

A Crowd-Based Explosive Detection System with Two-Level Feedback Sensor Calibration

Chengmo Yang
University of Delaware
chengmo@udel.edu

Patrick Cronin
University of Delaware
ptrick@udel.edu

Agamyrat Agambayev
Arizona State University
aagambay@asu.edu

Sule Ozev
Arizona State University
sule.ozev@asu.edu

A. Enis Cetin
University of Illinois at Chicago
aecyy@uic.edu

Alex Orailoglu
University of California, San Diego
alex@cs.ucsd.edu

ABSTRACT

Large, open, public events, such as marathons and festivals, have always presented a unique safety challenge. These sprawling events, which can take up entire city blocks or stretch for many miles, can draw tens to hundreds of thousands of spectators and in some cases have open admission. As it is impracticable to guarantee the subjection of every event-goer to a security screening, we propose a crowd-based explosive detection system that uses a multitude of low-cost ChemFET sensors which are distributed to attendees. As the sensors offer limited accuracy, we further propose a server-based decision-making framework that utilizes a two-level feedback loop between the sensors and the server and explores spatial and temporal locality of the collected data to overcome the inherent low-accuracy of individual sensors. We thoroughly explore two distinct detection schemes, stressing their performance under a myriad of conditions, thus showing that such a crowd-based detection system comprised of low-cost and low-accuracy sensors can deliver high detection accuracy with minimal false positives.

CCS CONCEPTS

• **Computer systems organization** → **Sensor networks**; • **Computing methodologies** → *Simulation tools*.

KEYWORDS

Sensor-based detection, open-space event monitoring, simulator design and implementation

ACM Reference Format:

Chengmo Yang, Patrick Cronin, Agamyrat Agambayev, Sule Ozev, A. Enis Cetin, and Alex Orailoglu. 2020. A Crowd-Based Explosive Detection System with Two-Level Feedback Sensor Calibration. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD '20)*, November 2–5, 2020, Virtual Event, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3400302.3415670>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICCAD '20, November 2–5, 2020, Virtual Event, USA

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8026-3/20/11...\$15.00
<https://doi.org/10.1145/3400302.3415670>

1 INTRODUCTION

The combination of low-cost sensors in conjunction with expanding computing power is starting to underpin the strategic plans of major semiconductor manufacturers and design houses. Such computing and sensor pairs can operate individually or in a crowd-sensing manner to synthesize the readings and observations of individuals into actionable information. Domains of timely interest where the use of such crowdcasting technologies can be envisioned include medical surveillance systems to tackle outbreaks of contagious illnesses or bomb detection systems to preclude terrorist attacks, among others.

The construction of such voluntary participatory sensor networks necessitates the resolution of multiple technical aspects. On the sensor front, voluntary participation by numerous individuals constrains these systems to low-cost components, which manifest multiple idiosyncrasies, such as validity radius, duration, and recalibration. The computing system, whose principal task is to synthesize the sensor data into information, must prevent noisy data from contaminating the decision making process by structuring both the temporal and spatial windows in which it is examined.

To explore these questions and examine the feasibility of such a system, we outline a case study in the specific context of improvised explosive devices (IEDs) detection. IEDs, which can cost less than \$30 in raw materials, can wreak disproportionate damage, causing loss of life and begetting terror in a population [1]. Detection of IEDs has been the focus of many research projects with state-of-the-art systems employing ion-mobility spectrometry and photoacoustic spectroscopy equipment using high-power laser beams [15, 16, 20]. However, the large size and cost of these technologies precludes their use in public events.

Recent technical revolutions in chemical and biological sensors provide a promising avenue towards the design of an early-warning detection system for homemade IEDs. Specifically, we envision a cyber-physical system (CPS) wherein small and low-cost chemical sensors are distributed to event-goers, which pair with their mobile phones to pass data to a decision-making server. Figure 1 shows the various components and their interactions in the envisioned CPS. Each *node*, composed of a willing participant with a smartphone and a distributed sensor, gathers readings of chemical compounds frequently associated with home-made explosives. The sensor is based on a low-cost CMOS-based sensing device that targets commonly found explosive derivatives [34, 37]. Raw sensor readings are sent to the *host* (smartphone) via a common interface such as Bluetooth BLE, processed, and forwarded to a *server* via the cellular

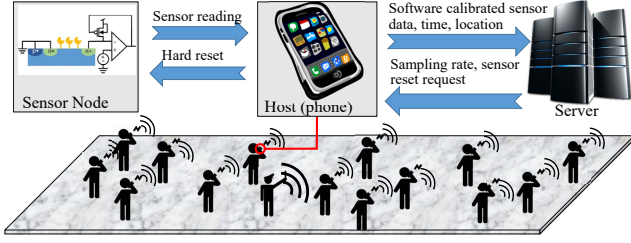


Figure 1: The proposed crowd-based explosive detection system

or WiFi network. The server runs a specialized detection algorithm to decide if there is a potential threat in the defined coverage area.

To realize this system, this paper addresses two technical challenges. First, state-of-the-art, low-cost chemical sensors are prone to drifting, and our CPS must intelligently utilize spatial and temporal correlation between sensors to make decisions. Second, we must explore the rich design space of the detection system; yet no systems are available to model complex crowd dynamics, raw chemical concentrations, and thousands of mobile sensors. To this end we implement a simulator framework that utilizes the outputs of industry-standard tools, and explore the design space of two detection systems. The contributions of this work are three-fold. First, we develop a system-level simulator to combine simulated pedestrian movement, the diffusion of explosive chemical vapor, and the drifting characteristics of chemical sensors. Second, we develop two crowd based detection algorithms to determine the presence (or absence) of target molecules under sensor process variations, sensor drift, and environmental noise. Last but not least, we evaluate the detection algorithms within the system-level simulator to demonstrate the promise of this technique.

The rest of this paper is organized as follows. Section 2 provides a brief description of our chosen sensor technology, ChemFETs. Section 3 details the theoretical basis of the proposed detection system, while Section 4 describes the simulator design. Section 5 presents the experimental setup and results. Section 6 reviews related works, and finally Section 7 concludes the paper.

2 SENSOR MODELING AND CALIBRATION

The envisioned CPS relies on low-cost and unobtrusive sensors to achieve crowd-sourced sensing in a large public gathering. Sensors must be highly sensitive as, in open areas, the diffusion of explosive vapor from an IED may be limited to a quite small area and present in rather low quantities. The most commonly used explosive compound in IEDs is ammonium nitrate, and the gas vapor with the highest concentration within this compound is ammonia. We use ammonia as the target detection component for this work.

In determining the most appropriate sensors, we choose an architecture that can be easily implemented with standard commercial CMOS processes, such as *Chemically sensitive Field Effect Transistors* (ChemFETs) [5, 9, 10, 23]. The sensor model in this work is based on ChemFETs, although the proposed system can work with other sensor architectures that can provide similar sensitivities.

ChemFET Sensor Modeling: A chemical sensor converts the concentration of the target molecule to current, which is subsequently converted into the digital domain by a current-to-voltage converter

and an analog-to-digital converter (ADC). The *measured* concentration (C_{sens}) can be expressed in terms of the actual concentration (C), a conversion ratio ($R(t)$), and sensor noise $n(t)$:

$$C_{sens}(t) = R(t) \cdot C + n(t) \quad (1)$$

The conversion ratio $R(t)$ will deviate from its ideal value of 1 due to process variations and the drift experienced by the sensor after deployment. $n(t)$ is the noise of the sensor response which sets a lower bound on the sensitivity as any concentration variations smaller than the noise will be indistinguishable from it. In the literature, CMOS-based chemical sensors for ammonia have been shown to yield sensitivities of 0.1–100 *parts per billion* (ppb) depending on manufacturing technology [3, 28, 32]. In this work, we assume a sensitivity of 50ppb and set the ADC resolution to 1LSB = 50ppb. Thus, the added quantization noise standard deviation, σ_n , can be expressed in terms of the ppb as: ($\sigma_n = LSB/\sqrt{12} = 14ppb$ [7]). Our sensor model also mimics process variations by assigning a random deviation to the conversion ratio, $R(t)$, with a standard deviation σ_R of 20%, thus setting it in the range of 0.4–1.6 under a 3σ range for variation.

ChemFET Drift Modeling: Unfortunately, the sensitivity of low-cost sensors drifts significantly over time, limiting their usefulness to minutes unless the readings are calibrated to compensate for this drift. Drifting within ChemFETs has been studied extensively in the literature [10, 11]. The drift response is generally split between a fast and a slow response, which can be modeled as an exponential decay [6, 19, 22] in the current signal output. Based on hardware experiments on manufactured ChemFETs (without surface coating) [24], the following drift model applies to this work:

$$R(t) = (R_0 + \epsilon_R) + R_f e^{-\frac{t}{\tau_f + \epsilon_f}} + R_s e^{-\frac{t}{\tau_s + \epsilon_s}} \quad (2)$$

where $R(0) = R_0 + \epsilon_R + R_f + R_s$ is the sensor conversion factor at the initial time after deployment/reset, R_f , R_s are fast and slow drift coefficients, τ_f , τ_s are fast and slow drift time constants, and ϵ_R , ϵ_f , ϵ_s are the corresponding error terms for each of the model variables. The fast and slow drift coefficients each have a nominal value of 0.2 with a standard deviation of 10% to account for process variations. The fast and slow drift time constants are set to 10s and 500s, respectively. Each time constant is also assigned a 10% standard deviation to account for process variations¹. This process variation profile enables us to generate a random population of sensors with individual sensitivities and individual time constants that can differ greatly. Thus, each sensor's *usable* time will vary depending on its own assigned parameters.

In practice, the model parameters can be estimated for each manufactured sensor and stored within the sensor device to pass to the host upon activation. Besides drift, each sensor parameter is also assigned a random 10% error term that represents errors during the characterization of the manufactured sensors. This error term is unknown to the host device and will generate a mismatch between the actual concentration and measured concentration.

Sensor Reset: The host can utilize the measured sensor parameters, including the conversion ratio, drift coefficients, and drift time constants, to calibrate the sensor in software, combining the raw

¹ While the drift coefficients and time constants are based on hardware measurements, the process variation model is based on experience with CMOS parameteric variations (e.g. threshold voltage for large feature-size devices).

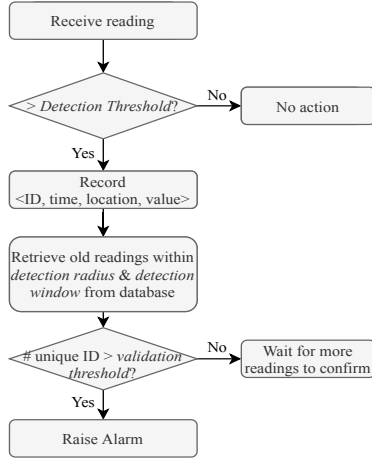


Figure 2: Validator-based Detection

Name	Description	Values
Validation Threshold	# of unique nodes required to confirm a high reading.	0–4
Detection Window	Maximum age (sec) of old readings kept for validation.	50, 100, 200 seconds
Validation Radius	Maximum distance to search for old readings to validate.	2-meter
Detection Threshold	Lowest sensor reading that is considered “high”.	5 ADC LSB

Table 1: Parameters of Validation System

Name	Description	Values
Grid Size	Size of grid squares.	1m–8m
Moving Avg. Period	Number of readings used to calculate grid average.	5, 10, 20
Detection Threshold	Lowest sensor reading that is considered “high”.	5 ADC LSB

Table 2: Parameters of the Grid System

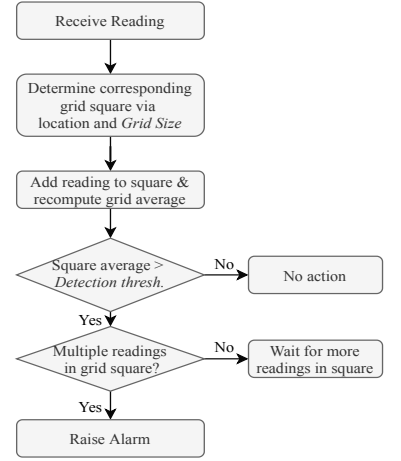


Figure 3: Grid-based Detection

data from the sensor with the model parameters to compensate for drifting. However, due to the unknown error term, the difference between the actual conversion ratio of the device and the compensated conversion ratio will grow over time. Moreover, software compensation cannot eliminate the reduction in sensitivity as non-target molecules begin to bond to the sensor. Once sensitivity falls below a threshold, the detection distance diminishes, and the sensor is effectively useless. To reverse the effects of sensor drift, we require a mechanism to release the trapped molecules at the insulating dielectric surface above the channel. It has been shown that the cycling of the vertical electric field by controlling the reference electrode voltage for a ChemFET [35] effectively resets the device back to its pre-drift state.

3 PROPOSED DETECTION ALGORITHMS

A high-quality IED detection algorithm must overcome the inherent inaccuracies of the ChemFET sensors described in Section 2. As sensor readings are known to drift, relying on readings obtained from a single sensor will lead to many false positives or negatives. Instead, the proposed detection system treats the sensors as a population and exploits both *spatial* and *temporal* information. We propose and examine two methods of detection: validator based detection and grid average based detection. We also employ a *two-level feedback* mechanism (server-level and host-level) to initiate sensor reset requests for tackling the observed sensor drift.

3.1 Spatial and Temporal Detection Validation

The major challenge encountered in developing an accurate detection algorithm is the potential high rate of false positives, as it is more likely for high readings to come from sensor drift rather than real detection of an IDE. Previous works in sensor networks [4, 14, 26] rely on searching for smooth reading gradients over a large area from stable sensors. However, this solution does not apply to our crowd-based IED detection system as nodes move fast while the detection range of ChemFET sensors is limited (around 2m). To reduce false positive rates, we explore two approaches to record sensor readings and quickly identify neighboring nodes to “confirm” high readings from individual sensors.

3.1.1 Validator Based Detection. This approach exploits the idea of corroboration, where each node searches for *validation* from its peers. Figure 2 shows the flow of this algorithm. In this approach, the server maintains a map of node readings, retaining high readings (i.e. preliminary detections) and recording them with a vector composed of four values: $\langle ID, time, location, value \rangle$. When the server receives a reading above the *detection threshold*, it places it into this map and searches within a *validation radius* (set to the IED detection range of 2m) around the (x,y) position (spatial locality) of the current reading for older (temporal locality) readings that confirm the value. The system requires that the quantity of readings surpass a *validation threshold*, e.g., a high reading must be validated by readings taken from N unique neighboring nodes. If the number of required validators is increased, the system becomes more specific and more likely to suppress false positives, while also being less sensitive to true positives.

Confirming high readings only partially solves the false-positive rejection problem. If the system retains unconfirmed readings for too long, it will validate old readings caused by drifting sensors. To control the temporal correlation between data, the system also maintains a *detection window* which defines the maximum age of readings that can be used for validation. This shifting temporal window keeps the data in the system fresh, removing old, unconfirmed, and likely drifting readings from consideration. A smaller window retains less stale data, thus increasing detection specificity but reducing sensitivity. Table 1 summarizes the parameters used in this approach that will be explored in our experiments.

3.1.2 Grid Average Based Detection. This approach overlays a grid structure onto the monitored region, breaking it into equally sized regions which maintain a *moving average* of the readings within the area. For example, overlaying a $2m \times 2m$ grid onto a $100m \times 100m$ area would result in $50 \times 50 = 2,500$ grid squares.

Figure 3 presents the flow of this algorithm. When a node takes a reading, the server determines which grid square the reading is from based on its (x,y) location and appends the reading to the moving average of that square. If the average of the square rises above the *detection threshold* and if at least two readings have been taken, an alarm will be raised. A smaller grid size creates a more sensitive

and less specific system as each square encompasses fewer nodes, allowing a single node with consecutive high readings to increase the square average above the *detection threshold*. To accurately detect true positives while still suppressing false positives, each grid square utilizes an independent moving average. The size of the *moving average period* also impacts the sensitivity and specificity of the detection system. A larger *moving average period* is more likely to encompass data from multiple nodes as they move through the square, and the readings from a single drifting node are less likely to influence the grid average. Thus, a larger period decreases system sensitivity and increases its specificity. Parameters used in the grid system are summarized in Table 2.

3.2 Two-level Feedback for Sensor Reset

As explained in Section 2, it is possible to revert a drifted sensor back to its initial state by applying a reverse vertical field, thereby releasing trapped molecules/ions from the sensor surface. However, our system must judiciously use this reset capability as it has several downsides. First, the reset process requires the sensor to be off line for 2 seconds (empirically determined in experiments using ISFETs [35]). Assuming an average human walking speed of 1 meter per second, a node can easily pass through the detection range (2m) of the explosive source during the reset process, missing a chance to detect it. Second, to reset the sensor, the host device must apply a strong reverse electrical field to the sensor, generating a power consumption overhead.

We use a two-level feedback to determine when a sensor needs to be reset and incorporate the ensuing downtime into our system-level simulations. Sensor reset is initiated in two ways: (1) based on the mathematical drift model and the elapsed time, the host determines if the sensor conversion ratio, $R(t)$, has diminished below 50% of its original value; and (2) based on the readings of neighboring sensors, the server determines if a sensor is an outlier.

3.2.1 Host Initiated Sensor Reset. In the proposed CPS, each sensor undergoes a short testing phase during manufacturing to determine its drift parameters and stores them. Later when it is paired with a host device, the host extracts these parameters and uses them to estimate the sensor conversion ratio via Eqn. (2). As mentioned in Section 2, the sensor eventually will experience sensitivity loss via the binding of non-target molecules. The host device uses this drift model to trigger a hard reset if $R(t)$ has diminished below 50% of $R(0)$ resulting in the sensor becoming unavailable for 2 seconds.

3.2.2 Server Initiated Sensor Reset. While host initiated reset can utilize the sensor parameters learned in production, there are multiple factors that the drift model cannot correct for, such as imperfect testing (reflected as the multiple error terms ϵ in Eqn. (2)), environmental noise (non-target molecules binding to the sensor), and temperature and humidity variations. As more time passes, the impact of these error terms and noise grows, and the sensor readings (even after software calibration) become increasingly unreliable.

To remedy this situation, we employ *server-level feedback*, wherein the server exploits both spatial and temporal data correlation to detect outliers and instruct outlier nodes to reset early. Specifically, the server maintains a history of readings with references to both time and location in a grid based system of the arena. If the server determines that readings from a node are vastly different from the



Figure 4: Simulated city block to determine the concentration profile of target molecule, ammonia

grid average (either too low or too high), it sends a message to the host requesting a sensor reset.

4 SYSTEM LEVEL SIMULATOR DESIGN

This section describes our simulator, designed to faithfully model the crowd-based explosive detection system, allowing the user to evaluate sensor drifting and explore the impact of different system-level parameters on detection sensitivity and specificity. It incorporates an accurate diffusion model for the explosive (generated in COMSOL [2]) and individual movement agents to simulate the movement of each crowd member (generated by Menge [12]). The simulator also implements the sensor model described in Section 2, as well as a server architecture to simulate the reception and processing of data from each node following the detection and sensor calibration algorithms described in Section 3.

4.1 Explosive Diffusion Modeling

To simulate the chemical diffusion of a homemade IED, we require the ability to simulate a scenario with a set of variables such as the effects of starting concentration, diffusion over time, and diffusion around obstacles such as walls or buildings.

We model the diffusion characteristics of various explosive vapors in the COMSOL Multi-Physics simulator [2]. The molecular diffusion model begins in a simple cylindrical source with a small circular opening (e.g. in the shape of a pipe bomb). The transient behavior of gas diffusion, based on Fick's law (Eqn. (3)), is simulated using the transport of diluted species interface tool of COMSOL multiphysics, where D_i is the diffusion coefficient and C_i is the concentration (mol/m^3). Since we target ammonia in this work, its diffusion coefficient ($D_i = 0.26cm^2/s$) is used in COMSOL simulations.

$$\frac{\partial C_i}{\partial t} = D_i \nabla^2 C_i \quad (3)$$

The chemical source is placed in a realistic environment including obstructions, such as buildings, and random convection models. Figure 4 shows one such realistic environment: an 800m×150m portion of the Boston marathon track. The track bisects the area into northern and southern regions. The black blocks represent buildings and roads that people cannot enter, and white space represents areas where the crowd can stand or move freely. This structure is designed in AutoCAD and input into COMSOL. COMSOL will then, according to the diffusion model, generate an output which specifies the expected concentrations of the chemical vapor at each point in time for each location in the simulated area.

4.2 Pedestrian Motion Modeling

To simulate realistic pedestrian motion, it is necessary to design complex scenarios that mimic the motions of event goers at large outdoor events such as marathons and music festivals. At such

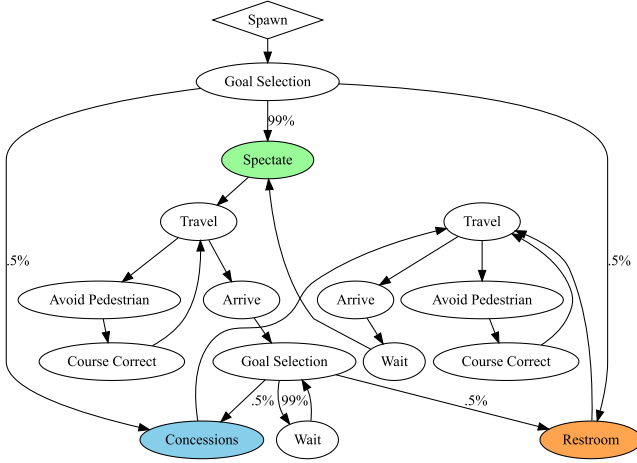


Figure 5: A state machine used for pedestrian movement simulation

events, some attendees may head directly to their seats, while others will prefer to visit concession or restroom areas before moving to their assigned areas. To model such behavior, we use the Menge pedestrian simulator [12] with a custom scenario generation tool. Menge provides the ability to simulate pedestrian groups with behaviors that follow a probabilistically defined state machine. Figure 5 presents an example. Nodes (i.e., pedestrians) enter this state machine and follow it until the end of the simulation. Nodes follow a shortest path algorithm to reach their “goals” (colored nodes in Figure 5) while routing around dynamic obstacles such as other pedestrians. The simulator is also able to model detailed behavior customization such as node preferences for social distance maintenance and differing routing strategy utilization. As simulations are probabilistic, no two successive simulations within Menge will be the same. We have modified the Menge simulator to output a record of each node’s movements for use within the system simulator.

4.3 Our System-Level Simulator

To combine and utilize the data from the previous two simulators for IED detection, we have designed and implemented a custom, modular, and event driven simulator. To ease design space exploration, the simulator is designed to be highly reconfigurable, controlled by a number of command line flags. As Figure 6 shows, the simulator takes as inputs the crowd movement data generated by Menge, the concentration data from COMSOL, and a list of configuration commands that configure detection algorithm parameters. It then runs the simulation and halts upon the detection of a true positive or the end of the simulation time. Upon completion, the simulator generates a series of log files to aid in statistical analysis of the run. The simulator consists of two major pieces, nodes and the server, each defined below.

4.3.1 Nodes. Each member of the crowd is simulated as a unique entity. Upon receiving data from the crowd simulation, the simulator assigns each member of the crowd a unique ID. Each node is then initialized with its own sensor model parameters (as discussed in Section 2), and embarks upon its pregenerated (Menge) movement path, taking sensor readings of the areas it passes through

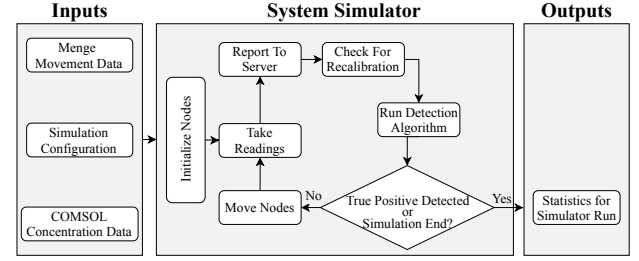


Figure 6: Flowchart of the system simulator

(COMSOL) and reporting them to the server. Every sensor reading incurs the following updates to the simulator state and calculations:

- (1) The node position is updated. If the exact time of sampling is between movement simulation time steps, the node position is bilinearly interpolated.
- (2) The explosive vapor concentration is retrieved from the COMSOL simulation. The sensor model transforms the raw concentration according to the unique drift model of the node. The resulting value is then transformed to an ADC value to simulate the response of the ADC within the sensor. Both transformations model the errors and accuracy loss incurred in the process.

After taking a reading, the node sends the reading value to the server along with its ID, (x,y) location, and time of the reading. As each node maintains its own sensor model values (randomly generated following normal distributions), it also maintains its own sense of *age*, which is utilized within the sensor drifting model (variable t in Eqn (2)). Drift begins when the sensor is first ‘unwrapped’ and exposed to air. Each host is aware of its sensor parameters (but not the error factors) and therefore can predict how degraded the sensor is. When the sensor reaches what the host believes to be 50% degradation, it undergoes a self-triggered reset as described in Section 3.2.1, which resets the internal degradation counter and restores the node to its initial sensitivity.

4.3.2 Server. The server is simulated as an independent element within the simulator that can only receive information sent by individual nodes. This client-server architecture allows for the testing of server-based detection algorithms. Information is sent to the server in “packets” where each packet contains one or more readings.

Upon receiving a reading, the server runs one of the detection algorithms (Section 3.1) to determine whether to flag the reading as an explosive detection. The system uses a pessimistic detection threshold of 5 ADC LSBs (equivalent to 250ppb), as listed in Tables 1 and 2. We choose this value to ensure that random noise does not trigger detection due to software compensation when the sensitivity is degraded. After the server determines whether to raise the alarm, it logs the results of its decision and the corresponding node ID and timestamp. To ensure high accuracy and to prevent mistakes caused by drifting sensors, the server maintains a history of readings with references to both time and location. If the server rules a sensor reading to be an outlier, it will send a reset request to the node (Section 3.2.2). This feedback mechanism largely reduces false positive rates and improves the overall specificity of the detection system.

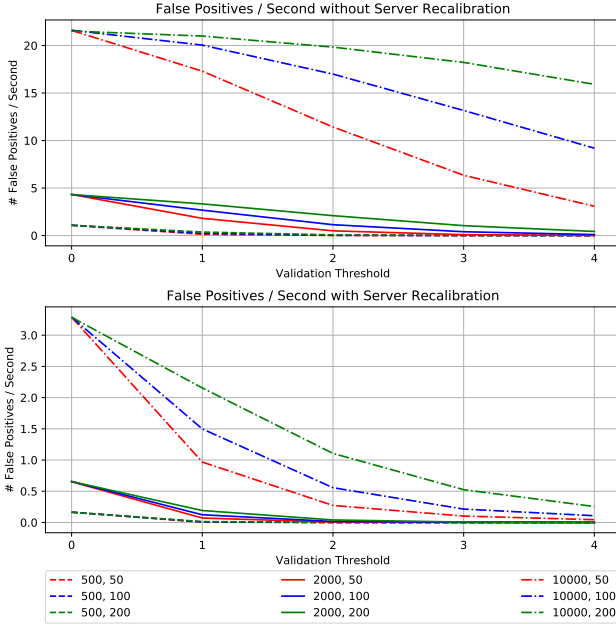


Figure 7: False positive rate of Validator-Based System. Each line is named as (node population, size of detection window).

5 EXPERIMENTAL SETUP AND RESULTS

To evaluate the effectiveness and design trade-offs of the proposed detection algorithms, we employ the system-level simulator described in Section 4 and create a scenario which mimics the last 800m of the Boston marathon (Figure 4). Pedestrians arrive into either the northern or the southern region of the simulated area via entry points at the border. Each pedestrian then follows the state machine presented in Figure 5.

Our experiments explore both the validator-based and grid-based detection algorithms, while also examining the effects of server-based sensor reset. To evaluate the validator-based algorithm, we simulate a total of 4680 runs, testing the combinations of 5 validation thresholds, 3 window sizes, 3 node populations, 2 pedestrian movement patterns, 2 server calibration options, as well as 1 no-bomb and 25 different bomb location cases. The total number of runs is 7488 for the grid-based algorithm as it has 8 different grid sizes (while the other parameters are kept the same).

This section presents the experimental results of both a system stability test which examines the number of false positives over time, and the specificity and sensitivity of each detection algorithm.

5.1 System Stability Test

Our first set of experiments examines the impact of sensor drift on system stability. We run the system with no IED present and examine the effects of each detection algorithm on false positive rejection. We vary their associated parameters according to Tables 1 and 2. Each simulation process runs for 10,000 seconds to reasonably simulate the length of a spectator event and to provide the detection algorithm with ample opportunity to experience and handle sensor drift. We also vary the node population to better understand the effects of population density, and enable/disable server initiated

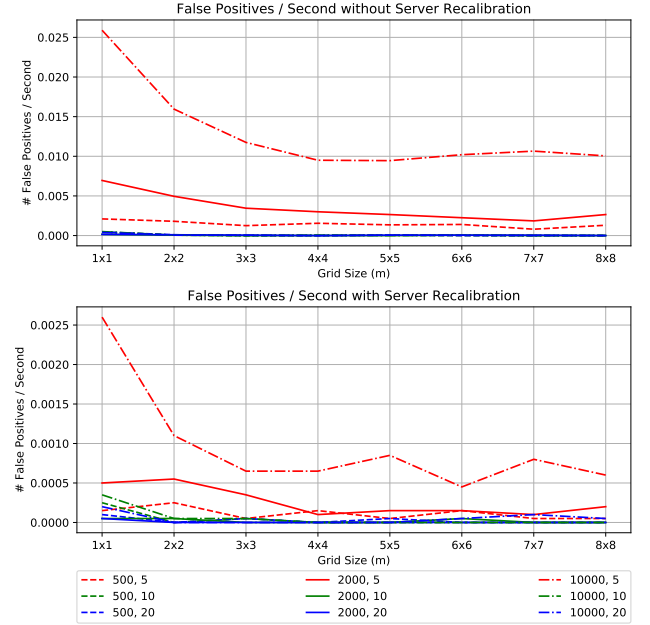


Figure 8: False positive rate of Grid-Based System. Each line is named as (node population, size of moving avg. period).

sensor reset to examine the effect of this feedback mechanism. Figures 7 and 8 present the results of these experiments.

The upper graph in Figure 7 demonstrates the need for a strong false positive rejection system. When the validation threshold is 0, the system immediately alerts on every high reading, with a population of 10,000 nodes reporting over 20 false positives per second. Upon increasing the number of validators, the false positive rate plummets, confirming the hypothesis that requiring agreement between independent, inaccurate sensors decreases false positives. The lower graph of Figure 7 further demonstrates that allowing the server to detect outliers and trigger sensor reset is critical, showing a 30x–100x reduction in false positive rate. Large validation thresholds and low node populations result in excellent performance; with a node population of 500 and a validation threshold above 2, the false positive rate decreases to zero.

Figure 8 demonstrates that the grid based scheme suppresses far more false positives than the validator based scheme. With server recalibration, the false positive rates are in the 10^{-4} per second range, a 10000x decrease as compared to the high population configurations of the validator scheme. As the grid squares grow larger, the false positive rate decreases to a nearly constant level, indicating that larger squares encompass more nodes and hence reduce the influence of a single node on the average. Furthermore, unlike the validator system, node population does not adversely affect the grid system, as it examines data from all nodes in a square, rapidly discarding stale data via a moving average. In comparison, the validator system focuses only on high readings. For a validation threshold of k , it takes false readings from k nodes within the detection window to generate a false positive, a phenomenon that grows more prevalent as more nodes are added to the simulation or as the size of the detection window increases.

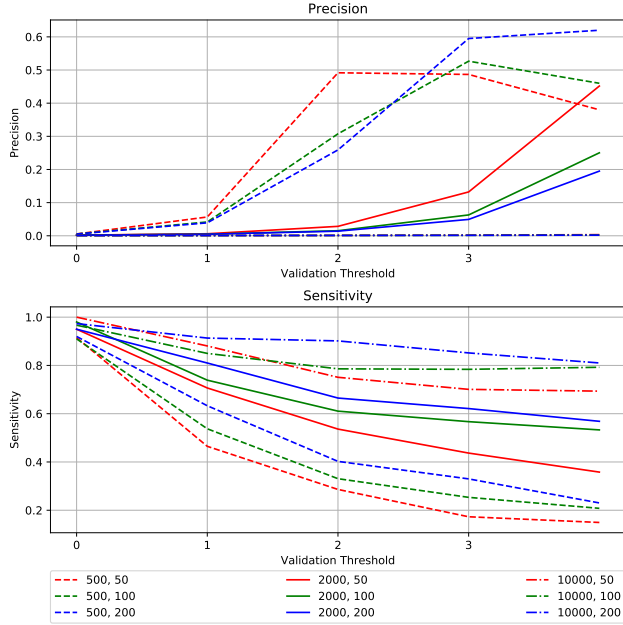


Figure 9: Precision and Sensitivity of validator-based system. Each line is named as (node population, size of detection window).

5.2 Explosive Detection

While the capability of suppressing false positives is essential to high system stability, it is equally important to ensure a high IED detection rate. To this end, we perform multiple simulation runs each containing a single explosive in one of 25 possible locations. Each run ends upon detecting the explosive (a true positive case) or after 10,000 seconds (a false negative). We evaluate the impact of different configurations and node population on the *sensitivity* and *specificity* (evaluated with *precision*²) of the system following the equations below:

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (4)$$

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} \quad (5)$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (6)$$

An ideal system should have both precision and sensitivity close to 1.0. As Figures 7 and 8 demonstrate that server-initiated sensor reset is highly beneficial, we report only experiments with server reset enabled. The precision and sensitivity statistics for each configuration are shown in Figures 9 and 10.

Precision and Sensitivity: The precision statistics demonstrate that the validator system is, again, less specific than the grid system. Furthermore, although a high validation number decreases the number of false positives greatly, thereby boosting precision, it is too strict, generating a large number of false negatives, and lowering the sensitivity. This reduced sensitivity is expected as

²Both *specificity* and *precision* quantify the impact of false positives. The reason for utilizing *precision* instead of *specificity* is that true negatives are extremely common: almost every node generates a true negative at every time step.

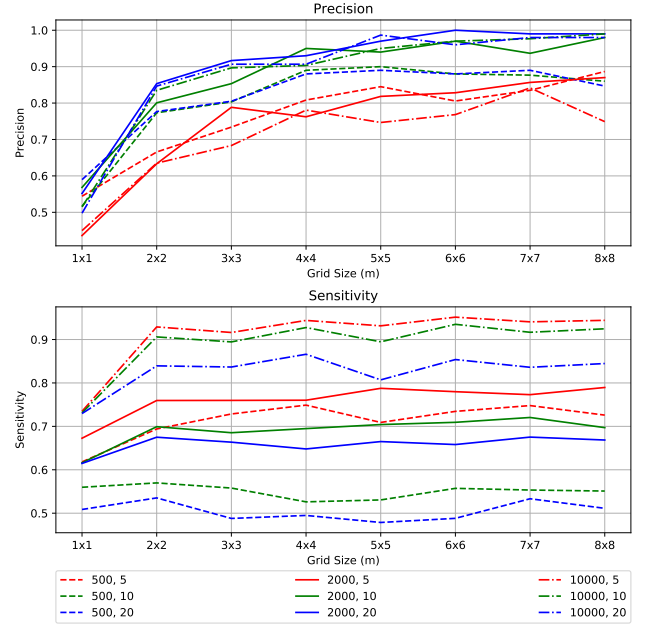


Figure 10: Precision and Sensitivity of grid-based system. Each line is named as (node population, size of moving avg. period).

more unique nodes must confirm high readings, requiring either high node traffic in an area or higher node density.

With respect to the grid system, larger grid squares increase precision to a point, reaching equilibrium around 6m×6m. While a large grid size is likely to encompass the effective sensing radius of a sensor (2m), it also encompasses the readings from more nodes, reducing the chance that a grid square is dominated by a few drifting nodes. The moving average period of the grid system greatly effects the precision and sensitivity, with a larger period leading to a higher precision value but a lower sensitivity. This conforms to expectations, as a larger period reduces the ability of a single node's readings to effect the average of a grid square and raise an alarm.

Population Density: In both systems, larger population densities increase sensitivity. This is expected, as a higher population density increases the chance that multiple nodes will pass near the IED, reducing the likelihood that a detection goes unconfirmed.

The precision values reported in Figures 9 and 10 are consistent with the false positive rates reported in Figures 7 and 8: the lower the false positive rate, the higher the precision. An in-depth examination shows that precision values are largely affected by the movement patterns (and resulting population densities) of the outdoor marathon scenario. In the beginning of the scenario, nodes enter the area from the edges with a goal of distributing themselves along the edges of the track. Importantly, this 'entry' phase leads to regions of both very high density as nodes jostle through constricted areas to reach their destinations and low density as nodes break out of these concentrated areas and are the first to reach their destinations. In the validator system, the high density areas result in a period of increased false positive activity, as there are more nodes which might generate validations. The early false positives

in the grid system can be attributed to the nodes that arrive first to their destination. These nodes may be the only nodes to influence the grid square until more nodes arrive. As shown in Figure 8, low densities increase the false positive rate of the grid system. As the majority of bomb detections occur within the first 1,800 seconds (30 minutes) of simulation, when the nodes are filling the area, the precision values reflect these higher, early, false positive rates for both systems.

5.3 Detection Rate and Time

The detection rate and time-to-detection vary from system to system. The grid system averages a 92% detection rate for 500 nodes and a 100% detection rate for higher populations. Neither high nor low population detection rates are effected by grid size or moving average period. The validator system struggles at low populations and high validation thresholds (50% detection rate with 500 nodes and a validation threshold of 4) as there are not enough nodes to confirm readings within the validation window. While the detection rate increases with higher densities and larger detection windows, only 10,000 nodes and a 200 second detection window are able to achieve 100% detection from 0-4 validators. The time taken for the grid scheme to detect the IED is always less than an hour, with high population configurations detecting the IEDs in a few minutes and low population configurations detecting them in 40 minutes. As expected, the higher sensitivity of smaller moving average periods decreases the time-to-detection relative to higher periods. The validator system achieves time-to-detection similar to the grid system for low validation thresholds, but more than 2x time-to-detection as higher validation thresholds are required.

5.4 Recommended Configurations

When protecting an open space event, it is important to carefully consider the different parameters of each detection algorithm. In examining the performance of the proposed detection systems, the grid scheme presents superior sensitivity and precision. In almost all node populations, a grid size of 6m×6m and a moving average period of 10 provides the best tradeoff between precision and sensitivity. This configuration reduces the false positive rate tremendously, while still maintaining a high detection rate and short time-to-detection. If the population of an event is known to be low and the event does not have the resources to check all alarms, a larger moving average period can be used to maintain high precision while slightly prolonging time-to-detection.

6 RELATED WORK

Multiple works examine the utilization of inaccurate sensors and event region (high chemical concentration) detection. However, these works either require large, smooth gradients to detect outliers [4] or the use of stationary sensors [8]. Other approaches [25, 36] attempt to rate sensors based on their correlation with other nearby, immobile, sensors. These systems are not applicable to our mobile sensor nodes which attempt to detect very small concentration gradients. Research in event region detection [14, 26] mainly focuses on the utilization of stable nodes and requires the processing of the readings of the entire sensor network at once or detections from multiple nodes at a time to reach a consensus. In contrast, our system targets small detection radius (2m) where

there is no guarantee that multiple sensors will be in range of the target at once.

The protection of public events is not a new area, and state-of-the-art systems fall into roughly three categories: pre-screening an area, pre-deploying stationary sensors, or the utilization of high-cost, operator required, mobile detection systems. In the pre-screening approach, there exist multiple commercial systems ranging from rapid mobile swab testing to continuous airborne detection [13, 17, 30]. These systems require highly trained users and closed event arenas where the area can be scanned ahead of time and people entering can be monitored. In contrast, our system is designed for areas where it is not possible to effectively create a secure perimeter and guarantee safety ahead of time. Other systems, which utilize varying sensor locations, such as [31, 33], focus on continually scanning the crowd with sensors and employing a team of professionals to monitor them. These stationary systems are usually placed in high traffic locations to monitor as many event goers as possible. In contrast, the proposed ChemFET based system is provided to anyone that enters the event, and can realize coverage wherever a user wanders. Another mobile method for event protection is the employment of explosive detection canines [27]. Full event coverage, however, is impractical as K9 teams would need to constantly and disruptively examine areas of an event such as seating areas, something that few event goers are likely to tolerate. Our proposed detection system is far less intrusive, passively monitoring a user wherever they go.

The proposed detection systems bear some similarities to sensor networks that deploy spatially distributed sensor nodes for habitat monitoring [29], military surveillance [21], and health care [18]. Similarly to [4, 14, 26], these approaches require the designer to determine where to place sensors or design algorithms that work with specific configurations etc. In our proposed scheme, the centralized server is able to exercise minimal control over the individual sensors and is unable to influence their location.

7 CONCLUSIONS

In this paper we proposed the utilization of low-cost ChemFET sensors for crowd-based explosive detection at large outdoor events. We described the design challenges inherent to the development of such a sensor network, motivated its necessity, and designed a system-level simulator to combine crowd simulations, chemical sensor models, and chemical diffusion data. We utilized the simulator to model and evaluate two detection algorithms, explored the design space for these detection systems, and evaluated their specificity and sensitivity in the face of inaccurate and drifting data from low-cost ChemFET sensors. We demonstrated that a grid-based detection system utilizing a two-level feedback mechanism can achieve not only high detection rates, but also suppress > 99% of the false positives generated by the sensors. Our future work will examine further design methodologies and algorithms to detect moving threats as well as the fusion of multiple different types of sensors to further enhance system accuracy.

8 ACKNOWLEDGMENTS

The work is supported by National Science Foundation under grant No. 1739390, 1739396, 1739451, and 1739684.

REFERENCES

- [1] 2012. IEDs: the home-made bombs that changed modern war. *Strategic Comments* 18, 5 (2012), 1–4.
- [2] 2018. COMSOL Multiphysics v. 5.4. <http://www.comsol.com>. COMSOL AB, Stockholm, Sweden.
- [3] Ayalew H Assen, Omar Yassine, Osama Shekhah, Mohamed Eddoudi, and Khaled N Salama. 2017. MOFs for the sensitive detection of ammonia: Deployment of fcu-MOF thin films as effective chemical capacitive sensors. *ACS sensors* 2, 9 (2017), 1294–1301.
- [4] Laura Balzano and Robert Nowak. 2007. Blind Calibration of Sensor Networks. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN)*. Association for Computing Machinery, 79–88.
- [5] P. Bergveld. 1970. Development of an Ion-Sensitive Solid-State Device for Neurophysiological Measurements. *Biomedical Engineering, IEEE Transactions on BME*-17, 1 (1970), 70–71.
- [6] L. Bousse, N.F. De Rooij, and P. Bergveld. 1983. Operation of chemically sensitive field-effect sensors as a function of the insulator-electrolyte interface. *Electron Devices, IEEE Transactions on* 30, 10 (1983), 1263–1270.
- [7] Mark Burns, Gordon W Roberts, et al. 2001. *An introduction to mixed-signal IC test and measurement*. Vol. 2001. IET.
- [8] Vladimir Bychkovskiy, Seapahn Megerian, Deborah Estrin, and Miodrag Potkonjak. 2003. A Collaborative Approach to In-Place Sensor Calibration. In *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks (IPSN)*. Springer-Verlag, 301–316.
- [9] S. Casans, Diego Ramirez Munoz, A.E. Navarro, and A. Salazar. 2004. ISFET drawbacks minimization using a novel electronic compensation. *Sensors and Actuators B: Chemical* 99, 1 (2004), 42–49.
- [10] Kow-Ming Chang, Chih-Tien Chang, Kuo-Yi Chao, and Chia-Hung Lin. 2010. A Novel pH-dependent Drift Improvement Method for Zirconium Dioxide Gated pH-Ion Sensitive Field Effect Transistors. *Sensors* 10, 5 (2010), 4643–4654.
- [11] D.Y. Chen and P.K. Chan. 2008. An Intelligent ISFET Sensory System With Temperature and Drift Compensation for Long-Term Monitoring. *Sensors Journal, IEEE* 8, 12 (2008), 1948–1959.
- [12] Sean Curtis, Andrew Best, and Dinesh Manocha. 2016. Menge: A Modular Framework for Simulating Crowd Movement. *Collective Dynamics* 1 (2016), 1–40. <https://collective-dynamics.eu/index.php/cod/article/view/A1>
- [13] DetectaChem. 2016. <http://detectachem.com/products>
- [14] M. Ding, D. Chen, K. Xing, and X. Cheng. 2005. Localized fault-tolerant event boundary detection in sensor networks. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 2. 902–913.
- [15] Gary Alan Eiceman, Zeev Karpas, and Herbert H Hill Jr. 2013. *Ion mobility spectrometry*. CRC press.
- [16] Robert Gordon Ewing, David Alan Atkinson, GA Eiceman, and GJ Ewing. 2001. A critical review of ion mobility spectrometry for the detection of explosives and explosive related compounds. *Talanta* 54, 3 (2001), 515–529.
- [17] FLIR Systems, Inc. 2013. <http://www.flir.com/threatdetection/>
- [18] Raghu K. Ganti, Praveen Jayachandran, Tarek F. Abdelzaher, and John A. Stankovic. 2006. SATIRE: A Software Architecture for Smart AtTIRE. In *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services (MobiSys '06)*. 110–123.
- [19] C.Z.D. Goh, P. Georgiou, T.G. Constantinou, T. Prodromakis, and C. Toumazou. 2011. A CMOS-Based ISFET Chemical Imager With Auto-Calibration Capability. *Sensors Journal, IEEE* 11, 12 (Dec. 2011), 3253–3260.
- [20] Frans JM Harren, Julien Mandon, and Simona M Cristescu. 2000. Photoacoustic spectroscopy in trace gas monitoring. *Encyclopedia of Analytical Chemistry* (2000).
- [21] Tian He, Sudha Krishnamurthy, Liqian Luo, Ting Yan, Lin Gu, Radu Stoleru, Gang Zhou, Qing Cao, Pascal Vicaire, John A. Stankovic, Tarek F. Abdelzaher, Jonathan Hui, and Bruce Krogh. 2006. VigilNet: An Integrated Sensor Network System for Energy-efficient Surveillance. *ACM Trans. Sen. Netw.* 2, 1 (Feb. 2006), 1–38.
- [22] Blake C. Jacquot, Nini Muñoz, Darren W. Branch, and Edwin C. Kan. 2008. Non-Faradaic electrochemical detection of protein interactions by integrated neuromorphic CMOS sensors. *Biosensors and Bioelectronics* 23, 10 (2008), 1503–1511.
- [23] S. Jamasb. 2004. An analytical technique for counteracting drift in ion-selective field effect transistors (ISFETs). *Sensors Journal, IEEE* 4, 6 (2004), 795–801.
- [24] Fatih Karabacak, Uwadiae Obahiagbon, Umit Ogras, Sule Ozev, and Jennifer Blain Christen. 2016. Making unreliable Chem-FET sensors smart via soft calibration. In *2016 17th International Symposium on Quality Electronic Design (ISQED)*. IEEE, 456–461.
- [25] Farinaz Koushanfar, M. Potkonjak, and Alberto Sangiovanni-Vincentelli. 2003. On-line fault detection of sensor measurements. *Proceedings of IEEE Sensors* 2, 974–979.
- [26] B. Krishnamachari and S. Iyengar. 2004. Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. *IEEE Trans. Comput.* 53, 3 (2004), 241–250.
- [27] Joshua Levine. 2013. The Education of a Bomb Dog. <https://www.smithsonianmag.com/innovation/the-education-of-a-bomb-dog-4945104/>
- [28] Amir Lichtenstein, Ehud Havivi, Ronen Shacham, Ehud Hahamy, Ronit Leibovich, Alexander Pevzner, Vadim Krivitsky, Guy Davivi, Igor Presman, Roey Elnathan, Yoni Engel, Eli Flaxer, and Fernando Patolsky. 2014. Supersensitive fingerprinting of explosives by chemically modified nanosensors arrays. *Nature communications* 5 (2014), 4195.
- [29] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. 2002. Wireless Sensor Networks for Habitat Monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*. 88–97.
- [30] Nanosniff Technologies Pvt. Ltd. [n.d.]. <http://www.nanosniff.com/>
- [31] QinetiQ Group plc. 2019. <https://www.qinetiq.com/what-we-do/c4isr/stand-off-threat-detection>
- [32] W.P.R.V. Stauthamer, J.F.J. Engbersen, W. Verboom, and D.N. Reinhoudt. 1994. Influence of plasticizer on the selectivity of nitrate-sensitive CHEMFETs. *Sensors and Actuators B: Chemical* 17, 3 (1994), 197–201.
- [33] Thruvision Ltd. 2019. <http://thruvision.com/>
- [34] Joseph Wang. 2007. Electrochemical Sensing of Explosives. *Electroanalysis* 19, 4 (2007), 415–423.
- [35] David Welch, Sahil Shah, Sule Ozev, and Jennifer Blain Christen. 2013. Experimental and Simulated Cycling of ISFET Electric Fields for Drift Reset. *Electronics Device Letters* (2013).
- [36] Xiang-Yan Xiao, Wen-Chih Peng, Chih-Chieh Hung, and Wang-Chien Lee. 2007. Using sensorranks for in-network detection of faulty readings in wireless sensor networks. *International Workshop on Data Engineering for Wireless and Mobile Access*, 1–8.
- [37] Jehuda Yinon. 2002. Field detection and monitoring of explosives. *TrAC Trends in Analytical Chemistry* 21, 4 (2002), 292–301.