IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 6, 2015, accepted December 31, 2015, date of publication February 11, 2016, date of current version April 12, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2529562

Sensing and Classifying Roadway Obstacles in Smart Cities: The Street Bump System

THEODORA S. BRISIMI[1], (Student Member, IEEE), CHRISTOS G. CASSANDRAS[1], (Fellow, IEEE), CHRIS OSGOOD[2], IOANNIS CH. PASCHALIDIS[1], (Fellow, IEEE), AND YUE ZHANG[1], (Student Member, IEEE)

[1]Division of Systems Engineering, Center for Information and Systems Engineering, Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215, USA
[2]City of Boston, Boston, MA 02201, USA

Corresponding author: I. C. Paschalidis (yannisp@bu.edu)

This work was supported in part by the National Science Foundation under Grant CNS-1239021, Grant IIS-1237022, Grant CCF-1527292, and Grant IIP-1430145, in part by the Army Research Office under Grant W911NF-11-1-0227 and Grant W911NF-12-1-0390, in part by the Air Force Office of Scientific Research under Grant FA9550-15-1-0471, in part by the Office of Naval Research under Grant N00014-10-1-0952 and Grant N00014-09-1-1051, and in part by Mayor's Office of New Urban Mechanics through the City of Boston.

ABSTRACT We develop an infrastructure-free approach for anomaly detection and identification based on data collected through a smartphone application (Street Bump). The approach is capable of effectively classifying roadway obstacles into predefined categories using machine learning algorithms, as well as prioritizing actionable ones in need of immediate attention based on a proposed anomaly index. We explore some novel variants of classification algorithms that combine clustering with classification and introduce appropriate regularization in order to concentrate on a sparse set of most relevant features, which has the effect of reducing overfitting. Furthermore, the anomaly index we introduce combines novel metrics of obstacle irregularity computed based on the data captured by the Street Bump smartphone application. Results on an actual data set provided by the City of Boston illustrate the feasibility and the effectiveness of our system in practice.

INDEX TERMS Classification, anomaly detection, machine learning, smart cities.

I. INTRODUCTION

A. A SMART CITY OVERVIEW

As of 2014, 54% of the earth's population resides in urban environments, a percentage that is expected to reach 66% by 2050. This increase would amount to about 2.5B people added to urban populations [1]. At the same time, there are now 28 mega-cities (with $\geq$10M people) worldwide, accounting for 22% of the world's urban population and projections are for more than 41 mega-cities by 2030. It stands to reason that managing urban areas has become one of the most critical challenges our society faces today.

The emerging prototype for a Smart City is one of an urban environment with a new generation of innovative services for transportation, energy distribution, health care, environmental monitoring, business, commerce, emergency response, and social activities. The term "Smart City" is used to capture this overall vision as well as the intellectual content that supports it. From a technological point of view, at the heart of a Smart City is a cyber-physical infrastructure with physical elements (e.g., roads, vehicles, power lines) which are continuously monitored through various sensors to observe, for instance, air/water quality, traffic conditions, occupancy of parking spaces, the structural health of bridges, roads, buildings, as well as the location and status of city resources including transportation vehicles, police cars, police officers, and municipal workers. The data collected need to be securely communicated (mostly wirelessly) to information processing and control points. These data may be shared and the control points can cooperate to generate good (ideally, optimal) decisions regarding the safe operation of these physical elements (e.g., vehicles guided through the city).

It is important to emphasize that what ultimately makes the city "smart" is not simply the availability of data but the process of "closing the loop" consisting of sensing, communicating, decision making, and actuating. Figure 1 is a high-level illustration of this process, which must take place while taking into account important issues of privacy, security, safety, and proper energy management necessitated by the wireless nature of most data collection and actuation mechanisms involved. Finally, equally important as the development of a cyber-physical infrastructure is the necessity for Smart Cities to engage – but not coerce – their citizens. Unlike other

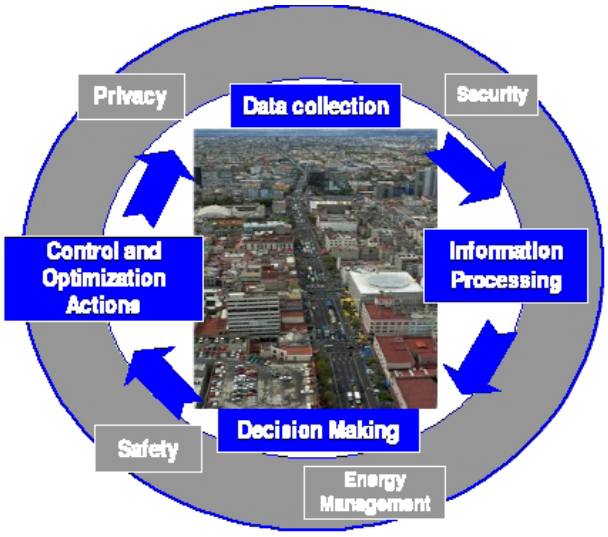

FIGURE 1. Cyber-physical infrastructure for a Smart City.

organizations (e.g., corporations or military units) which can often assume compliance of their human constituents, cities must resonate with their population's goals, means, desires, and freedom of choice. Thus, there is a crucial trade-off between technological efficiency and user engagement and an associated challenge of integrating a crucial *social* aspect into the *Cyber-Physical* environment that constitutes a Smart City.

Although a cyber-physical infrastructure is instrumental in realizing the Smart City vision described above, such infrastructure comes at a significant cost. Embedding sensors in an urban environment (e.g., induction loops in roadways to measure traffic flows or sensors monitoring the state of power lines under ground) does not only entail an installation expense, but also significant maintenance costs. For wirelessly networked sensors, for instance, battery life is limited, so that a battery replacement plan must be in place or additional intelligence must be present in the sensors to manage their energy usage. As another example, to monitor the structural health of roads, one approach is to build specialized vehicles heavily equipped with a variety of sophisticated sensing devices and design patrol paths in urban environments through which such vehicles can perform this function. Clearly, the cost of building and maintaining such vehicles is significant, not to mention their operation cost.

An exciting feature of Smart Cities, however, is the potential to exploit the ubiquitous availability of wireless devices and new technologies embedded in vehicles in order to meet several Smart City goals in an *infrastructure-free* manner. The majority of urban dwellers nowadays carry a smartphone, a device that contains three important functionalities: (*i*) the ability to locate itself through GPS, (*ii*) an accelerometer which can provide several forms of movement information, and (*iii*) a wireless Internet connection which enables it to communicate with other devices or with servers in an already existing network infrastructure. Finally, the shear volume of these devices provides the opportunity to process such "big data" in ways that can bypass inaccuracies or errors. Looking into the not-so-distant future, the connected vehicle initiative will be transforming vehicles into mobile nodes in a network which does not require a Smart City to build or maintain it, but simply to take advantage of the vast amount of data from the vehicles which will allow them to be self-driven. The advent of such *Connected Autonomous Vehicles* (CAVs) provides the opportunity for new approaches to realize, for example, automated vehicle intersection control based on requests and information received from the vehicles located inside some communication range [2]. There are several efforts recently reported in the literature involving the coordination and optimal control of multiple CAVs [3]–[7].

While there are many examples of systems recently developed to handle chronic transportation-related problems in urban environments, such as parking [8] and traffic light signaling [9], these assume the existence of an infrastructure, e.g., sensors to detect if a parking space is vacant or not and traffic lights. In this paper, we choose to focus on an *infrastructure-free* system (called *Street Bump*) which has been developed in Boston to sense and classify roadway obstacles (e.g., potholes) based exclusively on data collected through a smartphone application, as long as the smartphone resides in a vehicle.

B. SENSING AND CLASSIFYING ROADWAY OBSTACLES

According to the American Association of State Highway and Transportation Officials, as much as 50% of US roads and highways are in bad condition, which increases the likelihood of accidents. In the northern parts of the US, in particular, potholes are a recurring problem. As an example, the Massachusetts State Transportation Department [10] received about 1,700 pothole complaints over the first quarter of 2014 and spent more that $800,000 filling them, while the City of Boston filled more than 10,000 potholes.

To detect road obstacles in an automated and cost-effective manner, the City developed the smartphone application *Street Bump*, which records information from the phone's accelerometer and GPS. This information can adequately describe and locate "bumps" as a smartphone-carrying vehicle drives through the streets of Boston. We use the term "bump" in a generic sense to describe various obstacles which include potholes, sunk castings (manhole covers), utility patches, catch basins (drains), train tracks and speed bumps, all substantial enough to be clearly sensed by a driver and potentially cause damage to the wheels or other parts of a vehicle. An innovative component of this approach is the collection of real data through crowd sourcing, an approach which allows citizens to contribute to a massive and continuous data collection process without the need to build and deploy any infrastructure. Moreover, there is the added social benefit of reinforcing a feeling of participation in the betterment of a local community.

Our objective in this paper is to develop an *anomaly detection* and *decision support* system that utilizes the collected *Street Bump* data. Based on the setup introduced in [11], the present paper explores a larger variety of machine learning

classification methods, provides complete technical details of our analysis, and validates the proposed algorithms on an actual dataset provided to us by the City of Boston. The raw data include both "actionable" and "non-actionable" bumps. "Actionable" are bumps due to potholes and sunk castings which are caused by nature or accident and require prompt repair. "Non-actionable" are bumps due to either expected, known, or benign obstacles, such as train tracks, speed bumps, and relatively flat castings, which do not require immediate attention. Our goal is to accurately classify all detected roadway obstacles into predefined categories: actionable obstacles whose severity can be quantified and non-actionable ones. Further, we seek to develop a way of quantifying the severity of actionable obstacles. This can ultimately be used to maintain a prioritized list based on which the City can be dispatching repair crews in a timely and economical manner.

We emphasize the infrastructure-free nature of this system, since it relies exclusively on data collected from smartphones located in vehicles driven throughout an urban setting. Some may be municipal vehicles (e.g., buses, police cars) but most are driven by citizens who can therefore become engaged in the maintenance of their own public resources. This is in contrast to infrastructure-dependent approaches such as the specialized vehicles equipped with sophisticated sensing devices that were mentioned in the previous section.

The starting point in our approach is a "labeled" training set at our disposal which contains information on the type of bumps in that set. Processing the data collected by *Street Bump*, we extract features, that is, various functions of the data computed over time-windows associated with a specific bump. We then follow two complementary approaches. First, we formulate a binary classification problem to differentiate actionable from non-actionable bumps. To that end, we use *supervised learning* algorithms that have been shown to be effective in a number of applications (e.g., prediction of heart-related hospitalizations [12]). These algorithms include Support Vector Machines (SVM), AdaBoost, logistic regression, random forests (see [13]), and some novel variants we introduce that combine classification with clustering. In some instances, especially when we perform clustering and the number of training data per cluster is small, we introduce specific sparsity-inducing regularizations that identify a subset of most relevant features.

Our second approach is *unsupervised* and inspired by *anomaly detection* problems arising in a variety of applications (see e.g., [14]–[17] and references therein). Anomaly detection methods consist of modeling "normal" behavior and detecting deviations from it, called the anomalies. In this work, we will exploit several structural properties of the problem and specifically the fact that most actionable obstacles are due to natural phenomena that produce random obstacle configurations. In contrast, non-actionable obstacles are human-made which results in a significant degree of regularity. As an example, driving over a flat casting produces vibration data leading to a signal closely resembling that of a harmonic oscillation compared to signals obtained from vibrations due to a pothole. We are then able to quantify such a measure of regularity through an "anomaly index." We develop two methods for obtaining such an index. The first method is based on a *Mean Squared Error (MSE)* measure of a bump signal's deviation from that of a simple harmonic oscillation. The second method relies on the entropy [18] of a bump signal, which captures the extent of an anomaly; in particular, a higher anomaly index is assigned to bumps with larger entropy values.

Although our work is focused on locating and identifying street bumps, our methods have much broader applicability. As an example, we mention the problem of remotely detecting non-typical body motion of humans, such as a fall. This is particularly useful in monitoring the elderly and alert caregivers in case of a fall or other accident [19]. Once again, accelerometer and GPS data from a smartphone carried by such individuals may be used to both detect a potential fall and localize the incident without the presence of an underlying infrastructure.

The remainder of the paper is structured as follows. In Sec. II, we describe the *Street Bump* data and how we extract features to be used in bump classification. In Sec. III, we present our decision support and anomaly detection system. In Sec. IV, we define various performance evaluation criteria for our system. In Sec. V, we list extensive numerical results based on actual data provided to us by the City of Boston through controlled data collection using the *Street Bump* application, thus, illustrating the feasibility and effectiveness of our system. Conclusions and directions for future work are discussed in Sec. VI.

On a notational remark, lower case bold letters correspond to vectors. All vectors we use are column vectors and we will write $f = (f_1, \ldots, f_n)$ for $f \in \mathbb{R}^n$.

II. FEATURE EXTRACTION

A. ATTRIBUTES RECORDED BY THE SMARTPHONE

The Mayor's *Office of New Urban Mechanics* in the City of Boston, in partnership with the *Connected Bits* company have developed an iPhone application which can collect roadway obstacle data from sensors (3-axes accelerometer and GPS) embedded in the iPhone so as to ultimately identify "bumps," which the city can then fix. Drivers start up the app and place their smartphones in a stable location in the vehicle, such as the dashboard. While driving through the streets, the app automatically utilizes the phone's accelerometer and GPS receiver and registers a "bump" when the speed of the car exceeds 5 mi/hour and the accelerometer records an absolute value reading of 0.4 g or higher along the z-axis. It then transmits to a remote server information related to every such bump, including a time-stamp that captures the instant when the aforementioned "trigger" conditions were met.

Specifically, the information recorded and transmitted is: (1) latitude and (2) longitude of the bump location, (3) speed of the vehicle (meters per second), (4) course, which is the heading of the vehicle at the time of the bump (i.e., the

angle between the driving direction and a reference direction taken to be North), (5) x-axis, y-axis and z-axis readings from the accelerometer during a time window that includes the bump time-stamp. In particular, this time window starts 0.25 seconds before the time-stamp (recalled from a buffer), is 1 second long, and includes accelerometer readings sampled at 50 Hz (i.e., 50 samples).

According to the iPhone settings, in these readings, the x axis points North, the y axis points West and the z axis is in the direction of gravity. We rotate this coordinate system so that the x-axis aligns with the driving direction, the y-axis is perpendicular to it, and the z axis is unchanged. Henceforth, we will refer to these three time series (one for each coordinate) as the *signatures* of the bump. The x-coordinate signature of an anomalous bump is shown in Fig. 2, where the horizontal axis represents time in seconds and the vertical axis measures the acceleration in the x-axis (i.e., driving) direction.

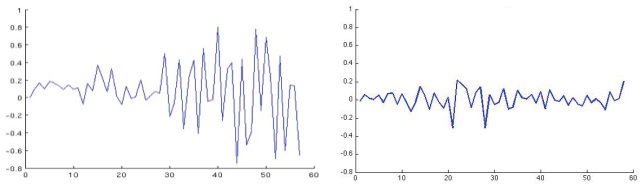

FIGURE 2. (Left): x-coordinate signature of an anomalous (actionable) bump. (Right): x-coordinate signature of a flat casting (non-actionable).

B. FEATURE ENGINEERING FOR THE DECISION SUPPORT SYSTEM

Our first step is to derive a set of features from the collected data which are informative, non-redundant and enable the subsequent learning steps. We divide each of the three n-dimensional coordinate time series into a number of K bins of length $d = \lfloor n/K \rfloor$. Because the total number of samples varies across bumps, we truncate each time series to maintain an identical number of samples for each coordinate and each bump. For each bump, let $\mathbf{x} = (x_1, \ldots, x_{dK})$ and similarly $\mathbf{y}, \mathbf{z}$ denote the vector of samples for the x, y, z-coordinates, respectively. Let $\mathbf{x}^{(k)} = (x_{d(k-1)+1}, \ldots, x_{dk})$, and similarly $\mathbf{y}^{(k)}, \mathbf{z}^{(k)}$, $k = 1, \ldots, K$, denote the vector of samples in the k^{th} bin of the x, y, and z coordinate signatures, respectively. Finally, let us define the operators $M[\cdot]$, $R[\cdot]$ and $\sigma[\cdot]$ to denote the average, range and standard deviation of the elements of a vector. We derive the following features:

- **Basic bump features**: From the set of bump attributes defined above, we retain the latitude and longitude of the bump as well as the speed of the vehicle.
- **Bump distributional features**: From the x-coordinate signature, we calculate $M[\mathbf{x}]$, $\sigma[\mathbf{x}]$, $R[\mathbf{x}]$ and $M[\mathbf{x}^{(k)}]$, $\sigma[\mathbf{x}^{(k)}]$, $R[\mathbf{x}^{(k)}]$, where $k = 1, \ldots, K$. We also compute:

$$D_{\mathbf{x}} = |\arg\max_i x_i - \arg\min_i x_i|,$$

and use it as an additional feature. Next, we quantize the range of $\mathbf{x}$ into B bins, and define the empirical measure $\mathbf{h}_{\mathbf{x}}^{B} = (h_{\mathbf{x}}(1), \ldots, h_{\mathbf{x}}(B))$ where

$$h_{\mathbf{x}}(b) = \frac{\sum_{t=1}^{dK} 1_{\{x_t \in b^{th} \text{ bin}\}}}{dK}, \quad b = 1, \ldots, B,$$

where $1_{\{\cdot\}}$ is the indicator function that takes the value 1 if its attribute is true and 0 otherwise. Last, we create a mapping: $\mathbf{x} \to \mathbf{x}_\mu = (M(\mathbf{x}^{(1)}), \ldots, M(\mathbf{x}^{(k)}))$, and we include $M[\mathbf{x}_\mu]$, $\sigma[\mathbf{x}_\mu]$, $R[\mathbf{x}_\mu]$ into the feature set. We repeat these calculations for the y- and z-coordinate signatures.

- **Temporal dependency features**: This set of features captures the intra-coordinate correlations among bins at different times. For the x-coordinate signature, we add to the feature set the covariance of the signals in consecutive bins, i.e., $cov(\mathbf{x}^{(k)}, \mathbf{x}^{(k+1)})$, $\forall k$. We also account for covariances between bins further away by including the following features: (a) the maximum covariance

$$C_{xx} = \max_{d+1 \le j \le d(K-1)} cov(\mathbf{x}^{(1)}, (x_j, \ldots, x_{j+d})),$$

and (b) the corresponding time lag

$$L_{xx} = \arg\max_{d+1 \le j \le d(K-1)} cov(\mathbf{x}^{(1)}, (x_j, \ldots, x_{j+d})) - d + 1.$$

We also calculate corresponding features for the other two coordinates.

- **Cross-coordinate dependency features**: This set of features captures dependencies between the three coordinates. We include $cov(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$, $cov(\mathbf{x}^{(k)}, \mathbf{z}^{(k)})$, $cov(\mathbf{y}^{(k)}, \mathbf{z}^{(k)})$, $\forall k$, and C_{xy}, L_{xy}, C_{yx}, L_{yx}, C_{yz} and L_{yz}, where definitions are similar as above. For example, $C_{xy} = \max_{d+1 \le j \le d(K-1)} cov(\mathbf{x}^{(1)}, (y_j, \ldots, y_{j+d}))$.

Let us now denote by $\mathbf{f}^{(i)}$ the feature vector we have constructed for each bump i as described above, where $i = 1, \ldots, N$, and N is the total number of bumps in the dataset. The dimensionality of the feature vector is denoted by D. To avoid significant mismatch in the ranges of each element of $\mathbf{f}^{(i)}$, we scale appropriately so that each element is in the [0, 1] range.

C. FEATURE CONSTRUCTION FOR THE ANOMALY DETECTION SYSTEM

For the anomaly detection system, we pre-process the collected data in a different manner. Still, we use the accelerometer measurements captured in the same vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ used earlier. We start with the observation that simple inspection of the bump signatures cannot reveal whether a bump is actionable or non-actionable. This is illustrated in Fig. 3, where the original signatures of a pothole (actionable) and that of a flat casting (non-actionable) cannot be differentiated in any obvious way. In order to enhance the differences between the two bump categories, we proceed as follows.

Let $\xi(k)$ be the original accelerometer values at sample time k associated with a bump (in either x, y or z-axis). Let $\delta(k)$ be the amplitude difference measured over two consecutive time steps:

$$\delta(k) = \xi(k) - \xi(k-1). \quad (1)$$

In order to magnify these amplitude increments, we further define a *differential* signal, which we refer to as the "Δ-Signature Filter:"

$$\Delta(k) = \begin{cases} \Delta(k-1) + \delta(k), & \text{if } \delta(k)\delta(k-1) > 0, \\ \delta(k), & \text{if } \delta(k)\delta(k-1) \leq 0, \\ 0, & \text{if } \delta(k) < c, \end{cases} \quad (2)$$

where we can see that $\Delta(k)$ either accumulates the increments $\delta(k)$ if there is no change of sign, or it resets the value to the latest difference otherwise. Through such accumulation, we therefore boost the effect of continuous movement along the same direction.

Intuitively, the sequence $\{\delta(k)\}$ captures the variability in the signature and $\{\Delta(k)\}$ attempts to capture the *trend* in the signal. If the signal increments $\delta(k)$ are positive or negative over some time interval consistently, this yields a large $\Delta(k)$ value; otherwise $\Delta(k)$ resets itself. Noise appears as small random oscillations in the signature. To reduce noise, we pass $\Delta(k)$ through a high-pass filter that sets its value to zero if the signal is below a threshold c as seen in the third branch of (2) (in our specific system seen in Fig. 3, we have used $c = 0.4$.) Clearly, more obvious patterns are now revealed through the Δ-filtered signature, as shown in the green curves in Fig. 3.

To further test and verify quantitatively the Δ-Signature Filter, we have conducted two additional experiments which have confirmed that these filtered signatures enhance classification performance:

1) The original signature and the Δ-filtered signature of a bump are added as feature vectors and used in a binary classification system based on Support Vector Machines (SVMs); see Sec. III. These additional features yielded superior classification performance.
2) The Fourier transforms of the signature and of the Δ-filtered signature of the bumps have also been used as feature vectors, which further improved classification performance.

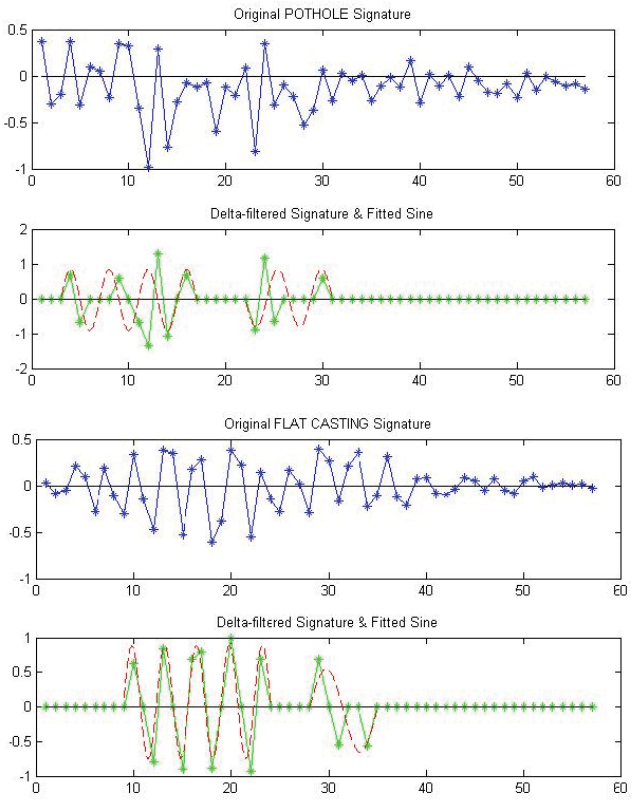

FIGURE 3. Top two figures: Pothole (actionable) signature and associated Δ-filtered signature with fitted sinusoid. Bottom two figures: Flat Casting (Non-actionable) signature and associated Δ-filtered signature with fitted sinusoid.

III. METHODOLOGY-DECISION SUPPORT AND ANOMALY DETECTION SYSTEM

In this section, we describe the methods that comprise the decision support and anomaly detection system. We aim at distinguishing between the actionable and the non-actionable (anomalous) bumps. We use two approaches: (*a*) a *supervised* binary classification approach, which classifies bumps as actionable or non-actionable, and (*b*) an *unsupervised* anomaly detection approach which attempts to identify bumps that are significantly different from the rest. As we will see, these approaches are complementary having different strengths and weaknesses.

A. SUPERVISED CLASSIFICATION METHODS

In the first approach, we formulate the problem as a binary supervised classification problem and experiment with various methods. Because of the limited data set size (the labeling of the bumps is tedious and requires human intervention) and to avoid over-fitting, we aim at limiting the number of variables based on which the classification decision is made, which leads us to introduce regularizers that induce sparsity. By combining results from the various supervised methods, we build a unified prioritized decision support system.

1) SOFT-MARGIN SVM AND SPARSE SOFT-MARGIN SVM

Let $\mathbf{f}^{(i)}$ be the D-dimensional feature vector of the ith bump. SVMs compute a separating hyperplane $\mathbf{w}'\mathbf{f}^{(i)} + b$, between sample vectors (bumps) of different classes [20]. The solution to the SVM problem is the hyperplane that maximizes the margin between the two classes, i.e., the distance from the hyperplane surface to the closest data points. Often in practice, the sample points are not linearly separable. For this reason, the original space is mapped through a kernel function into a higher dimensional space, where presumably linear separation can be achieved [20], [21]. Furthermore, we prefer a hyperplane that better separates the majority of the data even if it ignores a few misclassified samples [21]. Such an SVM is called *soft-margin* SVM. This is achieved through a penalty term in the objective function of the SVM problem multiplied by a weight parameter C that controls the trade-off between the goals of maximizing the margin and minimizing the number of misclassified examples. The parameters of the kernel as well as the regularization parameter C (cf. (5)) are selected through cross-validation.

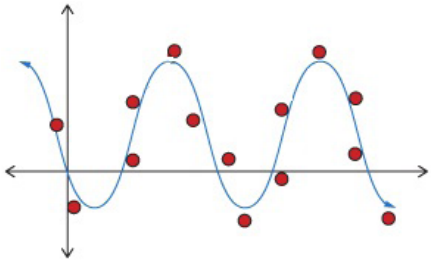

FIGURE 4. Sinusoid function fitting.

The soft-margin formulation of SVM is as follows:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \ & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t. } & v^i(\mathbf{w}'\mathbf{f}^{(i)} + b) \geq 1 - \xi_i, \quad \forall i, \\ & \xi_i \geq 0, \quad \forall i, \end{aligned} \tag{3}$$

where ξ_i are slack variables and v^i is the label for the bump corresponding to feature vector $\mathbf{f}^{(i)}$, taking a value of 1 if the bump is actionable and -1 otherwise. This problem is a convex quadratic programming problem with linear constraints. There exist Lagrange multipliers $a_i \geq 0$ and $\mu_i \geq 0$, for all i, for which (3) is equivalent to minimizing the Lagrangean function L_p with respect to $\mathbf{w}$, b and ξ_i, where

$$\begin{aligned} L_P(\mathbf{w}, b, \xi_i) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N} a_i[v^i(\mathbf{w}'\mathbf{f}^{(i)} + b) \\ - (1 - \xi_i)] - \sum_{i=1}^{N}\mu_i\xi_i. \end{aligned}$$

Imposing the Karush-Kuhn-Tucker optimality conditions and eliminating μ_i we obtain the following dual formulation of the problem:

$$\begin{aligned} \max_{\mathbf{a}} \ & \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j v_i v_j \mathbf{f}_i'\mathbf{f}_j \\ \text{s.t. } & 0 \leq a_i \leq C, \quad i = 1, \ldots, N, \\ & \sum_{i=1}^{N} a_i v_i = 0. \end{aligned}$$

We observe that in the above optimization problem only the inner products $\mathbf{f}_i'\mathbf{f}_j$ of the original features are involved. Employing the *kernel trick* to map the features from the original feature space into a higher dimensional space where features are closer to be linearly separable, makes the framework more flexible and accommodates non-linear boundaries:

$$\begin{aligned} \max_{\mathbf{a}} \ & \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j v_i v_j K(\mathbf{f}_i, \mathbf{f}_j) \\ \text{s.t. } & 0 \leq a_i \leq C, \quad i = 1, \ldots, N, \\ & \sum_{i=1}^{N} a_i v_i = 0, \end{aligned} \tag{4}$$

where $K(\mathbf{f}_i, \mathbf{f}_j)$ is the kernel function. Further and in an effort to limit the number of features used by the classifier, we introduce a *Sparse soft margin SVM (SSVM)*. To induce sparsity, and in addition to the aforementioned misclassification regularization penalty weighted by C, we impose an ℓ_1-norm penalty $\sum_{t=1}^{D}|w_t|$ for the vector of coefficients $\mathbf{w}$ that define the SVM hyperplane. The optimization problem is formulated as follows:

$$\begin{aligned} \min_{\mathbf{z},\mathbf{w},b,\boldsymbol{\xi}} \ & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i + P\sum_{t=1}^{D} z_t \\ \text{s.t. } & v^i(\mathbf{w}'\mathbf{f}^{(i)} + b) \geq 1 - \xi_i, \quad \forall i, \\ & \xi_i \geq 0, \quad \forall i, \\ & z_t \geq w_t, z_t \geq -w_t, \quad t = 1, \ldots, D. \end{aligned} \tag{5}$$

In this formulation, P is a parameter that controls sparsity; the larger P is the more of the w_t will end up being zero.

2) LOGISTIC REGRESSION AND ℓ_1 REGULARIZATION

Logistic regression [22] is a linear, fairly simple classifier, widely used in many classification applications. The basic idea is that for each bump we model the posterior probability of the actionable class as a logistic function with parameters $\boldsymbol{\theta}$ that weight the features $\mathbf{f}$ and an offset β. The parameters $(\boldsymbol{\theta}, \beta)$ of the model are selected by maximizing the log-likelihood using a gradient method. For the test samples, decisions are made by thresholding the log-likelihood ratio of the actionable class over the non-actionable class.

In the ℓ_1 regularized Logistic regression [23], when maximizing the log-likelihood we impose in the objective function an extra penalty term proportional to $|\boldsymbol{\theta}|$, which has the effect of "selecting" a sparse set of features. This leads to a lower complexity model and can avoid overfitting in situations where features are non-informative and/or we have few data on which to train.

3) ADABOOST WITH STUMPS

Boosting is an ensemble supervised learning method that constructs a classifier as a linear combination of simpler weak classifiers [13]. In this work, we will use decision stumps as the component classifiers used by AdaBoost. A decision stump makes a prediction based on the value of a single input feature. AdaBoost maintains a distribution of weights for the training sample points. During each iteration, a weak classifier is trained by focusing on the data points that have been misclassified by the previous weak classifier, and the weights get updated based on the misclassification error. When these iterations terminate, AdaBoost combines the decisions of these weak classifiers using an optimally weighted majority vote. The number of iterations is selected through cross-validation.

4) RANDOM FORESTS

Bagging is a technique for reducing variance of an estimated predictor by averaging many noisy but approximately unbiased models. A random forest is an ensemble of de-correlated

trees [13]. Each decision tree is grown on a training set constructed by sampling (with replacement) a random subset of the original data. On each split, among the full set of the original variables only a subset of fixed size is considered and the best split using these is selected to split the node. Each tree is fully grown until a minimum size is reached, i.e., there is no pruning. While the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Bootstrap sampling achieves de-correlating the trees by constructing them using different training sets. To make a prediction at a new point, random forests take the majority vote among the outputs of the grown trees in the ensemble. Random forests run very efficiently on large data sets, do not have the risk of overfitting as AdaBoost does and can handle data sets with unbalanced classes. The number of trees in the ensemble is selected through cross-validation.

5) A HIERARCHICAL APPROACH OF CLUSTERING AND CLASSIFICATION

Because bumps naturally fit into different categories with distinct signatures, we introduce a new hierarchical approach that first clusters the bumps into a pre-determined number of clusters L and then trains a different classifier for each cluster. For clustering, we use the widely used method of k-means++ [24] that is based on a heuristic to find centroid seeds for k-means clustering. For clustering, we employ Sparse Support Vector Machines (SSVMs), that base the classification decision on a subset of features only. Due to the limited size of the data set we have in our disposal, the use of a sparsity-inducing classifier is critical; a classifier that uses all features (like SVM) would not be able to learn all its parameters from a small training set. We will use the notation C-SSVM to refer to this clustering and classification method. We conduct various experiments to select the optimal number of clusters L. For clustering, a correlation-based distance metric yielded the best results. Specifically, for any two bumps i, j with feature vectors $\mathbf{f}^{(i)}, \mathbf{f}^{(j)}$ we use $1-cov(\mathbf{f}^{(i)}, \mathbf{f}^{(j)})/(\sigma(\mathbf{f}^{(i)})\sigma(\mathbf{f}^{(j)})$ as their distance metric, where $\sigma(\mathbf{f}^{(i)})$ is the sample variance of the feature vector $\mathbf{f}^{(i)}$.

6) A DECISION SUPPORT SYSTEM FOR PRIORITIZING ANOMALIES

All the methods we have outlined in this subsection, classify a test bump by comparing a decision function of its features $g(\mathbf{f}^{(i)})$ to a threshold ϵ. The bump is classified as actionable if $g(\mathbf{f}^{(i)}) \geq \epsilon$. The distance $g(\mathbf{f}^{(i)}) - \epsilon$ can in fact be used to prioritize among actionable bumps; the larger this distance the more confident we are about the bump being actionable. From the methods we considered, logistic regression provides explicitly the likelihood of a bump being actionable, which can then be used to order actionable bumps. Another way to make more robust predictions is to informally combine several methods, seeking for instance consensus among various classifiers in order to declare a bump as actionable.

B. UNSUPERVISED ANOMALY DETECTION METHODS

In the second approach, which focuses on anomaly detection methods, we define a "normal" bump signature in two different ways: (*i*) a normal signal with a sinusoidal pattern, or (*ii*) a normal signal with an expected range of amplitude. We then propose metrics to measure how different a test bump is from a normal pattern.

1) SINUSOIDAL FITTING AND A MEAN SQUARED ERROR METRIC

The key idea is that the Δ-filtered signature defined in (2) exhibits a pattern very similar to a sinusoidal function for non-actionable bumps, whereas actionable bumps do not exhibit this behavior. To explore this apparent separation, we fit a sine (or cosine) function (see Fig. 4) the Δ-filtered signature of a bump (see the red curves in Fig. 3) and calculate a *Mean Squared Error* (MSE) as a goodness-of-fit metric. Specifically, the lower the MSE is, the more actionable we expect a bump to be.

A typical sine (or cosine) function

$$f(t) = A\sin(\omega t - \theta_0) + b$$

is characterized by the four parameters A, ω, θ_0, and b which have to be determined in order to perform a sinusoid curve fitting. Traditionally, one can use nonlinear least squares optimization or sinusoid regression tools to determine the parameters. However, given the fact that all sampling points are approximately equally distributed around zero, we can determine the parameters through a much simpler computation procedure based on the Δ-filtered signature of a bump $\Delta(k)$ in (2).

We begin the fitting process by identifying time intervals such that $\Delta(k) = 0$ over more than one continuous sample points. Since they contain no valuable information, such intervals can be eliminated and we can concentrate on a typical interval $[t_0, t_0 + T_d]$ over which the continuous signal $\Delta(t)$ (see the green curves in Fig. 3) satisfies $\Delta(t) \neq 0$ except, possibly, at a finite number of points $t_i \in [t_0, t_0+T_d]$, $i = 0, 1, \ldots, N$, with $t_N = t_0 + T_d$. The steps for fitting $\Delta(t)$ to a sinusoid function are as follows.

Step 1: Determine A, the amplitude of the function.

Let z_j be the jth zero crossing of $\Delta(t)$ in $[t_0, t_N]$ and there are $N + 1$ zero crossings in total. Define $\Delta_j^{\max} = \max_{t\in(z_{j-1},z_j)} \|\Delta(t)\|$, $j = 1, \ldots, N$, which represents the extreme values between each pair of zero crossings. Then, we have

$$A = \frac{1}{N}\sum_{i=1}^{N} \|\Delta_j^{\max}\|. \tag{6}$$

Step 2: Determine b, the vertical shift (or mean level) of the function.

The parameter b is simply the mean of all the extreme values in $[t_0, t_N]$:

$$b = \frac{1}{N}\sum_{i=1}^{N} \Delta_j^{\max}. \tag{7}$$

Step 3: Determine the frequency ω, measuring the time a sinusoid function takes to repeat a cycle.

Since we have $N+1$ zero crossings, there are $\frac{N}{2}$ periods in total. Therefore, recalling the definition of T_d above, we have

$$\omega = \frac{N\pi}{T_d}. \tag{8}$$

Step 4: Determine the phase θ_0 (or horizontal shift) of the function.

This is simply given by

$$\theta_0 = \omega t_0 = \frac{N\pi t_0}{T_d}. \tag{9}$$

Solving for these four parameters, the fitting sinusoid function $f(t) = A\sin(\omega t - \theta_0) + b$ is fully determined. There are several approaches to measure how good the fit is, including R^2, defined as the coefficient of determination or proportion of variance explained by the model, or the p-value based on an F test [25]. Here, we utilize a Mean Squared Error (MSE) metric. Given the sinusoid function $f(t)$, we discretize it based on the sampling points $t_i \in [t_0, t_0 + T_d]$ and obtain the values $f(t_i)$ corresponding to $\Delta(t_i)$. Comparing $f(t_i)$ with $\Delta(t_i)$, the MSE is defined as

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[f(t_i) - \Delta(t_i)]^2}. \tag{10}$$

The MSE can be interpreted as a measurement of the proximity of $\Delta(t)$ to $f(t) = A\sin(\omega t - \theta_0) + b$. We expect that non-actionable bumps have a better fit to a sinusoid, hence, lower MSE. Therefore, bumps with larger MSE, are identified as anomalies (see the green and red curves in Fig. 3).

2) A BUMP ENTROPY METRIC

In this approach, we resort to basic information theory where systems are modeled by a transmitter, channel, and receiver. Messages are sent from the transmitter through the channel, which in turn will modify the message in some way. The receiver attempts to infer the information contained in each message. *Entropy* (more specifically, *Shannon entropy*) is a measure of the expected value of the information [18]. Generally, entropy refers to disorder or uncertainty.

Messages can be modeled by any flow of information. In our context, the bump signature (original message) can be modeled as a smooth signal modified by actionable/non-actionable bumps (different channels). Since entropy is a measure of unpredictability of information content, we expect that more regular signatures are more predictable and contain less information, which means lower entropy. This suggests that one way to measure the degree of irregularity of a Δ-filtered signature is through the concept of entropy.

The simplest definition of entropy H is to associate it to a discrete random variable X taking on values $\mathbf{x} = (x_1, x_2, \cdots, x_N)$ with a probability mass function $P_X(\mathbf{x})$. Letting $I_X(\mathbf{x}) = E[-\log(P_X(\mathbf{x})]$ be the information content of X, we define

$$H_X(\mathbf{x}) = E[I_X(\mathbf{x})] = E[-\log P_X(\mathbf{x}))], \tag{11}$$

which can be also written as:

$$H_X(\mathbf{x}) = \Sigma_i P_X(x_i) I_X(x_i) = -\Sigma_i P_X(x_i)\log P_X(x_i). \tag{12}$$

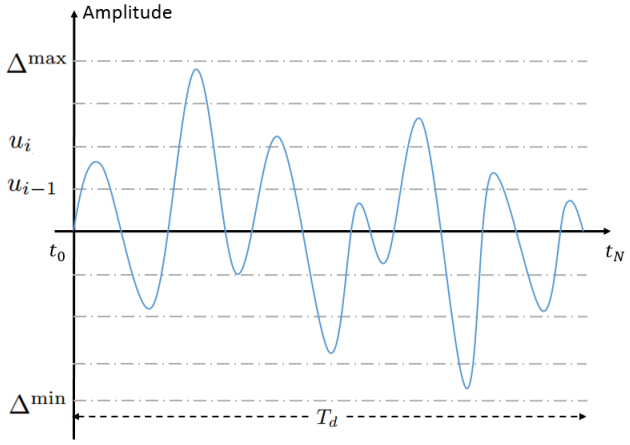

FIGURE 5. Amplitude partition of Δ-filtered signatures.

The question now is how to define the discrete random variable X and its corresponding probability mass function. Since we want to concentrate on different amplitude ranges of $\Delta(t)$, we set a maximal range $[\Delta^{\min}, \Delta^{\max}]$, where $\Delta^{\min}$ is the global minimum (lower bound) of all $\Delta(t)$ in the dataset and $\Delta^{\max}$ is the global maximum (upper bound). Then, we partition $[\Delta^{\min}, \Delta^{\max}]$ into sub-intervals $[u_{i-1}, u_i]$, $i = 1, \ldots, n$ (see Fig. 5), so that $u_0 = \Delta^{\min}$ and $u_n = \Delta^{\max}$. Regarding $\Delta(t)$ as a continuous-time signature, we then define X as the index of the interval into which $\Delta(t)$ falls, and the probability $p_i \equiv P_X(x_i)$ is defined as the fraction of time during which $\Delta(t) \in [u_{i-1}, u_i]$. Therefore, bump entropy is defined as

$$H = -\sum_{i=1}^{n} p_i \log(p_i). \tag{13}$$

Letting $\Delta^{-1}(\cdot)$ the inverse of the mapping $\Delta(\cdot)$, it follows that

$$p_i = \frac{1}{T_d}\sum_i \left|\Delta^{-1}(u_{i+1}) - \Delta^{-1}(u_i)\right|, \tag{14}$$

where T_d is the length of the time interval considered (similar to that of the sinusoid fitting approach). Finally, and in effort to further amplify the effect of extreme amplitudes, i.e., unusually large positive (near $\Delta^{\max}$) or negative amplitudes, we modify H for our purpose to

$$H = -\sum_{i=1}^{n} \log(p_i). \tag{15}$$

In this manner, a small number of extreme amplitudes can increase the bump entropy.

IV. PERFORMANCE EVALUATION

A. SUPERVISED CLASSIFICATION

We evaluate the performance of the learning algorithms by measuring two standard probabilities of error. The probability of *miss detection* corresponds to the fraction of

actionable bumps that are classified as non-actionable. The probability of *false alarm* corresponds to the fraction of the non-actionable bumps that are erroneously classified as actionable. By altering the decision threshold when classifying samples from a test set, we produce multiple pairs of the two error rates. We visualize the performance of the classifier by plotting the detection rate (1 minus the miss-detection rate) versus the false alarm rate. This produces the so called *Receiver Operating Characteristic (ROC)* curve. The point $(0, 0)$ on the curve corresponds to a decision rule which predicts all bumps as non-actionable, leading to 0% detection rate and 0% false alarm rate. Similarly, the point $(1, 1)$ on the ROC curve corresponds to a rule that predicts all bumps as actionable, leading to 100% detection rate and 100% false alarm rate. The ideal ROC curve, which would correspond to the best possible prediction method, is a piecewise linear curve that passes through points $(0, 0)$, $(0, 1)$ and $(1, 1)$. In contrast, the worst possible ROC curve, which corresponds to random guessing about the label of each bump, is a line that passes through points $(0, 0)$ and $(1, 1)$. All other classifiers produce ROC curves that lie between these two extreme curves. Depending on the tolerable false alarm rate (or detection rate), one can operate the classifier at a specific point of an ROC curve. We also report the *Area Under the ROC Curve (AUC)* (with values between 0 and 1), which is a summary statistic and is often used for model comparison. The AUC of the ideal classifier is 1, so the higher the AUC value of a classifier, the better.

B. ANOMALY DETECTION

The MSE metric in (10) and bump entropy in (15) both measure the degree of irregularity of a bump. Even though they are implemented in different ways, they share a common property: the larger the MSE/entropy, the more irregular a bump is. Therefore, it is simple to combine the MSE metric and bump entropy. Hence, we define the following *Anomaly Index* (*AI*):

$$AI = \lambda(MSE) + (1 - \lambda)H \tag{16}$$

associated with each bump, where the parameter $\lambda \in [0, 1]$ is selected to place more or less emphasis on the MSE or the entropy. Since we have signatures from the x, y and z accelerometer axis respectively, we can can possibly combine MSE and entropy metrics for each axis into an aggregate AI, or simply focus on a specific axis (z in our experimental results) which contains the most relevant information.

Based on AI, we can generate a prioritized list, such that the entries on the top of the list are the most likely to be actionable bumps requiring immediate attention, while those at the bottom of the list are the most likely to be non-actionable. The importance of this list lies in the fact that it provides simple information to the City Department of Public Works based on which it can prioritize bumps and dispatch limited resources (in the form of repair crews) where repairs are actually needed, while preventing unnecessary dispatching to non-actionable bump locations.

C. CLASSIFICATION AND ANOMALY DETECTION SYSTEM COMPARISON

The decision support and the anomaly detection systems can be seen as complementary but distinct approaches to the same problem. The first distinction is goal-oriented: the decision support system focuses on differentiating actionable bumps from non-actionable ones, while the anomaly detection system concentrates on identifying the most urgent actionable bumps.

Another key distinction relates to how these systems can be used. The first (decision support) system is based on machine learning/classification methods which are *supervised*, that is, they require a "labeled" training set to learn the various classifier parameters and thresholds. The anomaly detection system, on the other hand, is *unsupervised*; it simply ranks bumps based on the anomaly index we introduced. It provides no guarantees but suggests that higher ranked bumps are more likely to be actionable.

The consistency of the results of the two methods can be assessed by comparing whether the bumps at the top of the ranked anomaly detection list are also classified as actionable based on the decision support system. Moreover, as we discussed in Sec. III-A6, the supervised classification methods can also provide a metric of confidence the classifier has in a positive decision and this can be used to evaluate the top ranked bumps from the anomaly detection system.

V. EXPERIMENTAL RESULTS

Our results are based on an actual dataset with 813 bumps collected by the City of Boston. As we described in Sec. II, the number of samples in a bump varies between 48 to 65 samples, with the mode of the distribution being equal to 57. In our experiments, we set K (the number of bins we use to partition the signatures) to 3, and B (the number of bins we use to calculate the empirical measure $\mathbf{h}_{\mathbf{x}}^{B}(\cdot)$) to 5.

The number of features we use for the binary classification (actionable vs. non-actionable) is equal to 90. The data set is roughly balanced with 59% of the samples being actionable and 41% being non-actionable. To label the data, a camera was attached to the cars that drove in the city streets and based on the videos recorded the labeling was then done by experts in the City of Boston administration. From the bumps in the dataset we have omitted the ones in "screenable" categories, such as crosswalks, expansion joints, train tracks, speed bumps and road distortion/depression, since their location is known to the City and can thus be matched just by using location. We have also omitted data for bumps described as "unidentifiable" or otherwise ill-conditioned.

A. FALSE ALARM RATE VS. DETECTION RATE

In Fig. 6, we compare the various supervised classification methods. The ROC for each method captures the trade-off between the detection rate and the probability of false alarms. Table 1 provides the AUC for the various classification methods.

Some remarks are in order. AdaBoost achieves better performance than either logistic regression or SVM but is

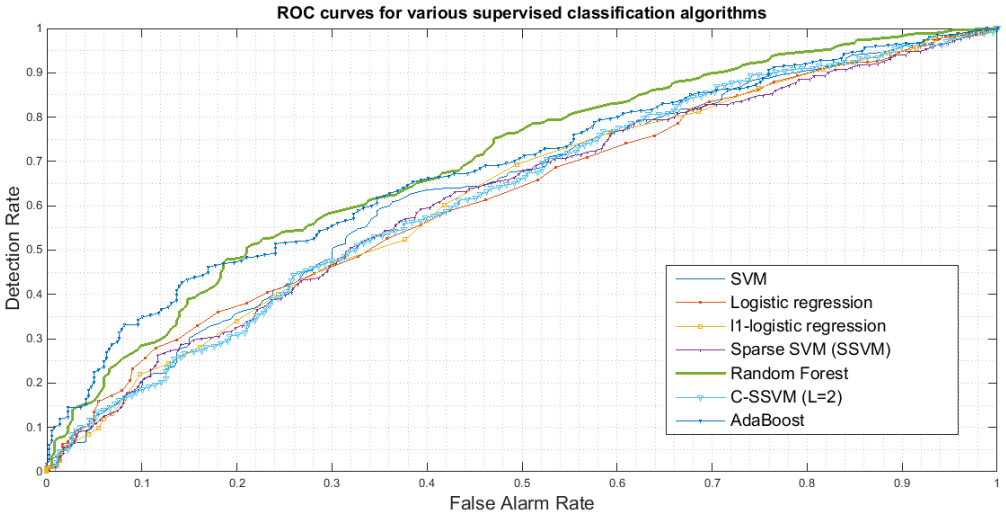

FIGURE 6. ROC curves for the classification methods.

TABLE 1. Area Under the ROC Curve (AUC) for various classification methods of the Street Bump decision support system.

Method	AUC
Logistic Regression	0.6219
ℓ_1 Regularized Logistic Regression	0.6225
SVM	0.6302
Sparse SVM	0.6205
AdaBoost	0.683
Random Forests	0.697
C-SSVM ($L = 2$)	0.6245
C-SSVM ($L = 3$)	0.6146
C-SSVM ($L = 4$)	0.6018

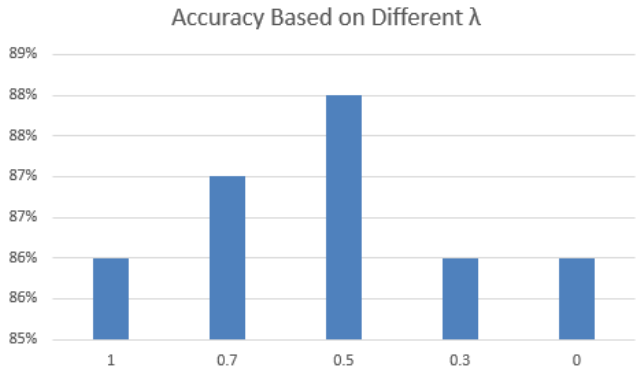

FIGURE 7. Parameter λ selection.

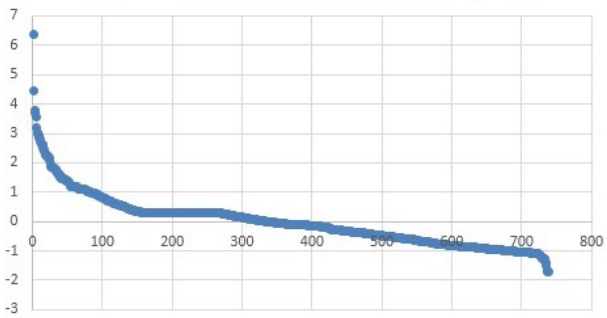

FIGURE 8. Normalized anomaly index values ($\lambda = 0.5$).

computationally more expensive to train. Regularization in either SVM or logistic regression does not appear to improve performance. As we have noted though earlier, it is critical in the context of the clustering and classification methods due to the (small) size of the training set for each cluster. Random forests yield the best overall performance among the various algorithms. We point out that with just 20% false alarm rate, we can correctly identify almost 50% of the actionable bumps. According to City of Boston officials, this is a level of performance that can enable the use of our algorithm in practice. Combining clustering and classification does not improve the performance, indicating that the bumps are not well separated into groups, which clearly shows how challenging the problem is.

B. TOP-N BUMPS ON THE ORDERED ANOMALY LIST

Since our anomaly detection system concentrates on the most urgent detected actionable obstacles, we limit ourselves to reporting a top-*N* list of actionable bumps. In order to determine the top-*N* actionable bumps, we use different parameters $\lambda = 1, 0.7, 0.5, 0.3, 0$ in (16) and plot the anomaly index *AI* values that correspond to each bump (see Fig. 8). Comparing over different weights, we can observe that the first 100 bumps usually have much larger *AI* values; for the rest, the rate with which *AI* decreases is much lower. Therefore, we limit ourselves to the top-*100* bumps on the list. Finally, in what follows we select the weight $\lambda = 0.5$ which yields the highest accuracy rate (88% as shown in Fig. 7) for detecting actionable bumps.

In Fig. 9 we limit ourselves to showing the top-27 of the bump prioritization list generated in descending order of

	OBSTACLE_NAME	ACTIONALBE vs. NON-ACTIONABLE	ANOMALY INDEX (λ=0.5)
2	Pot Hole	A	6.3581
3	Sunk Casting (Immediate repair)	A	4.4582
4	Pot Hole	A	3.7874
5	Cracking Around Casting (pothole)	A	3.6965
6	Bad Utility Patch (permanent)	A	3.5533
7	Pot Hole	A	3.2099
8	Pot Hole	A	3.0358
9	Bad Utility Patch (permanent)	A	2.9633
10	Flat Casting	N	2.9284
11	Pot Hole	A	2.836
12	Bad Utility Patch (permanent)	A	2.7836
13	Sunk Casting (Immediate repair)	A	2.7262
14	Catch Basin (repair)	A	2.6266
15	Pot Hole	A	2.6265
16	Sunk Casting (repair)	A	2.4961
17	Bad Utility Patch (permanent)	A	2.442
18	Pot Hole	A	2.3939
19	Sunk Casting (Immediate repair)	A	2.303
20	Sunk Casting (repair)	A	2.2712
21	Pot Hole	A	2.2538
22	Sunk Casting (repair)	A	2.2252
23	Sunk Casting (Immediate repair)	A	2.204
24	Cracking Around Casting (repair)	A	2.187
25	Pot Hole	A	2.1692
26	Bad Utility Patch (temporary)	A	2.0969
27	Pot Hole	A	1.9614
28	Sunk Casting (repair)	A	1.8764
29	...	...	...

FIGURE 9. **Bump list in descending *AI* order with weight $\lambda = 0.5$.**

anomaly index *AI*. Note that among the top 27 bumps in this list only a single non-actionable case is included (marked in yellow), illustrating the accuracy of the *Street Bump* system.

VI. CONCLUSIONS AND FUTURE WORK

The goal of this paper is to demonstrate how the ubiquitous availability of wireless devices can enable the development of effective infrastructure-free approaches for solving problems in Smart Cities. In particular, we have concentrated on the problem of detecting and classifying roadway obstacles (bumps) so as to differentiate between actionable bumps which correspond to obstacles that require immediate attention, and non-actionable bumps (e.g., cobblestone streets, speed bumps) for which no immediate action is needed. This classification enables City officials to efficiently and effectively prioritize repairs. We developed two complementary methods to that end. The first method uses classification algorithms. The second method introduces an anomaly index which captures the degree of regularity of a bump, and uses this index to differentiate between more "normal" bumps (non-actionable) from the "anomalous" (actionable) bumps.

As a next step of this work, it is important to be able to differentiate between different types of obstacles; for example, to distinguish a pothole from a poorly repaired sunk casting. The vision is that the accelerometer and GPS data collected by the app can be used in additional applications. An example is detecting wet or icy road conditions or obstacles causing vehicles to experience abrupt motions in a horizontal/lateral, rather than vertical direction. All these results, combined with the ones by our decision support system, could potentially be integrated to create a global "road smoothness" or "road quality" metric, available to all citizens through appropriate web sites, or specialized apps, or even integrated into map/navigation applications (Google maps, Waze, Apple maps, etc.), that can then be used to select the best route.

ACKNOWLEDGMENT

Special thanks to the City of Boston for collecting the data, and to the Connected Bits company for designing and developing the smartphone application.

REFERENCES

[1] United 865 Nations Dept. Economic Social Affairs, New York, NY, USA, (Jul. 2014). *World Urbanization Prospects*. [Online]. Available: http://esa.un.org/unpd/wup/highlights/wup2014-highlights.pdf

[2] K. Dresner and P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism," in *Proc. 3rd Int. Joint Conf. Auto. Agents Multiagents Syst.*, 2004, pp. 530–537.

[3] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *J. Artif. Intell. Res.*, vol. 31, pp. 591–656, Mar. 2008.

[4] A. de La Fortelle, "Analysis of reservation algorithms for cooperative planning at intersections," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2010, pp. 445–449.

[5] S. Huang, A. W. Sadek, and Y. Zhao, "Assessing the mobility and environmental benefits of reservation-based intelligent intersections using an integrated simulator," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1201–1214, Sep. 2012.

[6] K. Zhang, A. de La Fortelle, D. Zhang, and X. Wu, "Analysis and modeled design of one state-driven autonomous passing-through algorithm for driverless vehicles at intersections," in *Proc. IEEE 16th Int. Conf. Comput. Sci. Eng.*, Dec. 2013, pp. 751–757.

[7] Y. J. Zhang, A. A. Malikopoulos, and C. G. Cassandras, "Optimal control and coordination of connected and automated vehicles at urban traffic intersections," in *Proc. Amer. Control Conf.*, 2016. [Online]. Available: arXiv: 1362458

[8] Y. Geng and C. G. Cassandras, "New 'smart parking' system based on resource allocation and reservations," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1129–1139, Sep. 2013.

[9] Y. Geng and C. G. Cassandras, "Multi-intersection traffic light control with blocking," *J. Discrete Event Dyn. Syst.*, vol. 25, nos. 1–2, pp. 7–30, Jun. 2015.

[10] (Apr. 2014). *Massachusetts To Set Aside $40 Million To Fix Potholes*. [Online]. Available: http://boston.cbslocal.com/2014/04/09/894massachusetts-to-set-aside-40-million-to-x-potholes/

[11] T. S. Brisimi, S. Ariafar, Y. Zhang, C. G. Cassandras, and I. C. Paschalidis, "Sensing and classifying roadway obstacles: The street bump anomaly detection and decision support system," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Gothenburg, Sweden, Aug. 2015, pp. 1288–1293.

[12] W. Dai, T. S. Brisimi, W. G. Adams, T. Mela, V. Saligrama, and I. C. Paschalidis, "Prediction of hospitalization due to heart diseases by supervised learning methods," *Int. J. Med. Inform.*, vol. 83, no. 3, pp. 189–197, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.ijmedinf.2014.10.002

[13] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

[14] J. Wang, D. Rossell, C. G. Cassandras, and I. C. Paschalidis, "Network anomaly detection: A survey and comparative analysis of stochastic and deterministic methods," in *Proc. IEEE 52nd Annu. Conf. Decision Control (CDC)*, Dec. 2013, pp. 182–187.

[15] I. C. Paschalidis and G. Smaragdakis, "Spatio-temporal network anomaly detection by assessing deviations of empirical measures," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 685–697, Jun. 2009.

[16] J. Wang and I. C. Paschalidis, "Statistical traffic anomaly detection in time-varying communication networks," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 2, pp. 100–111, Jun. 2015.

[17] J. Zhang and I. C. Paschalidis, "An improved composite hypothesis test for Markov models with applications in network anomaly detection," in *Proc. IEEE 54th Annu. Conf. Decision Control*, Osaka, Japan, Dec. 2015, pp. 3810–3815.

[18] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[19] I. C. Paschalidis, W. Dai, and D. Guo, "Formation detection with wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 10, no. 4, 2014, Art. ID 55.

[20] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[21] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[22] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
[23] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, no. 11, pp. 1119–1125, 1994.
[24] S. Vassilvitskii and D. Arthur, "k-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2006, pp. 1027–1035.
[25] R. G. Lomax and D. L. Hahs-Vaughn, *Statistical Concepts: A Second Course*. London, U.K.: Routledge, 2013.

THEODORA S. BRISIMI received the Diploma degree from the National Technical University of Athens, Athens, Greece, in 2011. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Boston University, working with Prof. I. C. Paschalidis on developing and applying new techniques in machine learning with applications to healthcare and smart cities. She is also involved in Boston's Street Bump Project, work in collaboration with the City of Boston, to detect fixable bumps on city streets. She also has industrial experience, working during the summer of 2014 as an Intern with Mitsubishi Electric Research Labs and during the summer of 2015 as an Intern with the Palo Alto Research Center, a Xerox company. Her current research focus is on using classification methods to predict future hospitalizations of patients based on their electronic health records history (in collaboration with the Boston Medical Center).

CHRISTOS G. CASSANDRAS (F'96) received the B.S. degree from Yale University, New Haven, CT, USA, in 1977, the M.S.E.E. degree from Stanford University, Stanford, CA, USA, in 1978, and the M.S. and Ph.D. degrees from Harvard University, Cambridge, MA, USA, in 1979 and 1982, respectively. He was with ITP Boston, Inc., Cambridge, from 1982 to 1984, where he was involved in the design of automated manufacturing systems. From 1984 to 1996, he was a Faculty Member with the Department of Electrical and Computer Engineering, University of Massachusetts at Amherst, Amherst, MA, USA. He is currently a Distinguished Professor of Engineering with Boston University, Brookline, MA, USA, the Head of the Division of Systems Engineering, and a Professor of Electrical and Computer Engineering. He specializes in the areas of discrete event and hybrid systems, cooperative control, stochastic optimization, and computer simulation, with applications to computer and sensor networks, manufacturing systems, and transportation systems. He has authored over 350 refereed papers in these areas, and five books.

Dr. Cassandras is a member of Phi Beta Kappa and Tau Beta Pi. He is also a fellow of the International Federation of Automatic Control (IFAC). He was a recipient of several awards, including the 2011 IEEE Control Systems Technology Award, the 2006 Distinguished Member Award of the IEEE Control Systems Society, the 1999 Harold Chestnut Prize (IFAC Best Control Engineering Textbook), a 2011 prize and a 2014 prize for the IBM/IEEE Smarter Planet Challenge Competition, the 2014 Engineering Distinguished Scholar Award at Boston University, several honorary professorships, a 1991 Lilly Fellowship, and a 2012 Kern Fellowship. He was the Editor-in-Chief of the IEEE Transactions on Automatic Control from 1998 to 2009. He serves on several editorial boards and has been a Guest Editor for various journals. He was the President of the IEEE Control Systems Society in 2012.

CHRIS OSGOOD is currently with the City of Boston. He is pursuing degrees with City Year, Haverford College, and the Harvard Business School. He joined Mayor Menino's administration in 2006, serving as a Policy Advisor and working on the team that implemented the City's performance management program and rebuilt its 24 h hotline. He serves as a Boston Mayor Walsh's Chief of the Streets, working with the Public Works and Transportation Departments to deliver City services, build/maintain streets, and implement a transportation plan that works for everyone. He co-founded, in 2010, the Mayor's Office of New Urban Mechanics, a nationally replicated civic innovation group that experiments with new ways of using data, design, and technology to improve the constituent experience.

IOANNIS CH. PASCHALIDIS (M'96–SM'06–F'14) received the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 1993 and 1996, respectively. In 1996, he joined Boston University. He co-founded, in 2010, the Mayor's Office of New Urban Mechanics, a nationally replicated civic innovation group that experiments with new ways of using data, design, and technology to improve the constituent experience. He is currently a Professor and Distinguished Faculty Fellow with Boston University with appointments in the Department of Electrical and Computer Engineering, the Division of Systems Engineering, and the Department of Biomedical Engineering. He is also the Director of the Center for Information and Systems Engineering. His current research interests lie in the fields of systems and control, networking, applied probability, optimization, operations research, computational biology, and medical informatics.

Dr. Paschalidis was a recipient of the NSF CAREER Award (2000), several best paper and best algorithmic performance awards, and a 2014 IBM/IEEE Smarter Planet Challenge Award. He was an invited participant at the 2002 Frontiers of Engineering Symposium by the U.S. National Academy of Engineering and the 2014 U.S. National Academies Keck Futures Initiative Conference. He is the inaugural Editor-in-Chief of the IEEE Transactions on Control of Network Systems.

YUE ZHANG (S'15) received the B.S. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2013.

She is currently pursuing the Ph.D. degree with the Division of Systems Engineering and the Center for Information and Systems Engineering, Boston University. Her research interests include discrete event systems and optimal control of hybrid systems, with applications to intelligent transportation systems; machine learning, and big data analytics.