a new road? Can we do without training using unsupervized learning? These are the *usability issues* of *RoadSoundSense*.

In this paper, we seek to answer the above questions. We present the detailed design of an acoustic sensing hardware prototype which has been deployed by the side of the road. This unit samples and processes road noise to compute various metrics like amount of vehicular honks and vehicle speed distribution and sends the metrics to a remote server every alternate minute. Data from deployment of this prototype in six different Mumbai roads, validated against manually observed ground truth, shows feasibility of per minute congestion monitoring from the remote server. K-means clustering gives on average 90% accuracy to group unlabeled data on a new road into two clusters of congested and free-flow. Deployment data from one road for six days shows the temporal variation in traffic state for that road. Our prototype has a moderate cost of $160 and is easy to install and maintain on road-side lamp-posts.

[4, 5, 6, 7, 8, 9] are a few examples of the extensive literature that study the negative impact of road noise on urban life and explore how to mitigate this by making better vehicles, roads and isolation of roads through proper urban planning. The problem is more acute in developing countries, where quality control of vehicle engines and tyres and also of road pavement are not strictly observed due to issues of funds and bureaucracy and lack of public awareness. Also, the non-lane based disorderly traffic causes excessive use of vehicle brakes and honks, adding to the noise. [10, 11, 12, 13] study the high amount of noise pollution in the four Indian cities of Asansol, Jaipur, Kolkata and Varanasi respectively, and such pollution is common in other Indian cities too. *Though we are using acoustic sensing, our aim is not to promote noise pollution.* We seek to use an already existing negative feature for the positive purpose of road congestion monitoring.[1]

The rest of the paper is organized as follows. Section 2 discusses the related work for this application. Section 3 gives details of our prior work, on which the work in this paper is based. Section 4 presents the detailed design of our hardware prototype. Section 5 presents the deployment experiences and results. We discuss possible avenues of future work and conclude the paper in Section 6.

2 Related Work

Any congestion detection technique for developing regions should (1) handle chaotic traffic, (2) incur low cost (3) pose minimum hindrance to traffic while installation and maintenance and (4) minimize active participation from commuters and their vehicles. In this section, we discuss whether any of the existing techniques meet these requirements.

2.1 Fixed sensor based techniques

In these techniques, the sensors that gather various road related information are statically placed on or by the side of the road. Examples are inductive loop detectors [14], image sensors [15] and magnetic sensors [16]. These techniques can be prohibitive in terms of infrastructure and maintenance costs. As given in [17], the initial installation cost of a vehicle loop detector is approximately $26,100. Secondly, the inherent assumption of lane-based orderly traffic makes these techniques inapplicable for chaotic road conditions. Thirdly, the assumption of low variability in vehicle speeds also does not hold in developing regions where heavy slow moving trucks and high speed motorbikes ply on the same road at the same time. Finally loop and magnetic detectors need to be placed under the road. If roads need to be dug up to install and maintain the infrastructure, that will adversely affect traffic, as alternate routes are often times unavailable. Moreover, road lifetime in India is less than ideal. Each time the road is relaid, the sensors will have to be reinstalled.

2.2 Probe vehicle based techniques

In these techniques, the sensors are mobile and placed in a subset of vehicles that ply the road. Examples are GPS-enabled probe-vehicles [18, 19, 20] and multiple sensor enabled smartphones in vehicles [21]. In India, proliferation of GPS receivers in vehicles is quite low. Only a small number of taxi fleet companies and state transport companies have GPS units installed in their vehicles in a few metropolitan cities. Smartphone penetration in India is also quite low [22], though mobile phone penetration is extremely high. Most people have low end phones and are unable to take part in participatory sensing. Even for those who have smartphones, it is difficult to think of an incentive model to attract them to take part in GPS sensing as it involves sensing as well as communication costs.

As we show in this paper, our technique *RoadSoundSense*, with easy installation and maintenance of sensing units on road-side lamp-posts and a moderate cost of about $160 per unit, can detect congestion in chaotic traffic without intervention from commuters and vehicles.

3 Prior Work: "Horn-Ok-Please"

In [3], we developed a technique to estimate vehicle speed using differential Doppler shift of vehicle honks recorded at roadside recorders. The system architecture is given in Fig. 1. Suppose that a sound source moves with speed v_s, and the receiver (observer) is stationary. Denote the emitted audio frequency as f_0 and speed of sound as v. When the source is moving away from the receiver, the frequency observed at the receiver is given by,

$$f_1 = \frac{v}{(v + v_s)} f_0 \tag{1}$$

And when the source is moving towards the receiver, the frequency observed at the receiver is given by,

$$f_2 = \frac{v}{(v - v_s)} f_0 \tag{2}$$

If f_0 is known, v_s can be estimated easily from Eqn. 1 or Eqn. 2, and one sensor would suffice. But it is not easy to guess f_0, as different honks have different base frequencies.

[1] In future, if public awareness and government policies change in India, making noise pollution control more feasible, the techniques developed in this paper can still be used, but for different applications. For example, the prototype hardware, developed in this paper, detects vehicle honks. Suppose chaos on Indian roads subsides in future, so that drivers don't need to use honks to alert other drivers or pedestrians and honks become banned by law. Then the same prototype hardware can be used to enforce this law, as automatic honk detection will enable automatic fine imposition.

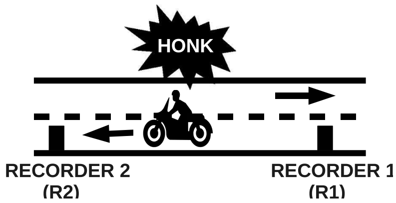

Figure 1. Original System Architecture

We thus used a two-sensor architecture: Fig. 1 depicts a deployment of two recorders by the side of a two-way road. When a moving vehicle blows honk in between the two receivers, it is approaching one receiver and receding from the other. Substituting the value of f_0 from Eqn. 1 in Eqn. 2, we get following equation,

$$v_s = \frac{(f_2 - f_1)}{(f_2 + f_1)} v \tag{3}$$

In [3], we developed algorithms for **(1) Honk detection:** The two recorders record and *detect* honk from the noisy road recording independently. **(2) Honk matching:** We then have to *match* same honks between the two recorders, so that we apply Eqn. 3 for the same honk. **(3) Frequency extraction:** We have to *extract* f_1 and f_2 and apply Eqn. 3 to get the speed estimate. Using over 18 hours of recordings from two roads in Mumbai, we showed that there are enough honk samples for our method to be useful and that our speed estimation technique is effective in real conditions. Further, we used our data to characterize traffic state as free-flowing versus congested using a variety of metrics: the vehicle speed distribution, the amount of honks and noise level. We used the voice recording application N79 phones for roadside recordings. Two persons had to be present for hours on the road to manually record road noise. Back in lab, the hour long recordings would be analyzed to get the information described above.

Contributions of this paper:
(1) We design and develop an acoustic sensing hardware prototype, comprising of Recorder1 (R1) and Recorder2 (R2), and show that computation intensive acoustic signal processing and reliable data communication to remote server in near real time are feasible on resource constrained embedded platform. The enhanced system architecture is shown in Fig. 2, where each prototype implements the architecture given in Fig. 1.
(2) Data from deployment of this prototype in six different Mumbai roads gives valuable insight into choosing appropriate acoustic metrics for congestion detection on particular roads.
(3) We address the problem of unsupervised learning of the traffic classification model by our unit, on a new road, using K-means clustering.
(4) Deployment data from one road for six days shows the temporal variation in traffic state for that road.

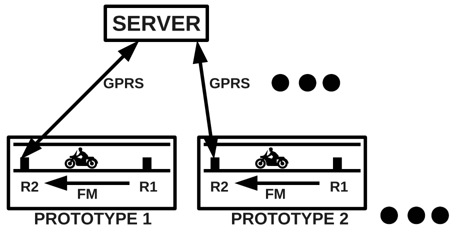

Figure 2. Enhanced System Architecture

4 Prototype Design

Our two sensor based congestion detection technique, that this prototype has to implement, has three functional requirements - (1) **Sensing** - sampling road noise, (2) **Processing** - filtering of sensed noise, detecting honks, matching honks between R1 and R2 after time synchronization and detecting speeds and (3) **Remote communication** - sending various metric values like amount of honks and speeds to remote server.

4.1 Design choices

The design choices available to us for R1 and R2 are as follows.

Both R1 and R2 having sensing and remote communication capabilities. Sampled raw audio signal will be sent to a central server which will do the processing. The primary disadvantage here is that, the amount of data to be communicated to the central server will be huge, though central server would need only the honk related information. This will unnecessarily increase communication cost and delay.

Both R1 and R2 having sensing, computation and remote communication capabilities. Only computed metric values will be sent to the server. The two units will also need local communication capabilities to communicate between each other, as honk matching and speed calculation cannot be done individually. This seems plausible but has scope of further optimization as specified in the next design choice.

R1 having only sensing and R2 having sensing, computation and remote communication capabilities. The units will have local communication capabilities for R1 to send sampled analog signal to R2 over a small inter-sensor distance. R2 will do processing and remote communication. This choice seems good, provided we can handle the various technical challenges discussed next.

First, the quality of audio signal should not be affected during communication from R1 to R2. Secondly, absence of computation capabilities on R1 would prevent time stamping of the audio signal. But R2 should know which sensing event happened when for matching honks and sending timestamped metrics to server. So propagation delay of audio signal from R1 to R2 has to be negligible. Thirdly, R2 should have enough capabilities to perform computation intensive operations like FFT on two audio signals. We now show how we choose our hardware components to meet these challenges.

4.2 Hardware

Recorder 1 (R1) - To meet the first two technical challenges, we take idea from any commercially available wireless microphone, that has an FM transmitter to transmit voice signal to the amplifier, which has an FM receiver. We seek to use FM to transfer analog signal from R1 to R2. Using commercially available WR-601 FM transmitter-receiver pair [23], we achieve good quality analog signal transmission, i.e. the frequency spectrum of original signal and signal transmitted over FM does not vary.

We experimentally test propagation delay using two recorders - an N79 phone and a laptop with an FM receiver. A square wave pattern is played from a speaker. The N79 phone and an FM transmitter with microphone are placed very near to the speaker. The laptop with FM receiver is first held close to the speaker and then gradually moved to 25m, the square wave pattern being played every 1m. Synchronizing the first recorded wave pattern in the laptop and the N79, to remove any time offset in starting the two recorders, synchronizes all 25 recorded patterns, with 3μs offset only in one case. The offset is 0μs in all other cases. This shows that analog signal propagation delay over FM is negligible.

But the range obtained using FM with line of sight between the transmitter and the receiver is around 25m, and not 30m, as used in [3]. Smaller range means less distance between R1 and R2, reducing number of vehicles honking between them. But since the advantages of using FM in terms of hardware complexity reduction, far exceed this small drawback, we decide to use it.

Another component required at R1 is the microphone, for which we use CTP-10DX miniature omnidirectinal 15 m range electret condenser microphone from Ahuja [24]. Fig. 3 shows our prototype hardware for R1.

Recorder 2 (R2) - Acoustic signal processing motivates the use of a Digital Signal Processor (DSP) instead of an ordinary microcontroller. We use an evaluation module from TI, C5505 ezdsp [25], which has a small form factor (3.15 x 1.5 inches), C5505 DSP chip with 100 MHz clockrate at active power less than 0.15 mW/MHz, AIC3204 codec chip, a line-in socket and a 60 pin expansion connector including SPI and UART pinouts. It also has an FFT Hardware Accelerator, a tightly coupled coprocessor, that we use extensively in our audio signal processing. We use off-the-shelf GPRS modem having SIM300 GSM/GPRS module from SIMCOM [26], with serial connectivity options. We include a flash memory in our unit, in case computed data needs to be stored before sending. For this we use 128 Mbit SPI flash from Spansion [27]. The same microphone, as used in R1, is used in R2 too. We design a PCB to interface the different components which also has the powering circuitry described next. Audio connections for R2 are shown in Fig. 4 and non-audio connections in Fig. 5. Fig. 6 shows the overall hardware block diagram comprising of R1 and R2.

Power supply - The GPRS modem oprates at 12V with 2A peak current. The DSP module operates at 5V with peak current in mA. The flash has an operating voltage of 2.7-3.6 V with 26 mA peak current. We use a pack of 6 Lithium-ion batteries, with 12V output voltage, 3A peak current and 4AH capacity. The interfacing PCB takes input from battery, directly powers up the GPRS modem and feeds the DSP module through a AP1512AT5 DC-DC converter [28] to step down from 12V to 5V. The flash is powered up from VCC pin of DSP module supplying 3.3V.

Enclosure - As shown in Fig. 7, we use a 15.5cm x 12cm x 6.5cm waterproof ABS plastic enclosure [29] for R2. The audio connectors to plug in the microphone and the FM receiver and the GPRS antenna remain outside the enclosure.

Cost - Table 1 gives the cost breakup for the different components. All the components are off-the-shelf modules, which has somewhat increased the cost. If we make our own custom board, with only the required ICs and connectivity, we believe that the cost can be kept under $100. The GPRS communication cost is Re 0.1 per 10 KB and we send about 50 bytes of data per minute. Hence this cost is negligible.

Item	Unit Price($)	Quantity	Cost($)
DSP module	50	1	50
GPRS modem	50	1	50
FM tx-rx	15	1	15
Microphone	5	2	10
Interfacing PCB	5	1	5
Battery	20	1	20
Enclosure	5	1	5
Flash	3	1	3
Connectors	0.4	5	2
Total			160

Table 1. Prototype Cost Breakup

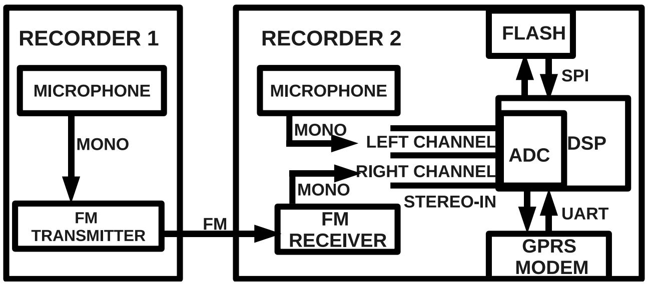

Figure 6. Block Diagram of Hardware

4.3 Software

The DSP module comes with a CCStudio IDE [30] for programming and debugging. The boot image of user program is loaded to 64 KB EEPROM. In our program, 16-bit stereo (composed of two mono signals as shown in Fig. 6) samples are captured by the AIC3204 codec at 16 KHz sampling frequency and copied to the DSP memory using DMA. We use ping-pong buffering to avoid missing samples during processing. Processing involves filtering of the noisy signal, detecting honks in each individual mono channel, matching honks between two channels and extracting $f1$ and $f2$ to compute speed. The details of detection, matching and frequency extraction algorithms can be found in [3].

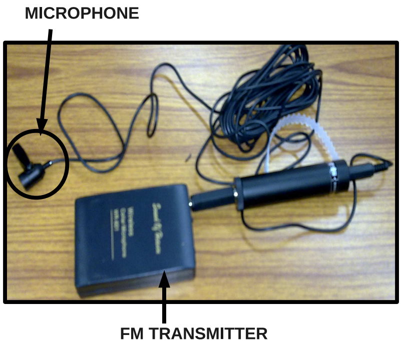

Figure 3. R1

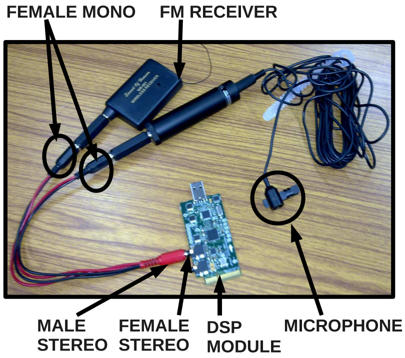

Figure 4. R2: Audio Connections

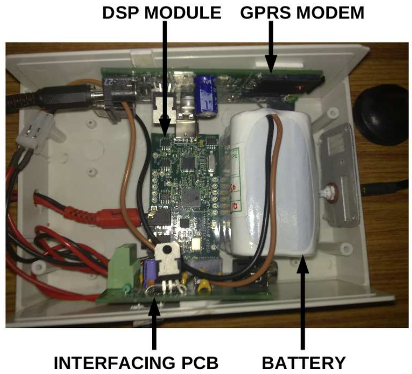

Figure 5. R2: Non-Audio Connections

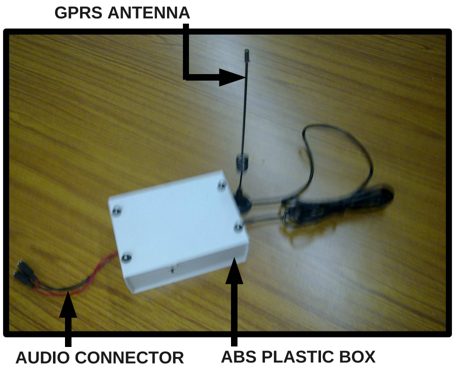

Figure 7. R2 Packaging

The sequence of events that happen after powering up our unit is as follows. The flash memory is erased. The DSP sends a request for current time-stamp to the remote server over GPRS and initializes the RTC based on the received time-stamp. Then the following loop starts. Audio is sampled, filtered and honks are detected for time t_{sample}; honks are matched, speeds are calculated and metrics are stored in flash for time $t_{process}$. This continues for time $t_{workLocal}$. Next in time $t_{sendRemote}$, metrics are read from the flash and sent to the remote server over GPRS. This loop of sampling, processing, storing for time $t_{workLocal}$ and sending for time $t_{sendRemote}$ continues indefinitely.

t_{sample}, $t_{process}$, $t_{workLocal}$ and $t_{sendRemote}$ are parameters, whose values are chosen either by resource constraints or by repeated experiments to detect worst case processing delay. For example, t_{sample} is decided to be 4 secs because of resource constraint. In honk detection, we use 128 point FFT, for which 128 audio samples are needed. Each audio sample is 2 bytes. In 1 sec, at 16KHz sampling frequency, there are 125 windows of 128 samples. Each window has a 2 byte time-stamp (10 bits for millisecs and 6 bits for secs). During detection, we store the time windows having honk characteristics, with time-stamp, to be used later for honk matching and frequency extraction. In the worst case, each of the 125 windows can have honk characteristics, for which we have to store (((128x2)+2)x125) bytes/sec = 32KB/sec. For 2 channels, we need to store 64KB/sec. Given a 320KB RAM, after setting aside storage for code, stack and temporary variables, we can use about 256KB to store honk windows. This limits t_{sample} to (256KB)/(64KB/sec) = 4 secs. $t_{process}$ is set at 5μs, after repeated experiments to ascertain flash writing time. $t_{workLocal}$ is set at 1 min, because we want updates at the server at least every alternate minute. $t_{sendRemote}$ is set at 50 secs, after repeated experiments to ascertain combined time for reading flash, setting up new TCP socket connection over GPRS in case there is no existing connection and sending data.

5 Prototype Deployment

Our hardware prototype is moderate in cost and is able to sample and process sound and send metrics to server every alternate minute. This solves the *implementability issues* of our technique. We next seek to answer the questions pertaining to the *usability issues*. Will acoustic metrics portray traffic state on a wide variety of roads? Should choice of metrics be road specific? In that case, will standard machine learning tools like Support Vector Machine (SVM) be able to automate the metric choice for a particular road? Labelling data instances with classes for training, generally involves human judgement, hence it is costly and time consuming. So, will unsupervised learning techniques (which do not require training data) be able to find patterns in the data for our system?. We deploy our prototype on different roads in Mumbai to seek answers to these questions.

5.1 Deployment locations

The server is kept within IIT Bombay campus in Powai. The hardware prototype units are deployed at six locations in Mumbai. The locations are listed in Table. 2 and their positions on Googlemap of Mumbai are shown in Fig. 8. The locations are near important road junctions or railway stations, where congestion occurs daily during the peak hours. The roads have variable width and traffic type. The locations are chosen within a distance of 5-6 Km from IIT to reduce trip time from our lab to each location.

The positioning of the prototype on a road, in relation to the location of the traffic signal along that road, matters significantly. Vehicles are expected to stop at the signal, so standing traffic very near to signal is not considered congested. On the other hand, traffic queue length on that road never exceeds a certain limit even in the worst case. Thus the road stretch very far from signal never gets congested. So the prototype should be at an intermediate position where congestion monitoring makes sense. Also, it does not make sense to put the prototype after traffic signal, as there is mostly no congestion there. In our deployments, we use a distance of 150-200m before a traffic signal, after manually observing the queue length on that road.

No.	Location	Road bi-directional	Road width (each way)	Vehicle type
1	Bhandup	Yes	10m	All+
2	Vikhroli	Yes	10m	All+
3	Gandhinagar	Yes*	25m	All+
4	Chandivali	Yes	15m	All+
5	Ghatkopar	Yes	10m	All+
6	Powai (Hiranandani)	Yes	8m	Light

Table 2. Deployment Location Details

* Road width prevents sensing of noise from opposite direction.
\+ Heavy (trucks, buses) and light (cars, motorbikes, autorickshaws)

To avoid the hassles of getting permission from city authorities to put up the units on lamp-posts, we deploy them at road-side shops. We use locks with steel chains through clamps, to prevent stealing of the deployed units.

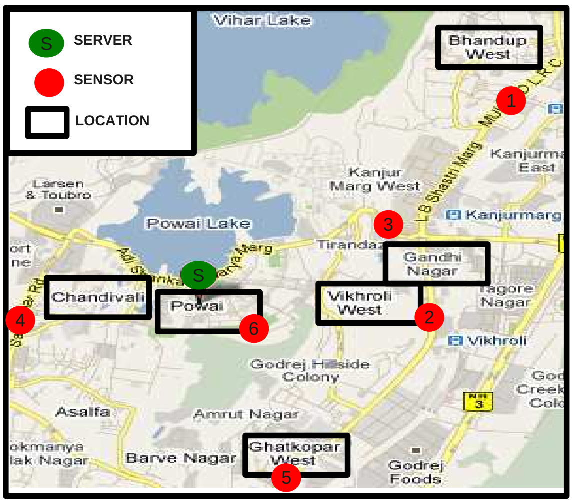

Figure 8. Sensor Deployment Locations in Googlemap

5.2 Results

We program our prototype to send the following timestamped values computed over 1 minute to the remote server in the next minute - (1) number of honks in R1 (numhonks1), (2) duration of honks in R1 (duration1), (3) number of honks in R2 (numhonks2), (4) duration of honks in R2 (duration2) and (5) all vehicle speed samples. The server uses the first four as *honk-based metrics*. From the last, it computes 70^{th} percentile speed (70speed) and percentile speeds less than 10 Kmph (10speed), the *speed based metrics*.

From 22^{nd} to 27^{th} Nov, 2010, we remain for 2-3 hours on road, manually observing the traffic state, one day at each deployment location. Two videos of traffic for each location, one showing free-flow and the other showing congestion, can be found at [2]. Fig. 9 - Fig. 20 show plots of first three deployment locations. The captions signify {Location, time}. Each vertical bar represents 1 minute and we color code the bars according to our manual observation. The horizontal black lines are provided in each plot to aid the height comparison among bars. As we can see, number and duration of honks are higher in congestion than in free-flow. In congestion, 70^{th} percentile speeds are low, while percentile speeds less than 10 Kmph are high.

5.2.1 Road specific choice of metrics

Though all four metrics somewhat correspond to manually observed traffic state, there are some metrics which bring out the traffic state at particular locations, more accurately than other metrics. Here we seek to intuitively understand the reasons behind this, as it will help us in selecting good attributes to build traffic classification model for each location.

(1) Fig. 9, Fig. 12, Fig. 15 and Fig. 18 show good difference between two traffic-states at Bhandup based on all four metrics. This is found in the other three deployment locations of Chandivali, Ghatkopar and Powai (Hiranandani) also. But these roads are bidirectional and not very wide, so honks from opposite direction are recorded too. The speed based metrics are direction sensitive, as Equation 3 gives signed speeds based on direction. Hence we can filter out speed values in opposite direction. But we cannot do this for honk based metrics, which might give inflated values. Thus though all four metrics are good for these locations, speed based metrics should be given higher weightage than honk based metrics in any road state classification.

(2) As seen in Fig. 10 and Fig. 13, the honk based metrics show no difference between congested and free-flowing traffic states at Vikhroli. This is because, just in front of the shop where our unit is deployed, there is a small cut in the divider between the bi-directional road. The pedestrians use this for road crossing. Thus, even in free-flowing traffic, many vehicles blow honk for alerting pedestrians and the road being quite busy, number and duration of honks in free-flow is high. But speed based metrics bring out the difference accurately here, as ample honking gives good number of matched honks and speeds, whose values vary widely between free-flow and congested. Thus only speed-based metrics are good for this location.

(3) As seen in Fig. 11 and Fig 14, the honk based metrics like number and duration of honks show clear difference between two traffic states at Gandhinagar. The width of the road being 25m for this location, number of vehicles increases almost five-fold in congestion from free-flow. Hence number and duration of honks go up drastically. The speed based metrics are more complex to calculate than the honk based metrics as it involves matching honks between R1 and R2 and computing 1024 point FFT for frequency extraction. So using only honk based metrics will be good for this loca-

Bhandup (Accuracy 93.2%)	Vikhroli(Accuracy 98.3%)	Gandhinagar (Accuracy 100%)
$+1.38 * numhonks1$ $+0.38 * duration1$ $-0.17 * numhonks2$ $+1.19 * duration2$	$+0.71 * numhonks1$ $-0.21 * duration1$ $+0.15 * numhonks2$ $-0.57 * duration2$	$+1.45 * numhonks1$ $+0.97 * duration1$ $+1.59 * numhonks2$ $+0.91 * duration2$
$-2.94 * 70speed$ $+2.25 * 10speed$	$-2.71 * 70speed$ $+2.71 * 10speed$	$-0.58 * 70speed$ $+0.49 * 10speed$
-1.73	-0.94	-1.98

Table 3. Attribute Weights using Binary SMO with Linear Kernel

tion.

Whether standard classification models like SVM, built using training data for a particular road, incorporates this road-specific weightage of attributes, is investigated next.

5.2.2 Classification models

We build a training set for each road, having 90 instances for Bhandup, 58 instances for Vikhroli and 60 instances for each of the other four locations. Each instance of the training set has 6 attributes, four honk-based and two speed-based metrics, and 1 class label, based on manual observation. We use WEKA, a widely used open source package for machine learning tools [31]. We input each training set to WEKA and get a binary Sequential Minimal Optimization (SMO) SVM model with linear kernels as output. The models for Bhandup, Vikhroli and Gandhinagar are given in Table 3. We highlight some cells to show that the weights assigned to the attributes are in accordance with our discussion in the previous section. For Bhandup and Vikhroli, the speed based metrics are given more weightage than the honk based. For Gandhinagar, the honk based metrics are given more weightage. Thus standard machine learning tools, can incorporate the road specific metric choice in the traffic classification model for that road, with some amount of training data.

The accuracy values for 10-fold cross validation are also given in the table. This accuracy is minimum (92.7%) for Hiranandani (not shown in the table).

5.2.3 Self-learning on new road

Next we seek to address the problem of absence of any training data, if we deploy our unit on any arbitrary road. Manual collection of ground truth, to build a training model for each individual road, will be cumbersome. Hence we test the unsupervised learning method of K-means clustering on our 6 roads' data. We use classes to cluster evaluation, where class labels of instances in training set are ignored during clustering. After clusters are made, all instances in a cluster are assigned the same class label. In the evaluation phase, assigned class label of each instance is compared with its actual class label, and accuracy values are reported. For Gandhinagar, we obtain 100% accuracy. For Bhandup, Chandivali, Ghatkopar and Powai (Hiranandani), accuracies obtained are between 85.7-94.33%.

For Vikhroli, we obtain accuracy of only 65.52%. This is because the honk based metrics are unsuitable for this road. Using only speed-based metrics, we get 96.55% accuracy. But what metrics are suitable for which road will not be evident without any training. Thus Vikhroli presents a corner case, where our system, without training, will perform poorly.

5.2.4 Temporal variation of traffic

The final question that we seek to answer in this paper is what optimizations and enhancements can we make to our system, based on the temporal variation in traffic state at a particular location. We deploy our unit at Bhandup for six days, Dec1-Dec3 and Dec6-Dec8, from 10.30 am to 9:30 pm. On Dec1, we remain at the deployment location from 10:30am-12:30pm and again from 4pm-9:30pm, to observe the ground truth of traffic state. This manual observation is necessary to have an idea of correctness of traffic pattern reported by our unit on the other five days. Our observations on Dec 1 are listed in Table 4, where 'F' signifies free-flow and 'C' signifies congestion. Video clips, corresponding to each entry in the table, can be found at [2].

Time	State	Time	State	Time	State
10:30am	F	11:00am	F	11:30am	C
12:00noon	C	12:30pm	F	–	–
4:00pm	F	4:30pm	F	5:00pm	F
5:30pm	C	6:00pm	C	6:30pm	C
7:00pm	C	7:30pm	C	8:00pm	C
8:30pm	C	9:00pm	C	9:30pm	C

Table 4. Traffic State at Bhandup on Dec 1, 2010

Our observations from the six days' data, along with their implications in enhancing our system are listed below.
(1) We get meaningful data only between 10:30am-12:30pm and 5:00pm-10:00pm everyday. The other times have hardly any honk detected, indicating the road to be mostly empty. This implies that after identifying the periods when a particular road remains empty, our system on that road can be shut down in those periods. This will save a lot of power.
(2) Some stray minutes can have excessive honks in free-flow due to some temporary reason like a car parking on the road-side. Similarly, congested traffic can have few silent minutes without honks. This implies that, though metric values should continue to be reported to the server every alternate minute for regular updates, the server should look at the data as a time-series, instead of seeing per-minute data in isolation. This will help in removing outliers and take correct decision about traffic state .
(3) During some times of the day, traffic state fluctuates between congested and free-flowing every few minutes. This happens when state is changing from free-flow to congested

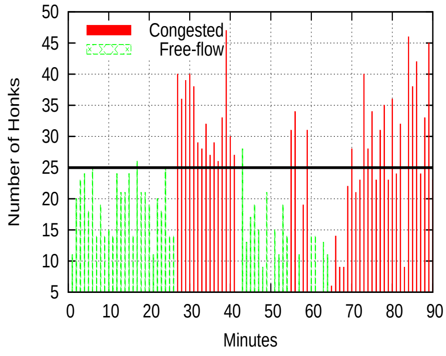

Figure 9. Bhandup, 5:30-8:30pm

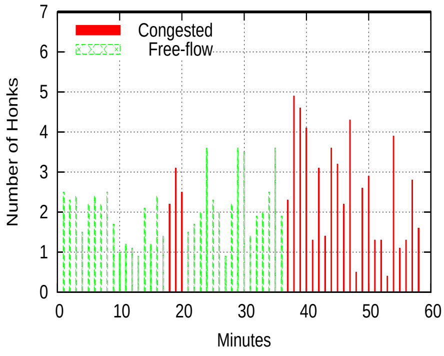

Figure 10. Vikhroli, 5:45-7:40pm

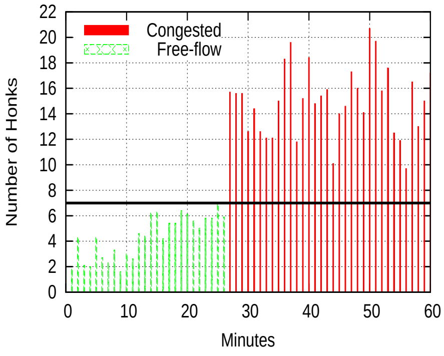

Figure 11. Gandhinagar, 6-8pm

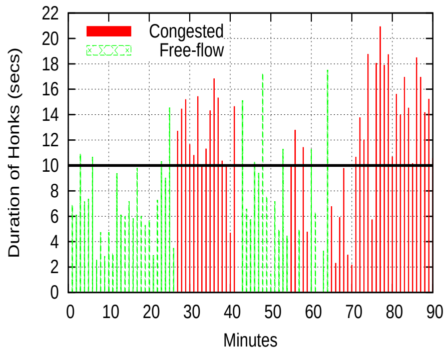

Figure 12. Bhandup, 5:30-8:30pm

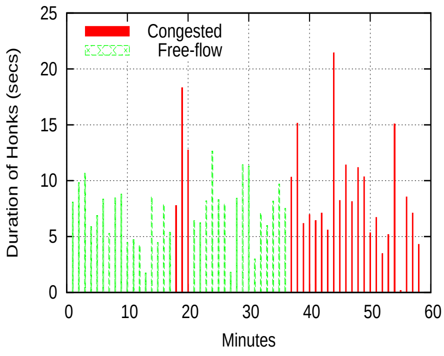

Figure 13. Vikhroli, 5:45-7:40pm

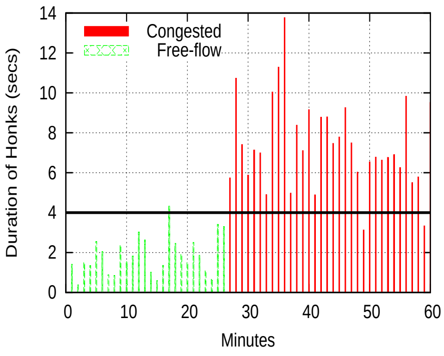

Figure 14. Gandhinagar, 6-8pm

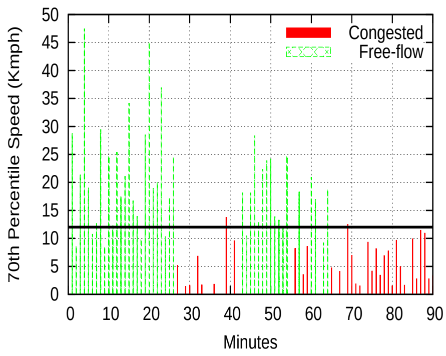

Figure 15. Bhandup, 5:30-8:30pm

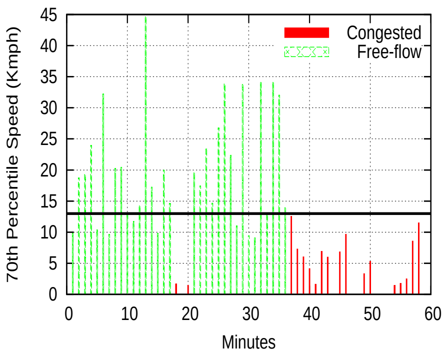

Figure 16. Vikhroli, 5:45-7:40pm

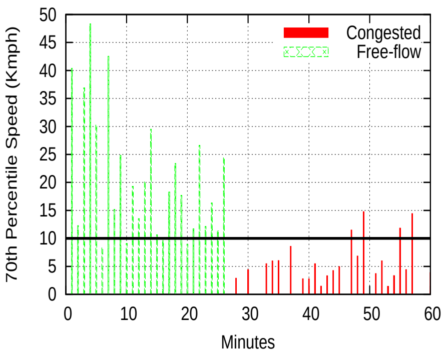

Figure 17. Gandhinagar, 6-8pm

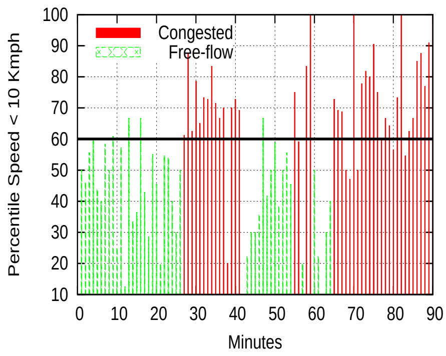

Figure 18. (1), 22/11, 5:30-8:30pm

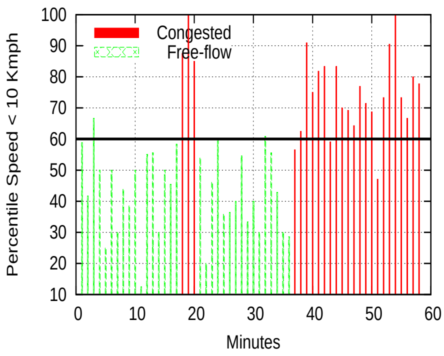

Figure 19. (2), 23/11, 5:45-7:40pm

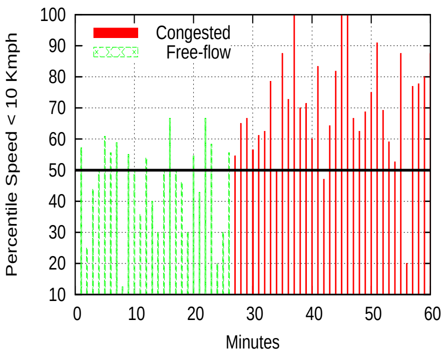

Figure 20. (3), 24/11, 6:00-8:00pm

or vice-versa, as traffic queue buildup and clearance do not happen instantaneously. Traffic is slow at these times. This implies that, time series analysis of per-minute updates at the server will help, where fluctuations over small time intervals can be categorized as a third traffic state intermediate between free-flow and congested states - *slow*.

6 Future Work and Conclusion

Deployment of *RoadSoundSense*, over a large geographical area for long months, is necessary for thorough evaluation of its strengths and weaknesses. This will also help in generating data for the time-series analysis requirements identified in the last section. But large scale deployment will need a more compact and robust prototype version. In our current prototype, the microphone and the FM receiver dangling from audio connectors, make them easy target to be stolen or spoiled by rain. Improper impedence matching between the two mono inputs, sometimes creates problems in the stereo line-in of the DSP module. Most importantly, the 4 AH capacity battery lasted for at most 49 hours in our deployments. Changing batteries every two days is cumbersome. The ideal solution will be to use grid power and keep batteries only for power backup. Switching off the unit when congestion monitoring is not needed and designing a solar-rechargeable photo voltaic circuit to recharge batteries during daytime are other possibilities.

In *Horn-Ok-Please* [3], we proposed an idea of using acoustic sensing to do congestion detection. In this paper, *RoadSoundSense*, we investigate the practical applicability of that idea through prototype development, deployment and data analysis. There are several open questions, as we ourselves have identified, that remain to be answered before large scale adoption of our method. But nevertheless, our results show good promise in solving the complex problem of congestion monitoring in disorderly traffic, existing solutions for which are scarce.

References

[1] http://www.nhai.org/roadnetwork.htm.

[2] http://www.cse.iitb.ac.in/~riju/rss-videos.

[3] Rijurekha Sen, Bhaskaran Raman, and Prashima Sharma. Horn-ok-please. In *Mobisys*, June 2010.

[4] http://www.trafficnoise.org/.

[5] http://www.transportenvironment.org/module-htmlpages-display-pid-20.html.

[6] http://www.conti-online.com/generator/www/com/en/continental/csr/themes/ecology/product_responsibility/tire_road_noise/reifengeraeusch_en.html.

[7] http://www.inderscience.com/search/index.php?action=record&rec_id=6094&prevQuery=&ps=10&m=or.

[8] http://www.soundfighter.com/content.asp?page=11.

[9] http://en.wikipedia.org/wiki/Noise_barrier.

[10] S. Bhattacharyya D. Banerjee, S. K. Chakraborty and A. Gangopadhyay. Evaluation and analysis of road traffic noise in asansol: An industrial town of eastern india. In *International Journal of Environmental Research and Public Health*, Sept 2008.

[11] S. Agarwal and B.L. Swami. Road traffic noise annoyance in jaipur city. In *International Journal of Engineering Studies Volume 1, Number 1, pp. 3946*, 2009.

[12] Abirlal Mukherjee Biva Roy Debashis Chakrabarty, Subhas Chandra Santra and Prasun Das. Status of road traffic noise in calcutta metropolis, india. In *Acoustical Society of America*, 1997.

[13] Brahma D. Tripathi Vinita Pathak and Virendra Kumar Mishra. Dynamics of traffic noise in a tropical city varanasi and its abatement through vegetation. In *Environ Monit Assess: Springer Science + Business Media*, 2007.

[14] Benjamin Coifman and Michael Cassidy. Vehicle reidentification and travel time measurement on congested freeways. *Transportation Research Part A: Policy and Practice*, 36(10):899–917, 2002.

[15] Li Li, Long Chen, Xiaofei Huang, and Jian Huang. A Traffic Congestion Estimation Approach from Video Using Time-Spatial Imagery. In *ICINIS*, 2008.

[16] Sing Y. Cheung, Sinem Coleri, Baris Dundar, Sumitra Ganesh, Chin-Woo Tan, and Pravin Varaiya. Traffic measurement and vehicle classification with a single magnetic sensor. Paper ucb-its-pwp-2004-7, California Partners for Advanced Transit and Highways (PATH), 2004.

[17] http://www.techtransfer.berkeley.edu/newsletter/08-3/vehicle-detection-with-wireless-sensors.php.

[18] Jungkeun Yoon, Brian Noble, and Mingyan Liu. Surface street traffic estimation. In *MobiSys*. ACM, 2007.

[19] http://traffic.berkeley.edu/theproject.html.

[20] Katrina LaCurts Samuel Madden Hari Balakrishnan Sivan Toledo Arvind Thiagarajan, Lenin Ravindranath and Jakob Eriksson. Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In *Sensys*, November 2009.

[21] Prashanth Mohan, Venkata N. Padmanabhan, and Ramachandran Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *SenSys*. ACM, 2008.

[22] http://www.financialexpress.com/news/by-next-year-well-target-10-of-indias\\-smartphone-market/551198/0/.

[23] http://www.made-in-china.com/showroom/rosstone/product-detailioZJEvnDJmVW/China-VHF-Wireless-Tie-Clip-Microphone-WR-601.html.

[24] http://www.ahujaradios.com/Productdetail.asp?prd_id=128&id=6&sc_id=32.

[25] http://focus.ti.com/docs/toolsw/folders/print/tmdx5505ezdsp.html.

[26] http://wm.sim.com/Sim/FrontShow_en/wireless/detail.aspx?cid=6&nid=36.

[27] http://in.mouser.com/search/ProductDetail.aspx?qs=boflksbqthoLObwWTE7%252b%252bw==.

[28] http://www.datasheetdir.com/AP1512AT5+Step-down.

[29] http://www.deocomtech.com/water_proof.htm.

[30] http://focus.ti.com/docs/toolsw/folders/print/ccstudio.html?DCMP=dsp_ccs_v4&HQS=Other+OT+ccs.

[31] Geoffrey Holmes Bernhard Pfahringer Peter Reutemann Ian H. Witten Mark Hall, Eibe Frank. The weka data mining software: An update. In *SIGKDD Explorations, Volume 11, Issue 1*, 2009.