

RESEARCH

Detection and Localization of Harmful Atmospheric Releases via Support Vector Machines

R. Taylor Locke^{*} and Ioannis Ch Paschalidis[†]

^{*}Correspondence: rtl@bu.edu
Department of Electrical &
Computer Engineering, Boston
University, 8 St Mary's St., 02215
Boston, MA, USA
Full list of author information is
available at the end of the article
[†]yannisp@bu.edu

Abstract

Background: We present a Support Vector Machine (SVM) approach to the localization of hazardous particulate releases in an urban area using features constructed only from measurements obtained from a network of sensors.

Results: We find high levels of localization accuracy when a reasonable number of noisy sensors are deployed within the environment. We also compare SVM source localization performance to an existing stochastic localization technique over varying degrees of sensor noise and find it favorable for areas prone to urban canyon turbulence effects.

Conclusions: This approach is in contrast to earlier works which either use solutions to inverse dispersion problems for localization or apply maximum likelihood techniques. By using established SVM results, we also tackle the problems of release detection and optimal sensor placement.

Keywords: Environmental monitoring; urban monitoring; tracking; simulation; machine learning

Background

Modern cities face a multitude of atmospheric pollution threats from a plethora of sources, including large industrial plants in the outskirts of the city, smaller production or processing units in industrial parks or even within the city, warehouses and storage facilities, and large underground networks of gas pipes that are particularly leaky and can release methane.

In addition to the accidental release of some harmful pollutant in the atmosphere, an increasing concern are terrorist threats involving the deliberate release of some *Chemical, Biological, Radiological, or Nuclear (CBRN)* material.

With either accidental or deliberate releases, it is critical to not only provide a reliable indication of the presence of hazardous particles within the atmosphere (*release detection*) but to also indicate where the event originated (*release localization*). Understanding the source's point in time and space can guide emergency responders and enable fast corrective actions to be more precisely targeted. In the case of a CBRN attack, for example, antidotal medications which are potentially already in short supply can be administered only to people who were exposed and in danger of succumbing to attack related illness or injury. Similarly, in the case of an accidental release, source localization can guide forensics and enable the prevention of future releases.

Prior work in the literature has tackled source localization by tracing, through dispersion modeling, observed particle concentrations back in time and space to the presumed origin [1, 2, 3]. The alternative presented in [4] and [5], as well as the alternative presented in this article, differ in that they depend only on *sensor measurements* and therefore do not require solution to complicated inverse dispersion problems. Additionally, and quite importantly, these source localization techniques provide insight to the related problem of placing sensors.

This work, as well as the work in [4] and [5], does require some a priori computation in lieu of the design of an analytical dispersion model. Specifically, knowledge of particulate dispersion behavior is needed in order to make sense of sequences of particulate concentration observations. Considering that large scale, physical simulant releases in urban areas is prohibitively costly in both time and money, we employ accurate numerical simulations of particulate dispersions in a Monte Carlo fashion to develop a mathematical characterization of sensor measurements under a variety of scenarios. A technique that we have found useful, and adapt for our purposes, which is described in greater detail in the Results and Discussion Section, is the *Lattice Boltzmann Method (LBM)*. This numerical dispersion model is advantageous to the problem at hand because it handles complex geometries and changing phenomena typical of urban areas quite naturally. Also, LBM is easily parallelizable for use on a computational grid. This allows us to perform the numerous dispersion simulations required for enhanced accuracy of the presented localization methodology and enables large deployments of our methodology.

The major contributions of this article are:

- 1 a new, deterministic localization methodology that does not rely on solving any sophisticated inverse dispersion problem and is an alternative to the stochastic localization methodology presented in [5];
- 2 a novel sensor placement methodology that stems from a machine learning feature selection procedure; and
- 3 and a novel procedure inspired by machine learning techniques for the detection of hazardous atmospheric releases.

To our knowledge, no other machine learning hazardous release localization approaches have been presented in the relevant literature. The work contained in [6] and [7] could be considered similar, however the localization processes presented therein employ Bayesian statistics as opposed to SVMs. A component that separates this work from [6] and [7] is the development of feature vectors that could be applied to any machine learning technique. Also, our approach is not reliant on a forward propagation model.

The work presented here extends dispersion model-free localization into the realm of data-science and opens the door for application of continually evolving swath of machine learning algorithms to the problems of release detection and release localization.

While the focus of this work is urban environments, other environments, at all scales, are also amenable to the presented localization and detection techniques. For instance, downstream river contamination monitoring and underground chemical seepage are tangible problems. Indoor environments, such as nuclear power plants and chemical processing plants, also present potential monitoring applications.

Related Work

Existing source localization approaches [1, 2, 3] observe particulate concentrations and solve the inverse problem of tracing dispersion backward in time and space to the source of the release. Limitations of this methodology stem from the irregular and dynamic phenomena typically found in urban areas. For instance, buildings within a city tend to have irregular geometries and a plethora of external surface textures. Additionally, the micro-climate effects of urban canyon turbulence under generally uncertain weather conditions are challenging to model. All of these characteristics of urban environments work in unison to make source localization via solving inverse dispersion problems difficult. In [1] the presence of challenging geographies and wind turbulence is accommodated by incorporating Monte Carlo simulation of fluid dispersion. However, all of these works suffer from the difficulty of determining, without a detection process, the point in time in which the release began. Barring this information, these inverse problem approaches can lead to erroneous localizations.

The work presented here provides a deterministic accompaniment to the stochastic localization approach presented in [5]. There, considering that under a release, large (small) particulate concentrations observed at one time instance by a sensor are likely to be followed by similarly large (small) particulate concentrations, environmental sensor observations are modeled as a first-order Markov chain. This construct is made tenable by first encoding real-valued sensor concentration observations into a finite set of concentration states. Through the use of Monte Carlo simulation, marginal and transition probabilities of the concentration states are derived empirically to construct probability laws for concentration state evolution under a plethora of release scenarios. Localization is then performed through hypothesis tests that compare current empirical concentration state distributions to the previously derived probability laws. This approach is completely sensor measurement based and therefore does not fall victim to the inherent challenges of producing and then solving an inverse dispersion model.

The SVM-based localization technique presented in this article is similar to the stochastic approach in [5] in that it relies on previously conducted Monte Carlo simulation of release scenarios to build *training sets*. It is also entirely sensor measurement based, which brings the added benefit of not relying on solving an inverse problem. It differs, however, in that the training data is used to build *deterministic* decision boundaries that indicate the location of a detected release.

Methods

An SVM is a well established machine learning technique for binary classification problems [8]. The premise is to construct the *decision function*

$$f(\mathbf{x}) = \sum_{i=1}^m y^i \alpha_i K(\mathbf{x}, \mathbf{x}^i) - b. \quad (1)$$

We classify a test data point $\mathbf{x} \in \mathbb{R}^n$, by assigning it a label equal to the value of $\text{sgn}(f(\mathbf{x}))$. We review the aspects of SVMs pertinent to source localization in the remainder of this section.

The values y^i and \mathbf{x}^i , $i = 1, \dots, m$, come from a *training set*. The parameters α_i , $i = 1, \dots, m$, and b are found via a training procedure. The function $K(\cdot, \cdot)$ denotes any of a set of *kernel functions*. Common choices include the Gaussian kernel, $K(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{(\mathbf{u}-\mathbf{v})^T(\mathbf{u}-\mathbf{v})}{\sigma}\right)$, and the polynomial kernel of degree d , $K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^T \mathbf{v} + 1)^d$, where $(\cdot)^T$ denotes transpose.

By labeling a pattern \mathbf{x} by the sign of the decision function (1), we are classifying it according to which side of a hyperplane it falls on. The use of kernel functions effectively augments the decision making space, thus allowing for accurate classification even in cases where two classes of data are not linearly separable.

Feature Selection for SVMs

Feature selection can be seen as a technique for dimension reduction. The objective is to choose a subset of the existing features by excluding those that provide little benefit in differentiating between two classes of data. Features selected in this way (i.e., features that were not excluded) remain intact. This is in opposition to techniques like principal component analysis in which the resulting features are linear combinations of the original features.

A feature selection method designed specifically for SVMs is presented in [9]. There, the authors find a binary vector $\boldsymbol{\sigma} \in \{0, 1\}^n$ whose positive elements indicate selection of k features. For a given $\boldsymbol{\sigma}$, a modified kernel,

$$K_{\boldsymbol{\sigma}}(\mathbf{u}, \mathbf{v}) = K(\mathbf{u} \bullet \boldsymbol{\sigma}, \mathbf{v} \bullet \boldsymbol{\sigma}),$$

is used to build an SVM, where $\tilde{\mathbf{u}} = \mathbf{u} \bullet \boldsymbol{\sigma} \in \mathbb{R}^k$ is a vector whose elements are those of \mathbf{u} corresponding to the elements of $\boldsymbol{\sigma}$ which are equal to one.

The *radius* of a modified kernel, R^2 , is the radius of the smallest sphere that includes the training patterns in \mathcal{X} after translation into the modified feature space. That is, R is the minimum non-negative value such that

$$[\Phi(\boldsymbol{\sigma} \bullet \mathbf{x}^i) - \mathbf{a}]^T [\Phi(\boldsymbol{\sigma} \bullet \mathbf{x}^i) - \mathbf{a}] \leq R^2, i = 1, \dots, m,$$

where \mathbf{a} denotes the center of the sphere. Lagrange duality produces a QP which defines the radius of a modified kernel,

$$\begin{aligned} \max_{\boldsymbol{\beta}} \quad & \sum_{i=1}^m \beta_i K_{\boldsymbol{\sigma}}(\mathbf{x}^i, \mathbf{x}^i) - \sum_{i,j=1}^m \beta_i \beta_j K_{\boldsymbol{\sigma}}(\mathbf{x}^i, \mathbf{x}^j) \\ \text{subject to} \quad & \sum_{i=1}^m \beta_i = 1, \\ & \beta_i \geq 0, \quad i = 1, \dots, m. \end{aligned} \quad (2)$$

A useful theorem, found in [9] but proven, in effect, in [10] relates the likelihood of erroneous classification to the radius found by (2).

Error Rate Theorem *Let $E[P_{err}]$ denote the expected probability of erroneous classification of an SVM trained on a data set of m elements. If the data in $\mathcal{X} = \{\mathbf{x}^1, \dots, \mathbf{x}^m\}$ with radius R are separable with a corresponding margin of ρ ,*

$$E[P_{err}] \leq \frac{1}{m} E \left[\frac{R^2}{\rho^2} \right] = \frac{1}{m} E [R^2 W^2(\boldsymbol{\alpha}^*)],$$

where $W^2(\boldsymbol{\alpha}^*)$ denotes the optimal value of the SVM soft margin objective function.

This theorem grants us an excellent metric for which different modifications of a kernel can be measured. An optimal $\boldsymbol{\sigma}$ is found as the minimizer of

$$R^2(\boldsymbol{\beta}^*; \boldsymbol{\sigma})W^2(\boldsymbol{\alpha}^*, C; \boldsymbol{\sigma}), \quad (3)$$

subject to $\sum_{i=1}^n \sigma_i = k$, where $W^2(\boldsymbol{\alpha}^*, C; \boldsymbol{\sigma})$ represents the optimal objective function of the soft margin maximization problem with a kernel modified by $\boldsymbol{\sigma}$, and $R^2(\boldsymbol{\beta}^*; \boldsymbol{\sigma})$ is the radius of kernel $K_{\boldsymbol{\sigma}}(\cdot, \cdot)$. By minimizing (3), features are selected according to excluding features that contribute least to discerning between two classes of data.

Ordinarily, optimization of the function (3) would prove to be a computationally intense search through the large number of possible feature combinations. Fortunately, derivative information for (3) is provided in [9]. Relaxing $\boldsymbol{\sigma}$ to be a real-valued vector on the unit hyper-square, one can iteratively minimize (3) using an efficient gradient based solver. After each iteration the features whose corresponding elements of $\boldsymbol{\sigma}$ are sufficiently close to zero or one are excluded or included, respectively. This process continues until the desired number of features remain.

Source Localization

Source localization is the problem of locating the origin of a hazardous atmospheric release in an urban environment with SVMs that operate solely on information obtained from a network of sensors deployed throughout the city under surveillance. These sensors are assumed to provide sequences of measured concentrations of a target particle. We assume for the time being that sensors have already been placed at K locations from a discrete set of locations $\mathcal{B} = \{B_1, \dots, B_M\}$. We defer to the problem of selecting which locations from \mathcal{B} will provide better localization performance to a later section. Our goal is then to, upon observing sequences of sampled particulate concentrations, determine which location from the set $\mathcal{L} = \{L_1, \dots, L_N\}$ the particles originated.

Feature Representation

Assuming concentration values to be sampled according to a fixed time interval, we represent concentration evolution observed by a sensor at location B_k as a sequence

$$c(k) = c_1^k, c_2^k, \dots, \quad (4)$$

where c_t^k denotes the real-valued sampled concentration at location B_k at discrete time step t .

Several options are available to encode the sequences in (4) into feature vectors. For instance, the patterns could simply be composed of the first, say, n samples in $c(k)$, producing patterns of dimension nK of the form

$$\mathbf{x} = (c_1^{k_1}, \dots, c_n^{k_1}, \dots, c_1^{k_K}, \dots, c_n^{k_K}),$$

where k_1, \dots, k_K depict the K location indices where sensors are deployed. A potentially smaller feature vector is the quantization of concentration observations made by taking aggregate mean concentrations over several consecutive elements of $c(k)$. If these means are taken over, say, m elements, the resulting patterns of dimension $\lceil \frac{n}{m} \rceil K$, where $\lceil \cdot \rceil$ denotes rounding up to the next integer, take the form

$$\mathbf{x} = \left(\frac{1}{m} \sum_{t=1}^m c_t^{k_1}, \dots, \frac{1}{m} \sum_{t=m(n-1)+1}^{mn} c_t^{k_1}, \dots, \frac{1}{m} \sum_{t=1}^m c_t^{k_K}, \dots, \frac{1}{m} \sum_{t=m(n-1)+1}^{mn} c_t^{k_K} \right).$$

In the simplest case, where $m = n$, these features are K dimensional and take the form

$$\mathbf{x} = \left(\frac{1}{n} \sum_{t=1}^n c_t^{k_1}, \dots, \frac{1}{n} \sum_{t=1}^n c_t^{k_K} \right).$$

Another feature space we consider consists of the maximum value in the sequence $c(k)$ and its time step index. This feature contains only two elements per sensor, but still contains a differentiating feature (maximum concentration) as well as a temporal measure (time index), thus making it adequate for detecting differences in release scenarios. Patterns produced under this paradigm take the form

$$\mathbf{x} = (\max_t c(k_1), \arg \max_t c(k_1), \dots, \max_t c(k_K), \arg \max_t c(k_K)).$$

SVM Localization

A question of how to adapt a binary classifier to select one of N release locations remains. For localization, we look to methods that repeatedly call upon the outcomes of binary SVM evaluations to select, based on the feature spaces described in Section , a release location. In effect, we are constructing an N -class classifier out of several binary classifiers. Popular approaches revolve around the idea of setting up some sort of tournament of several binary classifiers. Each match within the tournament prevents a test pattern from being labeled as a particular class. The class remaining at the conclusion of the tournament is assigned to the test pattern.

In [11], a comparison of the so-called “one-against-all”, “one-against-one”, and Directed Acyclic Graph (DAGSVM) methods for multi-class SVMs is conducted. In the “one-against-all” approach, N classifiers of the form (1) are found, where N is the number of classes. For classifier i , an SVM is trained with patterns belonging to class i labeled as 1 and those patterns belonging to all other classes as -1 . When evaluating a test pattern \mathbf{x} , the class whose SVM achieves the maximum value of (1) is assigned.

In the “one-against-one” approach, $\frac{N(N-1)}{2}$ SVMs are trained. For each class combination (i, j) , $i < j$, a binary SVM is trained on data belonging only to classes i and j . A test pattern \mathbf{x} is assigned the class that was assigned the most number of times out of all of the binary classifications. DAGSVM is similar to the “one-against-one” approach, but instead of considering each of the $\frac{N(N-1)}{2}$ binary SVMs,

it assigns labels by starting with the decision made on a particular class pair. Based on the results of the first classification, binary classification between the previously selected class and another particular class is performed. This process continues until no more prescribed comparisons remain.

Based on the results in [11] and ease of implementation, we use the “one-against-one” multi-class method for source localization throughout the following. From a large set of sensor concentration sequences \mathcal{C} , obtained from a particulate dispersion simulator, we form feature vectors according to one of the paradigms listed in Section to produce a training set \mathcal{X} . Out of \mathcal{X} we build $\frac{N(N-1)}{2}$ training sets denoted $\mathcal{X}_{1,2}, \mathcal{X}_{1,3}, \dots, \mathcal{X}_{N-2,N}, \mathcal{X}_{N-1,N}$, where $\mathcal{X}_{i,j}$ represents the subset of \mathcal{X} that consists only of training patterns obtained through simulation under release locations L_i and L_j . Then, for $i, j = 1, \dots, N$ and $i < j$, we use $\mathcal{X}_{i,j}$ to train a binary classifier of the form (1), denoted as $f_{i,j}(\mathbf{x})$. For a new test vector \mathbf{x} , produced from newly observed concentration sequences, we select either location L_i or L_j according to the sign of $f_{i,j}(\mathbf{x})$ for all i and j , with $i < j$. The location out of \mathcal{L} that was selected the most number of times out of all of the $\frac{N(N-1)}{2}$ selections is chosen as the release location.

Sensor Placement

[The question remains](#) of, if we are allowed to place sensors anywhere within the set of possible sensor locations \mathcal{B} , which sensor locations should be selected? A straightforward extension of the SVM localization method provides a guideline for placement of available sensors. Using the feature selection methods previously discussed, we can select K of the M sensor locations.

To illustrate this procedure, assume patterns are originally formed by M features of dimension d , with all potential sensor locations’ associated features included. That is, a pattern is originally of the form $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_M)$ where $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,d})$ is a d -dimensional feature of one of the forms discussed in Section or some other form. We select sensor locations by following the feature selection procedure of minimizing (3) with respect to $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_1, \dots, \boldsymbol{\sigma}_M)$, where $\boldsymbol{\sigma}_i \in \{0, 1\}^d$ for $i = 1, \dots, M$. More specifically, allowing the notation $\boldsymbol{\sigma}_i = (\sigma_{i,1}, \dots, \sigma_{i,d})$, we minimize

$$R^2(\boldsymbol{\beta}^*; \boldsymbol{\sigma})W^2(\boldsymbol{\alpha}^*, C; \boldsymbol{\sigma}) \tag{5}$$

subject to the constraints

$$\begin{aligned} \sum_{i=1}^M \sum_{j=1}^d \sigma_{ij} &= Kd, \\ \sigma_{i,1} = \dots = \sigma_{i,d}, & \quad i = 1, \dots, M, \\ \boldsymbol{\sigma}_i &\in \{0, 1\}^d, \quad i = 1, \dots, M. \end{aligned}$$

Here, R^2 is the radius of the modified kernel $K_{\boldsymbol{\sigma}}(\cdot, \cdot)$ and is the optimal solution to

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \sum_{i=1}^{|\mathcal{X}|} \beta_i K_{\boldsymbol{\sigma}}(\mathbf{x}^i, \mathbf{x}^i) - \sum_{i,j=1}^{|\mathcal{X}|} \beta_i \beta_j K_{\boldsymbol{\sigma}}(\mathbf{x}^i, \mathbf{x}^j) \\ \text{subject to} \quad & \sum_{i=1}^{|\mathcal{X}|} \beta_i = 1, \\ & \beta_i \geq 0 \quad i = 1, \dots, |\mathcal{X}|, \end{aligned} \tag{6}$$

and W^2 is the optimal solution to

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{|\mathcal{X}|} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{|\mathcal{X}|} \alpha_i \alpha_j y^i y^j K_{\boldsymbol{\sigma}}(\mathbf{x}^i, \mathbf{x}^j) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, |\mathcal{X}|, \\ & \sum_{i=1}^{|\mathcal{X}|} \alpha_i y^i = 0, \end{aligned} \quad (7)$$

for some prescribed C . In both (6) and (7) we use the convention introduced previously, where

$$K_{\boldsymbol{\sigma}}(\mathbf{u}, \mathbf{v}) = K(\mathbf{u} \bullet \boldsymbol{\sigma}, \mathbf{v} \bullet \boldsymbol{\sigma})$$

is the kernel modified by $\boldsymbol{\sigma}$, where $\mathbf{u} \bullet \boldsymbol{\sigma}$ returns a vector whose elements are those of \mathbf{u} that correspond to the elements of $\boldsymbol{\sigma}$ that are equal to one. The minimizer $\boldsymbol{\sigma}^*$ of (5) becomes the vector indicating which K out of the M present d -dimensional features are to be used in localization. Thus, $\boldsymbol{\sigma}^*$ effectively selects which K locations sensors should be placed.

This process is replicated for each of the $\frac{N(N-1)}{2}$ binary SVMs used in the “one-against-one” localization technique. Sensors are placed at the K locations that are selected most frequently among these replications.

Release Detection

An important component to locating the origin of a particulate release by concentration sampling is the amount of time between subsequent observations. It is hard to construct a feature space that does not rely on the starting time of the release. What is needed is a trigger mechanism that alarms immediately upon detecting a hazardous atmospheric dispersion.

Detection problems involving noisy observations have long been an area of research. When hazardous particulates are not present, sensor observations are purely due to sensor noise. If we make the assumption that sensor noise is independent and identically distributed (iid) and behaves according to a known distribution, an elementary approach is to set, through analysis of the noise distribution and a tolerable false alarm rate, a concentration threshold. Whenever any sensor observes a value greater than this threshold, the time of the alarm can be used as the originating time of the release.

An SVM approach to release detection is the so-called “one-class SVM” method of novelty detection[12]. One-class SVM is similar to the binary SVM in its form and training, except that only representative training elements from a single set are available for analysis. The premise is, in the absence of one class’s training patterns, to treat the origin of the higher dimensional space identified through the choice of a kernel function as the only member of the opposite class. Then, a hyperplane that separates all but a controlled number of training patterns from the origin effectively becomes an anomaly detector.

In the case of release detection, we train a one-class SVM using only features of the form described in Section that represent sensor observations when no release is present. These patterns would therefore represent the perturbations in concentration observations that result from sensor noise or routine false alarming due to the presence of some non-target particulate. When any of the sensor observations are

declared anomalous by the one-class SVM, we would presume that this is because a sensor's observations depart too largely from the usual behavior and declare a release.

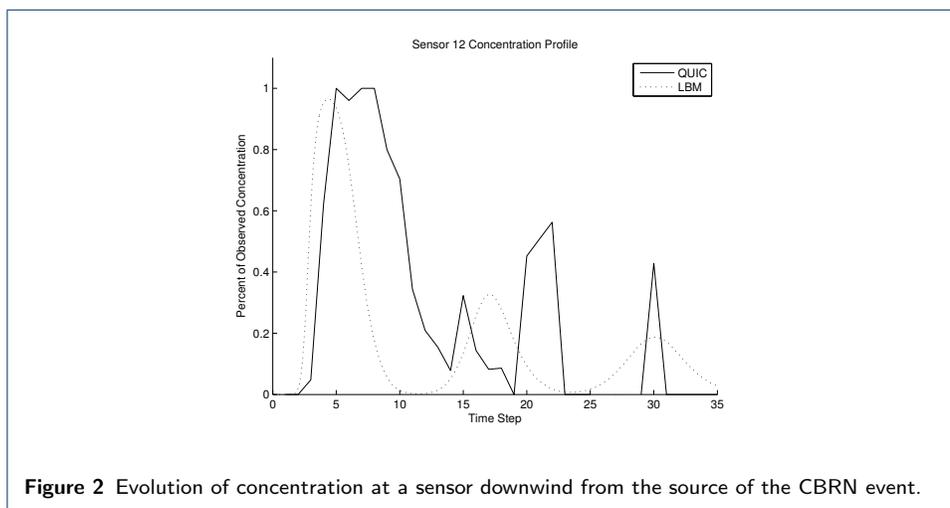
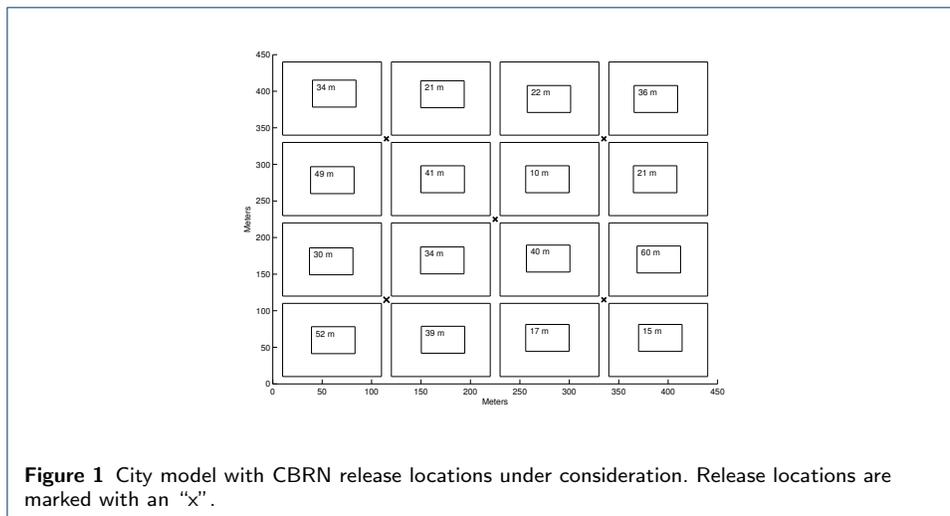
Results and Discussion

To demonstrate the performance of the sensor placement approach and the localization methodology, we simulated several point release scenarios in an illustrative environment. We simulated CBRN releases in the *Quick Urban & Industrial Complex (QUIC)* Dispersion Modeling System [13] developed at the Los Alamos National Laboratory. QUIC first solves the fluid dynamics problem of determining local wind eddies throughout a modeled three-dimensional, outdoor setting using the methods of Röckle [14]. Using the fluid flow solution, QUIC simulates the travel of CBRN particulates via a Lagrangian random walk. Previously, the QUIC codes have been tested and validated for real-world situations [13].

Additionally, we simulated CBRN releases using the *Lattice Boltzmann Method (LBM)*. LBM evolved from the numerical fluid modeling technique *Lattice Gas Automata (LGA)*, in which parcels of air adhere to microscopic laws which dictate their movement. Macroscopic values of flow velocities and densities are then derived by the underlying microscopic properties propagated by the algorithm [15]. Unfortunately, LGA often falls victim to instability in the face of statistical noise [16]. LBM extends LGA by considering air parcel movement more notionally by modeling microscopic air parcel velocities as distributions in the *Lattice Boltzmann Equation (LBE)*. It has been shown that under reasonable starting conditions LBM provides accurate approximations to fluid flows. The presented work continues the precedent set in [4] by using LBM in Monte Carlo simulation for analysis of CBRN events. It is not hard to imagine that it will be of use to future analysts as well. The macroscopic Navier-Stokes equations can be recovered from the microscopic LBE and LBM is easily adapted to parallel computation [17]. This last characteristic is of particular interest, as large-scale real-world applications have the potential to require large amounts of computation.

Our modeled environment consists of geometries typical of dense urban areas, consisting of a city grid, four blocks-by-four blocks. Each block is 100 meters-by-100 meters with 10 meter-wide through-ways. Each block's height is drawn randomly from the uniform distribution ranging from 20 to 60 meters. Sensors are allowed to be placed at any intersection and five intersections are considered as potential release locations. The shape of the grid, as well as the location of simulated releases, is shown in Figure 1. The agent concentration profiles of the two different dispersion models at a CBRN sensor located down wind of the point release are compared in Figure 2. These models have different discretizations of the three-dimensional model and hence produce concentration values that differ in scale. This has been accommodated in Figure 2 by reporting the percentage of all observed concentrations reported by a single sensor downwind from a release. The LBM model produces a much smoother agent concentration evolution than the QUIC model. Thus, noise within the evaluation of the proposed methodologies when the LBM data are used is primarily due to the sensor false alarm model rather than the dispersion model.

QUIC is selected for testing purposes as it is representative of popular and traditional dispersion simulation approaches. LBM serves as a contrast to QUIC in



modeling as well as in emerging atmospheric modeling trends that are intended to be scalable through parallelized computation. But it is important to note, the purpose of the presented work is hazardous atmospheric release detection and localization. The requisite data simulation process is simulator agnostic. We present results obtained by using training data generated from both simulators as a means to empirically demonstrate that assertion.

We consider 40 unique CBRN event scenarios, each containing a single point release of the same mass, spanning the five different release locations within the grid with wind blowing at 1 m/s or 5 m/s and originating from the four cardinal directions. A large set of training data was constructed via Monte Carlo simulation over each combination of wind direction, wind speed, and release location (data is available upon request.)

We use a real valued sensor model with additive white noise. We model these sensors as

$$\hat{C} = C + N(0, \sigma_\epsilon) \tag{8}$$

where C is the actual concentration of particulate present at the sensor's location, $N(0, \sigma_\epsilon)$ denotes a normally distributed random variable with mean 0 and standard deviation σ_ϵ , and \hat{C} denotes the sensor's reported concentration observation. First, in what will be referred to as the *mild noise case*, we use a value of σ_ϵ which produces an equivalent false alarm rate of 0.125. Also, in what will be referred to as the *large noise case*, we replace the normal random variate in (8) with $N(0, 10\sigma_\epsilon)$. The error term is intended to model both measurement error in the sensor and random perturbations in release concentration realizations. If a higher fidelity sensor is used, then the low-noise sensor data is a better model. Likewise, low-fidelity sensors correspond to the high-noise case.

SVM CBRN Localization Evaluation

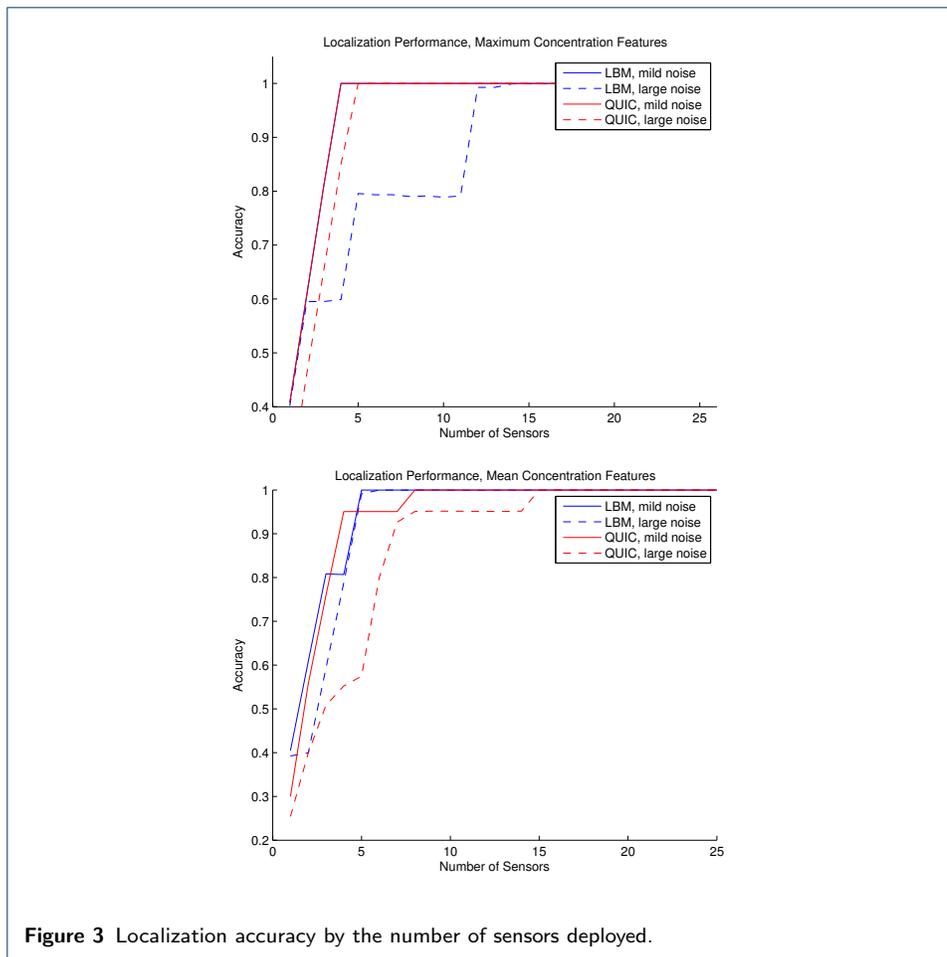
We construct a test set for localization and placement performance evaluation for each release location by first selecting a wind direction and speed according to a “wind rose” which describes the likelihood of each unique wind speed and direction pair. Once a test simulation was run, we add sensor noise according to the sensor model in (8) for both the mild and large noise cases.

Since the sensor placement procedure is based entirely on the amount of sensors available for placement, we evaluate localization performance for various numbers of sensors, deployed according to our SVM feature selection adaptation to sensor placement. We train SVMs using both the maximum concentration feature space and mean concentration feature space outlined in Section and test them using data generated via QUIC and LBM. As the number of sensors available placed within the urban grid increases, so does localization accuracy. Figure 3 depicts this result for all of the numerical examples.

Localization performed using the maximum concentration feature space and data generated using QUIC requires only four, in the case of mild-noise, or five, for the large-noise case, sensors to observe perfect localization on the data in the test set. When LBM data with large sensor noise is used for analysis, a performance plateau appears once five sensors are placed in the environment that is not overcome until 13 sensors are employed for localization. This is the result of many different sensor locations being selected an equal number of times in the iterative SVM feature selection process. The 13th-most commonly selected features provide the sensor location that produces the discernible information for a large portion of the data in the test set.

Features consisting of a time-series average of observed concentrations produce localization accuracy similar to when maximum concentration features are used. However, weaker performance on the QUIC generated data sets suggests the averaged concentration features have a greater sensitivity to noisy data.

The placement solutions produced adhere to an intuitive strategy. The features selected most frequently among all binary SVMs lie either directly on top of or adjacent to the five release locations. Thus, by the time five sensors are deployed, the city grid is covered by sensors that are not more than one block away from a release location. As is demonstrated in the bottom plot of Figure 3, placing five sensors according to this strategy is not always enough to produce acceptable localization accuracy.



One-class SVM CBRN Detection Evaluation

To evaluate our presented CBRN detection technique based on a one-class SVM novelty detector, we constructed a test set that contained sequences of sensor observations that are purely the product of noise as well as sequences of sensor observations from the test set used in localization evaluation. The sequences consisting of only sensor noise depict cases in which no CBRN release is present within the simulated environment and allow us to compute the detection methodology’s probability of false alarm. The results appear in Figure 4 for the cases in which average concentration features and maximum concentration features are used. Sensor locations were determined according to the placement procedure found in the Methods Section. As shown, for both feature constructions and all simulated data sets, the probability of false alarm was consistently very low no matter how many sensors were deployed. Probability of detection, on the other hand, is only promising in cases in which low levels of measurement noise are considered. In general, when the features are constructed by using a sequence of average concentrations the detection procedure is more robust to large levels of measurement noise than when features are constructed from observed maximum concentrations.

By and large, in most real-world CBRN detection applications, the cost of a false alarm is prohibitively expensive. Emergency response when it is not necessary could

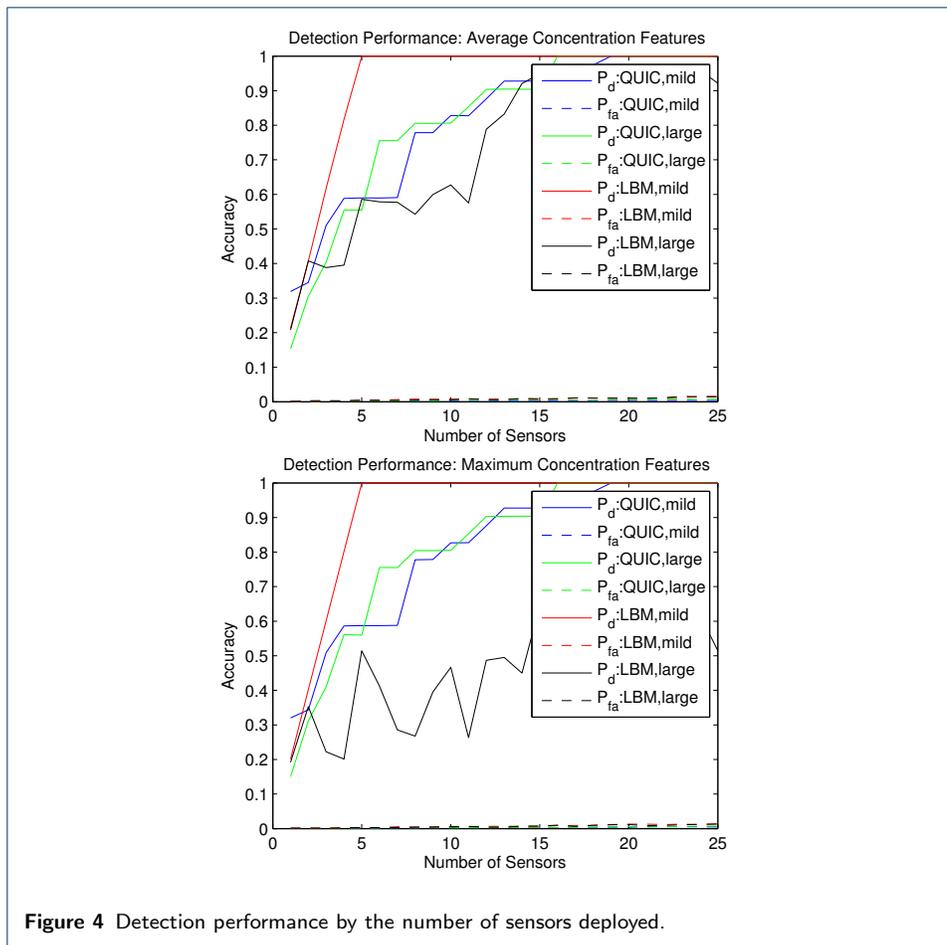


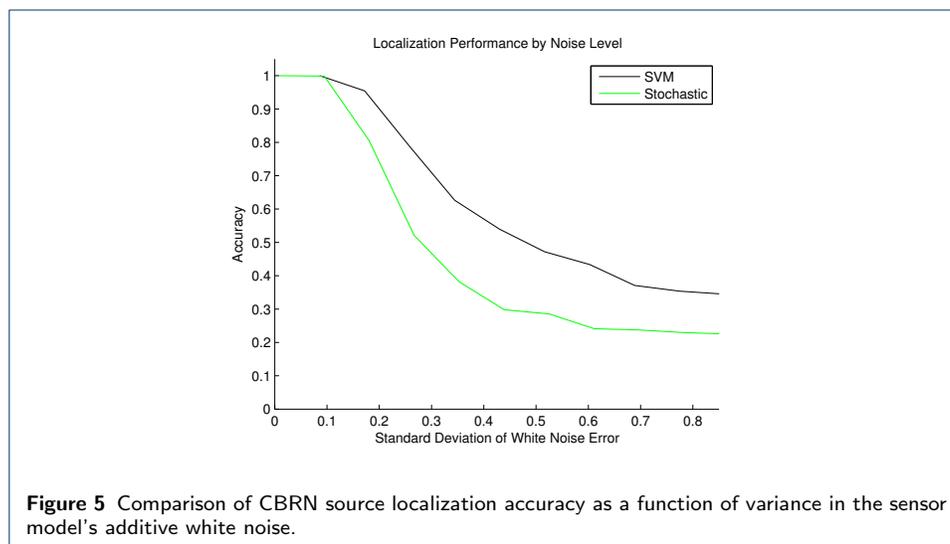
Figure 4 Detection performance by the number of sensors deployed.

mean that commercial or governmental buildings are closed to human access during critical times. Additionally, otherwise healthy people may be administered antidotal medicines which could lead to hazardous and undue side-effects. These facts, coupled with the extreme rarity of a CBRN event, cause extremely low probabilities of false alarm to be the governing factor in performance analysis and implementation. In other applications, such as pollution monitoring, where false alarms do not accrue the same level of costs, the probabilities of detection reported here could be increased by increasing the allowable false alarm rate. This is accommodated in the presented methodology by a tunable parameter which defines the detection procedure’s false alarm rate.

Comparison to Stochastic CBRN Localization

We conducted a comparison of the presented SVM CBRN localization methodology to the stochastic localization methodology presented in [5]. Both schemes, under the right conditions and with the right sensor placements, can locate the origin of a CBRN release quite accurately. The proper comparison entails evaluation of the two methodologies when ideal conditions begin to break down. To aid in this comparison, we conduct localization performance of the stochastic and SVM localization techniques under varying degrees of sensor noise. For varying values of σ_ϵ , we con-

struct training and test sets. The empirical probabilities of correct localization, as computed by localization performance on the test sets, appear in the Figure 5.



Clearly, as noise increases, both methodologies' localization accuracies deteriorate. However, it is evident that the SVM localization procedure is more robust to added noise than the stochastic localization procedure. To be fair, this may be a result of differences in fidelity. SVM features are based on real-valued concentration observations while the stochastic approaches rely on a discretization of concentration measurements. Adding fidelity to the stochastic approaches by expanding the alphabet depicting concentration samples would close the gap shown in Figure 5, albeit at the cost of added computational overhead.

What this translates to in real-world application is a robustness to urban canyon turbulence. In cities where avenues and streets are dwarfed by the tall, densely-packed buildings that line them, turbulence from these urban walls could play a greater role in sporadic particle concentration samplings and should be considered when selecting a localization methodology.

In terms of computational workload, while both methodologies suffer from requiring copious amounts of simulated dispersion data from the gambit of release conditions expected within the environment under surveillance, the SVM training procedure requires much more work than that of the virtually nonexistent training required by the stochastic methodology. However, SVMs in general are an ongoing research topic in an already well developed community. It is likely that advances in SVM training procedures and the development of new "off the shelf" SVM software packages could deaden this computational constraint.

Conclusions

This work is a step away from the inherently challenging approach of solving inverse dispersion problems in highly dynamic urban environments. At the same time, it is a deterministic compliment to the previously established stochastic localization technique in [5].

Numerical evaluation of SVM CBRN localization shows promising accuracy in most situations with even a small number of sensors. We also found that this deterministic strategy is more robust to applications where either sensor measurement noise or chaotic urban canyon turbulence is to be expected as compared to stochastic CBRN localization.

The detection methodology presented is robust to multiple release cases since it is based on determining simply whether a hazardous element is present in the atmosphere. In theory, a multiple release event would be easier to detect than a single release event due to the increased levels of particulate. Localization, on the other hand, is not. To locate the origin of multiple simultaneous releases requires either a fusion of several single-source localizers or sufficient simulation of multiple release events to use the localization approach we have presented. Considering the impact on computational load to the latter option, it would be wise to employ a computational grid when obtaining localizer training data via simulation.

However, one-class SVM release detection performance still has some room for improvement. While dependable concentration observations, as represented by our LBM dispersion simulations with mild sensor noise in this evaluation, leads to ideal performance characteristics, the probability of detection at reasonable numbers of deployed sensors needs to be higher. In the event that false alarm performance demands are less stringent than those imposed on CBRN attacks, such as in pollution monitoring applications, probability of detection using a one-class SVM is likely to improve dramatically.

While the focus on this work has been on urban environments, this need not be the only application. Any problems where detection and localization of the source of dispersed target particles are within the scope of application for our methods. These include finding a spurious pollution generating plant, downstream pollution monitoring, underground pollutant seepage tracing, and nuclear power plant monitoring, provided an accurate simulator is available for data set generation.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

IP advised and directed the presented research as well as contributed to drafting of the manuscript. RTL developed the and implemented the methods, conducted the numerical experimentation, and led the drafting of the manuscript. All authors read and approved the final manuscript.

Acknowledgements

This work is sponsored by the United States Air Force under Air Force Contract #FA8721-05-C-0002. Opinions, interpretations, recommendations and conclusions are those of the authors and are not necessarily endorsed by the United States Government.

Research partially supported by the NSF under grants CNS-1239021 and IIS-1237022, by the ARO under grants W911NF-11-1-0227 and W911NF-12-1-0390, and by the ONR under grant N00014-10-1-0952.

References

1. Ortner, M., Nehorai, A., Jeremic, A.: Biochemical transport modeling and bayesian source estimation in realistic environments. *Signal Processing, IEEE Transactions on* **55**(6), 2520–2532 (2007)
2. Atalla, A., Jeremic, A.: Localization of chemical sources using stochastic differential equations. In: *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference On*, pp. 2573–2576 (2008)
3. Ortner, M., Nehorai, A.: Biochemical transport modeling, estimation and detection in realistic environments. In: *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference On*, pp. 5169–5172 (2008)
4. Locke, R.T., Paschalidis, I.C.: Stochastic localization of cbrn releases. In: *Acoustics, Speech and Signal Processing, 2012. ICASSP 2012. IEEE International Conference On* (2012)
5. Locke, R.T., Paschalidis, I.C.: Model-free stochastic localization of cbrn releases. *Signal Processing, IEEE Transactions on* **61**(17), 4246–4258 (2013)

6. Vujadinovic, M., Rajkovic, B., Grisc, Z.: Locating a source of air pollution using inverse modelling and pre-computed scenarios. In: iEMSs 2008: International Congress on Environmental Modelling and Software (2008)
7. Wawrzynczak, A., Jaroszynski, M., Borysiewicz, M.: Data-driven genetic algorithm in bayesian estimation of the abrupt atmospheric contamination source. In: Computer Science and Information Systems (FedCSIS), 2014 Federated Conference On, pp. 519–527 (2014). IEEE
8. Cortes, C., Vapnik, V.: Support-vector networks. *Machine Learning* **20**, 273–297 (1995). 10.1007/BF00994018
9. Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., Vapnik, V.: Feature selection for svms. In: *Neural Information Processing Systems*, 2000, pp. 668–674 (2000)
10. Vapnik, V.N.: *Statistical Learning Theory*. John Wiley & Sons, Inc., New York, NY, USA (1998)
11. Hsu, C.-W., Lin, C.-J.: A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks* **13**, 415–425 (2002)
12. Schölkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J., Williamson, R.C.: *Estimating the Support of a High-Dimensional Distribution*. MIT Press, Cambridge, MA, USA (2001). <http://dx.doi.org/10.1162/089976601750264965>
13. Paryak, E.R., Brown, M.J.: QUIC-URB v.1.1: theory and users guide. Technical report, Los Alamos National Laboratory (2007). lanl.gov/projects/quic/open-files/QUICURB-UsersGuide.pdf
14. Röckle: *Bestimmung der strömungsverhältnisse im bereich komplexer bebauungsstrukturen*. PhD thesis, Technischen Hochschule, Darmstadt, Germany (1990)
15. Frisch, U., Hasslacher, B., Pomeau, Y.: Lattice-gas automata for the Navier-Stokes equation. *Physical Review Letters* **56**, 1505–1508 (1986)
16. Lallemand, P., Luo, L.-S.: Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, galilean invariance, and stability. *Physical Review E* **61**, 6546–6562 (2000)
17. Chen, S., Doolen, G.D.: Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics* **30**(1), 329–364 (1998)