Road-RFSense: A Practical RF-Sensing Based Road Traffic Estimation System for Developing Regions

RIJUREKHA SEN, Department of Computer Science and Engineering, IIT Bombay
ABHINAV MAURYA, Department of Computer Science and Engineering, IIT Bombay
BHASKARAN RAMAN, Department of Computer Science and Engineering, IIT Bombay
AMARJEET SINGH, Indraprastha Institute of Information Technology, Delhi
RUPESH MEHTA, Department of Computer Science and Engineering, IIT Bombay
RAMKRISHNAN KALYANARAMAN, Department of Computer Science and Engineering, IIT Bombay

Unprecedented rate of growth in the number of vehicles has resulted in acute road congestion problems worldwide, especially in many developing countries. In this paper, we present RoadRFSense, a practical RF sensing based road traffic estimation system for developing regions. Our first contribution is a new mechanism to sense road occupancy, based on variation in RF link characteristics, when line of sight between a transmitter-receiver pair is obstructed. We design algorithms to classify traffic states into two classes - free-flow vs. congested, at time scales of 20 seconds with above 90% accuracy.

We also present a traffic queue length measurement system, where a network of RF sensors can correlate the traffic state classification decisions of individual sensors, and detect traffic queue length in real time. Deployment of our system on a Mumbai road gives correct estimates, validated against 9 hours of image-based ground truth. Our third contribution is a large scale data-driven study, in collaboration with city traffic authorities, to answer questions regarding road specific classification model training. Finally, we explore multi-level classification into seven different traffic states using a larger set of RF-based features and careful choice of classification algorithms.

Categories and Subject Descriptors: C.3 [**Special-Purpose And Application-Based Systems**]

General Terms: Sensors, RF, Performance, Model training, supervised learning

Additional Key Words and Phrases: Wireless sensor networks, road traffic monitoring, RSSI, SVM

ACM Reference Format:
Rijurekha Sen, Abhinav Maurya, Bhaskaran Raman, Amarjeet Singh, Rupesh Mehta, and Ramkrishnan Kalyanaraman, 2013.Road-RFSense: A Practical RF-Sensing Based Road Traffic Estimation System for Developing Regions. ACM Trans. Sensor Netw. V, N, Article A (January YYYY), 30 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

The problem of road congestion is currently plaguing most cities in the world. Long traffic queues at signalized intersections cause unpredictable travel times and fuel inefficiency [Yoon et al. 2007; Koukoumidis et al. 2011]. The problem is felt more acutely in growing economies like India, primarily because infrastructure growth is slow compared to the growth in vehicles due to space and cost constraints.

Prior publication: Rijurekha Sen, Abhinav Maurya, Bhaskaran Raman, Rupesh Mehta, Ramakrishnan Kalyanaraman, Nagamanoj Vankadhara, Swaroop Roy, Prashima Sharma, "KyunQueue: A Sensor Network System To Monitor Road Traffic Queues", 10th ACM Conference on Embedded Networked Sensor Systems, SenSys'12, Toronto, Canada, Nov 6-9, 2012.
Author's addresses: Rijurekha Sen, (Current address), School of Information Systems, Singapore Management University; Abhinav Maurya and Bhaskaran Raman, Computer Science Department, IIT Bombay; Amarjeet Singh, Indraprastha Institute of Information Technology, Delhi; Rupesh Mehta, (Current address), 5733, Phillips Avenue,Floor 2, Pittsburgh, Pennsylvania 15217, USA; Ramakrishnan Kalyanaraman, (Current address), 5600, Heather Drive, Apartment "D", Blacksburg, Virginia 24060, USA;

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

Secondly, automated traffic management solutions, developed for western traffic (see Fig. 1), are at best an accidental fit for roads in developing countries. Traffic in many developing regions is distinctly different from western roads in two ways (see Fig. 2): it is (1) non-lane based and (2) highly heterogeneous. Four wheeler heavy vehicles like buses and trucks, four wheeler light vehicles like cars, three wheeler auto-rickshaws and two-wheeler motorcycles ply the same road, intermingled with each other, without any lane discipline [Sen et al. 2011].[1]

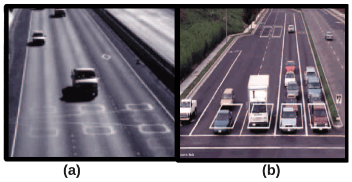

Fig. 1: Lane based traffic

Fig. 2: Chaotic traffic

Fig. 3: Issues with infrared

An ideal traffic sensing system for developing regions has the following stringent requirements: (1) it should sense road occupancy even if traffic is chaotic and non-lane based, (2) should be deployable without interrupting traffic flow, (3) be capable of real time sensing and classification to support applications like traffic signal control and (4) should have low installation and maintenance costs. Section 2 discusses the shortcomings of existing sensing systems like magnetic loops, camera, infrared sensors, acoustic sensors and probe sensors, vis-a-vis these requirements.

In this paper, we present RoadRFSense, a practical sensing methodology for real-time traffic monitoring in developing countries. Our first contribution is a new mechanism to sense road occupancy, based on how RF link quality suffers in absence of line of sight. Our system comprises of an IEEE 802.15.4 transmitter-receiver pair across the road, where the transmitter continuously sends packets and the receiver measures metrics like signal strength and packet reception ratio. We show that these metrics show a strong correlation with the occupancy level on the road between them. Our second contribution is investigation of which wireless link characteristics, what time windows and what algorithms can give a highly accurate *real time* classification of traffic states.

Building on this binary classification of traffic states using one pair of sensors, our third contribution is a system of linear array of such sensors, which can detect the length of a traffic queue on a given road stretch in real time. This system, deployed on a Mumbai road for 9 hours, gives upto 96% accurate queue length estimates.

The fourth contribution of this paper is the exploration of model training overhead of this traffic state classification system for different roads. In collaboration with the traffic authorities [Mapunity] in Bengaluru, India, we deployed RF sensors for 6 weeks during March-May 2012, 12 hours a day, on roads visibly different in width and vehicle types. Analyzing this significant volume of sensor data, and the corresponding video ground truth from traffic cameras [BTIS], we examine the necessity of road specific model training, the suitability of unsupervised learning and the minimum amount of training necessary for accurate classification.

The fifth and final contribution in this paper is the extension of binary traffic state classifications to multilevel classifications into seven different traffic states, using an enhanced set of classification features derived from the RF link characteristics. Our queue length estimates can potentially be used in a range of applications like automated signal control, coordinated signal control in a city or detecting traffic bottlenecks, and our multilevel classification can support applications such as building multi-level traffic congestion maps, all handling the challenges of chaotic non-laned traffic of developing regions.

[1] [Sen et al. 2011] has several representative videos of chaotic traffic at http://www.cse.iitb.ac.in/r̃iju/rss-videos/

2. RELATED WORK

We discuss our related literature in this section, under four different categories - (1) existing traffic sensing systems, (2) use of RF sensitivity to obstacles in different application domains, (3) subject specific model training in the context of traffic sensing and other RF sensing based systems and (4) multi-level traffic state classification. In each category, we also point out the key novelties of our work with respect to prior literature.

2.1. Traffic sensing systems

Automated traffic management solutions have been deployed in many developed countries since several years. In this section, we discuss the traffic sensing methods primarily used to provide input to the traffic management systems and their usability in chaotic traffic.

There are a lot of research projects [Coifman and Cassidy 2002; Varaiya] and several deployed systems [Scats] that do vehicle counting using **magnetic loops** under the road. A vehicle loop detector costs $700 for a loop, $2500 for a controller, $5000 for a controller cabinet and 10% of the original installation cost for annual maintenance [Yoon et al. 2007]. Loops need to be placed under the road. Digging up roads to install and maintain the infrastructure intrudes into traffic flow. Also re-laying the road surface needs re-laying the sensors. Moreover, loops have been traditionally placed under each lane as seen in Fig. 1(a). How should the placement be in absence of lanes is yet to be explored.

Video surveillance based traffic monitoring is fairly common [Video]. [Kastrinaki et al. 2003] gives a comprehensive survey of the major computer vision techniques used in traffic applications. But the traditional setting for which vision algorithms exist can be seen in Fig. 1(b). For usability in developing countries, algorithms are needed for scenarios like Fig. 2. [Quinn and Nakibuule 2010] is a preliminary work on image processing algorithms for chaotic traffic sensing. The algorithms are offline, so the trade-off between computation and communication is not yet understood. Also the sensing accuracy itself has been tested on only 2 minutes of video clip. [Jain et al. 2012] is another recent work to use low quality images from CCTV for traffic sensing. But computational overhead, real-timeliness and accuracy of the designed algorithms are yet to be evaluated. [Sen et al. 2013] estimates both density and speed from non-laned traffic videos. But the computations are offline and video needs to be transferred from road to remote location, for computation intensive video processing. Thus though image-based sensing shows promise, with advantages such as ease of deployment and potential low cost, there are several aspects that need careful evaluation and validation, especially in chaotic traffic conditions.

Active infrared sensors placed across the road suffer beam cuts by vehicular movement in between. [Tirtl] is a commercial product that measures speed, length and lanes of vehicles from beam cuts. However, infrared is overly sensitive to even small obstacles, because the receiver gets a binary result of either receiving or not receiving the infrared beam. When we tried using it on Mumbai roads, pedestrians who frequently use the roads (Fig. 3(a)) or irregularities of the road surface (Fig. 3(b)) would not allow the tx-rx pair to communicate. Also [Tirtl] has been tested only in lane-based traffic. Beam cuts in high density, non-lane based and heterogeneous traffic as seen in Fig. 2, would need careful study to be used for chaotic traffic measurement.

Some recent research is being done to use **acoustic sensors** for traffic state estimation, especially in developing regions, where traffic being chaotic is noisy [Sen et al. 2010]. But standing traffic does not have any uniform sound signature and depends largely on the driver behavior and type of vehicles. For example, if vehicles are standing in a long queue awaiting a green signal, many drivers might just shut down the engines and wait silently or might blow honk impatiently, producing two very different sound signatures for same traffic state. This reduces the sensing accuracy [Sen et al. 2011]. Also, to attain a valid signature correlating to a traffic state, sensing time window of at least a minute or so is needed, making these systems slow in response.

Significant research is being done to leverage **cell phones and participatory sensing** to generate traffic related information. Focus has been given to research both in networking issues [Skordylis and Trigoni 2011; Leontiadis et al. 2011; Skordylis and Trigoni 2008] and machine learning issues [Jing Yuan and Sun 2012; Yoon et al. 2007; Balan et al. 2011]. But participatory sensing data is inherently noisy [Le et al. 2011]. Also probe vehicles might not be present at a given intersection at all times. Travel time estimates, transit vehicle information and congestion maps to be disseminated to commuters for route selection, can tolerate aperiodicity and noise. Such commuter applications are thus highly suitable using probe data.

In comparison to loops, our sensors are to be placed on roadside, without affecting the flow of traffic. We can sense road occupancy even if traffic is chaotic and non-lane based. Our technique is cheap, with each sensor pair costing about $200.

In comparison to acoustic, our technique works in the order of 10-20 seconds. The classification accuracy is also much higher than acoustic, as vehicles being heavy metallic obstacles always affect RF signal quality. This high accuracy also makes it possible, to take the binary classification to queue length measurement using sensor arrays or multilevel classification using an enhanced set of RF based features and classification algorithms.

In comparison to probe sensing, our work is aimed at providing strictly periodic, accurate input to traffic management systems. Our system, based on static sensors, would potentially be used at some key intersections of a city, where intelligent and adaptive control would positively affect traffic flow. This is orthogonal and complementary to the commuter applications like travel time estimation, handled by probe sensing.

In comparison to image sensors, we present an alternative traffic sensing system that has been thoroughly evaluated in chaotic traffic. Also the algorithms are very low overhead and we have implemented them on low end embedded sensor platforms. Sensing and computation are done on the road. Only the computed traffic density values or length of traffic queues, have to be communicated to the remote server, avoiding the communication overhead of video transfer.

In comparison to IR systems, the setup of sensor pair across road in our system uses 802.15.4 radios which are more robust to noise and thus more suitable for disorderly road conditions such as in Fig. 2. This is because the 802.15.4 receivers report a wide range of RSSI values, which are less affected by small obstacles and more by heavy vehicles. This range of decision values can give better inference than the binary decisions from infrared. We also enhance the sensing method with queue length measurement using sensor arrays and explore associated questions of road specific model training and multilevel traffic state classification, aspects not explored in the IR-based systems.

2.2. RF sensitivity to obstacles

It is well known in the wireless networking domain, that characteristics of wireless links like 802.11 and 802.15.4 are affected in the absence of LOS [Shrivastava et al. 2007]. Obstacles cause reflection, absorption and scattering of the RF signal, degrading the link quality. Researchers have exploited these vagaries of RF links in different kinds of application scenarios, the most common being that for indoor localization [Youssef and Agrawala 2005; Chintalapudi et al. 2010; Patwari and Wilson 2010; Xu et al. 2011].

The effects of vehicular traffic on wireless links were briefly examined in [Mottola et al. 2010], where the authors seeked to quantify the link quality in harsh deployment scenarios, like tunnels with traffic moving through them. A small scale study has been done in [Al-Husseiny and Youssef 2012], to differentiate between vehicles and humans using on-road RF sensors. In this paper, we want to apply this effect of traffic on wireless links for a completely novel application, to measure road traffic density.

2.3. Subject specific model training

Road specific training has been considered for acoustic sensing [Sen et al. 2011] based traffic estimation. But the negative results of the work, in terms of high training overhead and low classification accuracy (67%) has been a subsequent deterrent for acoustic sensing. Traditional sensors like loop-detectors and image-sensors however, work on a per-lane basis and the width of a lane is standard for different roads. Thus the question of re-training sensors for different roads is irrelevant in their context. Image sensing based binary classification for non-laned traffic [Jain et al. 2012] needs manual labels to build models, but no work has been reported yet about the manual training overhead for a new road.

The fact that RSSI based sensing needs different classification models depending on the environment is apparent from a different sensing application: RSSI based indoor localization [Bahl and Padmanabhan 2000; Youssef and Agrawala 2005]. To build the indoor localization database, RSSI values of packets from Wi-Fi access points have to be manually collected and labeled for different locations within a building. [Chintalapudi et al. 2010] seeks to reduce the manual overhead by building models based on constraints of RF propagation in place of manually collected RF fingerprints. [Rai et al. 2012] attempts a crowd-sourcing based approach to easily collect the manual labels. The same constraints or crowd-sourcing opportunities (e.g. visitors in a shopping mall for crowd-sourcing indoor localization data) do not arise in the traffic-sensing scenario. We therefore, explore our own alternatives for practical manual supervision in this paper.

2.4. Multilevel traffic state classification

As proposed in NRC[2] manual, the most comprehensive scale of traffic states consists of six ordinal classes (A to F), distinguished by the average congestion delay in seconds per vehicle. Inspite of this theoretical definition, the authors in [Yoon et al. 2007] state that"it is practically impossible to know the perfect ground truth of traffic states, because there is no objective and absolute definition of good or bad traffic that everyone agrees on." This lack of a standard and practical way of measuring congestion is reflected in literature, where anything between two and six congestion classes are considered.

In [Lozano et al. 2009], road images are used to classify traffic into one of the six classes using a supervised algorithm called K-Means based Total Recognition by Adaptive Classification Experiments (K-TRACE). In [Porikli and Li 2004], the authors extract feature vectors from traffic video data and use a Gaussian Mixture-Hidden Markov Model (GM-HMM) to predict two-, three-, and five-level traffic state classification. [Yoon et al. 2007] uses a technique called threshold-based quadrant clustering to characterize and classify binary traffic states and predict incidents. [Pattara-Atikom et al. 2006] employs thresholding on weighted exponential moving average of speeds measured using GPS devices to classify three levels of congestion.

Comparison Parameter	Yoon et. al. [Yoon et al. 2007]	Pattara et. al [Pattara-Atikom et al. 2006]	Lozano et. al. [Lozano et al. 2009]	Porikli et. al. [Porikli and Li 2004]	This Work
Type of Sensors	GPS Probe	GPS Probe	Image	Video	RF Link
Algorithm	Quadrant Clustering	Moving Average	K-TRACE	GM-HMM	SVM
Supervised?	No (Hand-crafted Rule)	Yes	Yes	No (Baum-Welch)	Yes
Data	36 traversals	~3 hours	6 images	75+600 minutes	1170 minutes
Cost	Low	Low	Medium	High	Low
Number of Roads	1 (10.5 miles)	3	1	1	3
Number of Levels	2	3	6	2,4,5	2,4,7
Models Ambiguity?	No	No	No	No	Yes
Accuracy (Levels)	90% (2)	Error: 0.43 (3)	63.30-86% (6)	94-97% (5)	92.25-98.52% (4)

Table I: Comparative overview of related work in the context of multilevel classification

The comparison of this work, with respect to the above mentioned prior works, in terms of number of classification levels, accuracy, cost and scale of study i.e. number of roads and duration in minutes, is presented in Table. I. Also, as seen from the table, the handling of ambiguous or noisy manual labels is a novel contribution of this paper.

[2] Highway Capacity Manual, Transportation Research Board, National Research Council, Washington, D.C., 2000

3. CORE QUESTIONS EXPLORED

In this section, we briefly describe the five core questions explored in this paper.

3.1. Can binary traffic states be quantitatively inferred from RF link characteristics?

We first seek to see if the two traffic states of free-flow and congestion be inferred using RF sensors across road. By *congested traffic*, we mean when vehicles have to brake and stop vs *free-flowing traffic*, when vehicles move according to the driver's intended speed, bounded by the road speed limit. Our basic intuition is as follows. If a wireless transmitter-receiver pair is kept across a road and made to communicate, the link characteristics, observed at the receiver, should be affected by the vehicles on the road. Does road traffic affect wireless links, and can that effect be quantitatively captured using a setup as in Fig. 4, to differentiate the two traffic states? This proof-of-concept study will be the first question under exploration.

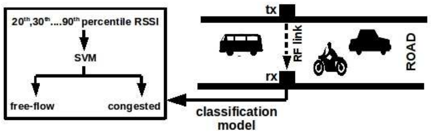

Fig. 4: Wireless communication across road

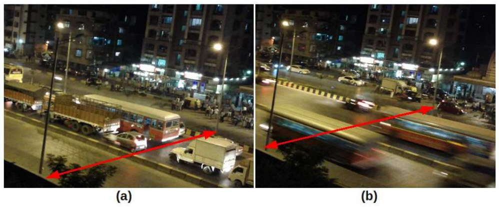

Fig. 5: Mismatch of sensing and traffic state

3.2. Can binary traffic state classifications be done in near real time?

From our observations of Mumbai and Bengaluru traffic lights, traffic signal cycles typically last for about a minute. This one minute cycle time is divided into slots, in which different contending flows get their respective green times. Green time for any flow lasts for about 10-30 secs, though it can go over a minute for critical flows.

Any system like ours, aiming to provide traffic state information to traffic lights, would need an input parameter about the frequency at which traffic density or queue length estimates are needed. We wish to determine the lowest classification time window at which a sufficiently high accuracy (say 90%) can be obtained. Very low time windows give noisy predictions. This noise comes from two sources - (a) the inherent stochastic nature of wireless links which causes link quality to be intermittently bad though the tx-rx are in perfect line of sight (b) the instantaneous traffic condition between the tx-rx are contrary to the actual traffic state. For e.g. tx-rx may be in line of sight between several standing vehicles in congestion (Fig. 5(a)) or several heavy vehicles may pass the tx-rx pair simultaneously in free-flow obstructing line of sight (Fig. 5(b)). Thus we need to choose the prediction time window, henceforth referred to as t, carefully.

Also, for a given value of t, machine learning algorithms for traffic state classification have to be carefully chosen, to balance ease of model training vs. accuracy tradeoffs.

3.3. Can length of traffic queues be estimated aggregating multiple traffic state information?

We seek to extend this pairwise sensing of binary traffic states to build an array of sensors, which can perform co-ordinated sensing and detect length of traffic queue on a given stretch of road. In the context of our goal of traffic queue length measurement, an important point to note is that, at almost all intersections in developing countries, a variety of vehicles stand as a coagulated mass waiting for green signal, as shown in Fig. 2. On getting green signal, they all move forward almost bumper to bumper, with little variability in speeds. So queue length is proportional to traffic volume, the road width being the proportionality constant. We thus assume, as is intuitive, that finer granularity information like exact vehicle counts are unnecessary. To tune traffic lights proportional to waiting traffic volumes, queue lengths are sufficient. Thus we aim to detect road occupancy in chaotic traffic, and build upon that to detect length of queues.

The proposed system architecture is shown in Fig. 6.

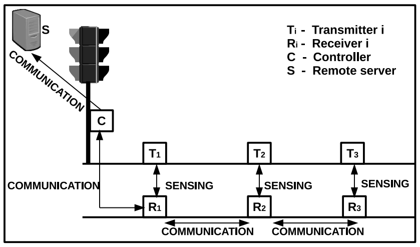

Fig. 6: System Architecture

(1) Three pairs of transmitter (T_i) - receiver (R_i) are shown as example. Each pair performs sensing and computation to know the traffic condition between them. The number of pairs to be placed on a given road stretch, would depend on the worst case length of traffic queues on that road.
(2) The individual observation of each T_i-R_i pair, can be communicated to a central controller unit (C), that may reside on the traffic light. C, upon receiving the road occupancy observation values from sensors on each incoming lane, can compute the optimal green light distribution. This can handle applications local to a particular intersection. e.g. minimizing worst case waiting delays.
(3) C can communicate with a server (S), that may reside in a remote traffic control office. The server, upon receiving road occupancy information from several controller units of the city-wide road network, can implement other applications like co-ordinated signal control, bottleneck identification and congestion mapping.

3.4. What is the overhead of classification model training on different roads?

When an RF based traffic sensor is placed on a new road, it would need a binary traffic state classifier for that road to be programmed on it for classifying traffic states in real time. This is shown in Fig. 4, as classification model pointing to an SVM classifier. Hence we also explore the question of training overhead for such classification models for different roads. Will road characteristics like width and vehicle type affect the RSSI values, on which this sensing mechanism is based? In that case, does the binary classifier need to be trained separately for different roads? Is manual supervision to train the classifier a must or will unsupervised methods like clustering work well? And if manual supervision is a necessity, how can that be provided in practical deployment scenarios? We analyze large scale empirical data in in collaboration with city traffic authorities to answers to explore this question.

3.5. Can binary traffic state classification be extended to multilevel classification?

The final question that we explore is whether this RF sensing mechanism can handle multilevel traffic state classification. Roads typically show more than two traffic states of free-flow and congested. They might be completely empty or have fast moving traffic. Vehicles might be moving slowly or waiting in a stand-still queue. Occupancy, in fact, can be thought to increase continuously from an empty road to a completely packed road. A more precise classification of this occupancy level into multiple traffic states can therefore add more information to applications like congestion maps. Also the binary decisions correlated across sensor pairs to estimate queue length, can be enhanced with multilevel decisions, to detect the tail of the queue more accurately. Is multilevel classification even possible, in chaotic traffic, using a sensing mechanism based on RF? This is a significant question, especially given the vagaries of RF propagation.

4. HARDWARE PLATFORMS AND PROTOTYPES

To explore the different questions under consideration in this paper, we need hardware platforms and prototypes to be deployed for on road experimentation and data collection. In this section, we outline the specific hardware requirements for each question under exploration, and present the details of the off-the-shelf platforms and implemented prototypes used in the studies.

4.1. Can binary traffic states be quantitatively inferred from RF link characteristics?

For this basic proof-of-concept study, we keep two 802.15.4 compliant Telosb motes across the road, one as transmitter (tx) and the other as receiver (rx), on a line perpendicular to the length of the road, as shown in Fig. 4. The tx sends 25 packets per second, each having a payload of 100 bytes, at -25dBm transmit power. The rx logs the number of packets received and Received Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI) for each received packet.

4.2. Can binary traffic state classifications be done in near real time?

The same setup as for the previous question, with two 802.15.4 compliant Telosb motes across the road, one as transmitter (tx) and the other as receiver (rx), are used for this study.

4.3. Can length of traffic queues be estimated aggregating multiple traffic state information?

For this question, off-the-shelf Telosb motes are not suitable and customized prototypes have been designed and implemented. The reasons for the necessity of customized prototypes, and the details of the platforms built are given below.

In our envisioned architecture as shown in Fig. 6, we have two types of wireless links – (1) *sensing links*, across the road, from T to R and (2) *communication links*, along the road, from one R to another R. The *sensing links* should be affected by the traffic flow on the road, so that wireless link characteristics measured on them reflect the traffic volume. This necessitates the radios on T and R to be at a low height of about 0.5 m from the ground level, such that the packets are blocked by the body of the vehicles. On the other hand, in a typical deployment scenario, the R units will be mounted on road-side lamp-posts. Inter lamp-post distance is in the order of 30 m. Even if we put our units on each lamp-post, any network fault might need units on alternate lamp-posts to communicate. Thus, we should keep provision for at least 60 m long *communication links*. These links have to be reliable as well. The question thus arises: can a single radio handle both sensing and communication links with conflicting requirements? This is verified next, through a set of experiments.

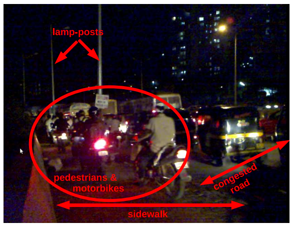

Fig. 7: Obstacles on sidewalk: pedestrians and impatient motorcyclists too!

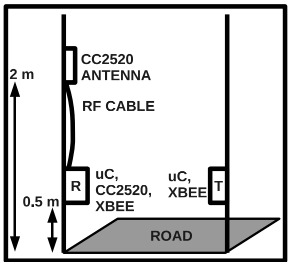

Fig. 8: Sensor pair setup

We keep a Telosb mote stationary which transmits 25 packets per second. Another mote is kept at distance x m from the transmitter; we vary x=30 m to x = 100 m, in steps of 10 m, logging number of packets received for 5 minutes at each distance. Both motes are 0.5 m above the ground. They are placed along the road, on the sidewalk of Adi Shankaracharya Marg, instead of being placed across the road like in previous experiments, since we want to evaluate the communication and not the sensing links. We do the experiment once at 6 am in the morning, when the sidewalk is empty and again at 8 pm in the evening, when the sidewalk is crowded with pedestrians (see Fig. 7) and repeat both experiments over four days.

The median packet reception ratio (PRR) is 60% at 60 m at 6 am. The link quality degrades at 8 pm when pedestrians on the sidewalk block the line of sight between the motes. Median PRR is at most 40% at any distance above 30 m. Such low PRR would make our communication links unreliable, and hence unsuitable for applications like adaptive traffic signal control, where timeliness is critical.

To resolve this issue, we choose to use *two* 802.15.4 radios in our receiver (R) units - one XBEE radio for sensing and a CC2520 radio for communication. The setup for a tx-rx pair is shown in Fig. 8. T is the tx unit across the road containing a micro-controller and an XBEE radio to transmit sensing packets. R is the rx unit with a micro-controller, an XBEE radio to receive sensing packets and a CC2520 radio to communicate to other R's. The CC2520 antenna is placed higher using RF cable, at about 2 m from the ground, clear of pedestrians on the footpath. Other possible design choices to handle the sensing-communication conflict, are discussed in Section 8.

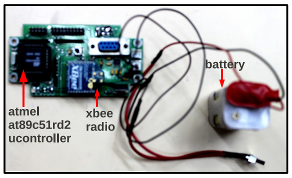

Fig. 9: Transmitter (T)

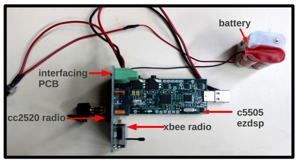

Fig. 10: Receiver (R)

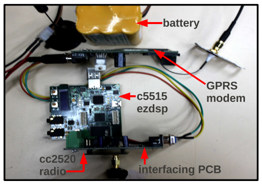

Fig. 11: Controller (C)

Based on the design choices outlined above, we have implemented three hardware prototypes for T, R and C. The T units have (1) an Atmel AT89C51RD2 micro-controller and (2) an XBEE radio on UART to transmit sensing packets. In the R units, we have used (1) a TI C5505 ezdsp stick for the computation, (2) a TI CC2520 radio connected on SPI for communication with other R units and the C unit and (3) an XBEE radio on UART for sensing. Four 1.5V batteries are connected in series to provide 6V power to each circuit. The C unit has (1) a TI C5515 ezdsp stick for processing with (2) an integrated Micro-SD card for data storage. There is (3) a TI CC2520 radio on SPI to communicate with the R units and (4) a SIMCOM SIM300C GPRS modem to communicate to a server. An 11.1V, 1 Amp peak current battery is used for powering up this circuit.

The choice of the TI ezdsp sticks are for their small form factors, low cost, low power requirements and convenient pinouts of several interfaces like UART and SPI. These make hardware integration easy. The vast resource of TI chip support library functions and up to 100 MHz clock rates make programming on these platforms convenient as well. The two IEEE 802.15.4 compliant radios, needed in the R units, are chosen to be CC2520 and XBEE. This is because they use two different interfaces of SPI and UART respectively, which simplifies prototype design and implementation.

A C5505 ezdsp stick costs $50, a C5515 ezdsp stick costs $80, a GPRS modem costs $70, an XBEE radio costs $18, a CC2520 radio costs $50, an AT89C51RD2 micro-controller costs $4. With interfacing PCB's, connectors and batteries, a receiver (R) - transmitter (T) pair costs about $200 and the controller (C) costs about $250.

4.4. What is the overhead of classification model training on different roads?

Our goal for this question is to explore the classification model for a single sensor pair, and how to train such a model. Thus unlike the requirements in of communication among multiple receivers to measure queue length, here the focus is on long-term RF data collection and storage, for off-line analysis. Hence we build the appropriate hardware platforms to suit the requirement of long-term data collection.

The transmitter (tx) unit (as shown in Fig. 9) is the same, as used in the queue measurement system. It has (1) an Atmel AT89C51RD2 micro-controller and (2) an XBEE radio on UART to transmit packets. The receiver (rx) unit (as shown in Fig. 12), is however different. It has (1) a TI C5515 ezdsp stick with in-built support for Micro SD card, to store the RSSI values of received packets, (2) an XBEE radio on UART to receive packets and (3) a SIM548C GPS-GSM module, that is used to initialize the RTC and time-stamp the collected packets according to GPS time and query the status of the hardware unit remotely using SMS to the GSM radio. Both units are packaged in waterproof ABS plastic boxes and clamped to lamp-posts across road.

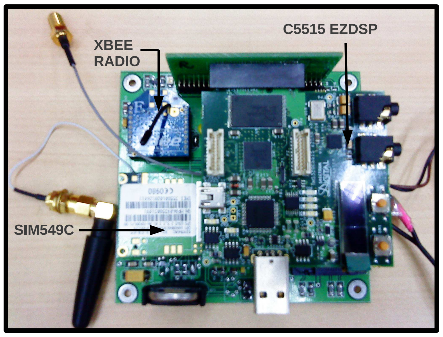

Fig. 12: Receiver (rx)

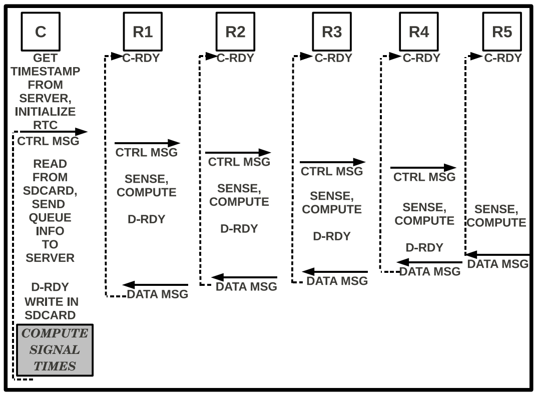

Fig. 13: Software flow

4.5. Can binary traffic state classification be extended to multilevel classification?

We place 802.15.4 compliant Telosb motes as used in exploration of first two questions for this study. The choice of Telosb instead of using XBEE radios as in the third and fourth questions, is to explore the possible use of LQI feature, along with RSSI based features, in multilevel traffic state classification. XBEE radios, unlike Telosb do not report LQI velues of received packets, and hence the choice.

5. ALGORITHMS AND SYSTEM SOFTWARE

For each question under exploration, there is a software component involved in the study. These include exploration of different machine learning algorithms for traffic state classification into binary and multiple levels, and also communication protocols among multiple receivers and controller in the queue estimation system. In this section, we outline the specific software requirements of each question and describe the designed and implemented softwares in detail.

5.1. Can binary traffic states be quantitatively inferred from RF link characteristics?

There is no specific software requirement for this proof-of-concept study. CDF of RSSI information, of 802.15.4 packets received over some time window, are plotted for both congested and free-flowing traffic. The nature of the CDF and the differences of CDF between the two traffic states are visually explored, to see if road traffic affects RSSI and whether the effect is quantifiable or not.

5.2. Can binary traffic state classifications be done in near real time?

To decide what binary classifier to use for real time traffic state classification, the trade-off is between (1) accuracy of classification, (2) implementability on a low end embedded platform, (3) complexity of the classifier models and (4) overhead of model training. Linear hyperplane classifiers are simple enough to implement on our platform (TI's C5505) and to train and test in near real time. SVM and K-Means-based classifiers belong to this category and are state-of-the-art supervised and unsupervised learning algorithms respectively. We build four possible algorithms based on these two classifiers and subsequently choose one based on accuracy and labeling overhead.

FeatureClassifier (FC) algorithm - In this algorithm, the wireless data for a certain t is transformed into a feature vector comprising 8 features: from the set of RSSI values of packets received in a time window, the eight percentile values corresponding to 20^{th}, 30^{th}, ..., 90^{th} percentile are drawn as features. The ground truth for the corresponding time window is appended to the generated feature vector. If no packet is received in a time window, a dummy packet having RSSI of -95 dBm, close to the radio sensitivity level, is considered to have been received. The collection of all data points thus obtained comprises the training dataset. This is used to train classification models either using SVM or K-Means. In the testing phase, similar feature vectors are created from wireless data over the same time window t. Then each t is labeled as free-flow or congested based on the training model.

SignalClassifier (SC) algorithm - This meta-classifier alternative to FeatureClassifier, uses majority voting on per-packet congestion predictions. In the training phase, we obtain a per-packet classifier using K-means or SVM, by considering only the packet RSSI as a feature. In the testing phase, for each t, we employ the per-packet classifier obtained in the training phase on each received packet, obtaining a label for each packet. Consider $count^{(t)}_{congestion}$ and $count^{(t)}_{freeflow}$ to be the number of packet-level congestion and free-flow predictions in a time slot t. We predict congestion in the time slot if $count^{(t)}_{congestion}$ is greater than or equal to $count^{(t)}_{freeflow}$ and free-flow otherwise.

5.3. Can length of traffic queues be estimated aggregating multiple traffic state information?

For this question, the software component required is a communication protocol among the multiple receivers (R) and the controller unit (C). To correlate the traffic state decisions of different R units and calculate the queue length, we have to ensure that all R's perform sensing and computation simultaneously, so that their individual measurements are co-ordinated in time. Since our architecture has a C unit, we choose to use centrally controlled measurement cycles, triggered by C, to achieve this. Thus we do not need any explicit time-synchronization mechanism among the units.

One possible way to design the software flow of our network is outlined in Fig. 13. The R units remain in receiver ready mode of CC2520 radio (C-RDY) on power up, waiting for a control message. C, on power up, asks for the current time from a server over GPRS and initializes its real time clock. C then sends a control message which each R receives and transmits to the next R. After transmission, each R starts sensing the incoming packets from T on the XBEE radio and computes its binary decision of traffic state based on the algorithm described in Section 5.2. Once this decision is ready, each R enters receiver ready mode of CC2520 radio (D-RDY), waiting for the data message from next R. The last R creates a data message to send its decision to the previous R. Each R, upon receiving the data message from next R, appends its own decision to the message and transmits it to the previous R.

When the data message reaches C, it writes the data message, along with the time of its reception, in the Micro-SD card. This ensures retention of data and control state information, in case C reboots or server communication goes down temporarily. *C can compute the traffic signal schedule from the queue length information in the data message.*[3] All R units go back to C-RDY after transmitting data message. C appropriately sends the next control message, when it intends to start the next

[3] The italicized parts of the software flow like traffic light schedule computation and co-ordinated traffic signal control have not been implemented. Negotiations for these are currently under progress with [Mapunity].

measurement cycle and this goes on in a loop. While the R's do sensing and computation, C reads the Micro-SD card and sends the data message of the previous cycle to the server over GPRS. The server can use this information *from different C units in the city for co-ordinated signal control* or for visualization of congestion maps.

Next we consider the MAC protocol to use in our network. TDMA needs strict time-synchronization among units. On the other hand, by design, both our control and data messages are transmitted sequentially by one R followed by the next R. Thus simple CSMA-CA can handle our MAC issues, and this is what we use in our network. To increase reliability, all messages are transmitted four times. If there is still a message loss, our design handles it by using a timeout in D-RDY state. Upon timeout, the unit goes back to C-RDY state and participates in the next measurement cycle when the next control message comes. Thus the fixed number of retransmissions allows us to achieve a good balance between resilience to stray wireless losses and implementation complexity.

C keeps track of the current measurement cycle number by generating and inserting a sequence number in the control message. If C reboots, it looks up the last sequence number from the Micro-SD card and generates the next one. If an R reboots, it simply waits in C-RDY and copies the sequence number of the first control message it gets, as the current sequence number. None of the R's can generate a sequence number. This ensures that though the sequence numbers wrap around after 0-255, there is no stale sequence number in the network. Thus units can confidently reject messages containing sequence numbers already seen or which are out of order, as retransmitted messages.

If the C and R nodes are arranged along a road, as shown in Fig. 6, this software protocol uses links between C and R_1, the first R_i along the road, and then between each R_i and R_{i+1}. Our system, using dual radio, has provision for longer communication links between R_i and R_{i+2}, which the current software does not utilize. All the messages are routed along the hardwired path of consecutive $\{R_i,R_{i+1}\}$ pairs. In future, the longer links may be used for RSSI based self-localization or fault detection, where if a particular R_i fails or link between any consecutive $\{R_i,R_{i+1}\}$ fails, it can be detected using the longer links.

5.4. What is the overhead of classification model training on different roads?

This is a data analysis question, and supervised classification algorithms like SVM and unsupervised algorithms like K-Means are evaluated on large datasets collected from multiple roads to analyze (1) the differences between classification models trained on different roads (2) the issues with unsupervised clustering in this context and (3) the minimum amount of manual supervision necessary to achieve good accuracy. The details of the data analysis process are described along with the analysis results in Section 7.

5.5. Can binary traffic state classification be extended to multilevel classification?

To explore this question, we first examine if multilevel classification is within the capabilities of an RF sensing system. Fig. 14, Fig. 15 and Fig. 16 show plots of the top-2 principal components in the four-level classification datasets for the three datasets - ADI_4L, KJR_4L, and SHR_4L. The details of these datasets will be given in Section 6.

Principal Component Analysis (PCA) is an unsupervised technique for discovering structure in a statistical dataset, specifically for finding the orthogonal basis vectors along which most of the variance in the dataset is distributed. Visualizing the top-2 principal components clearly shows the distinct regions and separability of various traffic states, indicating that multi-level classification using wireless signals is indeed feasible. As the difference between the classes can be seen using an unsupervised algorithm on our datasets, the inter-class variation must be an inherent characteristic of the RF data.

For multilevel classification, we experiment with a set of standard machine learning algorithms - the generative Naive Bayes algorithm and the discriminative Logistic Regression and Support Vector Machines (SVM) algorithms. For each of these three algorithms, we explore the three different settings possible for multilevel classification - *multiclass*, where a single classifier is trained

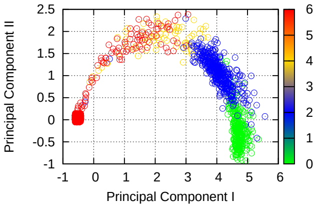

Fig. 14: PCA for 4 levels at *ADI*

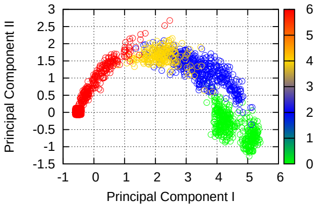

Fig. 15: PCA for 4 levels at *KJR*

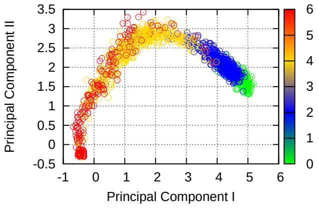

Fig. 16: PCA for 4 levels at at *SHR*

to distinguish all k classes, *1-versus-1*, where $\binom{k}{2}$ classifiers are trained to distinguish between each pair of the k classes and *1-versus-all*, where k classifiers are trained to distinguish each of the k classes from the remaining $k-1$ classes. We also experiment with two ordinal regression algorithms by Chu-Keerthi [Chu and Keerthi 2005] and Mark-Hall [Frank and Hall 2001]. The list of all algorithms evaluated for multi-level classification is given in Table II.

Algorithm	Acronym
Naive Bayes 1-versus-1	NB_1v1
Naive Bayes 1-versus-all	NB_1vR
Naive Bayes MultiClass	NB_MC
Logistic Regression 1-versus-1	LR_1v1
Logistic Regression 1-versus-all	LR_1vR
Logistic Regression MultiClass	LR_MC
Support Vector Machine 1-versus-1	SVM_1v1
Support Vector Machine 1-versus-all	SVM_1vR
Support Vector Machine MultiClass	SVM_MC
Mark-Hall SVM Ordinal Regression	SVOR_MH
Chu-Keerthi SVM Ordinal Regression	SVOR_CK

Table II: List of Multilevel Classification Algorithms

Though non-parametric algorithms like kernelized SVMs may yield better accuracies than their parametric counterparts, the space complexity of these models grows with the size of the training dataset, making them unsuitable for deployment in low-memory embedded systems. Moreover, the linear separability of traffic classes were apparent from the PCA analysis, reducing the necessity of non-parametric algorithms for our datasets.

In seven-level classification, the zero-one inter-label loss causes every wrong prediction to be counted as an error. This does not consider the fact that we have some *mixed* states in our dataset, where even a human observer has difficulty in discerning the correct traffic state. In such cases, it is unfair to expect the classification algorithm to work correctly.

SVM MultiClass[4] allows us to specify custom inter-label losses $\Delta(m,n)$ and to use these losses for margin scaling in learning a multi-class max-margin classifier. In learning the SVM MultiClass model, we would like to *not* impose a margin constraint between the score of a mixed state $(w_{y_i} \cdot x^{(i)})$ and that of one of its adjacent pure states ($w_y \cdot x^{(i)}$ where w denotes weights, $x^{(i)}$ denotes the features of the i^{th} instance and y is a pure state adjacent to the mixed state y_i). In prediction, we would like to incur *no penalty* on accuracy if a data point labeled with mixed state is classified as one of the

[4] http://svmlight.joachims.org/svm_multiclass.html

adjacent pure states. For example, classification of an EF data point as E, EF, or F should not be counted as an error. However, classification of the EF data point as FC, C, CS, or S should be counted as an error in calculating accuracy.

6. DATASETS USED IN STUDY

To empirically explore the questions under study, we collect extensive data from real roads in course of our study. The datasets are collected according to the requirements of the particular question under consideration, using appropriate hardware, ground truth and road settings. This section gives a brief overview of the data collection, preprocessing and labeling stages.

6.1. Can binary traffic states be quantitatively inferred from RF link characteristics?

To answer this proof-of-concept question, we create a setup shown in Fig. 4 on Adi Shankaracharya Marg, a road in Mumbai, about 25m wide in each direction. One person stands on the roadside footpath holding the rx and another stands across the road, on the road divider, with the tx, both rx and tx being at a height of about 0.5 m from the ground. These two persons also observe the road to note the ground truth of the traffic situation. We collect 14 logs of 5 minutes each from about 5:30 pm to 7 pm.

6.2. Can binary traffic state classifications be done in near real time?

To evaluate our choices of features and algorithms for real time traffic state classification and also the prediction time window t over which the features are to be computed, we need a data-set with labeled ground truth. For this purpose, we use a 16 hours data-set, collected using the setup in Fig. 4. The data is collected from two Mumbai roads, a 25 m wide Adi Shankaracharya Marg, henceforth referred to as wide-road and another road, 8 m in width, henceforth referred to as narrow road. Specifically we have 13676 secs of wide-road data labeled as free-flow and 14992 secs labeled as congested. Similarly, we have 13486 secs of narrow-road data labeled as free-flow and 16678 secs labeled as congested. The labeling was done at a larger time-scale of 5 minutes to reduce manual overhead and all the one second long windows, belonging to the same 5 minute window were uniformly labeled. The roads and the times of day of data collection were chosen in a way that traffic states did not toggle within 5 minutes. Thus the error in ground truth observation, even if present, is very small. Representative videos, showing free-flow and congested traffic on the wide road, can be found at [War-videos].

6.3. Can length of traffic queues be estimated aggregating multiple traffic state information?

To evaluate the accuracy and timeliness of queue estimates given by our network of sensors, we perform a set of experiments on a stretch of Adi-Shankaracharya Marg road in Mumbai. This road is about 25m wide and has fair amount of traffic throughout the day as it connects two express highways.. We deploy one C unit on a lamp-post near to the traffic signal and five T-R pairs on the next five consecutive lamp-posts. All units are packaged in ABS plastic boxes with holes to bring out the XBEE and CC2520 radio antennas. The C and the R units are clamped to lamp-posts on the sidewalk, at about 0.6m above the ground. Their CC2520 antennas, connected to the radio external ports with RF cables, are tied vertically to the lamp-posts at 2m above the ground. The C unit has an additional GPRS antenna. The T units are clamped to lamp-posts on the divider, perpendicularly opposite to the R units, such that a T-R pair face each other across the road. The deployed units and the deployment site are shown in Fig. 17.

We implement our designed software for this deployed network, such that C initiates a new measurement cycle through a control message every 30 secs. The T-R pairs are programmed to send the XBEE packets for sensing on different 802.15.4 channels to avoid interference[5]. The R units sense for 20 seconds each and compute individual binary decisions about the traffic state. These decisions are communicated to C in a data message. C stores the message in Micro-SD card and also sends

[5] The interference issue is discussed in more detail in Section 8.

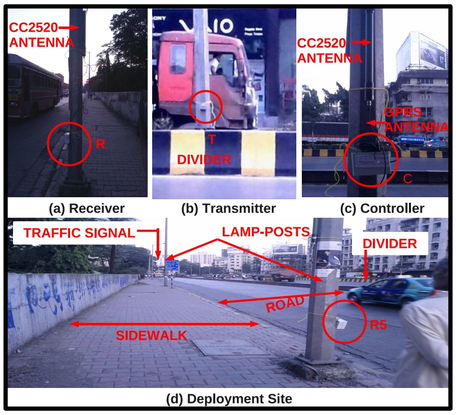

Fig. 17: Deployed Units

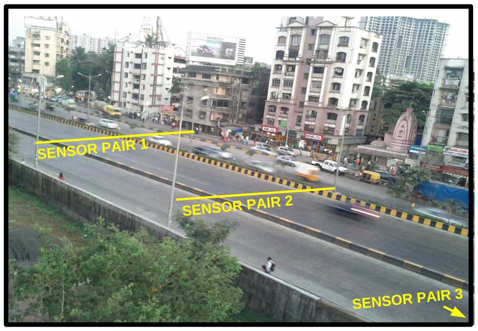

Fig. 18: View from Android Phone

it over GPRS to a remote server. The message is in the form of an array of 5 binary values, each signifying decision by an R, in increasing order of the lamp-posts from the C unit. The server logs these updates coming every 30 seconds and computes the queue length as the unit number of the last R reporting congested state. Thus our measured queue lengths can take 6 discrete values: 0, where all R's report free-flow, 1 when only R1 reports congestion while the others report free-flow, 2 when R1 and R2 report congestion while others report free-flow, 3, 4 and finally 5, when all R's report congestion. We run this deployed network on Nov 17, Thursday, Nov 18, Friday and Nov 19, Saturday, 2011, for 3 hours everyday, between 6-9 pm.

To know the accuracy of our measurements, we use an image-based manual verification scheme. We run an Android application on a Samsung Google Nexus phone to capture an image every 30 second and store it. The phone is placed on the roof of a four storeyed building by the road-side. The phone can cover T-R pairs 1, 2 and 3. We tried different apartment buildings by the roadside and different orientations and zoom-levels on the phone, but this was the maximum number of sensors that we could cover. The view from the phone is shown in Fig. 18. In the figure, C is further to the left of sensor pair 1 and T-R pairs 4, 5 are further to the right of sensor pair 3. The images for the three days can be viewed at [War-videos]. One person observes the images offline and estimates the queue lengths manually. In case this observer finds it difficult to estimate the length from the image, a second observer is consulted and the queue length is ascertained by their mutual consent.

6.4. What is the overhead of classification model training on different roads?

To empirically analyze the questions pertaining to model training, the primary requirement is to collect sensor data meeting certain criteria. The first criteria is to have data from a variety of roads, varying in width and vehicle types, so that the differences in models among different roads can be quantified. The second requirement is to collect some form of ground truth about the traffic state, simultaneously during the periods of sensor data collection, to label the sensor data for post-processing. The third requirement is to collect data for sufficiently long duration, so that the temporal variations in wireless signal due to temperature, rainfall and other environmental factors, are captured.

To meet the above requirements, we built a collaboration with the traffic control authorities [Mapunity] in Bengaluru, who have deployed about 180 PTZ cameras in different road intersections. These cameras are wire-connected to a central control room, where traffic personnels manually observe the incoming video feeds to detect incidents and bottlenecks.

We put up one tx-rx pair each near the intersections of **(1) Aurobindo Circle**, a 10 m wide road in a residential area, where only light vehicles like private cars, auto-rickshaws and motorbikes ply, **(2)**

JD Mara junction, an 8 m wide road where all vehicles including heavy trucks and buses ply and **(3) Udipi junction**, a 25 m wide road with all vehicle types. These locations are carefully chosen to test the model differences with variation in road width and vehicle types. We will henceforth refer to these roads as *AURO, JD* and *UDI* respectively. The approximate positions of the sensors at the deployment intersections are shown as light colored rectangles in Fig. 19 and Fig. 20. Data is collected for six weeks between March 25 and May 6, 2012, for about 12 hours everyday. The units are switched on at 8 am in the morning and switched off at 8 pm in the evening, followed by reading out the stored data in the SD card to a laptop and necessary battery replacement.

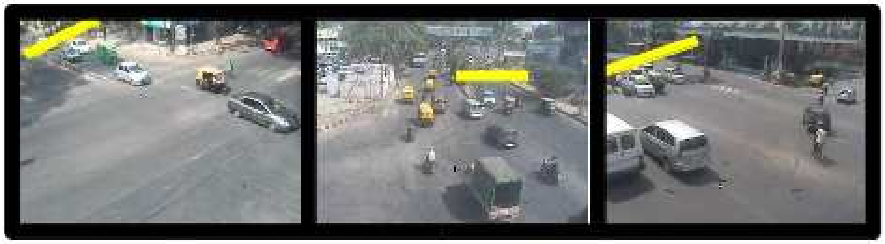

Fig. 19: Free-flow at *AURO, JD* and *UDI*

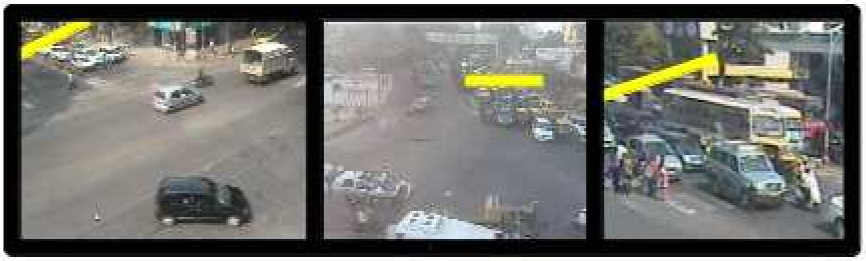

Fig. 20: Congestion at *AURO, JD* and *UDI*

Fig. 19 and Fig. 20 show snapshots from the video-feeds of the traffic cameras at the deployment intersections. These video-feeds are obtained from the traffic control room to be used as ground truth. If the video shows no traffic queue at the sensor location (as in Fig. 19), the ground truth is noted to be free-flow traffic. If the traffic situation is as shown in Fig. 20, and queued vehicles either stand or move bumper to bumper at the sensor location, the ground truth is noted to be congestion. The GPS time-stamp on the sensor data and the NTP time-stamp on the video feeds are used to time synchronize the sensor and the video data. From the central control room, the traffic cameras are rotated and focused on different incoming flows at the intersection. Thus only the videos, with the camera focusing on the road where our sensors were deployed, could be used for data labeling.

We take sensor data of prediction time window $t = 20secs$. The data contains the RSSI values of all packets obtained within that interval. If this interval has a consistent ground truth label as one of the two states of free-flow or congested, and there has not been any state transition within the interval, we compute the 20^{th}, 30^{th}, ... 90^{th} percentiles of the RSSI values. If no packet has been received in the 20 seconds, a dummy packet with minimum possible RSSI for XBEE radio of -94 dBm is considered. These 8 RSSI percentiles as features and the ground truth label as class, form

Road	Dates in Apr, 2012	Freeflow data-points	Congested data-points	Total data-points	Total hours
AURO	2,5,12,17,20	4259	2977	7236	40.2
JD	2,3,10,12,13,18,21	4153	2997	7150	39.7
UDI	5,14,15,17,18,19	4728	2491	7219	40.1

Table III: Labeled dataset

Road	Traffic State							Total
	E	EF	F	FC	C	CS	S	Data
ADI	2808	8940	4015	25	768	7341	2931	26828
KJR	3926	3571	4413	0	2074	2401	8457	24842
SHR	2880	56	4772	1108	4203	3398	2144	18561

Table IV: Labeled data per Traffic State (in seconds)

a data-point in our subsequent analysis. From the six weeks of collected data, we choose about 40 hours of data for each road, to manually label with ground truth.[6] The total number of such labeled data-points and the dates over which they were collected, are summarized in Table III.

6.5. Can binary traffic state classification be extended to multilevel classification?

For binary traffic state classification, data is collected mostly at road intersections. This is because at intersections, traffic states are typically binary. If there is a traffic queue waiting for green signal, then traffic state is congested with vehicles either standing or moving bumper to bumper. If there is no queue, then traffic is free-flowing, with vehicles moving fast either to cross the intersection within the available green cycle, or to move and stand ahead of other vehicles in a red cycle. The multiple traffic states that we wish to examine in this question, even if present briefly, do not persist near an intersection.

To collect RF data for multilevel classification, we therefore choose road locations after careful manual observation of several roads and locations, so that different traffic states occur in these locations frequently. Data is collected from three roads in Mumbai, India - **1. Adi Shankaracharya Marg**, a 25 meters wide road in a residential area with big apartments on either side of the road, **2. Kanjurmarg East Road**, which is roughly 18 meters wide with dense foliage on the roadside and **3. Shreyas Cinema Road**, a narrow road about 10 m wide. For convenience, we shall refer to these three roads as *ADI*, *KJR* and *SHR* henceforth.

To capture ground truth, we use the DailyRoads Voyager [7] application installed on a Samsung Google Nexus S device. The application records ground truth video clips throughout the data collection session, which are later used for manually labeling the traffic state. At the instant the receiver mote is powered on, the time-stamp from the Android device is noted to time-synchronize the video and the sensor data.

To determine the ground truth of traffic states, the occupancy level and vehicle movements are manually observed in the collected videos. We initially try to label videos into four intuitive traffic states – empty road, free-flow traffic, slow traffic and standing traffic. Examples of these four states are provided as images in Figure 21. But sometimes traffic states change within a very short time span of a few seconds, or it is visually confusing to categorize the video into any one of the four states. Thus we introduce three intermediate *mixed* states between the originally conceived four *pure* states to handle this confusion. The precedent for use of mixed states also occurs in the Natural

[6] The process of labeling the videos with ground truth is manual, due to the absence of any automated and evaluated tool to process video of non-laned traffic, at the time of this data collection, as noted in Section 2.

[7] http://www.dailyroads.com/voyager.php, Copyright © DailyRoads

Language processing task of part-of-speech tags where ambiguous tags are used by labelers to indicate their uncertainty or lack of knowledge about the part of speech of a word.

Thus our videos are labeled into seven traffic states as follows: **1. Empty (E)** : Pure state showing empty road, **2. Empty-FreeFlow (EF)** : Mixed state indicating that either the road is empty or having free-flowing traffic, **3. FreeFlow (F)** : Pure state indicating continuous free-flow of vehicles, **4. FreeFlow-Congestion (FC)** : Mixed state indicating either free-flowing or congested traffic, **5. Congestion (C)** : Pure state indicating that the road consistently experiences slow moving, congested traffic, **6. Congestion-Standstill (CS)** : Mixed state indicating that traffic is either congested or complete standstill, **7. Standstill (S)** : Pure state indicating consistent vehicular standstill. The durations of data collected for different traffic states are shown in Table IV.

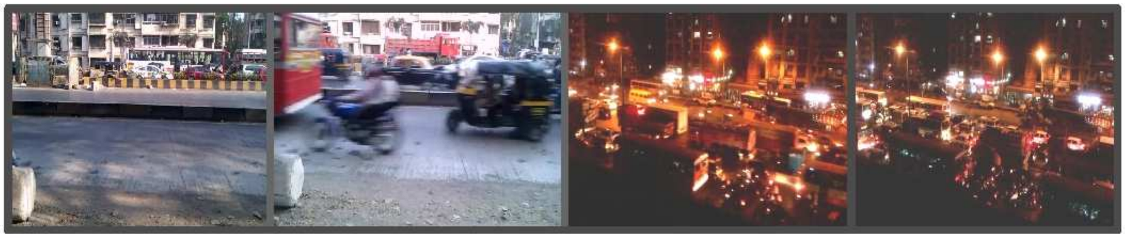

Fig. 21: Empty, free-flow, congested and standstill traffic states from left to right

We experiment with two modes of multilevel classification – **4L** considering only the four pure states E, F, C and S as classes 1, 2, 3, 4 respectively and **7L** considering both pure and mixed states E, EF, F, FC, C, CS and S as classes $1, 2 ... 7$ respectively. The RF dataset is broken into blocks of prediction time window $t = 10$ seconds, so that the corresponding ground truth class is consistent over that 10 seconds. t is varied later to experiment with the minimum prediction window.

To see the feasibility of multilevel classification, we consider 28 statistical features - arithmetic mean, standard deviation (std), variance, minimum value, maximum value, range, coefficient of variation (std/mean), coefficient of skewness, coefficient of kurtosis, counts of 10 histogram bins and 10^{th}, 20^{th} ... 90^{th} percentile values. The intuition behind using this extended set of features, compared to only the percentile based features for binary classification, is as follows. Stationary states like empty road or standing traffic should be differentiable with percentile values or minimum, maximum, average RF values, while these stationary states should be separable from the moving traffic states, like fast or slow traffic, using the RF variance based features.

These 28 features are extracted from each of RSSI, LQI, and Packet Reception Rate (PRR) values over the 10 seconds data, yielding 84 features in all. If no packet has been received in that 10 seconds interval, a dummy packet with RSSI -95 (minimum RSSI value for CC2420 radio) and LQI 55 (minimum LQI value) is considered to have been received. These features along with the corresponding class label form a single data-point in dataset used subsequently to evaluate the multilevel classification algorithms.

7. ANALYSIS AND EVALUATION

7.1. Can binary traffic states be quantitatively inferred from RF link characteristics?

Figures 22 and 23 show the CDF of RSSI and packet reception rate respectively. Each graph shows 14 plots: each a CDF calculated over 5 minutes. As seen from the figures, the curves in each graph can be classified into three distinct groups – *Group1* between 5:37-6:21pm, *Group2* between 6:22-6:27pm and *Group3* between 6:30-7:05pm. The ground truth of traffic state noted is free-flowing till 6:20pm, slow for about 5 minutes and then heavily congested till the end of the experiment. Thus *Group1* corresponds to free-flowing traffic, *Group2* to slowly moving traffic, intermediate between free-flowing and congested and *Group3* to heavily congested traffic. The high correlation of the CDFs with traffic state is apparent visually. For e.g., (a) the 50^{th} and 70^{th} percentiles of RSSI are around -93dBm in congestion and -78dBm in free-flow. (b) the 60^{th} and 80^{th} percentiles of reception rate are around 0 packets/sec in congestion and 24 packets/sec in free-flow. We see similar trends in 16 hours of data collected over three weeks, from two Mumbai roads. Sample videos of free-flowing and congested traffic are available at [War-videos]. Thus wireless links are indeed affected by road traffic, and the binary traffc states are distinguishable based on RF characteristics like RSSI of received packets.

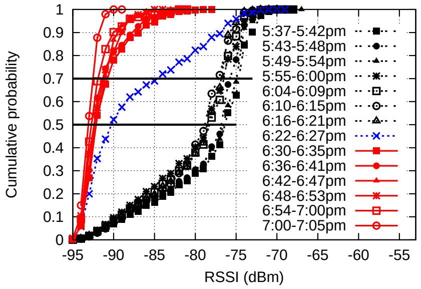

Fig. 22: CDF of RSSI (dBm)

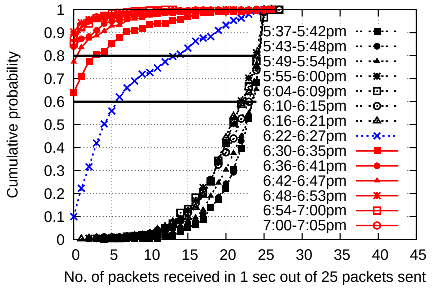

Fig. 23: CDF of PRR

7.2. Can binary traffic state classifications be done in near real time?

We evaluate the four algorithms:1) FeatureClassifier using SVM, 2) FeatureClassifier using K-Means, 3) SignalClassifier using SVM, and 4) SignalClassifier using K-Means, on our labeled dataset. Fig. 24 and Fig. 25 show the training errors of the FeatureClassifier and the SignalClassifier algorithms respectively. As we can see, FeatureClassifier using K-Means outperforms the other algorithms in terms of accuracy. Also, the classification error of SignalClassifier is not as well behaved as FeatureClassifier, and is surprisingly higher for the supervised SVM algorithm than the unsupervised K-Means algorithm. This is due to the increased label noise in translating the ground truth of a 5 minute time window to each packet received in that window.

The accuracy for the FeatureClassifier algorithm, using both SVM and K-Means is above 90%, when the classification time window is at least 20 seconds, shown by a vertical line in Fig. 24. The graphs presented here are for the wide-road dataset, but we observed similar results for the narrow-road dataset as well. Hence, we choose the FeatureClassifier algorithm and prediction period t as 20 seconds in the queue measurement system, and also in the exploration of road specific model training. As SVM and K-Means give similar results for the Feature classifier algorithm, K-Means is randomly chosen as the classification model in the queue measurement system. But the usability of unsupervised K-Means and the training overhead of supervised SVM are explored in depth in the question of road specific model training.

7.3. Can length of traffic queues be estimated aggregating multiple traffic state information?

The accuracy and break up of errors of our queue measurement deployment results are summarized in Table. V. Error of 1 unit indicates that our queue estimate and the ground truth differ by 1. This in turn can be a case of false positive (fp) or false negative (fn). The false positives and false negatives are determined in the following way. The viewer of the image determines the current queue length

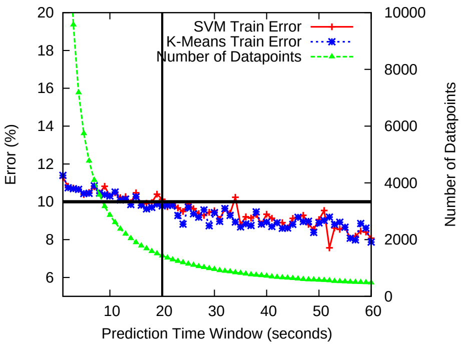

Fig. 24: FC algorithm training error

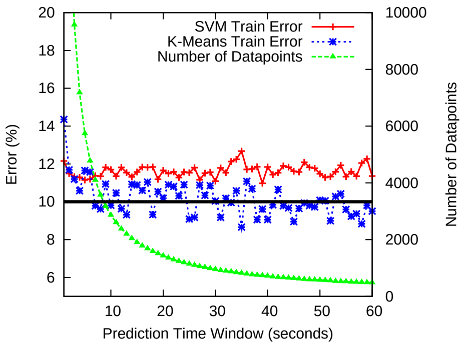

Fig. 25: SC algorithm training error

by seeing the image. This is considered as the ground truth. If the queue length reported by our sensor system is more than the ground truth queue length, that is considered as false positive, as we are overestimating the queue. Similarly, if the reported queue length is less than the ground truth queue length, that is considered as false negative, as we are underestimating the queue. Error of up to 3 units can occur, as images cover units 0-3.

Date	# of detections	% Exact matches	% Error of 1 unit (about 30m)	% Error of 2 units (about 60m)	% Error of 3 units (about 90m)
Nov 17	359	74.09	20.05 (fp=14.76, fn=5.29)	4.45 (fp=2.23, fn=2.23)	1.39 (fp=1.39, fn=0)
Nov 18	359	96.37	1.67 (fp=1.67, fn=0)	1.39 (fp=1.39, fn=0)	0.56 (fp=0.56, fn=0)
Nov 19	353	90.93	7.64 (fp=3.39, fn=4.24)	1.97 (fp=0.28, fn=1.13)	0 (fp=0, fn=0)
Overall	1071	87.11	9.8 (fp=6.62, fn=3.17)	2.4 (fp=1.3, fn=1.12)	0.65 (fp=0.65, fn=0)

Table V: Accuracy and error breakups of deployment results

Fig. 26, Fig. 27 and Fig. 28 show the length of the queue as measured by our system on the three consecutive days respectively. To aid visual comparison with ground truth which covers up to unit 3, we plot both the actual queue values reported by our system, termed as *actual sensed data* in the figures, and the min(3,actual sensed data), termed as *bounded sensed data* in the figures.

As we can see from Table V, the accuracy is upto 96% on Nov 18 and Nov 19. Of the three days, Nov 17 has minimum accuracy: 74%. But as seen from Fig. 26, queue buildup and clearing was very rapid on that day, increasing the challenge of deciding queue length by manual observation. So our low accuracy is likely a combined effect of our errors and errors of manual ground truth estimation. For example, the instant the image is taken, the queue might have cleared but it might have been present for most of the 20 seconds of sensing, leading to a false positive. A case of false negative would occur if a queue builds up the instant the image is taken, while most of the sensing time, traffic is free-flowing.

A better way of ground truth estimation would be to take continuous video, instead of an instant image, and we accept this to be a limitation of this work. For the subsequent studies of road specific model training and multilevel traffic state classification, we have overcome this limitation and used video based traffic ground truth.

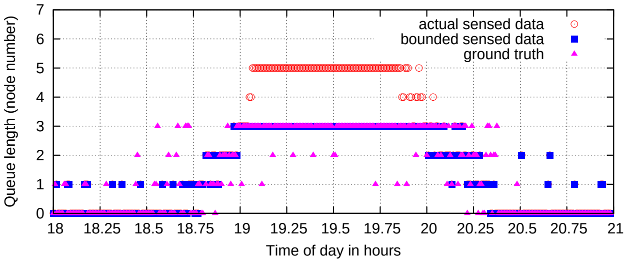

Fig. 26: 6 - 9 pm, Nov 17, 2011, Saturday

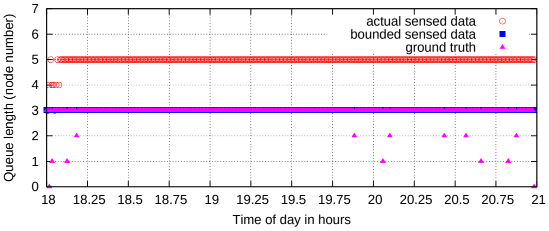

Fig. 27: 6 - 9 pm, Nov 18, 2011, Saturday

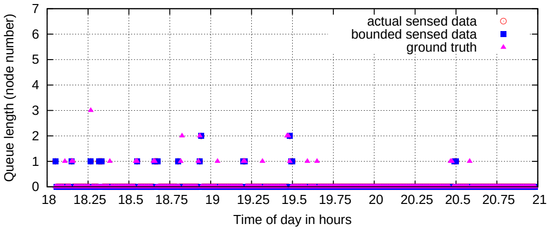

Fig. 28: 6 - 9 pm, Nov 19, 2011, Saturday

False negatives are rare for the error values of 2 and 3, as instant growth and reduction of long queues is non intuitive. But even on Nov 17, almost all the errors are only of one unit and higher errors of 2-3 units, given in the last two columns of Table V, are very low. The above results indicate that, the system is accurate in estimating traffic queue lengths, and shows good promise for use in applications such as automated traffic signal control.

An interesting point to note from the figures is: queue length can be fairly variable (1) over three hours on a single day, as seen in Fig. 26, (2) between two week days at the same time of the day, as seen between Fig. 26 and Fig. 27 and (3) between weekdays and weekend, as seen between Fig. 26, Fig. 27 and Fig. 28. This variability further motivates the usefulness of our system in building applications like dynamic traffic light control, to make traffic management more reactive to current traffic status.

7.4. What is the overhead of classification model training on different roads?

Cross-training Across Roads:
The first analysis that we perform is to examine the necessity of road specific training. We train an SVM on one road's data and test it on the data of the other two roads, and repeat this for all three roads. Congestion is defined to be the *positive* class and free-flow as the *negative* class. Thus, a true positive (TP) is a data-point marked as congested by both the ground truth and the SVM and a true negative (TN) marked free-flow by both. A false positive (FP) is a data-point marked as congested by the SVM and free-flow by the ground-truth and a false negative (FN) the reverse.

We compute three performance metrics, for each of which a higher value is better - **(1) accuracy** - as (TP + TN)/(#data-points), **(2) recall** - as TP/(TP + FN) and **(3) precision** - as TP/(TP + FP). Thus increase in FN decreases recall and increase in FP decreases precision. The computed values with the SVM trained on *AURO* and tested on all three roads is presented in Fig. 32. Similar results for training on *JD* and *UDI* are presented in Fig. 33 and Fig. 34 respectively.

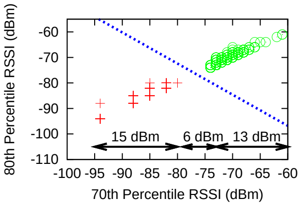

Fig. 29: 70^{th} and 80^{th} percentile RSSI on *AURO*

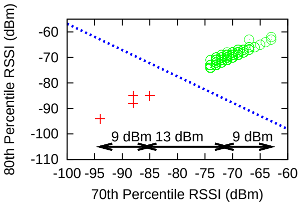

Fig. 30: 70^{th} and 80^{th} percentile RSSI on *JD*

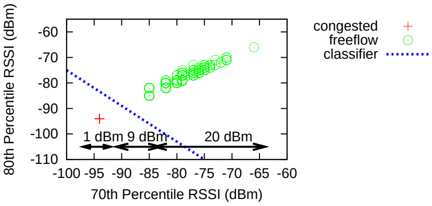

Fig. 31: 70^{th} and 80^{th} percentile RSSI on *UDI*

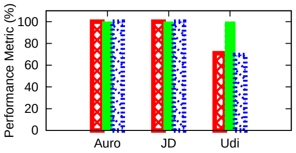

Fig. 32: Train at *AURO*, test on all roads

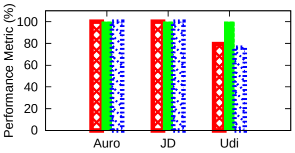

Fig. 33: Train at *JD*, test on all roads

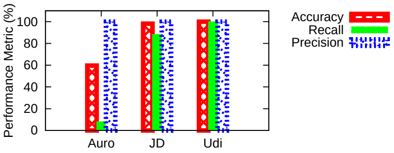

Fig. 34: Train at *UDI* and test on all roads, with performance metrics

As can be seen from Fig. 32 and Fig. 33, training on *AURO* gives 100% accuracy on *JD* and vice versa. But training on *AURO* or *JD*, gives low precision that is high FP on *UDI*. And training on *UDI* (Fig.34) gives low recall that is high FN at *AURO* and *JD*.

To understand the intuition behind the obtained results, we plot two of the eight percentile features, the 70^{th} and the 80^{th} percentiles of RSSI for each road. We also plot the SVM classifier computed on these two features. This is simply to be able to visualize the features in a 2D plot, which is not possible with all the features taken together. As seen from Fig. 29, Fig. 30 and Fig. 31, the free-flow RSSI values are higher for *AURO* and *JD*, than that for *UDI*. This results from the narrow widths of *AURO* and *JD* in comparison to that of *UDI* as RSSI degrades with distance. This leads to the classifier shift towards bottom left at *UDI* (Fig. 31), in comparison to the other two roads.

Intuitively, the road being very wide at *UDI*, the RSSI values are so low even in free-flow traffic, that they are equivalent to congested RSSI on the other two narrow roads. This results in increased FP when training is done on the narrow roads and testing on *UDI*. The reverse train-test procedure causes congested RSSI on the narrow *AURO* and *JD*, to be as high as free-flow RSSI on wide *UDI*, leading to increased FN. The difference between *AURO* and JD is in terms of vehicle types, with all types plying on *JD* and only light vehicles on *AURO*. This is visible in the lower distance between congested and free-flow RSSI at *AURO* in comparison to that on *JD*, though the classifiers for these two roads are similar and cross-training accuracies are 100%.

The above empirical analysis using 120 hours of data on 3 roads show that training models can be used across roads of similar widths, while roads of different widths need separate training models for high classification accuracy. Also against earlier intuition, the vehicle type does not have much effect as *AURO* and *JD* with similar widths but different vehicle types have similar classification models.

Issues with unsupervised clustering:
Thus it seems necessary to build separate classification models for roads of different widths. Can we then use unsupervised learning? Section 7.2 reported that unsupervised K-Means clustering gave comparable classification accuracy to supervised SVM models. Also, K-Means was successfully used in the queue measurement deployment in Section 7.3.

The 16 hours of data, used in Section 7.2 and also used as the training data for the queue measurement system, was however collected manually and carefully, with almost equal amounts of free-flow and congested data. This balanced and clean data distribution for the two intended clusters, made K-means work perfectly. But it was not *unsupervised* in the true sense of the term, as manual supervision had crept in, in the data collection process itself.

How would unsupervised clustering fare if the data collection is unsupervised too? We examine this as follows. Two hours worth of data-points (360 data-points, each having 8 RSSI percentiles computed over 20 seconds intervals) are selected from 8 am to 10:20 am data of Apr 17, 2012 at *AURO*. The manually observed ground truth is that, only 31 data-points (about 10 mins) have congested traffic and remaining 329 data-points have free-flow traffic. K-means clustering without any manual label gives 138 false positives (only 18.34% precision) on this dataset. This exhibits is a well-known issue of K-Means clustering, namely non-handling of unbalanced cluster sizes and irregular cluster shapes.

The two hours data described above is one of many possible datasets from Table. III, that give bad clusters with poor correlation to ground truth. The intuition behind the problem is as follows. Choosing data to give as input to the clustering algorithm, so that any particular cluster does not have excessive representation in the dataset compared to other clusters, is non-trivial in our context. The vague idea, that between 8 and 10:30 am, both traffic states of free-flow and congestion should have occurred in somewhat balanced proportions, might not hold, because congestion and free-flow time durations for a road are not predictable.

So any given time window might have widely varying congested and free-flow data points at such an intersection, and balanced clusters for both traffic states might be absent in the collected dataset. Thus clustering algorithm suffers from imbalance in cluster sizes, and choosing balanced number of data-points for free-flow and congestion is tricky because of traffic unpredictabilities.

Practical ways to provide supervision:
The RF sensors, when put on a new road for which the classification model is not yet built, would collect continuous RF data. This does not have any manual overhead. The manual overhead will come into picture for labeling *some fraction* of this RF-data as free-flow or congested, after seeing the traffic situation on the road. For roads with an existing camera infrastructure, where the RF sensors might be used to complement the cameras in traffic sensing, the process of seeing the traffic situation from the video feeds is easy. That is what we have done to ascertain ground truth in this paper. But on roads where RF sensors only will be used, as a low-overhead substitute for cameras, the human labeler would need something like a battery operated wireless camera [Portacams] to see the traffic situations remotely. Such a camera has to be put up at the RF sensor location and the videos can be seen on a website or a cellphone, based on the camera's wireless connectivity.

Now *how much data* needs to be labeled is interesting to analyze, to get a practical estimate of the video transfer and manual supervision overhead for a new road. If we can empirically estimate N_{min}, as the minimum number of *labeled* free-flow and congested data-points needed to train an acceptably accurate classification model, the human labeler has to look at traffic videos for enough time to accumulate those N_{min} free-flow and N_{min} congested labels. The times of the day to look at traffic videos to label RF data can be intuitively chosen by the human labeler, like 7 am to label N_{min} free-flow data-points and 8 pm to label N_{min} congested data-points and possible attempts at other times to handle the unpredictability of congestion times.

To empirically estimate N_{min}, we consider the dataset in Table III, taking one road at a time and then repeating this procedure for each road. For a particular road, we do the following experiment - choose N free-flow data-points, contiguous in time and N congested data-points, contiguous in time; train an SVM model with those $2N$ data-points and test the model on the remaining data for that road. We gradually increase N to see that when training data amount is gradually increased, how does classification accuracy improve?

This experiment however, gives almost zero test error even with $N = 1$. We felt however that this was not stringent enough, i.e. we could have been lucky in our specific choice of training points.

So we next ran a more stringent test, where we randomly chose N points each for congested and free-flow. This lets us examine how much training we need even if we are "unlucky" with respect to the choice of the 2N training data points.

We randomly shuffle the free-flow data-points (e.g. 4259 data-points for *AURO*) and select N data-points from them. We also randomly shuffle the congested data-points (e.g. 2977 data-points for *AURO*) and select N data-points from them. These $2N$ randomly selected data-points are then used to train an SVM which is tested on the all the remaining data-points ($7236 - 2N$ for *AURO*). N is varied from 3 (this amounts to 3 congested data-points and 3 free-flow data-points, each computed over 20 seconds, i.e. 120 seconds of manual supervision) to 15 (this amounts to 600 seconds of manual supervision) in steps of 3. $N = 1, 2$ are also considered in addition to the above range. For each value of N, the procedure of {shuffle, select $2N$, train SVM and test on unseen (total - $2N$) data-points}, is repeated 10,000 times.

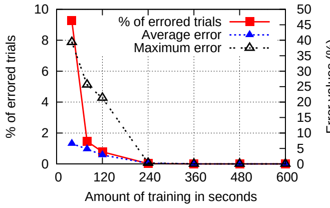

Fig. 35: Errors at *AURO*

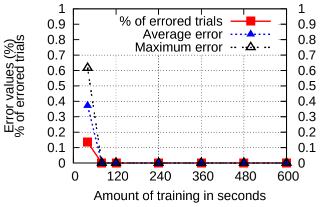

Fig. 36: Errors at *JD*

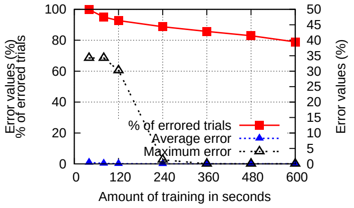

Fig. 37: Errors at *UDI*

The percentage of runs out of 10,000, that do not have 100% accuracy on unseen test data, is shown as % errored trials in Fig. 35, Fig. 36 and Fig. 37. The average and maximum error values over these errored trails are also shown in the same figures. The maximum error cases represent the "unlucky" choices of 2N data points for training. As we can see, even the maximum errors decrease to almost 0% with 240 seconds of manual supervision which amounts to $N = 6$, i.e. only 6 data-points each of congested and free-flow in the training dataset.

To intuitively understand how such small amount of training data achieves high classification accuracy, we need to look back at Fig. 29, Fig. 30 and Fig. 31. The left and right horizontal arrows in each plot, show for the feature 70^{th} percentile RSSI, the dispersion within the congested and the free-flow traffic classes respectively. High dispersion should necessitate more training data-points to better position the classifier with respect to the peripheral data-points. The central horizontal arrow in each plot shows the distance between the congested and the free-flow classes. Low distance should necessitate more training data-points to better learn the boundary between the classes.

For example, we can see the dispersion of the free-flow class is high for *UDI* in Fig. 31, as this wide road can have several levels of free-flow occupancies. The distance between the two classes is small for *AURO* as seen in Fig. 29. This narrow road has only light vehicles plying on it which cannot drastically change the RSSI values between free-flow and congested traffic states. Such characteristics of feature distribution lead to 240 seconds of necessary manual supervision needed for these two roads to reduce errors to 0% as seen from Fig. 35 and Fig. 37, while on *JD*, only 120 seconds of manual supervision seem sufficient (Fig. 36).

In addition to road width and vehicle type affecting the RSSI distribution, environmental factors also add to RSSI variations. But even with the different sources of RSSI variation, the distance between features of the two traffic classes is high enough and the dispersion of features within each class is low enough to make the classifiers accurate, even with small amount (2-4 minutes) of training data for all three roads. This makes the deployment of RF sensors over a city wide road network manageable in terms of model training. We discuss some possible future work in this direction in Section 8.

7.5. Can binary traffic state classification be extended to multilevel classification?

Fig. 38 and Fig. 39 show percentage errors of ten-fold cross-validation of the various multilevel classification algorithms listed in Table. II for the three roads. We make the following key observations from the figures.

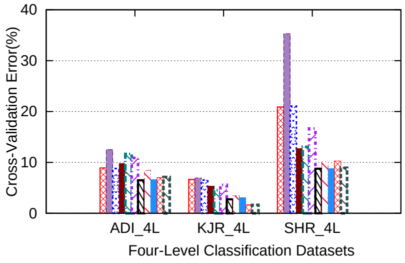

Fig. 38: Four-Level Errors

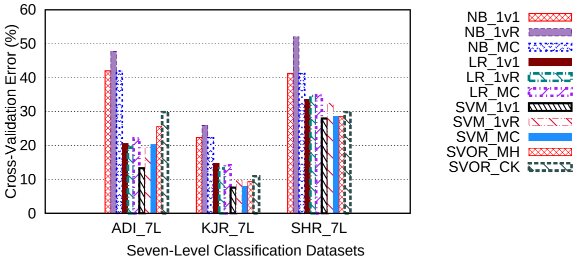

Fig. 39: Seven-Level Errors

(1) In Fig. 38, SVM 1-versus-1 algorithm shows least error on ADI_4L and SHR_4L datasets. In the cases of KJR_4L dataset, the difference in the accuracy of SVM 1-versus-1 and the best accuracy is within 1.20%. In Fig. 39, again SVM 1-versus-1 provides the best CV accuracies on all datasets. Thus **SVM 1-versus-1** seems to be the best algorithm for both four and seven level classification.

(2) For both four and seven level classifications, Naive-Bayes performs poorly while Logistic Regression, SVM and Ordinal Regression have similar accuracies. The well-known systemic issues of Naive-Bayes [Rennie et al. 2003] are intuitively apparent in our dataset. Firstly, the independence assumption among the features does not hold at all with many features derived from the same signal like RSSI or LQI or PRR. Secondly, there is imbalance in number of training samples for each class as seen from Table. IV, which Naive-Bayes is known not to handle well [Rennie et al. 2003].

(3) The three settings of 1-versus-1, 1-versus-all and Multiclass have almost same error values. There has been literature on higher training overhead of 1-versus-all than 1-versus-1 [Milgram et al. 2006], as each of the k classifiers in 1-versus-all needs to handle more instances than each of the $\binom{k}{2}$ classifiers in 1-versus-1. In our setting, model training overhead is of low importance as training will typically be offline and the trained model programmed into the sensors, will be used for real-time traffic state classification using unseen data.

(4) The errors on *SHR* is much higher than that on *ADI* and *KJR* for both four and seven level classifications. From our intuitions developed in road specific model characteristics in Section 7.4, the most probable reason is the narrow width of *SHR*, that reduces the distance between features representing different traffic classes, making classification more tricky.

We also experiment with various loss functions for SVM MultiClass to incorporate ambiguity aware inter label losses, as listed in Table 40a. The explicit formulation of Loss IV given in Table 40b. **Loss I** is the normal 0-1 loss where all wrong classifications suffer equal penalty. **Loss II** and **Loss III** are typical for ordinal classification tasks where there is an intuitive ordering among the classes and the losses mimic that ordering. For example, an E (class 0) classified as an F (class 2) is a less serious error than being classified as a C (class 4). **Loss IV**, **Loss V** and **Loss VI** are specific to our seven-level labeling taxonomy in addition to being ordinal. They do not penalize the classification of a data point if its true label y is a mixed state and it is classified as y' - one of the two adjacent pure states. This characteristic is marked in bold for the mixed classes 1, 3 and 5 in Table 40b.

$y, y' \in \{0, 1, 2, 3, 4, 5, 6\}$	
Loss I	$\mathbf{1}\{y \neq y'\}$
Loss II	$\lvert y - y' \rvert$
Loss III	$(y - y')^2$
Loss IV	as described in Table 40b
Loss V	$\mathbf{1}\{\text{Loss IV} \neq 0\}$
Loss VI	$\frac{\text{Loss IV}}{2}$ (rounded to the upper integer)

(a) SVM MultiClass Losses

y' y	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	**0**	**0**	**0**	1	2	3	4
2	2	1	0	1	2	3	4
3	2	1	**0**	**0**	**0**	1	2
4	4	3	2	1	0	1	2
5	4	3	2	1	**0**	**0**	**0**
6	6	5	4	3	2	1	0

(b) SVM MultiClass Loss IV

Road Dataset	SVM MultiClass Losses					
	I	II	III	IV	V	VI
ADI_7L	76.95	79.78	73.39	87.89	**89.39**	88.15
KJR_7L	90.28	91.96	91.12	**97.12**	**97.12**	97.00
SHR_7L	71.54	69.78	70.66	88.96	90.35	**90.98**

(c) Results on Seven-Level Datasets

Fig. 40: Ambiguity aware losses for SVM MultiClass

Ten-fold cross-validation accuracy results of SVM MultiClass on the seven-level traffic classification datasets ADI_7L, KJR_7L, and SHR_7L are shown in Table 40c. The best accuracy for each dataset is marked in **bold**. A considerable improvement in accuracy is observed in comparison with Fig. 39 by the realization that all mis-classifications of mixed state data points need not be penalized.

Prediction time window t: We compute 10 fold cross validation errors using the SVM 1-versus-1 algorithm by gradually increasing the prediction time window t, over which the 84 features are computed. The results for both four and seven level classifications for the three roads are given in Fig. 41. As expected, the errors decrease with increase in t, due to removal of high frequency noise. The errors flatten out after 10 secs in most cases, as shown by a vertical line. Also, since we are making a fine categorization of traffic states here, the multilevel traffic states occur for smaller time durations and inter-state transitions are faster. So lower time granularity is necessary, though higher time scales gives less error, to handle the quick traffic state transitions in ground truth. Hence in our evaluation and comparison of algorithms above, we have used $t = 10$ seconds.

Thus we are able to achieve about 90% accuracies in both four level and seven level traffic state classification, using cheap and noisy RF sensors. The features for multilevel classification can be computed in real-time on an embedded platform. With an SVM multiclass or 1-versus-1 model trained offline and programmed on the embedded platform, real-time multilevel classification on unseen wireless packets is achievable with little engineering efforts. Such multilevel classification can be used as input to applications like congestion maps and queue-length measurements at intersections, for intelligent traffic signal control.

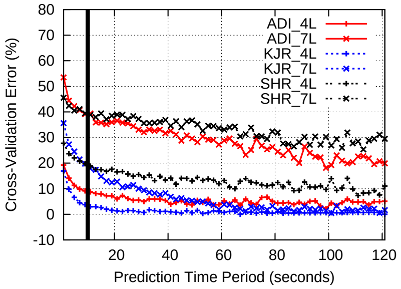

Fig. 41: Effect of Prediction Time Period on Error

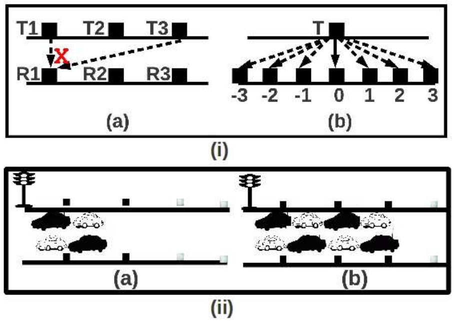

Fig. 42: Interference & duty-cycling

8. DISCUSSION AND FUTURE WORK

In this section we discuss some aspects relevant to our sensing system and outline a few areas of future work.

Interference issues:

Hidden node problems can occur in our queue measurement network. For example, as shown in Fig. 42(i)(a), T1 and T3 might transmit simultaneously, though XBEE radios use CSMA/CA, as they might not hear each other. Collision might happen at R1, if it can hear both T1 and T3, which is possible because of wireless link asymmetries. 802.15.4 has 16 orthogonal channels. Thus each T-R pair can potentially use a different channel to avoid intra-network interference. But inter-network interference can still occur, because of Wi-Fi interference from nearby 802.11 access points.

Fortunately, interference does not affect RSSI of received packets [Rayanchu et al. 2008; Liang et al. 2010]. If Clear Channel Assessment (CCA) is disabled at tx, it stops holding back packets in presence of interference [Liang et al. 2010]. If Cyclic Redundancy Check (CRC) is disabled at rx, all packets, even with bits corrupted by interference would be passed to the application layer. Since we use only RSSI for sensing, the actual packet content is irrelevant to us. Disabling CCA and CRC, increases number of received packets in presence of interference, using RSSI of which, our technique would work fine. Only the communication of sensed data needs an interference free channel, as there the actual packet contents are important. But for this we can use 802.15.4 channel 26, which lies outside the 802.11 spectrum. Our preliminary experiments disabling CCA and CRC show the usability of all channels for sensing, even in presence of heavy 802.11 interference. This will be explored further in future.

Power consumption:

Battery life saving is not a very crucial requirement for our system, as our units are deployed on street lamp-posts with constant power supply from grid lines. Still, to get a power budget for the batteries used in our prototype deployment, we explorer the power consumption of our queue measurement system. The T units perform 20 XBEE tx operations per second. The R units, in every measurement cycle spanning 30 secs, receive at most 400 XBEE packets during sensing, perform one classification operation and receive and transmit at most eight CC2520 messages. The C unit, in every measurement cycle of 30 secs, performs one GPRS communication, two SD card operations and receive and transmit at most eight CC2520 messages. The power consumed for each individual operation is given in Table VI.

A power optimization approach, that we may consider in future, is as follows. It might be unnecessary to keep all T-R pairs functioning at all times. For e.g., in our Nov 19 deployment (Fig. 28), the first two sensor pairs would have been enough to detect queues. A simple duty-cycling

Function	mW	Function	mW
C5505 operations	213	GPRS	2016
CC2520 tx (0 dBm)	167	CC2520 rx (0dBm)	610
XBEE tx (0 dBm)	390	XBEE rx (0 dBm)	540

Table VI: Function Specific Power Consumption

mechanism can be as follows. The pair nearest to the signal remains awake at all times. If this pair sees congestion for more than a threshold number of cycles, the next pair wakes up. This continues as the queue grows and the whole network gradually comes up. This is illustrated in Fig. 42(ii), where dark rectangles denote nodes which are awake and the others denote sleeping nodes. Some power can also be saved by suppressing updates that can be inferred by correlating other updates [Guitton et al. 2007].

Further exploration of model training:

Another direction of future work can be the exploration of transfer learning in the context of road specific model training. The variation of RSSI with distance is complex, so the width of roads will affect the RSSI based features in a complicated way. The features will also be affected by vehicle type, and mix of vehicles on a given road will be difficult to measure and quantify. But if such factors affecting RF features can be quantified and the relation between the factors and the RF features can be expressed mathematically, then classification model trained on one road, along with quantitative inputs like road width or vehicular mix, might be used to build models for other roads. The feasibility of this and comparison of the complexity and accuracies of such transfer learning vs. small amount of manual training overhead for a new road, will be interesting to explore. Also, classification model drift with time due to effect of weather and other environmental factors can be studied over several months of data, with special focus on automatic drift detection applying concept drift theory of machine learning.

Other design choices:

Finally, it will be interesting to try different network topologies for the queue estimation system, instead of the linear array, to reduce hardware overhead. An example star-topology is shown in Fig. 42(i)(b). A second design choice to explore, would be to replace the dual radio solution with single radio, which has two antenna ports [Jennic-radios]. We might attach an external antenna to one port and keep the other empty and can select the former for communication and latter for sensing, dynamically in software. A third design choice to explore is the use of directional antennas. Such antennas have fairly low costs and they have been shown to improve the stability of wireless links [Ostrom et al. 2010]. This may render the RSSI measurements to be less susceptible to oscillations, possibly further improving the detection accuracy. Further, customized platform design and cheaper radios and micro-controllers will be considered to bring down overall system costs.

9. CONCLUSIONS

In this paper, we present RoadRFSense, a practical RF sensing based road traffic estimation system for developing regions. We present a novel sensing mechanism to monitor road traffic conditions, based on change in RF signals. Building upon the sensing mechanism, we design, implement and deploy a real-time queue measurement system, that gives upto 96% accurate queue measurements. Thirdly, through large scale empirical data collection and analysis, we show that road specific training of classification models is needed for roads of different widths and unsupervised clustering is not feasible. We show that however, small amount of manual supervision is sufficient to train the classification models and outline practical ways to provide such manual supervision. Finally, we analyze RF features and algorithms to perform multilevel traffic state classification at above 90% accuracy over 20 hours of data from three roads. Traffic monitoring in developing regions poses a set of challenges, not met fully by existing systems. In that context, the RoadRFSense study shows good promise of our methods, to be immediately useful for a variety of applications in real situations.

REFERENCES

AL-HUSSEINY, A. AND YOUSSEF, M. 2012. Rf-based traffic detection and identification. In *Proceedings of the IEEE Vehicular Technology Conference (VTC'12)*.

BAHL, P. AND PADMANABHAN, V. N. 2000. Radar: an in-building rf-based user location and tracking system. In *Proceedings of 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*.

BALAN, R. K., NGUYEN, K. X., AND JIANG, L. 2011. Real-time trip information service for a large taxi fleet. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*.

BTIS. Bangalore transport information system. In `http://www.btis.in/`.

CHINTALAPUDI, K., IYER, A. P., AND PADMANABHAN, V. N. 2010. Indoor localization without the pain. In *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking (MobiCom '10)*.

CHU, W. AND KEERTHI, S. S. 2005. New approaches to support vector ordinal regression. In *Proceedings of the 22nd International conference on Machine learning (ICML'05)*.

COIFMAN, B. AND CASSIDY, M. 2002. Vehicle reidentification and travel time measurement on congested freeways. *Transportation Research Part A: Policy and Practice 36,* 10, 899–917.

FRANK, E. AND HALL, M. 2001. A simple approach to ordinal classification. In *Proceedings of the 12th European Conference on Machine Learning (ECML'01)*.

GUITTON, A., SKORDYLIS, A., AND TRIGONI, N. 2007. Utilizing correlations to compress time-series in traffic monitoring sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC '07)*.

JAIN, V., SHARMA, A., AND SUBRAMANIAN, L. 2012. Road traffic congestion in the developing world. In *Proceedings of the 2nd Annual Symposium on Computing for Development (DEV '12)*.

JENNIC-RADIOS. Zigbee radios with dual antenna port. In `http://www.jennic.com/products/modules/jn5148_modules,http://www.jennic.com/products/modules/jn5139_modules`.

JING YUAN, YU ZHENG, X. X. AND SUN, G. 2012. T-drive: Enhancing driving directions with taxi drivers intelligence. In *Transactions on Knowledge and Data Engineering (TKDE '12)*.

KASTRINAKI, V., ZERVAKIS, M., AND KALAITZAKIS, K. 2003. A survey of video processing techniques for traffic applications. *Image and Vision Computing 21*.

KOUKOUMIDIS, E., PEH, L.-S., AND MARTONOSI, M. 2011. Signalguru: Leveraging mobile phones for collaborative traffic signal schedule advisory. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*.

LE, H. K., PASTERNACK, J., AHMADI, H., GUPTA, M., SUN, Y., ABDELZAHER, T. F., HAN, J., ROTH, D., SZYMANSKI, B. K., AND ADALI, S. 2011. Apollo: Towards factfinding in participatory sensing. In *Proceedings of the 10th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN '11)*.

LEONTIADIS, I., MARFIA, G., MACK, D., PAU, G., MASCOLO, C., AND GERLA, M. 2011. On the effectiveness of an opportunistic traffic management system for vehicular networks. *IEEE Transactions on ITS 12*.

LIANG, C.-J. M., PRIYANTHA, N. B., LIU, J., AND TERZIS, A. 2010. Surviving wi-fi interference in low power zigbee networks. In *Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*.

LOZANO, A., MANFREDI, G., AND NIEDDU, L. 2009. An algorithm for the recognition of levels of congestion in road traffic problems. In *Mathematics and Computers in Simulation, Volume 79*.

MAPUNITY. Mapunity: Social technology at work. In `http://www.mapunity.in/`.

MILGRAM, J., CHERIET, M., AND SABOURIN, R. 2006. “one against one” or “one against all”: Which one is better for handwriting recognition with svms? In *Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition*.

MOTTOLA, L., PICCO, G. P., CERIOTTI, M., GUNĂ, C., AND MURPHY, A. L. 2010. Not all wireless sensor networks are created equal: A comparative study on tunnels. In *ACM Transactions on Sensor Networks (TOSN '10)*. Vol. 7.

OSTROM, E., MOTTOLA, L., AND VOIGT, T. 2010. Evaluation of an electronically switched directional antenna for real-world low-power wireless networks. In *Proceedings of the 5th Workshop on Real-World Wireless Sensor Networks (REALWSN '10)*.

PATTARA-ATIKOM, W., PONGPAIBOOL, P., AND THAJCHAYAPONG, S. 2006. Estimating road traffic congestion using vehicle velocity. In *Intertional Conference on ITS Telecommunications*.

PATWARI, N. AND WILSON, J. 2010. Rf sensor networks for device-free localization: Measurements, models, and algorithms. In *Proceedings of the IEEE*. Vol. 98.

PORIKLI, F. AND LI, X. 2004. Traffic congestion estimation using hmm models without vehicle tracking. In *IEEE Intelligent Vehicles Symposium*.

PORTACAMS. Wireless cameras. In http://portacams.com/3G_CCTV.html,http://www.alibaba.com/product-gs/564355996/battery_powered_wireless_camera.html.

QUINN, J. A. AND NAKIBUULE, R. 2010. Traffic flow monitoring in crowded cities. In *AAAI Spring Symposium on Artificial Intelligence for Development*.

RAI, A., CHINTALAPUDI, K. K., PADMANABHAN, V. N., AND SEN, R. 2012. Zee : Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking (Mobicom'12)*.

RAYANCHU, S., MISHRA, A., AGRAWAL, D., SAHA, S., AND BANERJEE, S. 2008. Diagnosing wireless packet losses in 802.11: Separating collision from weak signal. In *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM '08)*.

RENNIE, J. D. M., SHIH, L., TEEVAN, J., AND KARGER, D. R. 2003. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 20th International Conference on Machine Learning (ICML'03)*.

SCATS. Sydney coordinated adaptive traffic system. In http://www.scats.com.au/index.html.

SEN, R., CROSS, A., VASHISTHA, A., PADMANABHAN, V. N., CUTTRELL, E., AND THIES, W. 2013. Accurate speed and density measurement for road traffic in india. In *Proceedings of the 3rd Annual Symposium on Computing for Development (DEV '13)*.

SEN, R., RAMAN, B., AND SHARMA, P. 2010. Horn-ok-please. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*.

SEN, R., SIRIAH, P., AND RAMAN, B. 2011. Roadsoundsense: Acoustic sensing based road congestion monitoring in developing regions. In *Proceedings of the 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '11)*.

SHRIVASTAVA, V., AGRAWAL, D., MISHRA, A., AND BANERJEE, S. 2007. Understanding the limitations of transmit power control for indoor wlans. In *Proceedings of the 7th Internet Measurement Conference (IMC '07)*.

SKORDYLIS, A. AND TRIGONI, N. 2008. Delay-bounded routing in vehicular ad-hoc networks. In *Proceedings of the 9th International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*.

SKORDYLIS, A. AND TRIGONI, N. 2011. Efficient data propagation in traffic-monitoring vehicular networks. *IEEE Transactions on ITS, volume 12*.

TIRTL. The infra red traffic logger. In http://www.ceos.com.au/products/tirtl.htm.

VARAIYA, P. Publications in transportation. In http://paleale.eecs.berkeley.edu/~varaiya/transp.html.

VIDEO. Video based traffic sensing. In http://www.visioway.com, http://www.traficon.com.

WAR-VIDEOS. Anonymous picasa album containing ground truth images and sample videos from deployment location. In https://picasaweb.google.com/108285005574366399467.

XU, C., FIRNER, B., ZHANG, Y., HOWARD, R., AND LI, J. 2011. Statistical learning strategies for rf-based indoor device-free passive localization. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys '11)*.

YOON, J., NOBLE, B., AND LIU, M. 2007. Surface street traffic estimation. In *Proceedings of the 5th International Conference on Mobile Systems, Applications, and Services (MobiSys '07)*.

YOUSSEF, M. AND AGRAWALA, A. 2005. The horus wlan location determination system. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*.